CUSTOMER EXPERIENCE AND THE IMPACT TOWARDS CUSTOMER PERCEPTION IN INDONESIAN TELECOMMUNICATION COMPANY USING TWITTER SENTIMENT ANALYSIS: PT. SMARTFREN TELECOM, Tbk. CASE STUDY

UNDERGRADUATE THESIS

Submitted to the International Undergraduate Program in Industrial Engineering as the Requirement for the degree of Undergraduate Degree Industrial Engineering
at
Universitas Islam Indonesia

Arranged by:

Name	: Muhammad Arif Naufal
Student Number	: 18522032

INTERNATIONAL UNDERGRADUATE PROGRAM
DEPARTMENT OF INDUSTRIAL ENGINEERING
FACULTY OF INDUSTRIAL ENGINEERING
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2023

AUTHENTICITY STATEMENT

In the name of Allah, I declare that this research is a work based on research that I did by myself except for the citation and summaries that have their sources listed. If in the future, this research is proven to violate the rules and intellectual property rights, I am willing to accept the sanctions by Universitas Islam Indonesia.

Yogyakarta, April 7th, 2023

SEPULUH RIBU RUPIAH
10000
METERAI
TEMPEL
FB5E7AKX318605668

Muhammad Arif Naufal
18522032

UNDERGRADUATE THESIS APPROVAL OF SUPERVISOR

CUSTOMER EXPERIENCE AND THE IMPACT TOWARDS CUSTOMER PERCEPTION IN INDONESIAN TELECOMMUNICATION COMPANY USING TWITTER SENTIMENT ANALYSIS: PT. SMARTFREN TELECOM, Tbk. CASE STUDY

UNDERGRADUATE THESIS

By

Name	**:**	**Muhammad Arif Naufal**
Student Number	**:**	**18522032**

Yogyakarta, 15 May 2023

Supervisor,

(Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.)

UNDERGRADUATE THESIS APPROVAL OF EXAMINATION COMMITTEE

CUSTOMER EXPERIENCE AND THE IMPACT TOWARDS CUSTOMER PERCEPTION IN INDONESIAN TELECOMMUNICATION COMPANY USING TWITTER SENTIMENT ANALYSIS: PT. SMARTFREN TELECOM, Tbk. CASE STUDY

UNDERGRADUATE THESIS

By

Name	:	Muhammad Arif Naufal
Student Number	:	18522032

Has been defended in front of Examination Committee in Partial Fulfillment of the Requirement for Bachelor Degree of Industrial Engineering Department Universitas Islam Indonesia

Examination Committee

Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.
Examination Committee Chair

Dr. Drs. Imam Djati Widodo, M.Eng.Sc.
Member I

Dr. Taufiq Immawan, S.T., M.M.
Member II

Acknowledge by,
Head of Undergraduate Program
Department of Industrial Engineering
Faculty of Industrial Technology
Universitas Islam Indonesia

(Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.)

DEDICATION PAGE

This undergraduate thesis that spent a lot of time and resources is dedicated to my family that always support me in any situation and condition.

To all my truly friends who always share happiness to each other.

This thesis also would not be possible to be completed without the guidance of my supervisor, Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.

MOTTO

“Once something is a passion, the motivation is there” − Michael Schumacher

PREFACE

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Alhamdullillah, all praise to Allah SWT, because only with his permission the author can finish the undergraduate thesis. Shalawat and greetings to the Prophet Muhammad SAW who has saved the mankind from the Jahiliyyah era to the Islamiyah era and gave Syafa'at into Yaumul Akhir.

This report was made to fulfill the requirements for completing a degree in Industrial Engineering at Universitas Islam Indonesia. The author realizes that he cannot finish the undergraduate thesis without the support, prays, and motivation from all parties. Thus, the author would like to thank:

1. Mr. Prof. Dr. Ir. Hari Purnomo, M.T. as Dean of Industrial Technology Faculty, Universitas Islam Indonesia.
2. Mr. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. as the Head of the Undergraduate Program in Industrial Engineering, Universitas Islam Indonesia, as well as my supervisor who always provides guidance and knowledge to help the author finish this undergraduate thesis.
3. Mrs. Ir. Ira Promasanti Rachmadewi, M. Eng. as the Secretary of the International Undergraduate Program in Industrial Engineering, Universitas Islam Indonesia.
4. Mrs. Devy Nurrahmah, S.Kom as the staff of the International Undergraduate Program in Industrial Engineering, Universitas Islam Indonesia who always help the author in accomplishing administrative process during the undergraduate thesis report.
5. The author's beloved parents, Mr. Dr. Riyanto Wibowo, S.ST., M.Si. and Mrs. Nur Laila, the author's siblings, Amalia Khoirunnisa and Muhammad Zaidan Aulia who always gives support, motivation, and prayer.
6. IP students' batch 2018, and all the closest and truly friends for the togetherness and encouragement that cannot be mentioned one by one.
7. All the internship partners and mentors in Smartfren who have worked side by side to finish the project related to my thesis subject, as well as shared motivation and togetherness during the program.

The author realizes this undergraduate thesis cannot be said as perfect, so the author appreciates if the reader gives critics and recommendations. The author hopes that this report can give many benefits to all parties.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 15 May 2023

(Muhammad Arif Naufal)

ABSTRACT

Consumers of telecommunications are growing, and 97% of all users are assigned to prepaid cell plans. Moreover, social media is utilized by marketers and salespeople to connect with and reach their target clients, because it helps businesses quickly and affordably reach their target customers and understand their requirements and goals. The limitations of data deep dive analysis prevent insights from being sufficient for business questions, despite Smartfren starting to analyze social media sentiment data as of 2021. To meet the commercial needs of Smartfren, a Twitter sentiment analysis was done for this study. Sentiment analysis labeling is conducted using a distinct lexicon-based method to increase accuracy. This method divided the lexicon dictionary and labeling process based on churn-related tweets and non-churn-related tweets. Following training and testing using the Support Vector Machine (SVM) classification algorithm, which is divided into three feature extraction techniques (Count Vectorizer, Bags of Words, and TF-IDF), the results of the sentiment labeling are then used to evaluate the accuracy of the labeling using a parameter. The Bags of Words approach produces the best classification results, with 97% accuracy (98% precision, recall, and F1 score). The results of a sentiment analysis reveal that there are more tweets with negative sentiment than positive sentiment during a period of 2.5 years (70.5% of negative tweets). Product experience and network experience are the parts of the customer experience that are most frequently discussed, accounting for (37.8% and 37.5% respectively). Most of the network-related tweets have mostly discussed coverage, social media, and gaming experiences. Positive sudden sentiment changes and customer churn dominate negative ones in terms of consumer perception change. The researcher found that the number of sentiment changes and churn are higher with a higher level of average engagement rate, and vice versa.

Keywords: Telecommunication, Social Media, Customer Experience, Sentiment Analysis, Lexicon-based Method, Classification, Support Vector Machine, Churn, Engagement Rate

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Background

Telecommunication consumers are increasing year after year, as the total of internet subscribers emerge significantly. According to the GSM Association (2022), the total of mobile internet subscribers in 2021 was 4.2 Billion, and predicted to reach 5 Billion in 2025 worldwide. The country's telecommunications business, on the other hand, has promising development possibilities in data and value-added services. Prepaid mobile subscriptions account for 97 percent of all subscribers, making switching from one provider to another quite simple (International Trade Administration, 2021). As a result, service providers are competing to retain or acquire more customers, where mobile telecommunications service providers challenged themselves to compete in increasing innovation and quality.

According to KPMG (2020), one of the challenges of telecommunication company company is the quick shift in consumer behavior that may well create infrastructure and security threats for both consumers and telecommunication companies. There will be a need for developing innovative methods of addressing shifting customer wants. A case study of Pepephone, a Spanish telecommunication company (established in 2015), has successfully observed and analyzed the customer experience and needs, which resulted in a reconfiguration of product price without compromising its service quality (KPMG, 2020). Successfully addressing this problem and taking advantage of the available opportunities will have an impact on how telecommunication companies see the changing in customer preferences and behaviors, which are now necessary to be analyzed. Given the fact that understanding the changing in customer behavior and needs is inevitable, telecommunication companies need to know the biggest challenge of analyzing customer experience and how to measure it comprehensively.

Nowadays, social media become a more viable solution for marketing purposes, to analyze customer experiences. It has become an essential aspect to retain and acquire customers, as it allows companies to analyze the customer experience of using particular products and determine branding strategies (Diebes, H.M.Y, 2019). Users of social media frequently express themself by posting complaints about a variety of objects. Users can share information and produce content on social media networks. Brands, on the other hand, are recognized as social media influencers (Sharawneh, T., 2020). From social media, users' good feedback and experience can create a desire for a certain product, raise brand recognition, inspire a positive attitude about the product, and brand reputation, and influence purchasing intentions (D., Evans & J., Mackee, 2010). One of the social media platforms with the highest amount of users is Twitter. As of 2014, Twitter had 284 million monthly active users and sent 500 million Tweets each day, with around 80% of those Tweets coming from mobile devices (Carley et al., 2015). Even though it may not have as many users as platforms like Facebook and Instagram, which have billions of users each, Twitter nevertheless has a faithful following of people that log in frequently. In terms of user segmentation, Gen-Z (those born between 1997 and 2012) were active tweeters in 2020. According to Twitter, Gen-Z users posted 52% of all tweets that year. Although Twitter users range widely in age, individuals between the ages of 25 and 34 are the most dominant worldwide. Here is a summary of Twitter's global age distribution (The Social Shepherd, 2022):

- Ages 13 to 17: 6.6% of users
- Age 18–24: 17.1% of users
- 25 to 34-year-olds of users, 38.5%
- 35 to 49 years old: of users, 20.7%
- 17.1% of users are 50 or older

By analyzing customer experience on social media, companies would be able to discover the real needs and satisfaction of the younger generation customers which accounts for more than half of the total social media users.

However, reading through a massive number of text-based data on Twitter can take a long time. Due to the subjectivity of emotions and their polarity as either good, negative, or neutral, it is difficult to precisely evaluate and quantify sentiments of the customer experience stated within the tweet texts (Jayasanka, S. C., 2013). Additionally, some people speak in various jargon, slang, and abbreviations for words for convenience. It's tough to completely detect and integrate data on brand sentiment when relying only on manual processes, whether assessing solicited feedback via channels like surveys or investigating unknown feedback discovered on social media (Qualrtics, 2022).

Due to the complexity of assessing social media users' textual opinions, sentiment analysis would play an important role to solve these problems. Sentiment analysis is a natural language processing (NLP) technique that extracts significant informational patterns and features from a large text corpus (Lamba & Madhusudhan, 2022). It is also known as subjectivity analysis, opinion mining, or emotional artificial intelligence. Through the tweets that are shared on the platform, it extracts the sentiments of social media users in the form of subjectivity (objective and subjective), polarity (negative, positive, and neutral), and emotions (angry, happy, surprised, sad, jealous, and mixed) regarding their thought, attitude, views, opinions, beliefs, comments, requests, questions, and preferences. There are many methods to conduct sentiment analysis, which are categorized such as using Machine Learning method, Lexicon method, and Hybrid method, as stated by Sadia, A., et al. (2018). However, the common problem of sentiment analysis is that most of the sentiment analysis algorithm models are lacking contextualization of sentiment labeling when the text is processed by the algorithm (Ajayi & Sodha, 2020). When determining the positive or negative polarity of the text, the context of the text may differentiate the result. It's the reason that the sentiment analysis model must be adjusted into the context of the analysis, such as in this case telecommunication company. In this research, a modified lexicon-based method would be chosen, because of the ability to contextualize the users' opinion textual data (Kolchyna et al., 2015). It's because lexicon-based sentiment analysis used the words dictionary to calculate the polarity of the whole text (negative and positive).

In summary, as the competition between telecommunication companies become more competitive and customer behavior is changing constantly, telecommunication company like Smartfren can use social media sentiment analysis to measure and analyze customer experience. According to OpenSignal (2021), Smartfren has the lowest mobile experience quality compared to the other telecommunication companies, in terms of video experience, games experience, voice app experience, download speed experience, upload speed experience, and 4G availability. Responding to this fact, in 2021 Branding Department in Smartfren reached out to the Data Science Team in Smartfren, to continue the sentiment analysis projects that were requested in 2020. The project got stuck for a year due to the lack of human resources. By having this research, it's expected that Smartfren would be able to reflect on the insights about what the customers have thought and experienced so that the product could later be improved.

1.2 Problem Formulation

Based on the background, the problem formulation that can be made are:

1. How many positive and negative customer experiences from the tweets shared by Twitter users, and the most discussed topics?

2. How many sudden sentiment changes and what causes sentiment change?
3. How many tweets are categorized as churn-related tweets (positives and negatives churn-related tweets), as well as the reason behind it?

1.3 Research Scope

This research must be limited in numerous ways due to limited resources. The following determines the study's scope:

1. In this study, the Branding division and Data Science division of PT. Smartfren Telecom, Tbk. was used as a case study.
2. The modeling method is used with data from Twitter comments from a library that retrieves tweets via the Twitter API.
3. Examined Tweets from PT. Smartfren Telecom, Tbk. that referenced product keywords.
4. The Indonesian language is utilized in tweets.
5. The lexicon-based method is used for sentiment labeling, Support Vector Machine (SVM) is utilized for the classification algorithm, the Evaluation method is used to determine the sentiment labeling accuracy, and Post Engagement Rate (PER) calculation is used for measuring user-generated contents' engagement (tweets engagement).
6. Tweets data from January 2019 to September 2021.
7. Tweet sentiment analysis is only focused on the customer experience tweets.
8. The service topics that were examined included network, coverage, and product, as well as other telecommunication-related topics that would be discovered during the research.

1.4 Research Objective

This research aimed to give accurate and comprehensive insights by utilizing Twitter sentiment analysis, extracting the topic and sentiment of the tweets from Smartfren customers, as well as analyzing the significance of the sentiment's impact on customer perception. These methods are used to maximize the utilization of sentiment analysis in Twitter over certain telecommunication brands.

1.5 Research Benefits

This research is able to give benefits for the institution where the study was conducted, which is PT. Smartfren Telecom, Tbk., academician, and practitioner.

1. For University

Increasing knowledge and research related to sentiment analysis and its utilization in terms of customer experience analysis.

2. For Company

   Giving new comprehensive insights about a real-time customer experience in social media, so that the company could immediately react towards the upcoming branding and marketing planning in a more strategic way.

3. For Practitioner

   Having a new perspective to analyze customer experience in social media that could be implemented in the future.

CHAPTER II
LITERATURE REVIEW

This chapter covers the underlying theory, previous research, methods, and theoretical framework that will be used to help address the research challenge.

2.1 Data Mining

The amount and complexity of the data that is stored in an organization's database are expanding along with the advancement of the technology used by that company. Organizations must effectively utilize the data that is growing in volume and structure in order to give their business a competitive edge. More advanced methods and procedures are needed to extract information from this stored data in order to provide more meaningful and valuable information. The number of new techniques for obtaining data from this set is also increasing. By using these techniques, an organization can get access to more insightful data than they could before using more conventional techniques like straightforward descriptive queries. Data mining is one technique that has recently gained popularity in businesses.

The practice of extracting useful information and interesting patterns from vast amounts of data is known as data mining. The data sources can be databases, data warehouses, the Internet, other information sources, or data that is dynamically streamed into the system (Han et al., 2012). The patterns discovered must be meaningful in that they lead to some advantage, usually an economic one (Witten et al., 2011). According to Han et al. (2012), data mining techniques consist of four primary procedures:

1. Predictive Modelling

   This method is comparable to how humans learn, which involves making observations and identifying crucial characteristics in a phenomenon. The model created using this method employs a supervised learning methodology with two key learning and testing phases. Two similar methods are used in predictive modeling, namely categorization and value prediction using statistical science's linear regression methods.

2. Segmentation

   To divide the data set into a number of groups with distinct and homogeneous profiles, segmentation is used. To identify homogeneous groups, this technique employs an unsupervised learning methodology.

3. Link Analysis

This method seeks to establish a connection between records, or a group of records, in the database. This method aims to determine whether one factor influences how another appears in an event.

4. Deviation Detection

   This method is employed to locate anomalies in a data set. These activities can be carried out either directly as a byproduct of data mining or via statistical and visualization techniques. Fraud in the usage of credit cards, insurance claims, quality assurance, and defect tracking can all be found using this method.

Because the author believes that predictive modeling is the best applicable technique for categorizing sentiment, this study will use it to develop a sentiment model. Positive and negative opinions and sentiments can be categorized into two groups using the classification technique in predictive modeling.

Although data mining is simply one crucial step in the sequence of knowledge discovery procedures, data mining is generally referred to as Knowledge Discovery from Data (KDD). The procedure can be explained as consisting of numerous successive iterations (Han et al., 2012):

1. Data cleansing: Data cleansing seeks to eliminate errors and erratic data.
2. Data Integration: Data integration is the process of combining data from many sources.
3. Data selection: Data selection is the process of choosing relevant data from the database to be used in the study.
4. Data Consolidation: Data is consolidated and turned into a format that is acceptable for data mining requirements, such as by putting together a summary or aggregate procedure.
5. Data mining: Data mining is the practice of applying intelligence techniques to identify patterns in data.
6. Evaluation of Patterns: Pattern evaluation is used to Identify patterns that represent the discovered knowledge base.
7. Knowledge Presentation: Knowledge presentation is the process by which the user's requested knowledge is produced using visualization and knowledge representation approaches.

2.2 Twitter Crawling

To make data crawling easier, Twitter offers Streaming Application Programming Interfaces (APIs). The API makes it simple for users to get real-time tweet data. The

Twitter API's original goal was to understand user relationships and interactions, but it is also frequently used to gather data on particular communities to learn what they think about trending topics (Nguyen & Zheng, 2014).

Crawling involves fast transferring a huge number of web pages onto a local storage location and indexing them according to a number of keywords (Liu, Web Crawling, 2011). For this research, we will use opinions from certain Twitter accounts. Web search engines work by storing information about numerous web pages that are taken directly from the site. These pages are retrieved by an automated Twitter crawler that clicks on each link it encounters. Following then, each page's content is examined to determine how it is indexed (for example, words are extracted from titles, subtitles, or special fields called meta tags).

Since it’s an effective way to collect tweets data about certain topics, the researcher would use Twitter crawling to collect tweets regarding the discussed topic from 2020 until now. The data crawling would be based on specified keywords about related telecommunication companies and its product. With a main emphasis on business intelligence, Power BI is an interactive data visualization software tool. With the help of a number of software services, applications, and connections called Power BI, it is possible to transform disparate data sources into coherent, immersive visual insights that can be used in interactive ways. Users may easily analyze data using Power BI to create dynamic and simple visual representations in the form of graphs, scatter plots, maps, and other visual formats. Direct reading of data from a database, website, or structured files like spreadsheets, CSV, XML, and JSON are all acceptable methods for data input.

2.3 Power BI

With a main emphasis on business intelligence, Power BI is an interactive data visualization software tool. With the help of a number of software services, applications, and connections called Power BI, it is possible to transform disparate data sources into coherent, immersive visual insights that can be used in interactive ways. Users may easily analyze data using Power BI to create dynamic and simple visual representations in the form of graphs, scatter plots, maps, and other visual formats. Direct reading of data from a database, website, or structured files like spreadsheets, CSV, XML, and JSON are all acceptable methods for data input.

2.4 Sentiment Analysis

Textual information called sentiment is found online and includes both facts and opinions. Sentiment is a personal statement that expresses how someone feels about a situation (Dave, Lawrence, & Pennock, 2003). The study of calculating people's sentiments, views, and emotions using text-based entities and attributes is known as sentiment analysis or opinion mining (Liu, 2012).

In general, sentiment analysis is divided into 2 main categories (Schneider, 2005):

a. Coarse-grained sentiment analysis

   This group of categories handles document-level analysis. In short, we're attempting to categorize a document's orientation as a whole. Positive, Neutral, and Negative are the three classifications for this orientation. There are individuals who, however, make the value of this approach continuous and non-discrete.

b. Fine-grained sentiment analysis

   Currently, this second category is often being used by researchers. The idea is that this kind is where most study is concentrated. The item to be categorized is a sentence within a document, not the entire document. For example, a sentence like “Smartfren has a very slow connection when it comes to downloading apps” has a “negative” sentiment, and a sentence like “Compared to other providers, Smartfren’s internet package is very cheap and affordable” has a “positive” sentiment.

Due to the widespread availability of English-language tools and resources, most sentiment analysis research to date has only been conducted in English. SentiWordNet and WordNet are two tools that are often used for sentiment analysis.

According to Sadia, A., et al. (2018), there are three approaches for broadly categorizing sentiment analysis, visualized in Figure 2.1 below:

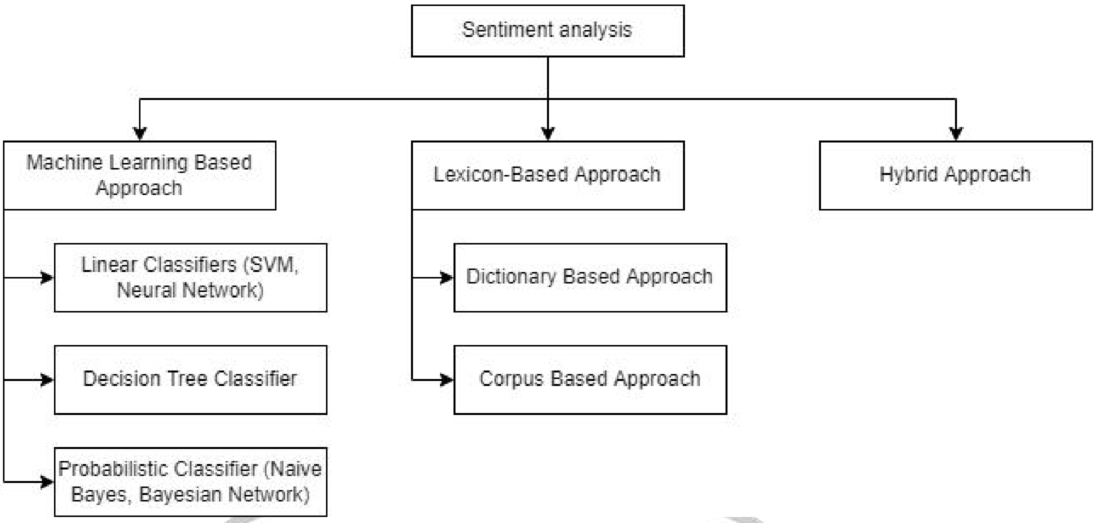

Sentiment analysis
Machine Learning Based Approach
Lexicon-Based Approach
Hybrid Approach
Linear Classifiers (SVM, Neural Network)
Decision Tree Classifier
Probabilistic Classifier (Naive Bayes, Bayesian Network)
Dictionary Based Approach
Corpus Based Approach

Figure 2.1 Sentiment Analysis Methods

1. Machine Learning based algorithms

   Through the use of manually labeled data, machine learning algorithms train the classifier. However, the classifier's effectiveness is greatly influenced by the quality and coverage of the training data; hence, the only drawback is that it needs a sizable database to function. Instead of using a lexicon, this method is more accurate.
2. Lexicon based approach

   In this method, the polarity (positive, negative, and neutral) of a text's content is described using a sentiment lexicon. As opposed to machine learning-based methods, this strategy is simpler to understand and execute. The disadvantage is that text analysis must be done by humans, which is a downside. The test will be more notable for cutting through the noise, detecting the sentiment, and separating useful data from diverse content sources the more significant the information volume. Dictionary-based approaches (based on words from dictionaries like WordNet or other entries) and Corpus-based approaches (using corpus data, which can be further divided into Statistical and Semantic approaches) are the next two categories under which lexicon-based approaches can be further subdivided.
3. Hybrid Approach.

   This strategy combines lexicon-based and machine-learning techniques. This review, which contains a study of several works on lexicon-based sentiment analysis, can be helpful for scientists who are new to the field.

In this research, the author would like to implement the Lexicon-based approach to do the sentiment analysis classification or sentiment labeling, and use machine learning to test and train the labeled tweets to be used for accuracy and precision evaluation afterwards. Therefore, the method would be a hybrid approach.

2.4.1 Data Preprocessing

The dataset often utilized for sentiment analysis classification takes the form of a raw document. For the sentiment analysis categorization procedure, this raw document includes portions that have no significance, including Indonesian stopwords. It is necessary to process the dataset so that it is prepared to be used as input at the document preprocessing stage in order to convert the raw document into a representation or document with an appropriate format for the learning algorithm used in the classification process from an opinion (Han et al., 2012).

a. Data Filtering

Document filtering, also known as document filtering, is the process of removing portions of unprocessed documents from consideration for the classification process (Dave, Lawrence, & Pennock, 2003). Dates, labels, topics, and other classification elements, for instance, may be present in the raw document, but they will be removed since they represent a value or category that is actually determined by another method, namely learning algorithms.

b. Case Folding

Case folding is the process of comparing cases in a document. This is performed to make searching simpler. The use of capital letters varies among different text documents. Consequently, case folding is necessary to transform the full text of the document into a standard form (usually lowercase).

c. Tokenization

Tokenization is the process of breaking up text into discrete sections, such as sentences or paragraphs (Manning, Raghavan, and Schütze, 2009). Kevin is very excited to be graduated, for instance, tokenizing the text yields six tokens: "Kevin," "is," "very," "excited," "to," "be," and "graduated." Typically, a space and a punctuation mark are used as the token separator. In linguistics, tokenization is frequently utilized, and the findings of tokenization are helpful for additional text analysis.

d. Stopword Elimination

A stop word is a word that frequently appears in a written document but has little importance on the content of the document (Patel & Shah, 2013). Conjunctions and prepositions are excellent candidates from the list of stopwords that need to be dropped. Documents connecting words in Indonesian include "yang," "di," "dan," "itu," "dengan," and others. This action helps to cut down on the number of features that must be used.

2.4.2 Part of Speech (POS) Tagger

By POS tagging, words in a text are given a portion of a language tag. Grammatical groups including verbs, nouns, adjectives, adverbs, and so forth are included in this language tag (Sari, Syandra, & Adriani, 2008). Many natural language processing applications, including word splitting, machine translation, and ambiguous words, depend on the part-of-speech tagger. Bing Liu (2010), stated that the procedure of assigning word classes to the POS Tagger, validated this claim. And based on his research, the verb, noun, adverb, and adjective word classes have been chosen. The four categories of words mentioned above are the ones that convey the most sentiments. Using the Big Indonesian Dictionary to determine word classes (KBBI).

2.4.3 Lexicon-based sentiment analysis

The Lexicon Based method was selected for this study because it is simple, practical, and useful for analyzing sentiment from data from social media. Data from surveys, tweets, posts on Facebook, or other social media sites that provide customer reviews of a good or service are examples of data that fit the lexicon method. Lexicon based method is based on the assumption that contextual sentiment orientation is the sum of the sentiment orientations of each word or phrase. The lexical method can be used to extract sentiments from blogs by combining lexical knowledge and text classification (Melville et al., 2011). The lexicon method can be created manually (Taboada et al., 2011) or expanded automatically from the seed of words (Kaji & Kitsuregawa, 2007).

For this technique, a lexicon (a dictionary) of words with assigned polarity is needed. The dictionary is a crucial part of a system that utilizes the lexicon-based approach. In order to normalize sentences and extract keywords, dictionaries are utilized. A few of the lexicons that are now in use are the Opinion Lexicon, SentiWordNet, AFINN Lexicon, Loughran McDonald Lexicon, NRC-Hashtag, and Harvard Inquirer Lexicon (Kolchyna et al., 2015). Here is an illustration of a dictionary's contents (Nurfalah et al., 2017):

1. Positive keywords (in Bahasa): baik, banyak, bisa, ok, best, pintar, lancar, cepat, bagus, senang.
2. Negative keywords (in Bahasa): bangkrut, banjir, bodoh, gagal, kurang, susah, lambat, parah, bohong.
3. Negation keywords: belum, bukan, tidak
4. Slang words conversion dictionary (in Bahasa): bgmn = bagaimana, bgs = bagus, beud = banget

The quality of the lexicon has a significant impact on categorization quality. Various methods can be used to generate a lexicon dictionary (Kolchyna et al., 2015):

1. Manually constructed lexicon: The most time-consuming method is to manually create a vocabulary and categorize words as positive or negative. However, this method is the most straightforward. According to Das and Chen (2001), this method involves reading through thousands of messages and hand-picking the words that expressed sentiment. Then, words from a training dataset that can be used for sentiment classifier purposes are found using a discriminant function. The remaining terms were enlarged to incorporate each word's final lexicon in all of its possible forms.
2. Constructing a lexicon from trained data: This method falls under the category of supervised methods because it requires a training dataset of labeled sentences (Kolchyna et al., 2015). The method creates a bag-of-words by tokenizing the sentences from the training dataset. The words are then further filtered to remove parts of speech and other non-sentimental components of speech. The frequency of each word in positive and negative phrases determines the words' past polarity. For instance, if the word "success" occurs more frequently in the sentences in the training data set that have been categorized as positive, its prior polarity will be given a positive value. Since

the words and their polarity are better suited to a specific type of writing, this method of lexicon creation has demonstrated good performance outcomes.

3. Extending a small lexicon using bootstrapping techniques: A limited lexicon of adjectives was to be expanded by Hazivasiloglou and McKeown by adding additional adjectives that were connected with the terms from the previous lexicon (Hatzivassiloglou & McKeown, 1997). The method is based on the syntactic connection between two adjectives linked with "AND," and it has been determined that "AND" typically connects words with the same semantic orientation. For instance: "Yesterday's weather was lovely and inspiring. Since "AND" joins the words "lovely" and "inspiring," it is assumed that both of them have a positive connotation. The word "inspiring" would be added to the dictionary if the only word in it was "lovely."

Here's the lexicon-based method's algorithm in general described based on Figure 2.2 below:

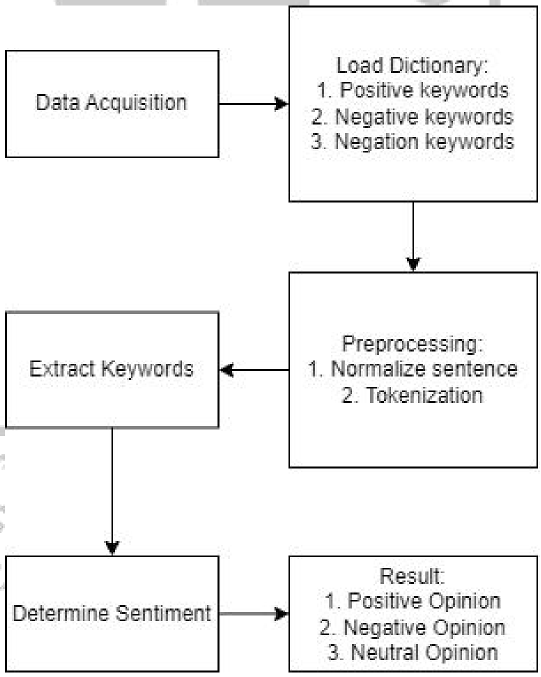


Figure 2.2 Lexicon-based Method Algorithm Flow

The algorithm illustration above in Figure 2.2 is explained below according to Nurfalah et al. (2017):

1. Data acquisition: Data collecting from social media in Indonesian was done during this phase. The procedure produces a collection of opinions together with metadata like username and time.

2. Load dictionary: The dictionary has been loaded in this process. Positive keywords, negative keywords, emoticon dictionaries, and language slang/joke dictionaries are among the dictionaries that are used.
4. Preprocessing: The purpose of this technique is to have sentences ready before performing keyword extraction and sentiment analysis. This technique involves tokenizing and normalizing sentences.
5. Extraction of Keywords: This procedure seeks to extract both positive and negative sentiments for the determiner keyword.
6. Determine sentiments: This procedure seeks to ascertain a sentence's sentiment or opinion’s sentiment. By assessing the likelihood of positive keywords and word negative keywords appearing, sentiment is determined.

Keyword and Emoticon Extraction Process Flow as shown in Figure 2.3 below:

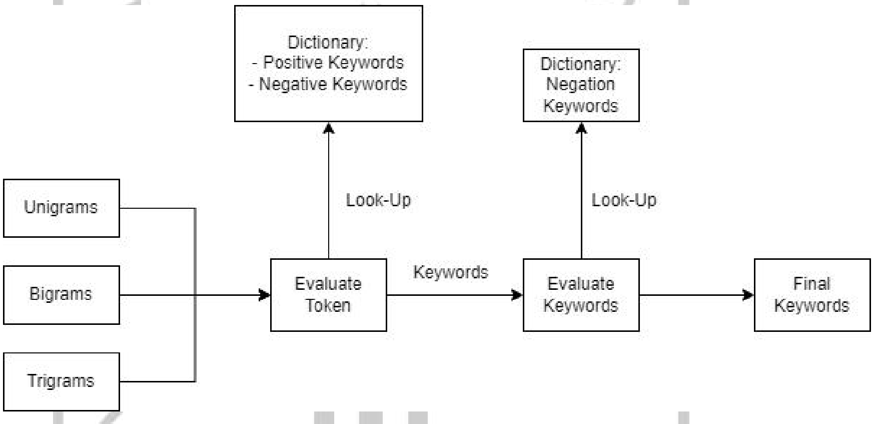


Figure 2.3 Keywords Extraction Flow

After the sentence is normalized, then the sentence is broken down into tokens using a delimiter. There are 3 types of tokens, namely (Nurfalah et al., 2017):

1. Unigrams: i.e. tokens consisting of just one word, for example: “lambat”
2. Biggrams: i.e. tokens consisting of two words, for example: “koneksi lambat”
3. Trigrams: namely tokens consisting of three words, for example: “koneksi smartfren lambat”

After forming the unigram, bigram, and trigram, then the keywords are extracted from the sentences using the three types of tokens and matched with a dictionary to get positive, negative, and neutral keywords.

According to Kolchyna et al. (2015), the ultimate sentiment score of the text is determined by dividing the sum of the scores of words caring the sentiment by the total number of such words, following the assignment of polarity scores to each word making up the text, described in the equation below:

$$Score_{AVG} = \frac{1}{m}\sum_{i=1}^{m} W_i.$$

The sentiment score can be calculated by averaging the results, which yield a value between -1 and 1, where 1 denotes a strongly positive sentiment, -1, a strongly negative attitude, and 0 denotes a very neutral sentiment. For instance, in the text:

"As [0.0] an [0.0] operator [0.0] which [0.0] have [0.0] a [0.0] good [+0.80] price Smartfren [0.0] has [0.0] a [0.0] disappointing [-1] signal and bad [-0.80] coverage [0.0]"

The sentiment would be calculated below:

$$Score\ AVG = \frac{0.0 + 0.0 + 0.0 + 0.0 + 0.0 + 0.0 + 0.80 + 0.0 + 0.0 + 0.0 - 1 - 0.80 + 0.0}{13} = 0.077$$

The sentiment score of 0.077 indicates that the sentence has a positive opinion. However, in the real scenario, all 13 words inside of the sentence above would be eliminated first through stopwords elimination to only calculate the words' sentiment score with significant sentiment meaning, before the words' sentiment is calculated. Therefore, the actual number of calculated words' sentiments would be less than that in the real case, which would make the average score even higher than 0.077.

However, there are several methods of lexicon sentiment calculation existed such as VADER, SentiWordNet, SentiStrength, Liu and Hu opinion lexicon, AFINN, etc (Al-Shabi, 2020). In this research, VADER will be used as the lexicon calculation method due to its higher accuracy among those mentioned methods. VADER is a lexicon and rule-based method, and it has been adjusted for analyzing the sentiments expressed on social media. This method was developed by Hutto and Gilbert (2014) to address the issue of assessing language, symbols, and writing style in the context of social media texts. ability to determine the polarity (positive, neutral, or negative) of the emotions present in the text in addition to their intensity. The lexicon is available as open source in Python code thanks to the authors. Three pre-built lexicons which are Linguistic Inquiry and Word Count (LIWC), Affective Norms for English Words (ANEW), and General

Inquirer, were examined and their features were chosen to create it. Along with commonly used acronyms in social media like WTF and LOL, there are also slang terms and facial expressions like: (to show a happy expression). It has been determined that 7,500 features belong in the vocabulary. Table 2.1 shows the detail of the calculation using VADER:

Table 2.1 VADER Classification

Sentence	Positive	Compound	Neutral	Negative
The bike was good	0.492	0.4404	0.508	0.0
The person is not smart nor funny	0.0	-0.7424	0.354	0.646
Today is kind of boring! But I'll get by	0.317	0.5249	0.556	0.127

The compound score was calculated by adding the equivalent values for each word in the lexicon and adjusted according to the rules to be between -1 (most negative) and +1 (most positive). The pos, neg, and neu values give the percentage of text occurring in each row. They also set a standard threshold for classifying sentences, either negative, neutral, or positive, as follows:

1. Positive: compound score >= 0.05
2. Neutral: compound score >-0.05 and compound score <0.05
3. Negative: compound score <= -0.05

2.5 Classification Algorithm using Support Vector Machine (SVM)

In order to forecast the class of an item whose class is unknown, classification is the process of identifying a model or function that represents and distinguishes class data (Han et al., 2012). A support vector machine is a set of supervised learning methods that analyze data and recognize patterns (Cortes & Vapnik, 1996), used for classification and regression analysis. The original SVM algorithm was created by Vladimir Vapnik and the current standard derivative (soft margin) was proposed by Clorinna Cortes (1995). The SVM standard takes a set of input data and predicts for each given input, the probability that the input is a member of one of the two classes, which makes SVM a binary linear non-probabilistic classifier. Since SVM is a classifier, it is then assigned a training set, each marked as belonging to one of two categories.

While the Support Vector Machine (SVM) according to Vapnik (1992) was first presented in 1996 at the Annual Workshop on Computational Learning Theory. The basic concept of SVM is actually a harmonious combination of computational theories that have existed decades before, such as the hyperplane margin, kernel, and other supporting concepts. However, until 1996, there had never been an attempt to assemble these components.

In contrast to the neural network strategy that seeks to find a dividing hyperplane between classes, SVM works on the principle of Structural Risk Minimization (SRM) intending to find the best hyperplane that separates two classes in the input space. The basic principle of SVM is linear classification, and was further developed to work on non-linear problems, by incorporating the concept of kernel tricks in high-dimensional workspaces, as illustrated in Figure 2.4 below:

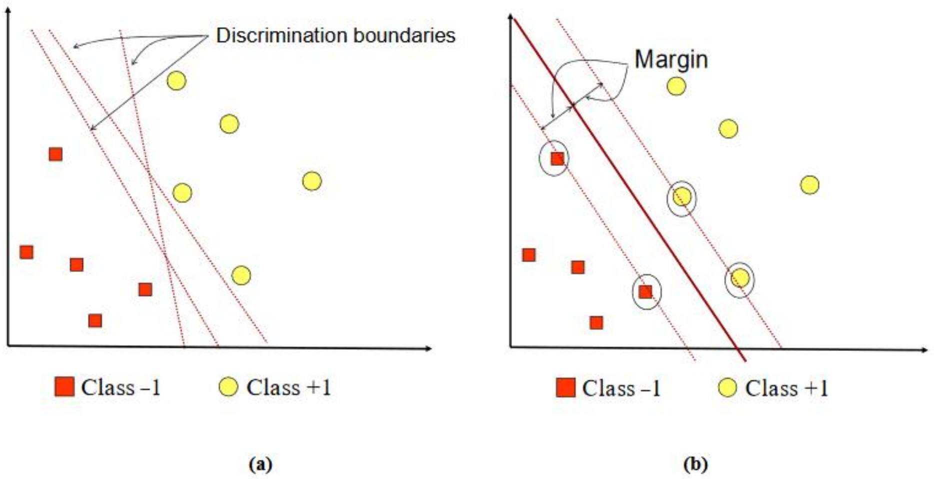


Figure 2.4 SVM Tries to Find The Best Hyperplane to Separate Classes

Figure 2.4 shows how the basic concept of SVM., the spread of data is indicated by red (box) and yellow (circle). Red data is a member of class -1 and yellow data is a member of class +1. The main problem of classification is finding the dividing hyperplane between the two classes. From Figure 3a, it can be seen that there are many alternative discriminatory boundaries between the two classes.

The best-separating hyperplane between the two classes is obtained by measuring the margin of the hyperplane and finding the largest margin. Margin is the distance between the hyperplane and the closest data from each class. The data closest to the hyperplane is called the support vector. The solid line in Figure 3b shows the best hyperplane, which is located right in the middle of the two classes, while the red and yellow dots in the black circle are support vectors. Efforts to find the location of this hyperplane are the core of the learning process in SVM.

2.6 Evaluation Parameter

The measurement parameters are used to evaluate the performance of the model that has been made. Measurements to evaluate performance are not only seen from how fast (time) the system builds a model or performs a classification but also look at the model's ability to make predictions (the system predicts/classifies correctly or not). The Text Classification method can be evaluated using accuracy, precision, recall, and F-measure (Han et al., 2012).

In the world of pattern recognition and information retrieval, precision and recall are calculations that are widely used to measure the performance of the system or method used. Precision is the level of accuracy between the information requested by the user and the answer given by the system. Recall is a presentation of the success of the system in retrieving information. While accuracy is defined as the level of closeness between the predicted value and the actual value (Han et al., 2012, p. 368).

F-measure is one of the evaluation calculations in information retrieval that combines precision and recall. Precision and recall values in a situation can have different weights. The measure that displays the reciprocity between precision and recall is the F-measure which is the average weight of precision and recall. In general, precision, recall, accuracy, and f-measure can be formulated as explained in Table 2.2 below:

Table 2.2 Confusion Matrix Table

		Actual value	
		TRUE	**FALSE**
Prediction value	**TRUE**	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FALSE	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

In addition, the measurement of the test evaluation can be seen visually by using the ROC curve. ROC is a visual curve that is very useful for comparing two classification models (Han et al., 2012, p. 374). The ROC curve is used to determine the characteristics of the classifier, by changing the adjustable parameters of the classifier so that it creates a lot of confusion table. of many confusion tables, then we can take the value of TP and FP from the table The reason ROC only takes TP and FP values is because there are several cases of a classifier created to guess class 1 as the correct class so that we can analyze directly the prediction process in class 1 only. Meanwhile, the plotted values are the TPR (True Positive Rate) and FPR (False Positive Rate). TPR is the proportion of positive tuples (or 'yes') labeled correctly by the model, whereas FPR is the proportion of negative (or 'no') tuples labeled as positive (Han et al., 2012, pp. 374-375). Each value can be calculated using the equation as follows:

$$TPR \frac{TP}{TP + FP}$$

$$FPR \frac{FP}{TP + FP}$$

2.7 Feature Extraction

In sentiment analysis, we need to convert features into numbers so that they may be used by a machine-learning classifier. The accurate presentation of tweets is required for machine learning classification methods, with each tweet converted into a feature vector that only contains unique terms. The fact that the classifier uses the feature vector as an input suggests that a strong feature vector would produce better classification outcomes.

CountVectorizer, Bag of Words (BoW), and TF-IDF (term frequency-inverse document frequency) are just a few of the algorithms that can be used for this.

b. CountVectorizer (Deepa et al., 2019): It is a simple vectorizer that converts each token in our data into a feature. It becomes a matrix and an array of features when taken as a whole. Each text in the document is taken into account, together with the number of times each word appears inside. The classifier can then receive this multiset of words as input. It is known as a sparse dataset because each file contains numerous zeros for each text that does not appear in the dataset.

c. Bag of Words (BoW) (Akuma et al., 2022): In order to be employed in modeling, such as in machine learning models, this technique removes features from textual expressions. It is said that a document is a "bag" of words because all indications as to the order or structure of words have been eliminated. The location of known phrases is insignificant to this model; it only cares about whether they appear or not in a document. The objective is to convert every document into a vector that may be used as input or as output from a machine learning model. Assigning a Boolean value of 0 or 1 to the presence of words, with 0 indicating absence and 1 indicating presence, is the simplest scoring method.

d. Term Frequency-Inverse Document Frequency (TF-IDF) (Deepa et al., 2019): In this type of feature extraction, words are vectorized by taking into consideration how frequently they appear in the given document. One issue with CountVectorizer is that there might be text that appears many times in the observations of the target class. This text can be removed so that it does not convey biased information about the data. By utilizing the following formula, TF-IDF is utilized to enhance the performance of the frequent words found in the document.

TF = Number of time the term occurs in the text/total number of words in the text.

IDF = Total amount of documents/number of documents with words inside of it

TF-IDF = TF*IDF

2.8 Post Engagement Rate (PER)

The concept of client engagement is currently a hot topic in the marketing literature. There is a movement in the view of a product-oriented organization to a customer-oriented organization in today's period of rising e-commerce, which is characterized by rapid internet users and social media in the corporate world. It causes the consumer connection to become a top priority. Customer engagement also takes on a new meaning in terms of maintaining customer relationships. Verhoef et al. and Kabadayi et al. (2014), stressed the relevance of a corporate awareness of consumer participation in social media as a way for businesses to learn about their customers' values.

The engagement rate is derived by dividing the number of likes, comments, and shares on a post by the total number of page fans, according to SocialBakers. The average engagement rate of all posts during a specified time period is used to compute a page's engagement rate.

$$Post\ Engagement\ Rate = \frac{likes + comments + shares}{total\ fans\ on\ a\ given\ day} \times 100\%$$

According to past research conducted, the PER could be the metric that measures how much attention the users pay to each post (tweet) because engagement rate takes into account the number of interactions compared to the total amount of followers to understand how many people are engaged with the information posted. Considering the sentiment of each post (tweet) can be analyzed, therefore, the engagement rate could also be incorporated with the sentiment of each tweet. Tweets that have positive or negative sentiments could influence other users with the engagement rate that the tweets have.

2.9 Twitter

A fresh and intriguing information source for handling data is microblogging (Bifet & Frank, 2010). Twitter is a social network or microblog that enables users to send and read 140-character tweets, which are text-based messages. Jack Dorsey launched Twitter and his social networking service in March 2006. Beginning in July 2006, Twitter has developed into a popular tool for many groups to immediately track the preferences of people for various situations. Millions of people could utilize this as a source of data. Everything on Twitter is made public as a data stream, which can be used to your advantage if you employ stream mining methods. This, in theory, alerts us to overall public sentiment.

Twitter, also known as "brief communications from the Internet," has grown to become one of the ten most frequently viewed websites online since its launch (eBizMBA, 2014). Unregistered users on Twitter can only read tweets, whereas registered users can post tweets via the web interface, SMS, or certain mobile applications. Twitter grew swiftly and became well-known all over the world. 200 million of the more than 500 million registered users on Twitter were active users as of January 2013. The seasonal increase in Twitter users typically coincides with a particular occasion or season. Due to its immense popularity, Twitter is utilized for a wide range of activities, including political campaigns, commercial promotion, educational institutions, and numerous emergency communication channels. Twitter is also dealing with a number of challenges and controversies, including censorship, lawsuits, and concerns about user security and privacy.

2.10 Social Media Marketing

Social media marketing is an online marketing strategy that involves increasing a website's visibility, existence, and presence on social media networks like Facebook, Twitter, Digg, Web 2.0, and others (PT. Amitra Visinet Indonesia, 2014). Social media is a highly effective medium for customer feedback and for engaging with customers. Social networking websites are crucial for boosting your website's search engine rating and bringing in a ton of high-quality visitors to the business website (Evans & Mckee, 2010).

Social media marketing initiatives typically center on producing content (posts, writings, images, and videos) that will catch readers' interest and persuade them to share it with their social networks. The marketing division of the business or the public relations department of the agency frequently utilizes social media to produce text, images, videos, graphics, or postings on the agency's social media accounts to advertise goods and services. In order to generate electronic Word of Mouth (eWoM) or be discussed by users of social media, it is intended that the information will be able to draw attention, be liked, and be shared as widely as possible. A favorable reputation and image among customers or business partners are the ultimate goals.

2.11 The Eight Aspects of the Dimensions of Quality

An important consideration when choosing a product is quality. The level of good or terrible items a company produces can be determined by the quality of the company. David A. Gavin introduced eight quality dimensions in 1987 in an effort to address the subjectivity and ambiguity that can arise when assessing the quality of a product. These eight dimensions are performance, durability, features, conformance, serviceability, reliability, perceived quality, and aesthetics. Hermansyah and Sarno (2021) also used these quality dimensions to further analyze the Twitter sentiment analysis result of a telecommunication company. Tweet texts were categorized into eight quality dimensions, using keywords that represent each category. The keywords for each quality dimension are derived below:

Table 2.3 Aspects of Quality Dimension

Aspect of Quality Dimension	Term Variables
Performance	‘lemot’, ‘lambat’, ‘lama’, ‘cepat’, ‘lancar’, ‘normal’, ‘hilang’, ‘unduh’, ‘unggah’
Features	‘wifi’, ‘game’, ‘telkomsel’, ‘netflix’, ‘poin’, ‘telepon’, ‘zoom’, ‘instagram’, ‘drakor’
Reliability	‘putus’, ‘patah’, ‘kedip’, ‘stabil’, ‘ganggu’
Conformance	‘tagihan’, ‘mahal’, ‘murah’, ‘bohong’, ‘jujur’, ‘aman’
Durability	‘rusak’, ‘awet’, ‘tahan’, ‘usia’, ‘waktu’
Serviceability	‘professional’, ‘ramah’, ‘respon’, ‘komunikasi’, ‘solusi’, ‘perilaku’
Aesthetics	‘kotor’, ’bersih’, ‘rapih’, ‘indah’, ‘buruk’
Perceived Quality	‘puas’, ‘senang’, ‘males’, ‘emosi’, ‘bosan’, ‘bahagia’, ‘rekomendasi’, ‘juara’, ‘henti’

2.12 Mobile Network Experience

In 2022, Opensignal was able to analyze consumer mobile experience to give an understanding of the true experience consumers receive on wireless networks. Opensignal categorized the mobile network experience into 6 experience categories, which are video experience, games experience, voice app experience, download speed experience, upload speed experience, and coverage experience. All of the experience aspects by Opensignal measure all mobile network experiences through an operator's networks to determine the level of video quality being transmitted to mobile devices. The statistic is based on an International Telecommunication Union (ITU) methodology and is based on in-depth research that has found a connection between technical factors of each experience category and the reported perceived experience by actual users.

2.13 Previous Research

Here are several previous research that become the foundation of the methodologies used in this research:

1. The previous research entitled “Sentiment Analysis of Customer Engagement on Social Media in Transport Online” (Saragih & Girsang, 2017) , aimed to discover the most discussed topic and its sentiment polarity from customer feedback in social media (Facebook and Twitter). This research used machine learning and TF-IDF methods. Data mining using API, Data Preprocessing (case folding, converting emoticons, stemming, removing stop words, tokenization), Sentiment categorization, Comment categorization, Analysis, and Comparison. The research focused on customer engagement sentiment analysis by analyzing comments on the Facebook fan pages of 3 transport online in Indonesia, namely Gojek, Grab, and Uber using the API service provided. The total amount of 1000 comments’ sentiments was tagged into positive, negative, and neutral by using the TF-IDF method with an accuracy of 80.1%. The result shows that there’s only a few users complain about the driver via social media. A further suggestion is delivered that the next research should be conducted using a machine learning algorithm for better accuracy. From this research, the researcher found out that sentiment analysis can be conducted to analyze further into the categorization of commentaries.
2. Next, the research entitled “Twitter Sentiment to Analyze Net Brand Reputation of Mobile Phone Providers” by Vidya et al. (2015). The research tries to measure customer loyalty in Indonesian Telco companies using Net Brand Reputation which calculates the percentage of positive mentions minus the percentage of negative mentions on Twitter. The method used is Net Brand Reputation, Sentiment Classification of Naïve Bayes, Support Vector Machine (SVM), Decision Tree, and Classifier Evaluation. The steps are (1) data description and mining, (2) Data Preprocessing, (3) Sentiment Classification (using naive bayes, support vector machine, and decision tree, (4) Classifier Evaluation (using Receiver Operating Characteristic and Area Under Curve), and (5) Measuring Brand Reputation. This research analyzed the brand reputation by analyzing customer satisfaction through sentiment analysis. It uses 10,000 raw Twitter messages data. The sentiment classification algorithm was evaluated and shows that the Support Vector Machine algorithm has a good classification with an AUC score of 0.854. The best F1-score

result was coming from the SVM algorithm with the value of 89.33% The NBR calculation was divided into 5 product categorizations (4G, 3G, Voice, SMS, and Data). It shows that XL Axiata has a better reputation in five products with an NBR score of 32.3%. It gives the researcher an insight to compare the result of the classification algorithm, to get the best accuracy of the model. The tweet texts could also be classified into categorizations based on telecommunication context, to better understand the polarity for each category of telecommunication topics. From the comparison of 3 machine learning algorithms, the researchers have proven that SVM can give the most accurate and precise result.

3. The research namely "Enterprise Competitive Analysis and Consumer Sentiments on Social Media Insights from Telecommunication Companies" by Afful-Dadzie et al. (2014) aimed to use opinion mining to do a competitive analysis using unstructured textual information on the Facebook and Twitter sites of the top 3 telecommunications company in Ghana. The method used is Data Mining and Lexicon-based Sentiment Analysis. The findings demonstrate (1) the rapid increase in social media users in Ghana (2) the significance and scale of active social media usage in the telecommunications sector (3) and the efficacy of social media opinion mining for competitive analysis. (4) How to derive corporate value from the vast amounts of unstructured textual data present on social media, and (5) The firm that responds to consumer issues the best. It has shown that sentiment analysis can be used to compare how well each of the telecommunication company handles customer issues.
4. A research created by Ranjan et al. (2018) entitled "Twitter Sentiment Analysis of Real-time Customer Experience Feedback for Predicting Growth of Indian Telecom Companies" aimed to predict the growth of telecommunication companies in India in terms of subscriber addition, using sentiment analysis. Several methods are used such as Data Mining, Semantic Analysis, and Correlation Analysis. The researchers also used the prediction of subscriber addition and validation method. This research analyzed the growth of telecommunication companies in India using 153,651 unique tweets from 5 telecommunication brands in India. Then, the tweets' sentiments are determined using the sentiment score. The subscriber addition prediction was conducted using the sentiment score. The validation results are within the significance limits of the prediction model. Therefore, the actual growth rate and predicted growth rate indicate a high degree of correlation validating the prediction

model. The addition of a sizable number of new subscribers to a business' subscriber base might be viewed as customer churn, where subscribers leave other businesses that didn't keep an eye on customer sentiment and take prompt action to stop it.

5. The research entitled “Sentiment Analysis of Customer Response of Telecommunication Operator in Twitter using DCNN-SVM Algorithm” created by Firdausi et al. (2020) aimed to know the classify sentiment of customer responses on Twitter using Deep Convolutional Neural Network (DCNN) and Word2Vec as a feature extraction and Support Vector Machine (SVM) as its classification. There are 4 Indonesian telecommunication operators’ tweets data taken. This research was able to prove that the combined machine-learning-based feature extraction and Support Vector Machine were able to give enough accuracy. The utilization of using 2 different feature extractions could be conducted to find the best result. The best performance results are 63% accuracy, 63% precision, and 50% recall of test data.
6. A research conducted by Kolchyna et al. (2015) entitled “Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method, and Their Combination” presents a comparative study of different lexicon combinations and machine learning to do a sentiment classification. This research provides a review of two primary approaches to sentiment analysis, which are a lexicon-based method and a machine-learning method. The findings demonstrate that the lexical method is outperformed by a machine learning technique based on SVM and Naive Bayes classifiers. The researchers provide a brand-new ensemble methodology that applies machine learning to a lexicon-based emotion score as an input feature. It turned out that the combination approach produced more accurate classifications.
7. Research entitled “Lexicon Addition Effect on Lexicon-Based of Indonesian Sentiment Analysis on Twitter” created by Saputra et al. (2020) tries to prove the lexicon-based sentiment improvement from the lexicon words added to the sentiment lexicon. This research provides a review of two primary approaches to sentiment analysis, which are a lexicon-based method and a machine-learning method. The findings demonstrate that the lexical method is outperformed by a machine learning technique based on SVM and Naive Bayes classifiers. The researchers provide a brand-new ensemble methodology that applies machine learning to a lexicon-based emotion score as an input feature. It turned out that the combination approach produced more accurate classifications.

8. Chorana and Cherroun (2021) created research entitled "User-Generated Content and Engagement Analysis in Social Media Case of Algerian Brands". The research used statistical analysis for the reaction-based assessment to measure Users/Brand Engagement Rates (ER) while considering the reactions and content. The importance of the current social signals and user-generated content in the Algerian environment might be stressed based on the findings. Latent Dirichlet Allocation (LDA) algorithm method is used to automatically categorize the topics of the content and determine the most frequent topic discussed. After that, the Engagement Rate of each content is calculated as well to further analyze the engagement level of the content. Based on statistical and linguistic analysis of user/brand-generated content on social media, the researchers offered an analytical study. Utilizing the standard User engagement formulas from the literature for reaction cases is one of two methods used to investigate the extent of User/Brand engagement. Therefore, an LDA Topic Modeling-based strategy for content-based user engagement is suggested. The results emphasize the social signals' quantitative and qualitative significance in the context of productivity in Algeria. This effectively aids brand owners in enhancing their online marketing strategy and efficiency. In the future, the researchers want to build a more accurate topic modeling for measuring users' engagement and normalize the entire content of the data set by automatically detecting the language and translating it to Modern Standard Arabic.
9. Research entitled "Engagement Analysis of Canadian Public Health and News Media Facebook Posts and Sentiment Analysis of Corresponding Comments during COVID-19" created by MacKay et al. (2022) aimed to prove several things. First, it has an objective to classify the engagement rate of Facebook posts over time. The second, is to evaluate the percentage of negative sentiment in comments over time. The third, is to evaluate the potential impact of trends in the percentage of trinary sentiment (positive, neutral, or negative emotional response) on the overall number of comments per post. The study looked at the sentiment of the Facebook comments and the engagement rates of the postings made by official actors during COVID-19. The posts and associating comments from the Healthy Canadians, CTV News, and CBC News pages were manually compiled using Facebook's advanced search feature. In comparison to public health, news media received a higher percentage of negative comments and had a lower total average post engagement. Public health postings had more likes, comments, and shares than those from the news media due to their

above-average engagement post rate. Even while the overall comments had no impact on the total number of comments for each post, it is still important to consider the bad impact that can have on other people's beliefs and actions.

10. Research by Abdillah et al. (2021) namely “Analisis Sentimen Penanganan Covid-19 dengan Support Vector Machine: Evaluasi Leksikon dan Metode Ekstraksi Fitur” has an objective to determine the performance of Indonesian lexicon dictionaries, which are InSet and sentistrength_id, towards the actual sentiment label. InSet lexical dictionary that the researcher used consists of 10,218 words in total. The data used has a total of 1,912 social media comments. The research used Support Vector Machine (SVM) for the classification algorithm and then, to determine the effect of selecting term presence, BoW, and TF-IDF feature extraction methods. InSet and Sentistrength_id were chosen because these are the only Indonesian lexicon dictionaries, which have research publications. The research process begins with the pre-processing of the text, selecting the feature extraction method, labeling words with the lexicon, and classification of sentiments with the Support Vector Machine (SVM). The final evaluation was carried out by k-Fold cross-validation using data synthesized by the SMOTE algorithm. Preliminary testing shows that the sentistrength_id lexicon word sentiment dictionary produces a slightly higher accuracy score (64.46%) than InSet (62.65%) when combined with TF-IDF. In the final evaluation stage, the sentistrength_id word sentiment dictionary still shows relatively better performance based on the average values of accuracy, precision, sensitivity, and f-measure (59.22%, 61.1%, 31.3%, 41.0%) compared to InSet (58.98%, 62.2%, 26.1%, 36.6%) when combined with TF-IDF. In general, the performance of the two lexicons is still below the data with actual labels which indicates that the two lexicons are not strong enough in determining word classes. The reason could be due to the relatively small amount of data or because text normalization has not been carried out optimally.
11. A study entitled “Peringkasan Sentimen Ekstraktif di Twitter Menggunakan Hybrid TF-IDF dan Consine Similarity” conducted by Wahid and Azhari (2016) aimed to combine SentiStrength, Hybrid TF-IDF and Cosine Similarity methods to automatically extract summaries of positive and negative public sentiments towards celebrity topics on Twitter automatically, with the artist Agnes Monica as a case study. The SentiStrength method is used to get a sentiment strength score and classify tweets into positive, negative, and neutral classes. Tweets with positive and negative

sentiments are summarized by ranking tweets using Hybrid TF-IDF combined with a sentiment strength score, then removing similar tweets using Cosine Similarity. The sentiment dictionary contains a collection of words that have been weighted with a sentiment strength of 1 (no positive sentiment) to 5 (has a very strong positive sentiment), and -1 (no negative sentiment) to -5 (has a very negative sentiment). The sentiment dictionary is obtained from the translation of the English sentiment dictionary which has experienced addition and subtraction of words based on observations in the process of developing this system. This study involved 3 expert respondents, Master of Linguistics students at Gadjah Mada University, to give weight to the strength of sentiment. The median value of the three respondents is used as the final weight for sentiment strength. The test results show that the combination of SentiStrength, Hybrid TF-IDF, and Cosine Similarity is able to produce sentiment summaries with better accuracy than using Hybrid TF-IDF alone, with an average accuracy of 60% and an f-measure of 62%. This is due to the addition of sentiment strength as a summary weight.

12. The last study entitled “Evaluating the Performance of the Most important Lexicons Used to Sentiment Analysis and Opinions Mining” by Al-Shabi (2020) aimed to assess the performance of several lexicon-based methods in Twitter polarity classification. This study uses lexicon-based sentiment analysis as its methodology, focusing on five of the most significant and well-known lexicons used in the field on Twitter data, including (VADER, SentiWordNet, SentiStrength, Liu and Hu opinion lexicon, and AFINN. It provides an assessment of the performance of these lexicons in Twitter polarity classification by comparing the overall classification accuracy and the F1-measure. It was proven that VADER has good accuracy.

2.14 Theoretical Framework

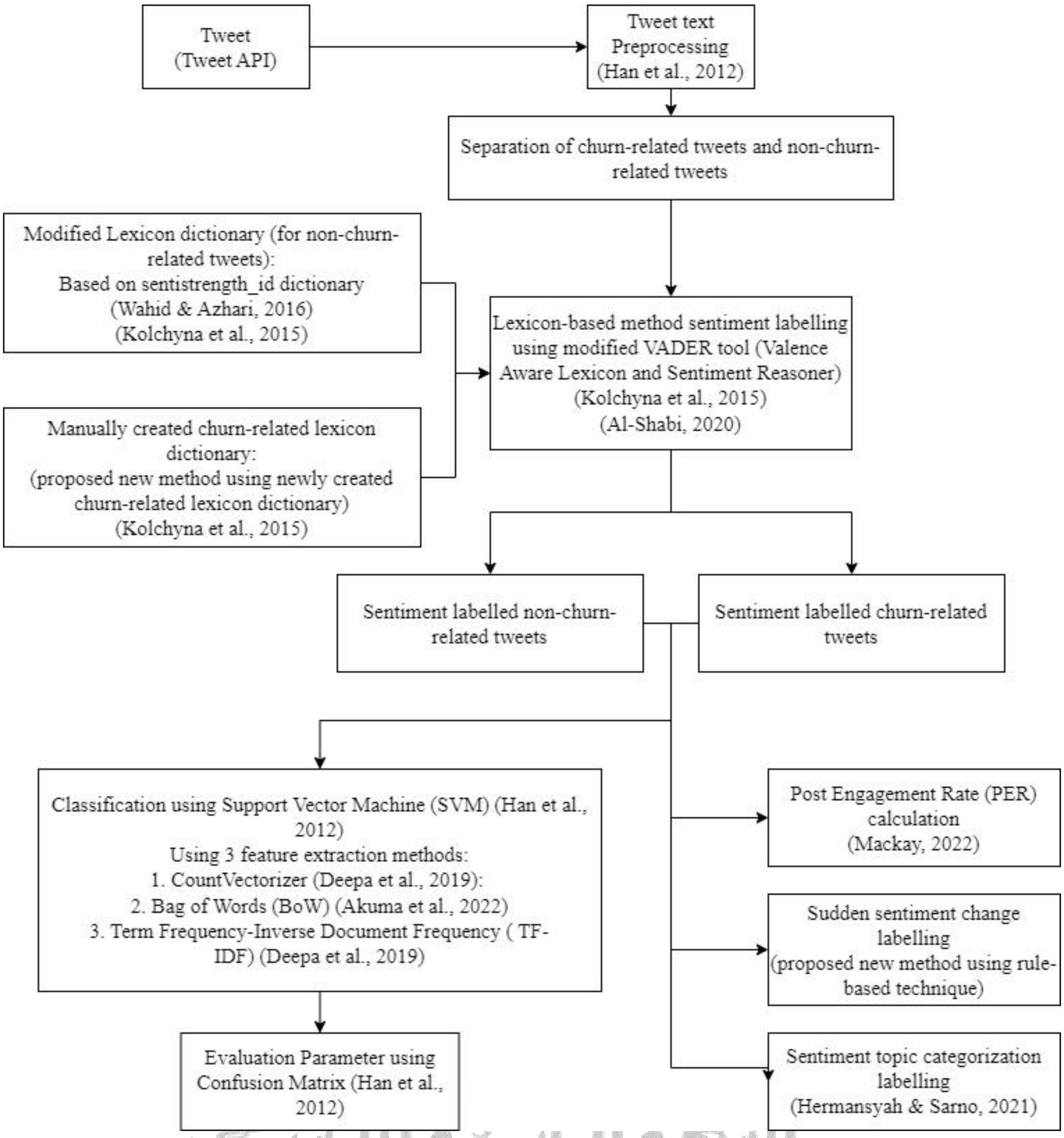

Tweet
(Tweet API)
Tweet text
Preprocessing
(Han et al., 2012)
Separation of churn-related tweets and non-churn-related tweets
Modified Lexicon dictionary (for non-churn-related tweets):
Based on sentistrength_id dictionary
(Wahid & Azhari, 2016)
(Kolchyna et al., 2015)
Manually created churn-related lexicon dictionary:
(proposed new method using newly created churn-related lexicon dictionary)
(Kolchyna et al., 2015)
Lexicon-based method sentiment labelling using modified VADER tool (Valence Aware Lexicon and Sentiment Reasoner)
(Kolchyna et al., 2015)
(Al-Shabi, 2020)
Sentiment labelled non-churn-related tweets
Sentiment labelled churn-related tweets
Classification using Support Vector Machine (SVM) (Han et al., 2012)
Using 3 feature extraction methods:
1. CountVectorizer (Deepa et al., 2019):
2. Bag of Words (BoW) (Akuma et al., 2022)
3. Term Frequency-Inverse Document Frequency (TF-IDF) (Deepa et al., 2019)
Evaluation Parameter using Confusion Matrix (Han et al., 2012)
Post Engagement Rate (PER) calculation
(Mackay, 2022)
Sudden sentiment change labelling
(proposed new method using rule-based technique)
Sentiment topic categorization labelling
(Hermansyah & Sarno, 2021)

Figure 2.5 Theoretical Framework

The figure above summarized the theoretical framework of this research. In the case study of customer experience and the impact towards customer perception using Twitter sentiment analysis of PT Smartfren Telecom, the research begins with data preprocessing which is carried out in three stages, namely the first data gathering (Twitter crawling) which aims to retrieve the sentences contained in the tweet, as well as the other variables of the tweets itself such as likes amount, retweet amount, etc. Twitter API is used to collect the tweets data.

The second is data preprocessing which consists of doing buzzers and promotion removal to remove tweets that are not relevant to this research, the third is eliminating stop words and slang word conversions to remove unimportant information and converting informal words into formal words in a tweet. This stage is necessary to discard data noise (text noise) and irrelevant data, which would be useful for sentiment labeling and classification (Han et al., 2012).

After the data has been cleaned, sentiment labeling for each tweet is conducted using a lexicon-based method. This method is chosen because, among all sentiment analysis methods available explained by Kolchyna et al. (2015), the lexicon-based method does not need as much data and resources to be used for the labeling process (Abdillah et al., 2021). The pure machine learning method needs a sufficiently bigger amount of data to classify the sentiment, in accordance with a big number of tweets data that we will be used. In this research, the lexicon calculation that we will use is VADER (Valence Aware Lexicon and Sentiment Reasoner). This lexicon calculation method is chosen due to its higher accuracy and precision, compared to other methods (SentiWordNet, Sentistrength, and Liu and Hu opinion lexicon) according to the latest research conducted by Al-Shabi et al. (2020). VADER has a decent possibility for the sentiment classification of short texts pre-process, positively and negatively, and neutral, where it can deal with all text cases.

Since the tweets are in the Indonesian language, we will use a particular Indonesian lexicon dictionary. Currently, there are only 2 lexicon dictionaries, which have been proven through research publications, which are InSet and Sentistrength_id (Abdillah et al., 2021). Based on the research conducted by Abdillah et al. (2021) about the evaluation of these 2 Indonesian lexicon dictionaries, it was found that Sentistrength_id has a higher accuracy (64.46%) compared to InSet (62.65%). Therefore, we would use the Sentistrength_id as the foundation of the lexicon dictionary. Although in this research, Sentistrength_id dictionary would also be modified to adjust the lexicon dictionary in the context of telecommunication company sentiment. This lexicon dictionary modification has been justified by Kolchyna et al. (2015) in the research which explains the methods of creating a lexicon dictionary, to improve the performance of lexicon-based sentiment labeling. During the lexicon modification, we will split the lexicon into non-churn-related and churn-related lexicon dictionaries. It's an additional proposed method, to differentiate tweet labeling context which is related to customer churn and not related to customer churn, based on the text of the tweets.

After the tweets have been labeled, Support Vector Machine (SVM) classification algorithm will be used to train and test the labeling result. The classification method is used to be analyzed for evaluation after the features (tweet text) have been labeled (Han et al., 2012). This step was also conducted by Abdillah et al. (2021) after the tweets have been labeled by the lexicon dictionary. The support vector machine method is considered suitable because it can minimize errors that will occur in the classification sentiment that will be carried out. Also, this method is also considered simple, and fast, but accurate (Vidya et al., 2015). In Vidya et al. (2015), SVM has resulted in a higher classification accuracy compared to the other machine learning algorithm for sentiment analysis, such as Naïve Bayes and Decision Tree. Several feature extraction methods are used for the classification. We will use Count Vectorizer (Deepa et al., 2019), Bag of Words (BoW) (Akuma et al., 2022), and TF-IDF (Deepa et al., 2019) for converting words into features for classification. The classification results from the 3 word embedding methods will be evaluated and compared. The evaluation will be carried out using the K-fold method, then the output values in the form of accuracy, precision, recall, and f-measure will be compared (Han et al., 2012).

Next, after the evaluation results have shown that the sentiment labeling is accurate, the results of the sentiment label will be used for the Post Engagement Rate calculation (Mackay, 2022), topic categorization (Hermansyah & Sarno, 2021), and sudden sentiment change labeling process (proposed additional method). Post Engagement Rate (PER) is calculated based on the number of likes, comments, and retweets of the tweet post divided by the follower amount of the user who posted the tweet. It is used to measure how engaging/influencing the tweets are to the other users who follow the account. Topic categorization is used to determine which topic the tweets belong to (network, price, etc.). Lastly, sudden sentiment change is used to determine whether the tweet's sentiment is changed compared to the previous tweet of the same user. So, it will tag the tweet as "positive change" if the tweet's sentiment is moving from negative to positive, compared to the previous tweet of the same user, and vice versa. Therefore, in the end, this research would be able to find the relationship between tweet influence (from PER), the impact on other users' sentiment (from sudden sentiment change tagging), the possibility of churn (from churn-related tweet tagging), and what causes the sentiment change or even a churn in a big picture.

To summarize all the methods that the researchers used in this research, the exclusivity of the research is summarized below:

1. Differentiate the sentiment lexicon tagging based on churn-related tweets and non-churn-related tweets, based on the tweets' text content using a modified sentistrength_id dictionary. This proposed method is important because not all tweets only discussing 1 brand only. Therefore, the sentiment lexicon tagging should be distinguished using a modified lexicon dictionary.
2. Able to gain insights from the relationship of tweets engagement level, the impact towards other users' sentiment, the churn possibility based on the tweet text, and the reason behind sentiment change or even churn. These insights would be able to be tracked periodically (daily, monthly, etc.).

   The previous research regarding sentiment analysis has not combined all these metrics to be summarized as an output, which makes the research exclusive. This insight is expected to be relatively more accurate because we will do the tagging for each tweet and summarized the whole thing through data visualization.

CHAPTER III
RESEARCH METHODOLOGY

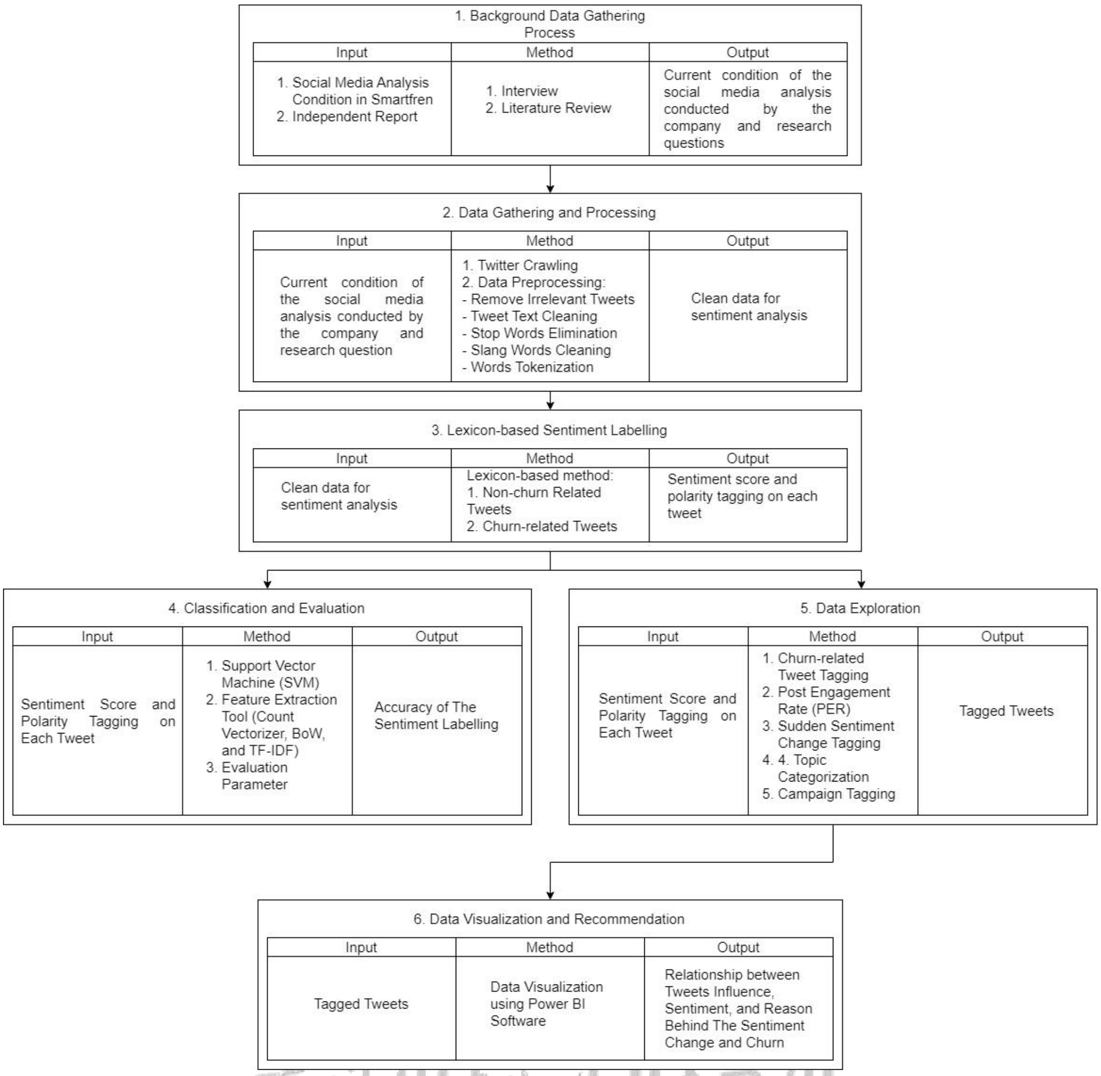
The design and procedures utilized in this investigation are covered in this chapter. A flow chart is used to visualize the multiple connected phases that make up the research design.

3.1 Category of Methodology

The case study research was the strategy used in this study. A case that is assumed to be a problem in one company, PT Smartfren Telecom, Tbk Telecommunications, serves as the basis for this case study research. The research is intensively focused on customer opinions through all Twitter accounts that mention "smartfren" in their tweets, as well as variations of these keywords. The Support Vector Machine (SVM) classification technique will be used to build the classification model. The performance of the algorithm's output model is compared after being quantitatively measured. An assessment of the accuracy was made at the conclusion of the experiment. In this study, a sample from a well-defined population was selected, and a random sampling procedure was used to choose the sample. The likelihood that a phrase will contain the most positive or negative terms determines the classification findings.

3.2 Research Stages

The Knowledge Discovery from Data (KDD) cycle, a data mining activity, was used to design this study. Figure 3.1 below explains the research's progressive framework.



1. Background Data Gathering Process

Input	Method	Output
1. Social Media Analysis Condition in Smartfren 2. Independent Report	1. Interview 2. Literature Review	Current condition of the social media analysis conducted by the company and research questions

2. Data Gathering and Processing

Input	Method	Output
Current condition of the social media analysis conducted by the company and research question	1. Twitter Crawling 2. Data Preprocessing: - Remove Irrelevant Tweets - Tweet Text Cleaning - Stop Words Elimination - Slang Words Cleaning - Words Tokenization	Clean data for sentiment analysis

3. Lexicon-based Sentiment Labelling

Input	Method	Output
Clean data for sentiment analysis	Lexicon-based method: 1. Non-churn Related Tweets 2. Churn-related Tweets	Sentiment score and polarity tagging on each tweet

4. Classification and Evaluation

Input	Method	Output
Sentiment Score and Polarity Tagging on Each Tweet	1. Support Vector Machine (SVM) 2. Feature Extraction Tool (Count Vectorizer, BoW, and TF-IDF) 3. Evaluation Parameter	Accuracy of The Sentiment Labelling

5. Data Exploration

Input	Method	Output
Sentiment Score and Polarity Tagging on Each Tweet	1. Churn-related Tweet Tagging 2. Post Engagement Rate (PER) 3. Sudden Sentiment Change Tagging 4. 4. Topic Categorization 5. Campaign Tagging	Tagged Tweets

6. Data Visualization and Recommendation

Input	Method	Output
Tagged Tweets	Data Visualization using Power BI Software	Relationship between Tweets Influence, Sentiment, and Reason Behind The Sentiment Change and Churn

Figure 3.1 Research Stages

Figure 3.1 above shows the stages carried out in this research. The following explanation below is a description of each stage of the research conducted.

3.2.1 Background Data Gathering Process

This stage was carried out to find the current condition of the company and figure out the main issues related to the customer experience analysis. The interview regarding the current condition of the company was conducted with the data science division in Smartfren. The interview result discovered the insufficient insights that the Brand and Communication department can get from the current social media analysis conducted by Branding Department and Third-party institutions. After the internal problems inside the company were ientified, the problems were validated from the Mobile Network Experience Report in June 2021, conducted by Opensignal, which is an independent global standard for analyzing consumer mobile experience. It was found that Smartfren remained in last place in all experience metrics (video experience, games experience, voice app experience, download speed experience, upload speed experience, and 4G availability) with the exception of 4G Coverage Experience.

3.2.2 Dataset Gathering and Preprocessing Process

According to the social media analysis problem, the next step is to gather the Twitter data from the Twitter API (Twitter scrapping), remove the irrelevant tweets, and clean the tweet's text. Twitter data scrapping is conducted using Python coding language, to specify the period of the tweets that would like to be gathered, the specific variables that would like to be included, as well as the keywords that would specify smartfren-related tweets only. The irrelevant tweets are the tweets that are not representing customers' experience/feedback, such as buzzer tweets and promotion tweets. The tweets text cleaning is used to remove the noise of characters inside of the text, for the sentiment analysis.

a. Twitter Crawling

Utilizing the Application Interface (API) resources offered, the crawling operation on Twitter is carried out, by first signing up for Twitter Developer via the application. In this study, sentiment analysis is restricted to Smartfren company. The URL used to use the Twitter API is https://api.twitter.com/1.1/search/tweets.json. After the account is saved in python, the query's returned data is in plain text format, along with the other variables that the researcher specified. All smartfren-related tweets were gathered from Twitter API, in a particular period of time (January 2019 to September 2021). Smartfren-related keywords were used to filter and pull the tweets data from the Twitter API directly, such as "smartfren", "@smartfrenworld" as smartfren's official account name, and "@smartfrencare" as smartfren's official customer care account name. Therefore, all tweets that talk about Smartfren or mention Smartfren's official account will be gathered all together. Several variables are gathered from the Twitter API, such as tweet text, tweet date, verified user, tweet username, user follower amount, user following amount, a user mentioned, tweet media link, reply amount, retweet amount, and like amount.

b. Data Preprocessing

Because high-quality data models demand high-quality data, data preparation is strongly advised for a variety of reasons, including database quality, the data analysis process, the potential application of related algorithms for removing noisy data and missing data, and increasing data reliability. Sentiment analysis entails locating a given text's subjective contents after preprocessing the text to find stop words, symbols, etc. The content is categorized according to sentiment as either favorable, negative, or neutral. Data cleaning is a crucial data preprocessing tool approach. It is a method involving DM methods. It eliminates poor errors from data and cuts down on extraneous data. Data cleaning strategies also take into account missing data. The details of data preprocessing are explained below:

1. Remove irrelevant tweets:

   There are a lot of tweets, which are not relevant to be analyzed and identified for customer satisfaction analysis. In the real case, the tweets that mentioned "smartfren" are not merely talking about Smartfren's product satisfaction or feedback. Many tweets came from buzzers who promoted Smartfren extensively, seller who sells smartfren products on Twitter, or even job connector who promoted job-related to Smartfren. These types of tweets should be eliminated because it does not represent the customer experience.

2. Tweet's text cleaning:

   Both testing data and training data (learning process) are required for the classification of tweet sentiment. Preprocessing of the corpus data is performed before this learning process. In order to reduce the dimensions of the vector space model, preprocessing is required before the classification process. The vector space model's classification process will proceed more quickly by reducing its dimensions. The homogenization and volume reduction of words are the goals of this preprocessing. In other words, text cleaning aims to clean tweets of words or symbols that are not needed to reduce noise during the classification process. The words that were cleaned are:

i. Non-ascii character
ii. URLs
iii. Mentions symbol or “@”
iv. Hashtag symbol or “#”
v. Non-alphabet symbols or “[!$%^&*@#()_+|~=`{}\[\]%\-:";\'<>?,.\/]”
vi. Numbers from 0 to 9
vii. Duplication corrections with more than 2 characters. For example, “yukkk” into “yuk”
viii. Double space between words
ix. Space in the beginning and/or in the end of the sentence
x. Convert all text to lowercase
xi. Convert all variations of providers’ names into 1 name per provider

3. Tokenization:

   A group of characters in a text will be divided up into word units using this procedure. This is accomplished by identifying specific characters that may or may not be used as separators. For instance, word separators include whitespace characters like enter, tabulation, and space.

4. Stopwords:

   Stopwords can add dimension to the data classification process. Data dictionary of commonly used stopwords (consisting of which, in, to, from, etc.) will be added with Twitter-specific stopwords, such as “wkkwk”, “hihihi”, “xoxoxox”, etc. The complete Indonesian stopwords and Twitter stopwords data dictionary are in the appendix. Especially for the Indonesian Twitter stopwords data dictionary, which is collected manually from Twitter.

3.2.3 Lexicon-based Sentiment Labelling

To determine the sentiment label of each tweet's text, the lexicon-based method is chosen. The labeling step would summarize the whole tweet's text and determine whether it has a positive or negative sentiment towards Smartfren. Therefore, each of the tweets would be labeled as positive or negative. A positive label indicates the positive sentiment that the tweet's text has, and vice versa. In this research, the lexicon sentiment labeling would be divided into 2 types, which are Non-churn-related tweets sentiment labeling and Churn-related tweets sentiment labeling. It's because users are not only talking about Smartfren inside of their tweets but also comparing Smartfren with other operators/providers. Since the lexicon-based method uses a dictionary that cannot differentiate which subject (provider) is being discussed, therefore the sentiment labeling would not be accurate if we don't treat these tweets differently. Here is the visualization of the sentiment labeling described in Figure 3.2 below:

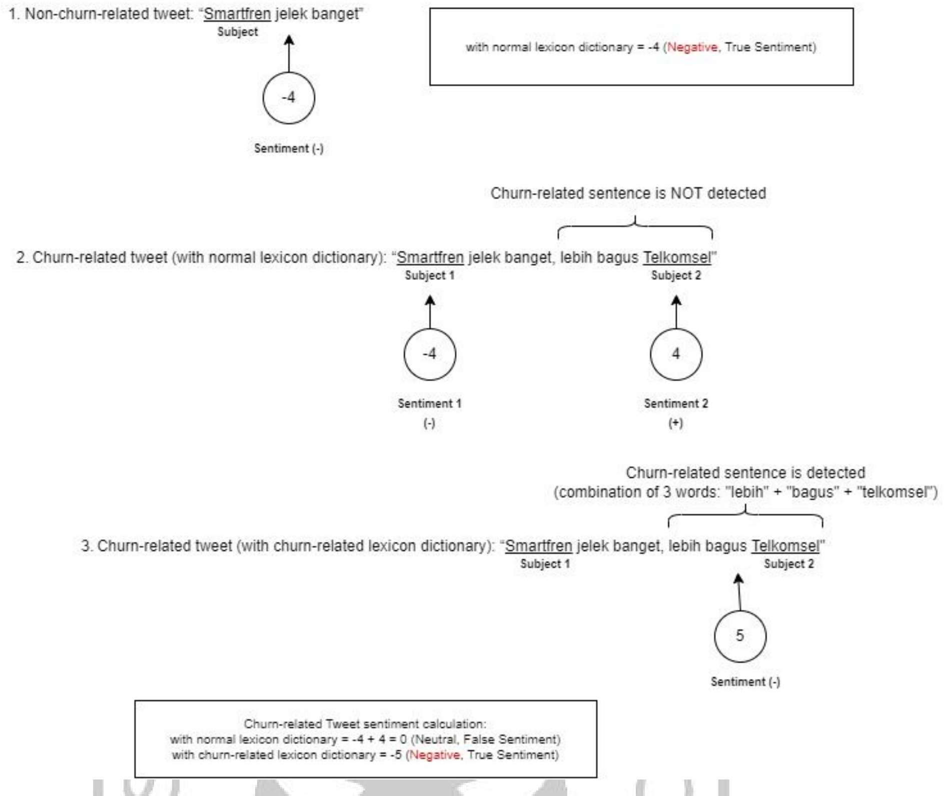

1. Non-churn-related tweet: "Smartfren jelek banget"
Subject
-4
Sentiment (-)
with normal lexicon dictionary = -4 (Negative, True Sentiment)
Churn-related sentence is NOT detected
2. Churn-related tweet (with normal lexicon dictionary): "Smartfren jelek banget, lebih bagus Telkomsel"
Subject 1
Subject 2
-4
4
Sentiment 1
(-)
Sentiment 2
(+)
Churn-related sentence is detected
(combination of 3 words: "lebih" + "bagus" + "telkomsel")
3. Churn-related tweet (with churn-related lexicon dictionary): "Smartfren jelek banget, lebih bagus Telkomsel"
Subject 1
Subject 2
5
Sentiment (-)
Churn-related Tweet sentiment calculation:
with normal lexicon dictionary = -4 + 4 = 0 (Neutral, False Sentiment)
with churn-related lexicon dictionary = -5 (Negative, True Sentiment)

Figure 3.2 Churn-related Tweets and Non-churn Related Tweets

Based on Figure 3.2 above, we can see what the churn-related tweets and non-churn-related-tweet look like. Here's the explanation:

a. Non-Churn-related tweets

Any tweets which have only 1 subject of operator/provider inside the tweet text are considered as non-churn-related tweets. Since the normal lexicon dictionary only has 1 word per sentiment score, tweets that only talks about 1 subject would be labeled correctly by the dictionary. The examples are described in Table 3.1 below:

Table 3.1 Example of Normal Lexicon Dictionary

Normal lexicon dictionary (only consists of 1 word per lexicon score)	
Non-churn-related lexicon	**Lexicon Score**
Jelek	-4
Lambat	-5
Mahal	-3

The non-churn-related tweets are going to use VADER calculation since the VADER rule-based calculation can only detect 1-word score per lexicon score. It cannot detect 2 words or more per lexicon score. At the end of the analysis, the tweets would be tagged as “non-churn” to summarize how many tweets are considered as non-churn.

b. Churn-related tweets

Any tweets which have more than 1 subject of operators/providers are considered churn-related tweets. A normal lexicon dictionary would not be able to recognize which subject is being talked about, and what sentiment the subject belongs to. Therefore, we need to make a new churn-related dictionary and separate this kind of tweet. The churn-related lexicon dictionary should have more than 1 word, because it must consist of minimum of 2 words (1 subject besides smartfren + sentiment over the other subject), such as for example “Telkomsel” + “Jelek”. Therefore, the calculation of the lexicon is different, because it uses different lexicon dictionaries. Then the resulting output is the sentiment of several topics that will be categorized. We will create the churn-related dictionary that has maximum words for 1 lexicon score of 4 words. Each of the churn-related lexicon score will have a score of -5 because the text sentiment is very specific since it has 2 to 4 words per lexicon score. The examples are described in Table 3.2 below:

Table 3.2 Example of Churn-related Lexicon Dictionary

Churn-related lexicon dictionary (consists of more than 2 words per lexicon score)	
Churn-related lexicon	**Lexicon Score**
Telkomsel Jelek	-5
Ganti kartu Telkomsel	-5
Mau pindah ke telkomsel	-5

For the calculation method, we will use the customized rule-based sentiment calculation. The rule would be able to detect 2 or more words per lexicon score, inside of the tweet text, unlike VADER which could only detect 1 word per lexicon score.

3.2.4 Classification and Evaluation

This study uses SVM as a classification algorithm (classifier). The training data and test data were obtained from each feature extraction result (Count Vectorizer, BoW, and TF-IDF) and lexicon labeling results (modified sentistrength_id). The algorithm used for sentiment classification utilizes the SVM, accuracy_score, model_selection, and LabelEncoder modules from scikit-learn (Pedregosa et al., 2011). SVM with a linear kernel was chosen as a classifier because the classification carried out in this study was in the form of binary or linear classification. For the record, the random_state parameter in the classifier is used to produce measurable randomization. For reasons of simplicity, the regularization parameter in the classifier is not configured, it is left at the default setting (C=1.0).

3.2.5 Data Exploration

In this stage, the sentiment-labeled tweets are analyzed further. 5 main processes happened during data exploration, which are churn-related tweets, engagement rate calculation, sudden sentiment change tagging, and topic categorizations. All these variables are important to be recognized and analyzed because it affects each other. Below is the illustration of the relationship between churn-related tweets, engagement rate calculation, sudden sentiment change tagging, and topic categorizations, described in Figure 3.3 below:

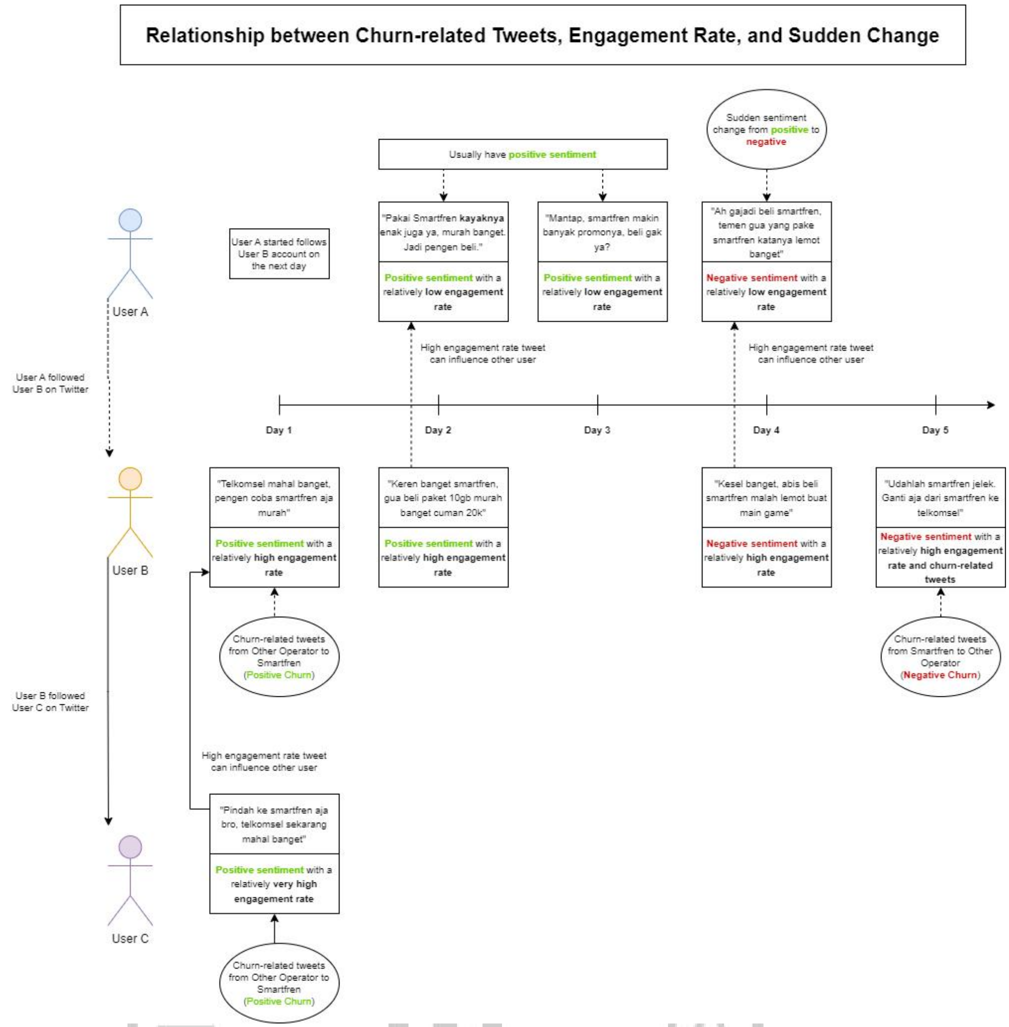

Relationship between Churn-related Tweets, Engagement Rate, and Sudden Change
Sudden sentiment change from positive to negative
Usually have positive sentiment
User A
User A started follows User B account on the next day
"Pakai Smartfren kayaknya enak juga ya, murah banget. Jadi pengen beli."
Positive sentiment with a relatively low engagement rate
"Mantap, smartfren makin banyak promonya, beli gak ya?
Positive sentiment with a relatively low engagement rate
"Ah gajadi beli smartfren, temen gua yang pake smartfren katanya lemot banget"
Negative sentiment with a relatively low engagement rate
High engagement rate tweet can influence other user
High engagement rate tweet can influence other user
User A followed User B on Twitter
Day 1
Day 2
Day 3
Day 4
Day 5
User B
"Telkomsel mahal banget, pengen coba smartfren aja murah"
Positive sentiment with a relatively high engagement rate
"Keren banget smartfren, gua beli paket 10gb murah banget cuman 20k"
Positive sentiment with a relatively high engagement rate
"Kesel banget, abis beli smartfren malah lemot buat main game"
Negative sentiment with a relatively high engagement rate
"Udahlah smartfren jelek. Ganti aja dari smartfren ke telkomsel"
Negative sentiment with a relatively high engagement rate and churn-related tweets
Churn-related tweets from Other Operator to Smartfren (Positive Churn)
Churn-related tweets from Smartfren to Other Operator (Negative Churn)
User B followed User C on Twitter
High engagement rate tweet can influence other user
"Pindah ke smartfren aja bro, telkomsel sekarang mahal banget"
Positive sentiment with a relatively very high engagement rate
User C
Churn-related tweets from Other Operator to Smartfren (Positive Churn)

Figure 3.3 Relationship between Churn-related Tweets, Post Engagement Rate, and Sudden Change

a. Churn-related tweet tagging

Churn-related tweet tagging is used to identify the tweets sentence which might have the indication of churning from Smartfren to the other provider, and vice versa. To identify the churn-related tweets, the churn-related keywords is used before the sentiment lexicon labeling process. So, any tweets which have churn-related keywords would be categorized as churn-related tweets. For example, tweets, which consist of words like “pindah ke telkomsel”, “ganti ke smartfren”, “telkomsel jelek”, indicate that the users might want to churn from one provider to another.

b. Post Engagement Rate (PER) Calculation

Engagement rate calculation is used to measure the level of influence that each tweet has when the followers of the user saw or read the tweet. By calculating the engagement rate of each tweet, the influence level of each tweet to influence people who see the tweets can be measured. The value of PER is between 0 and 1. A relatively high PER could potentially change other users' sentiment and causes sudden sentiment changes or even a churn from one provider to another.

c. Sudden Sentiment Change Tagging

Sudden sentiment change tagging would determine whether a particular user's tweet has a different sentiment compared to the previous user's tweet. It's because a user might usually have positive sentiment towards Smartfren, but then at some point in time, the same exact user might change his/her sentiment to a negative sentiment because of a particular reason, and vice versa. For example, in Figure 11 above, User A who has always had positive sentiment over Smartfren from day 1 to day 3, has a sudden sentiment change to negative sentiment at day 4. This User A tweet on day 4 would be tagged as "negative change". Sudden sentiment change is calculated using the rule-based method. The rule-based method will be used to determine the number of users who experience a change in sentiment from positive to negative, and vice versa, per time period.

d. Topic Categorization

Topic categorization is used to tag what topic the user talked/discussed about in each tweet. This might also highlight the reason why the user changes the sentiment from positive to negative, and vice versa. The method used is to categorize sentences in tweets with specific topic-related keywords. For example, to categorize tweets that discuss video streaming satisfaction, the researcher will use the keywords "youtube", "film", "streaming", and "watching" to label topics regarding video streaming satisfaction. In the categorization, the researcher used 2 levels of topic categories. However, only 1 category, which derived into the level 2 category. Here are all the topic categories:

i. Network Performance

All the tweets, which consist of internet connectivity topics. Since this metric is quite general, the network performance topic is then categorized further into the level 2 category. Here's the level 2 category of Network Performance:

1. Video Experience: Tweets, which consist of any video watching or streaming experience. It includes video streaming on any platform/application, like in Youtube, Social Media, and in Browser.
2. Games Experience: Any tweets which mention gaming-related feedback, such as mentioning the name of the games that the users are playing and how the network performs.
3. Social Media App Experience: Tweets that mentioned the experience in using certain social media apps. It includes any social media activity that users do such as chatting, video calls, seeing images and video contents, etc.
4. Video Call Experience: Tweets that mentioned video call or conference call experience, such as Zoom meetings, Google meetings, WhatsApp video calls, etc. It includes any app activity which uses voice and real-time video calls.
5. Voice Call Apps Experience: It includes tweets that mentioned users' experience in using apps' voice call service which requires an internet connection to do the voice call. For example, WhatsApp voice call, Line voice call, etc.
6. Download Experience: Any tweets which mentioned about download-related activity
7. Upload Experience: Any tweets which mentioned upload-related activity
8. Coverage Experience: To track the signal availability in Indonesia, the researcher uses this categorization as a way to directly get feedback from customers who use the Smartfren network in different parts of Indonesian territory. Tweets are categorized based on tweet text which mentioned cities, provinces, and specific areas (rural areas, urban areas, forests, etc.). Therefore, the network feedback in certain locations would be able to be detected.

ii. Product Conformance

This metric is used for the tweets which mentioned the product-related experience. Product-related experience refers to whether Smartfren products are worth buying, in terms of their values, price, product bundling, and content inside the products. This categorization will include any user experience on Twitter expressed by several keywords such as any Smartfren product name (youtube bundling package, internet unlimited package, etc.) and product price feedback (expensive, cheap, affordable, etc.).

iii. Customer Service Experience

Smartfren uses Twitter as one of its platforms to reach out the customers and communicate with the customers regarding certain issues. Oftentimes, customers give their complaints to the Smartfren official account on Twitter, such as (@smartfrenworld and @smartfrencare). This tweets categorization will represent the tweets' sentiment regarding how well Smartfren handles or answers the customers' complaints on Twitter, according to the customers' tweets.

Table 3.3 Topic Categorization Levels

Level 1 Category	**Level 1 Category Keywords**	**Level 2 Category**	**Level 2 Category Keywords**
Network Performance	"lemot", "lambat", "lama", "cepat", "lancar", "normal", "hilang", "putus"	Video Experience	"streaming", "video", "film", "movie", "nonton", etc.
		Games Experience	"main", "game", "mabar", "mobile legends", "ml" (abbreviation of mobile legends), "ping"
		Social	"whatsapp",

		Media App Experience	“line”, “telegram, “facebook”, “twitter”, etc.
		Video Call Experience	“zoom”, “meet”, “gmeet”, “google meet”, “vidcall”, “video call”
		Voice Call Apps Experience	“telepon” & “whatsapp”, “whatsapp & “call”, “line” & “call”, etc.
		Download Experience	“download”, “donlot”, “unduh”, “mengunduh”, etc.
		Upload Experience	“upload”, “kirim” & “file”, “ngirim” & “video”, etc.
		Coverage Experience	“4G”, 3G”, “kota”, “kabupaten”, “disini”, “di” & “kampung”, all of Indonesian city and province names, etc.
Product Conformance Experience	“tagihan”, “mahal”, “murah”, “terjangkau”, “produk”, “worth”		

	& “it”, “sepadan”, all the smartfren package names, etc.		
Customer Service Experience	“professional”, “ramah”, “respon”, “solusi”, “cs” (abbreviation of customer service), “customer” & “service”, “pelayanan”, etc.		

3.2.6 Data Visualization and Recommendation

After all the data output have been generated from the previous processes, the sentiment data can be visualized to see the data insights. The visualization would be conducted using Power BI software, to generate several kinds of charts from the data provided. The visualized data would be analyzed and so the recommendation for each period (each year or month) would be summarized for the company’s improvement. The visualization would be conducted using Power BI software.

CHAPTER IV
DATA COLLECTION AND PROCESSING

This chapter will discuss in detail the data collection, data pre-processing, lexicon tagging, model building, data exploration process, and data visualization. At the end of this chapter, a comparison is made of the models that have been made and the best model is selected to be implemented in the branding and marketing communication evaluation activities. The data processing would be conducted using Python and R programming languages, since it enables the researcher to process a large amount of data efficiently.

4.1 Data Gathering

In the data crawling, we will gather the tweets data from Twitter which consists of Smartfren keywords. Smartfren-related keywords were used to filter and pull the tweets data from the Twitter API directly, such as “smartfren”, “@smartfrenworld” as smartfren’s official account name, and “@smartfrencare” as smartfren’s official customer care account name. To gather the data from January 2019 to September 2021, the researcher used a library built on the Python programming language, namely Snscrape. It is a social media data scraper library that is able to collect the tweets alongside with other useful information regarding the tweets. Here are the tweet data variables that the researcher gathered from Twitter, as described in Table 4.1 below:

Table 4.1 Raw Data Variables

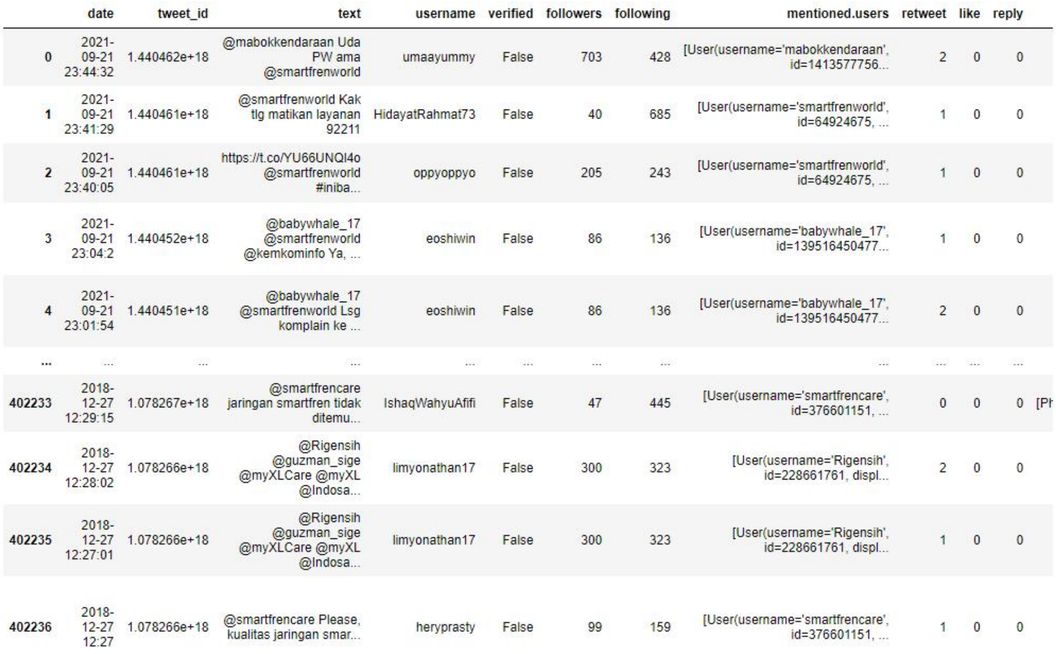
Variables	Description
Date	Date tweet was created
Tweet ID	Unique ID of the tweet
Content	The text content of the tweet
Username	Account username who uploaded the tweet
User Verified	Whether the account is a verified account or not
User Followers Count	The total amount of followers of the

	account
User Following Count	The total amount of followed accounts
Mentioned Users	The account username who are mentioned in the tweet content
Retweet Count	The total amount of retweets
Like Count	The total amount of like
Reply Count	The total amount of reply
Media	Media object containing the preview URL, full URL, and media type on the tweet content
Language	Machine-generated language of the tweet
Location	Location in which the account tagged on the tweet (not the real location of the account that tweeted the tweets)

Since the tweets' data size is massive, it could not be gathered all at once. It needs to be gathered sequentially. It needs to be gathered per 1 million tweets. Here are the codes for gathering the data using Snscrape library:

~~~
```
maxTweets = 1000000  # the number of tweets you require
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('smartfren OR @smartfrenworld OR @smartfencare' + 'since:2019-01-01 until:2019-12-31').get_items()) :
        if i > maxTweets :
            break
        print(tweet.date)
        csvWriter.writerow([tweet.date, tweet.content.encode('utf-8'), tweet.user.username.encode('utf-8'), tweet.user.verified, tweet.user.followersCount, tweet.user.friendsCount, tweet.mentionedUsers, tweet.retweetCount, tweet.likeCount, tweet.replyCount, tweet.media, tweet.lang.encode('utf-8'), tweet.location])
```
~~~

The total tweets that have been gathered is 2,168,136 tweets. This data would be considered as raw data and is going to be cleaned and processed further. Here's some of the result of the data scraping as shown in Figure 4.1 below:



	date	tweet_id	text	username	verified	followers	following	mentioned.users	retweet	like	reply	
0	2021-09-21 23:44:32	1.440462e+18	@mabokkendaraan Uda PW ama @smartfrenworld	umaayummy	False	703	428	[User(username='mabokkendaraan', id=1413577756...	2	0	0	
1	2021-09-21 23:41:29	1.440461e+18	@smartfrenworld Kak tlg matikan layanan 92211	HidayatRahmat73	False	40	685	[User(username='smartfrenworld', id=64924675, ...	1	0	0	
2	2021-09-21 23:40:05	1.440461e+18	https://t.co/YU66UNQI4o @smartfrenworld #iniba...	oppyoppyo	False	205	243	[User(username='smartfrenworld', id=64924675, ...	1	0	0	
3	2021-09-21 23:04:2	1.440452e+18	@babywhale_17 @smartfrenworld @kemkominfo Ya, ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	1	0	0	
4	2021-09-21 23:01:54	1.440451e+18	@babywhale_17 @smartfrenworld Lsg komplain ke ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	2	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
402233	2018-12-27 12:29:15	1.078267e+18	@smartfrencare jaringan smartfren tidak ditemu...	IshaqWahyuAfifi	False	47	445	[User(username='smartfrencare', id=376601151, ...	0	0	0	[P
402234	2018-12-27 12:28:02	1.078266e+18	@Rigensih @guzman_sige @myXLCare @myXL @Indosa...	limyonathan17	False	300	323	[User(username='Rigensih', id=228661761, displ...	2	0	0	
402235	2018-12-27 12:27:01	1.078266e+18	@Rigensih @guzman_sige @myXLCare @myXL @Indosa...	limyonathan17	False	300	323	[User(username='Rigensih', id=228661761, displ...	1	0	0	
402236	2018-12-27 12:27	1.078266e+18	@smartfrencare Please, kualitas jaringan smar...	heryprasty	False	99	159	[User(username='smartfrencare', id=376601151, ...	1	0	0	

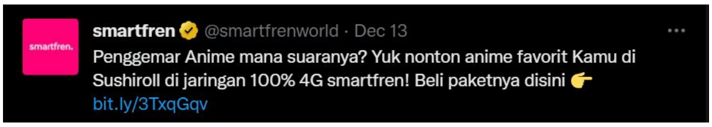
Figure 4.1 Raw Data Result

4.2 Data Pre-processing

At this stage, the documents generated from the crawler will be filtered. This process aims to change the correct data type so that it can be read by the sentiment labeling rules and machine learning. The crawled data or raw data that has been gathered previously, must be transformed into a data type suitable for the classification process. There are 3 steps in the data pre-processing part, which are irrelevant tweets removal, text cleaning, churn-related tweets and non-chun-related tweets separation, and slang words and stopwords cleaning.

4.2.1 Irrelevant Tweets Removal

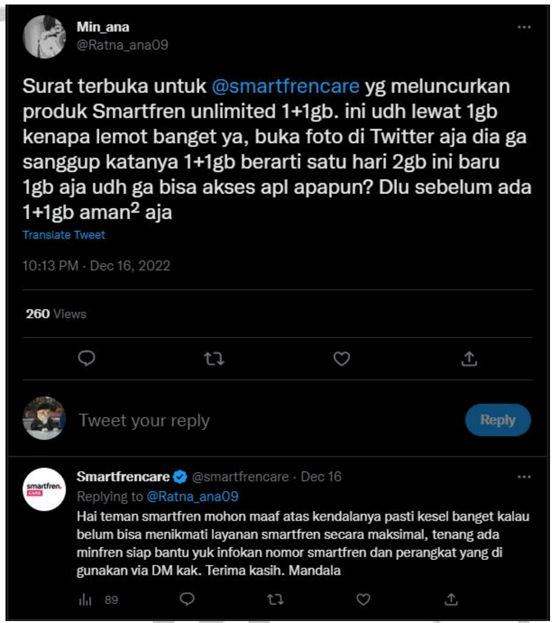
Although in the data crawling part the tweets that have been gathered are only for the smartfren-related tweets, there are still massive amounts of tweets that could not be considered as customer experience tweets. Not always the tweets which consist of "smartfren" word, are talking about the customer experiences. First, the researcher found out that there are tweets that came from Smartfren's official account. These tweets which came from Smartfren is going to be included in the analysis because it's obviously not coming from customer and are not customer experience tweets. Mainly, the Smartfren account aimed to promote Smartfren, engage with the customers' tweets (only a few customer tweets), and reply to the complaints from customers on Twitter. These are the tweets from Smartfren that need to be removed from the analysis, as shown in Figure 4.2 and Figure 4.3 below:



smartfren @smartfrenworld · Dec 13

Penggemar Anime mana suaranya? Yuk nonton anime favorit Kamu di Sushiroll di jaringan 100% 4G smartfren! Beli paketnya disini 👉 bit.ly/3TxqGqv

Figure 4.2 Smartfren Official Account Tweeted about Promotion



Min_ana
@Ratna_ana09

Surat terbuka untuk @smartfrencare yg meluncurkan produk Smartfren unlimited 1+1gb. ini udh lewat 1gb kenapa lemot banget ya, buka foto di Twitter aja dia ga sanggup katanya 1+1gb berarti satu hari 2gb ini baru 1gb aja udh ga bisa akses apl apapun? Dlu sebelum ada 1+1gb aman^2 aja

Translate Tweet

10:13 PM · Dec 16, 2022

260 Views

Tweet your reply

Reply

Smartfrencare @smartfrencare · Dec 16

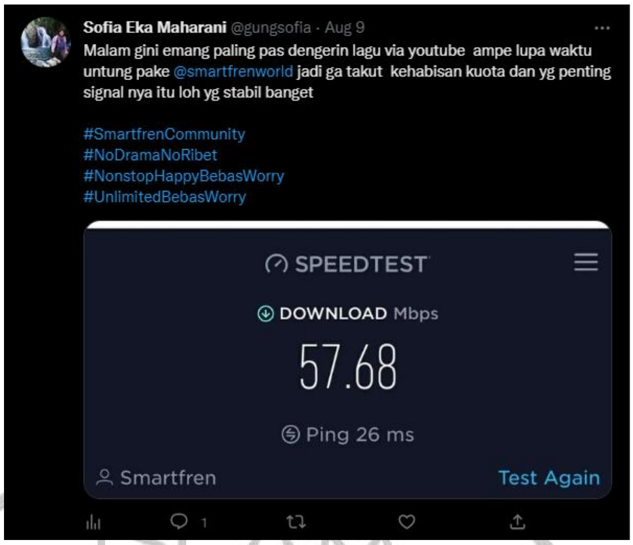
Replying to @Ratna_ana09

Hai teman smartfren mohon maaf atas kendalanya pasti kesel banget kalau belum bisa menikmati layanan smartfren secara maksimal, tenang ada minfren siap bantu yuk infokan nomor smartfren dan perangkat yang di gunakan via DM kak. Terima kasih. Mandala

89

Figure 4.3 Smartfren Official Account Replied Customer’s Complaint

Second, the researcher found many tweets which came from buzzers or users who have been affiliated with Smartfren or other institutions, to share “fake” positive content all over the Twitter platform. They aimed to polarize the people's opinion on Twitter about Smartfren, so that they may be able to get benefits from the higher selling of Smartfren products through buzzing on Twitter. The positive contents shared by the buzzers are too exaggerated. The words being used by the Buzzers are like made-up-words, unlike a typical customer’s feedback. Buzzers are also shared positive tweets much more frequently than normal users. They can send fake positive content daily or weekly, with a very low engagement rate. Below is an example of a buzzer’s tweet, as shown in Figure 4.4 below:



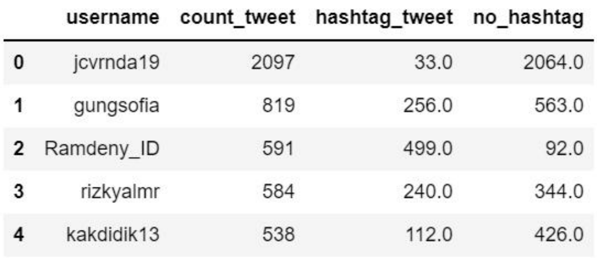
Sofia Eka Maharani @gungsofia · Aug 9

Malam gini emang paling pas dengerin lagu via youtube ampe lupa waktu untung pake @smartfrenworld jadi ga takut kehabisan kuota dan yg penting signal nya itu loh yg stabil banget

#SmartfrenCommunity
#NoDramaNoRibet
#NonstopHappyBebasWorry
#UnlimitedBebasWorry

Figure 4.4 Buzzers' Tweet

Besides the fact that buzzers posted many tweets about Smartfren, they also have characteristics in which they always use hashtags on their tweets post. This way, buzzers are able to polarize the sentiment of Smartfren-related tweets through keywords and hashtags. In conclusion, buzzers can be identified from the number of tweets. After the researcher dive through the tweets for each of the users from 2020 to 2022, there are massive number of buzzers' tweets that needs to be removed from the analysis. There are many users who have hundreds or thousands of tweets, which mentioning Smartfren during the 3 years, in which some of these tweets have hashtags, as described in Figure 4.5 below:

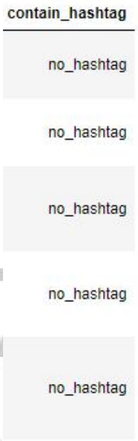


	username	count_tweet	hashtag_tweet	no_hashtag
0	jcvrnda19	2097	33.0	2064.0
1	gungsofia	819	256.0	563.0
2	Ramdeny_ID	591	499.0	92.0
3	rizkyalmr	584	240.0	344.0
4	kakdidik13	538	112.0	426.0

Figure 4.5 Several Twitter Accounts Have Hundreds of Tweets and Many Tweets with Hashtags

These buzzer tweets need to be removed from the analysis. First, the tweets which contain hashtags need to be tagged. The new hashtag variable called "contain_hashtag" will be created below, using the codes as follows, and the result is shown in Figure 4.6 below:

```
sf_tweet.loc[sf_tweet.text.str.contains('#'), 'contain_hashtag'] = 'hashtag_tweet'
sf_tweet['contain_hashtag'] = sf_tweet['contain_hashtag'].fillna(value='no_hashtag')
# Show the result
sf_tweet.head()
```

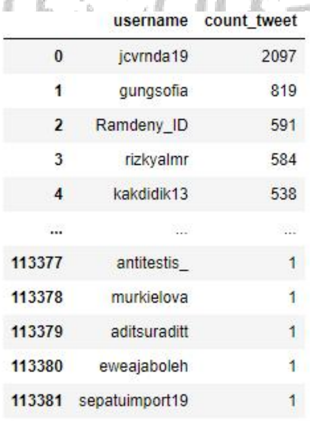


contain_hashtag
no_hashtag
no_hashtag
no_hashtag
no_hashtag
no_hashtag

Figure 4.6 New Hashtag Tagging Variable

Next, the number of tweets for each user needs to be calculated. This data would be used for the consideration of tweets, which should be removed. Here are the codes to calculate the number of tweets for each Twitter user, and the result is shown in Figure 4.7 below:

```
df_2 = sf_tweet['username'].value_counts()
df_2 = df_2.to_frame()
df_2 = df_2.reset_index()
df_2 = df_2.rename(columns={"index": "username", "username": "count_tweet"})
df_2
```

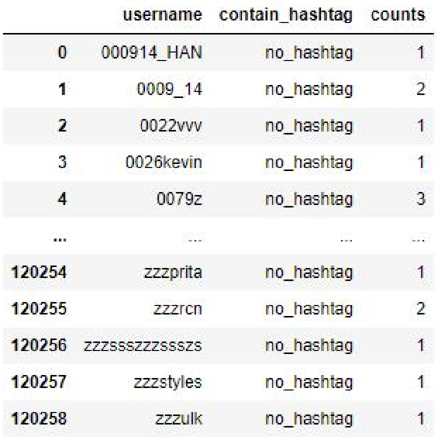


	username	count_tweet
0	jcvrnda19	2097
1	gungsofia	819
2	Ramdeny_ID	591
3	rizkyalmr	584
4	kakdidik13	538
...	...	...
113377	antitestis_	1
113378	murkielova	1
113379	aditsuraditt	1
113380	eweajaboleh	1
113381	sepatuimport19	1

Figure 4.7 Calculated Number of Tweets for Each Twitter User

After that, the number of tweets that have hashtags needs to be calculated per user. The calculation would be based on the new hashtag variable that was created previously and the result is shown in Figure 4.8 below:

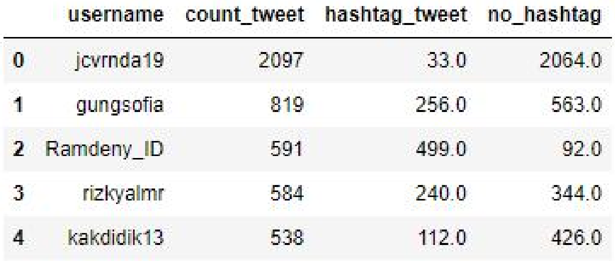
```
df_hashtag_count = sf_tweet.groupby(['username',
'contain_hashtag']).size().reset_index(name='counts')
df_hashtag_count
```



	username	contain_hashtag	counts
0	000914_HAN	no_hashtag	1
1	0009_14	no_hashtag	2
2	0022vvv	no_hashtag	1
3	0026kevin	no_hashtag	1
4	0079z	no_hashtag	3
...	...	...	...
120254	zzzprita	no_hashtag	1
120255	zzzrcn	no_hashtag	2
120256	zzzssszzzssszs	no_hashtag	1
120257	zzzstyles	no_hashtag	1
120258	zzzulk	no_hashtag	1

Figure 4.8 Tweets Which Have Hashtags per Twitter Account

After the tweets with hashtags and the number of tweets have been calculated, it can be summarized altogether as shown in Figure 4.9 below:



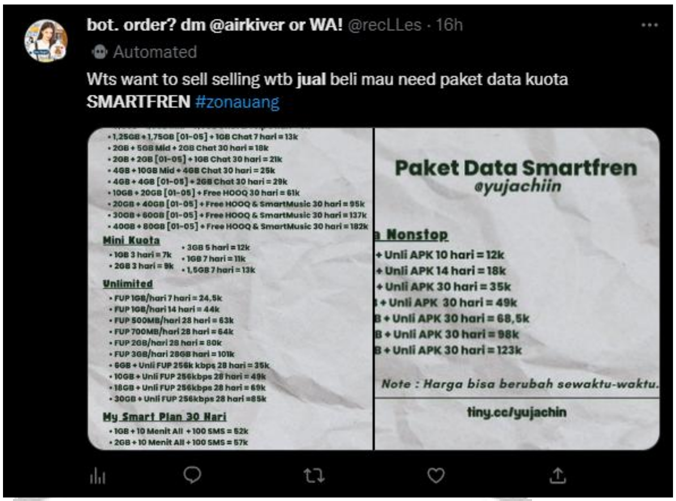
	username	count_tweet	hashtag_tweet	no_hashtag
0	jcvrnda19	2097	33.0	2064.0
1	gungsofia	819	256.0	563.0
2	Ramdeny_ID	591	499.0	92.0
3	rizkyalmr	584	240.0	344.0
4	kakdidik13	538	112.0	426.0

Figure 4.9 Tweets with and without Hashtags Total per Account

There are 2 ways to remove the buzzers' tweets. First, the researcher used the benchmark of more than 200 tweets during the 3 years period (more than 5 tweets mentioning Smartfren in a month) and more than 14 tweets, which have hashtag during the 3 years period, to indicate that the Twitter user is the buzzer account, in which all of their tweets need to be removed. Second, the researcher also listed manually Twitter accounts that have buzzer-related tweets. There are in total 3,881 tweets incorporated with listed buzzer accounts. These listed accounts are then removed as well. The result shows that there are 295,922 tweets without buzzers’ tweets. Here are the codes to remove the tweets from the listed buzzers’ usernames:

```
# After removing buzzers, there are still tweets that identified as buzzers (after manually look at the result)
# Therefore, buzzer accounts need to be dropped manually using the list of buzzer accounts
list_to_remove = ['@Kaum_pusing', '@Sofy_Beeeeee', '@NyaiiBubu', '@timun_renyah2', '@DandanNiLL', 'Bisniscom', 'HaloBCA', 'waferkedjoe', 'Rinddha', 'yqyu', 'YusmillahR', 'VIN_NOOO', 'dinithea', 'sedihterus_', 'mandiricare'
, 'Vitameansi', 'mahardinaaa','Cauzycanabiz', 'renahsetiawan8', 'LebahGanteng07', 'capanieee', 'Namigoreng', 'ayuniwang852', 'GoHaera', 'maz_echo'
, 'rahmat28adi', 'nisaamolla', 'ouuwsomm', 'shappyworld95', 'jaya_janwar', 'mandaaakk', 'dwik_Gen', 'RizalRosyadi93', 'r_wahyuindah', 'Iarismanis'
, 'lariesmanis', 'itsmedif', 'JarrFajarr_', 'alnazp139', 'SayYasha', 'mhmmdsairaji', 'afifahrahmatika', 'yukenkolmi', 'nabilaasingal']

list_buzzer_2 = sf_tweet[sf_tweet.text.str.contains('|'.join(list_to_remove))]['username'].tolist()
print(len(list_buzzer_2))
print(len(list_buzzer_1))

#drop all buzzers from the second list
sf_tweet = sf_tweet.drop(sf_tweet[sf_tweet['username'].isin(list_buzzer_2)].index).reset_index(drop=True)
sf_tweet.shape[0]
```

The next step after buzzer tweets removal is promotion tweets removal. There are many people who sell Smartfren products and other providers’ products on social media, especially Twitter. When the researcher took the tweets data from Twitter, the promotion tweets are also included since it also has “smartfren” word. Tweet in Figure 4.10 below is one of the examples:



bot. order? dm @airkiver or WA! @recLLes · 16h

Automated

Wts want to sell selling wtb **jual** beli mau need paket data kuota SMARTFREN #zonauang

- 1,25GB + 1,75GB [01-05] + 1GB Chat 7 hari = 13k
- 2GB + 5GB Mid + 2GB Chat 30 hari = 18k
- 2GB + 2GB [01-05] + 1GB Chat 30 hari = 21k
- 4GB + 10GB Mid + 4GB Chat 30 hari = 25k
- 4GB + 4GB [01-05] + 2GB Chat 30 hari = 29k
- 10GB + 20GB [01-05] + Free HOOQ 30 hari = 61k
- 20GB + 40GB [01-05] + Free HOOQ & SmartMusic 30 hari = 95k
- 30GB + 60GB [01-05] + Free HOOQ & SmartMusic 30 hari = 137k
- 40GB + 80GB [01-05] + Free HOOQ & SmartMusic 30 hari = 182k

Mini Kuota

- 1GB 3 hari = 7k
- 2GB 3 hari = 9k
- 3GB 5 hari = 12k
- 1GB 7 hari = 11k
- 1,5GB 7 hari = 13k

Unlimited

- FUP 1GB/hari 7 hari = 24,5k
- FUP 1GB/hari 14 hari = 44k
- FUP 500MB/hari 28 hari = 63k
- FUP 700MB/hari 28 hari = 64k
- FUP 2GB/hari 28 hari = 80k
- FUP 3GB/hari 28GB hari = 101k
- 6GB + Unli FUP 256k kbps 28 hari = 35k
- 10GB + Unli FUP 256kbps 28 hari = 49k
- 18GB + Unli FUP 256kbps 28 hari = 69k
- 30GB + Unli FUP 256kbps 28 hari =85k

My Smart Plan 30 Hari

- 1GB + 10 Menit All + 100 SMS = 52k
- 2GB + 10 Menit All + 100 SMS = 57k

Paket Data Smartfren
@yujachiin

a Nonstop

+ Unli APK 10 hari = 12k
+ Unli APK 14 hari = 18k
+ Unli APK 30 hari = 35k
+ Unli APK 30 hari = 49k
B + Unli APK 30 hari = 68,5k
B + Unli APK 30 hari = 98k
B + Unli APK 30 hari = 123k

Note : Harga bisa berubah sewaktu-waktu.

tiny.cc/yujachin

Figure 4.10 Promotion Tweet Example

Therefore, we need to exclude these tweets because it's not represent the customer experience. The way to remove it effectively is by using certain keywords which are often times used by the sellers to promote Smartfren products. So, any tweets which have the keywords will be removed. The keywords are the result of manual observation through some tweets which indicate a promotion activity. This will involve 2 words for each keyword to detect whether the tweets are included as a promotion or not. Here's the example for the promotion tweets keywords, as shown in Table 4.2 below:

Table 4.2 Example of Promotion Tweets Keywords

Keywords	Tweet Example
"jual" + "kuota"	"Jual kuota smartfren 20gb cuman 230 ribu"
"open" + "promo"	"open promo pulsa smartfren 20 ribu"

Here are the codes to remove the promotion tweets based on keywords:

```
# We also have discovered manually about the keywords that contains 2 words, regarding irrelevant tweets
df1 = sf_tweet.drop(sf_tweet[(sf_tweet['cleaned_text'].str.contains('promo')) & (sf_tweet['cleaned_text'].str.contains('fast'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuhuuw')) & (df1['cleaned_text'].str.contains('promo'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('pulsa'))].index)
```

```
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kuota')) & (df1['cleaned_text'].str.contains('jual'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('hayuk')) & (df1['cleaned_text'].str.contains('open'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('simak'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('kuota'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('promo'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('cek')) & (df1['cleaned_text'].str.contains('telkomsel'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kak')) & (df1['cleaned_text'].str.contains('promo'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('promo')) & (df1['cleaned_text'].str.contains('whatsapp'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('promo')) & (df1['cleaned_text'].str.contains('telkomsel'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('fast')) & (df1['cleaned_text'].str.contains('promo'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kak')) & (df1['cleaned_text'].str.contains('open'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('fast'))].index)
df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('kak'))].index)
```

Finally, the relevant tweets only count for 292,547 tweets from the raw total of 2,168,136 tweets, or only 13% of tweets that could be indicated as customer experience tweets. Although, according to the researcher's manual observation, there is still a chance that the relevant tweets result still has many irrelevant ones, since there's still a limitation to clean the whole tweets by only using a keyword filtering process.

4.2.2 Text Cleaning

During this stage, each tweet text or sentence would be cleaned from the characters' noise and words noise. In order to label the sentiment for each tweet, the text should be adjusted to a form that makes the lexicon words dictionary able to be detected inside the tweet text, so that the algorithm could calculate the sentiment score accurately from the detected words. This process is also used to exclude insignificant words for sentiment labeling.

Many words would not be carrying any sentiment meaning inside of the tweets. If these insignificant words are also calculated during the process, it will increase the burden to computing device, and make the processing duration become much longer, especially when it comes to the machine learning stage which would be conducted afterward.

In the text cleaning process, the researcher will clean the text sentences in several steps:

13. Non-Ascii removal: Removing text characters that are not categorized as ASCII (American Standard Code for Information Interchange), a standard data-encoding format for electronic communication between computers. A non-ASCII character that might be included during the data crawling process, would not be able to be processed by the codes.
14. URLs removal: Any website links, media links, and any other links inside of the tweets would be removed because it does not contain any sentiment.
15. Mentions removal: Twitter users used to tag their friends when they tweeted something using "@" character. Since it would become a noise for the sentiment labeling, this mentioned character would not be included.
16. Hashtag removal: Hashtags or "#" would be a noise for the sentiment labeling too, thus it would be removed as well. But the word of the hashtag would be kept inside the tweet text.
17. Symbol and numbers removal: There are many non-alphabetical characters, which contained in the tweet text. Any symbol and numbers included, especially which are not separated from the words are going to block lexicon sentiment labeling to detect the words.
18. Duplications and extra space cleaning: Oftentimes, Twitter users typed the text with exaggeration for each word. For example, "bagus" are often typed "bagusss". And a lot of times, they also unintentionally typed extra spaces at the end and/or beginning of the sentence, or in between words. These unclean words and spaces need to be normalized into the right words, and unexaggerated spaces.

19. Lowercase: The words would be easier to read if all words used lowercase.
20. Convert variations of providers’ names into the right name: In Twitter, the variations of providers’ names mentioned by Twitter users are commonly happened. For example, people used to type “tsel” to mention “telkomsel”, or “axiata” for “xl”. While in the lexicon dictionary, the providers’ name should be normalized into 1 chosen name per provider, otherwise, the lexicon logic would not be able to detect these names variations, to further calculate the sentiment.

Here are the codes to clean the tweet text, based on the 8 steps described above:

```
# Text Cleaning Process

import re

count=0
with open(output, 'w') as f:
    for line in text:

        # Step-1: Non-ascii
        res = re.sub(r'[^\x00-\x7F]+',' ', line)

        # Step-2: URLs
        res = re.sub(r'http[s]?\:\/\/.[a-zA-Z0-9\.\/\_?=%&#\-\+!]+',' ', res)
        res = re.sub(r'pic.twitter.com?.[a-zA-Z0-9\.\/\_?=%&#\-\+!]+',' ', res)

        # Step-3: mentions
        res = re.sub(r'@','', res)

        # Step-4_alt-2: Convert hashtags to sentence (string separation)**
        res = re.sub(r'((?<=[a-z])[A-Z]|[A-Z](?=[a-z]))', ' \\1', res)

        # Step-5: symbol
        res = re.sub(r'[!$%^&*@#()_+|~=`{}\[\]%\-:";\'<>?,.\/]', '', res)
        # Step-5: numbers
        res = re.sub(r'[0-9]+','', res)

        # Step-6: Duplications correction with more than 3 characters (example: "yukkk" to "yuk")
        res = re.sub(r'([a-zA-Z])\1\1','\\1', res)
    # Step-6: double space or more, to only one space
    res = re.sub(' +', ' ', res)
    # Step-6: space in the beginning and/or in the end of the sentence
    res = re.sub(r'^[ ]|[ ]$','', res)

    # Step-7: convert all text to lower case
    res = res.lower()

    # Step-8: Convert all kinds of provider's package names, official account names, and informal names to 1 provider's name
    # Note that since several provider's package names are perhaps included in the lexicon dictionary, only names with no sentiment polarity will be replaced
    res = re.sub(r'smartfrenworld','smartfren', res)
    res = re.sub(r'smartfrencare','smartfren', res)
    res = re.sub(r'connex','smartfren', res)
    res = re.sub(r'gokil max','smartfren', res)
    res = re.sub(r'telkomselhalo','telkomsel', res)
    res = re.sub(r'telkomcare','telkomsel', res)
    res = re.sub(r'tsel','telkomsel', res)
    res = re.sub(r'telkomsl','telkomsel', res)
    res = re.sub(r'tlkomsel','telkomsel', res)
    res = re.sub(r'tlkmsl','telkomsel', res)
    res = re.sub(r'myorbitid','telkomsel', res)
    res = re.sub(r'byu','telkomsel', res)
    res = re.sub(r'by','telkomsel', res)
    res = re.sub(r'combo sakti','telkomsel', res)
    res = re.sub(r'internet sakti','telkomsel', res)
    res = re.sub(r'internet omg','telkomsel', res)
    res = re.sub(r'kartuhalo','telkomsel', res)
    res = re.sub(r'kartu halo','telkomsel', res)
    res = re.sub(r'kartu as','telkomsel', res)
    res = re.sub(r'simpati','telkomsel', res)
    res = re.sub(r'internetmax','telkomsel', res)
    res = re.sub(r'internet max','telkomsel', res)
    res = re.sub(r'xlaxiatatbk','xl', res)
    res = re.sub(r'xlaxiata','xl', res)
    res = re.sub(r'axiata','xl', res)
    res = re.sub(r'xlaxiata_tbk','xl', res)
    res = re.sub(r'axiataxl','xl', res)
    res = re.sub(r'xlaxiataid','xl', res)
    res = re.sub(r'triindonesia','tri', res)
    res = re.sub(r'three','tri', res)
    res = re.sub(r'kompak','tri', res)
    res = re.sub(r'indosatcare','indosat', res)
    res = re.sub(r'isat','indosat', res)
    res = re.sub(r'ooredoo','indosat', res)
    res = re.sub(r'oredoo','indosat', res)
    res = re.sub(r'ooredo','indosat', res)
    res = re.sub(r'oredo','indosat', res)
    res = re.sub(r'idsat','indosat', res)
    res = re.sub(r'indsat','indosat', res)
    res = re.sub(r'indo','indosat', res)
    res = re.sub(r'indst','indosat', res)
    res = re.sub(r'freedom','indosat', res)
    res = re.sub(r'yellow','indosat', res)
    res = re.sub(r'askaxis','axis', res)
    res = re.sub(r'ask_axis','axis', res)
    res = re.sub(r'bronet','axis', res)
    res = re.sub(r'warnet','axis', res)

    # Re-write all the text into .txt
    f.write(str(res+"\n"))
    count+=1
```

4.2.3 Churn-related Tweets and Non-chun-related Tweets Separation

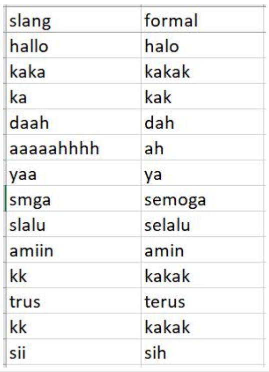
In the tweet separation stage, the tweets are going to be separated based on its churn-related and non-churn-related content included in the tweet text. As it has been explained in the previous chapter, this stage is necessary because lexicon words cannot identify the context of the text. The lexicon logic cannot identify which subject (which provider) Twitter users are complaining about or talking about. The tweet text content is going to be separated using keywords. It’s actually a simple process, in which, the researcher will use the other telecommunication providers’ names for the churn-related keywords. When the tweets consist of churn-related content, the text must mention the other telecommunication provider’s name, to compare Smartfren with the other provider. For example, “Smartfren jelek, mendingan koneksi Telkomsel” or “Telkomsel mahal banget, lebih baik pakai Smartfren”. Therefore, the researcher would categorize tweets as churn-related tweets if the content mentioned any other provider’s name. If not, then the tweets are categorized as Non-churn related tweets. Due to convenience reason, in this stage, the researcher will use R programming language to conduct the process, instead of Python programming language. Here is the code for this process:

```
library(dplyr)
library(data.table)
library("writexl")

df <- read.csv("cleaned_data_merged.csv")
View(df)


churnKeyword <- c('telkomsel', 'indosat', 'xl', 'axis', 'tri')


# churn-related tweets
df_churn <- df[grepl(paste(churnKeyword, collapse='|'), df$cleaned_text.1),]
nrow(df_churn) #total 41,147 tweets
View(df_churn)

# churn-related tweets
df_nonchurn <- df[!grepl(paste(churnKeyword, collapse='|'), df$cleaned_text.1),]
nrow(df_nonchurn) #total 251,400 tweets
View(df_nonchurn)
```

The tweets separation of 292,547 tweets, resulted in a total of 251,400 non-churn-related tweets and a total of 41,147 tweets. So, there are about 14% of tweets which categorized as churn-related tweets.

4.2.4 Slang Words and Stopwords Cleaning

Slang words cleaning aimed to normalize informal or slang words into normal words or formal words. This process is useful for the lexicon logic, to make the words able to be detected and identified. The researcher used Salsabila’s slang words dictionary to replace the informal words with formal words, which is currently the only Indonesian slang words dictionary that has publication (Aliyah Salsabila et al., 2018). The dictionary provides a lexicon for text normalization of Indonesian colloquial words. It consists of 3,592 unique colloquial words-also known as “bahasa alay”, and manually annotated with the normalized form. The words normalization lexicon was built from Instagram comments, which is quite relevant to this research which uses social media data. Not only the tweet text data would be cleaned, but the lexicon-sentiment-dictionary would also be processed with slang words cleaning so that the word normalization would be inlined between the tweet text and lexicon dictionary. Below is the example of slang and formal words dictionary data, as shown in Figure 4.11:



slang	formal
hallo	halo
kaka	kakak
ka	kak
daah	dah
aaaaahhhh	ah
yaa	ya
smga	semoga
slalu	selalu
amiin	amin
kk	kakak
trus	terus
kk	kakak
sii	sih

Figure 4.11 Slang Words Dictionary

First, load the slang words dictionary file into the Python kernel.

```
import json

# Using slang words dictionary Salsabila
with open("C:/Users/Dell/Documents/Thesis/sentiment_smartfren/_json_colloquial-indonesian-lexicon.txt") as f:
    data = f.read()
# Reconstruction data as 'dict'
lookp_dict = json.loads(data)
```

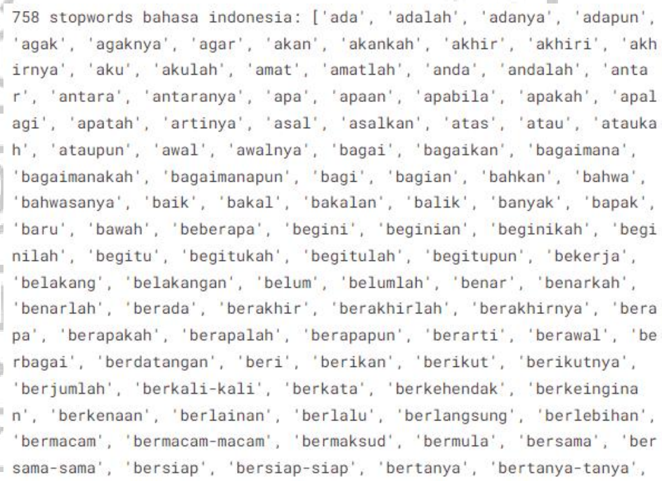
After that, load the tweet text data that has been cleaned previously, then the code will need to read all rows. In the code example below, the tweet text data is non-churn-related tweets. The same code will be applied for the churn-related tweets and lexicon sentiment dictionary as well.

```
import os

# Import the input file
os.chdir("C:/Users/Dell/Documents/Thesis/sentiment_smartfren")
base = "df_nonchurn_cut.txt"

# Open input file and read line by line
input_stream = open(base, 'r')
input_stream_lines = input_stream.readlines()
input_stream.close()

# Separate the text column
text = []
```

```
for line in input_stream_lines:
    text.append(line.split("\t")[0])
```

Next, the cleaned tweet text (output) would be saved in .txt format and ready for the next stage.

After the slang words process was conducted, the stopwords cleaning would be applied to remove meaningless words in terms of sentiment score calculation. Both tweet text and lexicon sentiment dictionary would be cleaned. The process of eliminating stopwords uses two data dictionaries, the first contains conjunctions in Indonesian and the second is a special Twitter data dictionary. The combined 758 Indonesian stopwords data example is as below:



```
758 stopwords bahasa indonesia: ['ada', 'adalah', 'adanya', 'adapun',
'agak', 'agaknya', 'agar', 'akan', 'akankah', 'akhir', 'akhiri', 'akh
irnya', 'aku', 'akulah', 'amat', 'amatlah', 'anda', 'andalah', 'anta
r', 'antara', 'antaranya', 'apa', 'apaan', 'apabila', 'apakah', 'apal
agi', 'apatah', 'artinya', 'asal', 'asalkan', 'atas', 'atau', 'atauka
h', 'ataupun', 'awal', 'awalnya', 'bagai', 'bagaikan', 'bagaimana',
'bagaimanakah', 'bagaimanapun', 'bagi', 'bagian', 'bahkan', 'bahwa',
'bahwasanya', 'baik', 'bakal', 'bakalan', 'balik', 'banyak', 'bapak',
'baru', 'bawah', 'beberapa', 'begini', 'beginian', 'beginikah', 'begi
nilah', 'begitu', 'begitukah', 'begitulah', 'begitupun', 'bekerja',
'belakang', 'belakangan', 'belum', 'belumlah', 'benar', 'benarkah',
'benarlah', 'berada', 'berakhir', 'berakhirlah', 'berakhirnya', 'bera
pa', 'berapakah', 'berapalah', 'berapapun', 'berarti', 'berawal', 'be
rbagai', 'berdatangan', 'beri', 'berikan', 'berikut', 'berikutnya',
'berjumlah', 'berkali-kali', 'berkata', 'berkehendak', 'berkeingina
n', 'berkenaan', 'berlainan', 'berlalu', 'berlangsung', 'berlebihan',
'bermacam', 'bermacam-macam', 'bermaksud', 'bermula', 'bersama', 'ber
sama-sama', 'bersiap', 'bersiap-siap', 'bertanya', 'bertanya-tanya',
```

Figure 4.12 Indonesian Stopwords Example

The codes for removing stopwords from the previous stage can be seen below. First, the text output from the previous step (slang words cleaning step) should be inputted.

```
base = output

input_stream = open(base, 'r')
lines = input_stream.read().splitlines()
input_stream.close()
```

After that, the Indonesian stopwords library should be imported into the kernel.

```
from nltk.corpus import stopwords
```

```
import pandas as pd

# Use stopwords NLTK module to use customized stopwords
stopwords = stopwords.words('indonesian')
print(stopwords[:3])
```

Then, remove stopwords that are included in the lexicon sentiment dictionary and in the tweet text. Eventually, the output of the stopwords cleaning would be created in a new .txt file format.

```
lexicon = pd.read_csv(r"C:\Users\Dell\Documents\Thesis\sentiment_smartfren\sentistrength_id\json_sentiwords_id_modified.csv", sep=",")
lexicon_list = list(lexicon['text'])
len(lexicon_list)

%%time
stopword_list_reduced = []
for stword in stopwords:
    if stword not in lexicon_list:
        stopword_list_reduced.append(stword)
len(stopword_list_reduced)

# Create output file
output = os.path.splitext(base)[0]+'-stop.txt'

count=0
with open(output, 'w') as f:
    # Delete stopwords founded in every lines
    for line in lines:
        s = ""
        words = line.split()
        for w in words:
            if not w in stopword_list_reduced:
                s+=str(w)+" "
        res = s

        # Rewrite text
        f.write(str(res+"\n"))
        count+=1
```

4.3 Sentiment Labelling

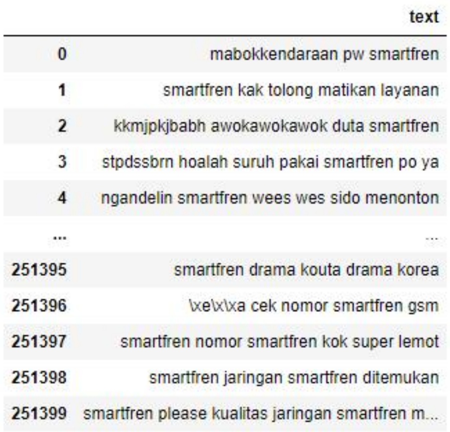
Lexicon-based sentiment analysis would be conducted in this stage. The output of this stage is the sentiment label (positive or negative), based on the sentiment score calculation. In this stage, the sentiment labeling is separated into 2 flows, which are non-churn-related tweets sentiment labeling and churn-related tweets sentiment labeling. As the researcher has explained in the previous chapter, the rules and logic for non-churn-related tweets and churn-related tweets are different, thus the flow should be differentiated. The lexicon dictionary that would be used is the modified sentistrength_id. It was built based on the sentistrength sentiment lexicon model, which is a model that uses linguistic rules and information to detect sentiment values in short English texts. This model lexicon includes a dictionary of sentiments, emoticons, expressions, booster words, negation words, and question words. Except for negation words and question words, all sentstrength_id dictionaries have a value range between -5 to +5. However, the modification of sentistrength_id is different between the non-churn and churn ones.

4.3.1 Non-churn-related Tweets Sentiment Labelling

Valence Aware Dictionary for Sentiment Reasoning (VADER) algorithm is used as the lexicon calculation method, to determine the sentiment label. The VADER lexicon algorithm has been built in the Python Natural Language Toolkit (NLTK) package and is an open-source package that can be customized by the user. Due to the limitation of only the English language which is compatible with the default VADER package, the researcher would replace the English lexicon dictionary built into the package, with the Indonesian lexicon dictionary (modified sentistrength_id). The Indonesian lexicon dictionary used in the sentiment labeling for churn-related tweets consists of a total of 2,323 lexicon keywords.

First, the tweet text needs to be separated from the other variables. It’s because the initial data which is formatted as .csv needs to be converted into .txt. Data which consists of more than 1 variable or column would not be able to be processed as .txt, specifically for the VADER algorithm that we would use for sentiment analysis. Here’s the code and the result of the separated text as shown in Figure 4.13 below:

```
import pandas as pd

# Converting text input from .txt format to dataframe format
Corpus = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/
df_nonchurn_cut-slang-stop.txt", encoding='latin-1',
header=None, sep="\t", names=['text'], usecols=['text'], dtype=str)
```



	text
0	mabokkendaraan pw smartfren
1	smartfren kak tolong matikan layanan
2	kkmjpkjbabh awokawokawok duta smartfren
3	stpdssbrn hoalah suruh pakai smartfren po ya
4	ngandelin smartfren wees wes sido menonton
...	...
251395	smartfren drama kouta drama korea
251396	\xe\x\xa cek nomor smartfren gsm
251397	smartfren nomor smartfren kok super lemot
251398	smartfren jaringan smartfren ditemukan
251399	smartfren please kualitas jaringan smartfren m...

Figure 4.13 Separated Tweet Text

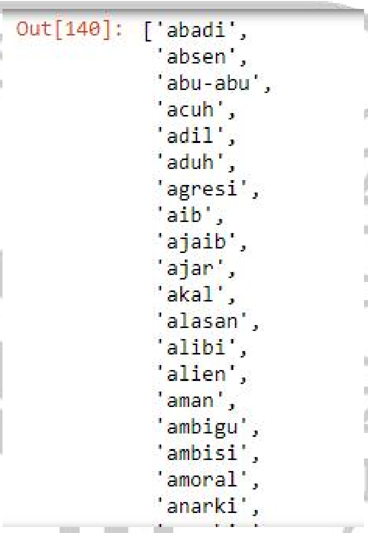
Next, the NLTK VADER package needs to be imported. Then, the default lexicon dictionary could be modified using the Indonesian lexicon. During the process, the lexicon dictionary would be inputted with the separated keywords column and scores column. Below are the codes:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.downloader.download('vader_lexicon')
import json
import reprlib

# using nltk VADER to customize the lexicon
sia1A, sia1B, sia2 = SentimentIntensityAnalyzer(), SentimentIntensityAnalyzer(), SentimentIntensityAnalyzer()
# membersihkan leksikon VADER default
sia1A.lexicon.clear()
sia1B.lexicon.clear()
sia2.lexicon.clear()

separated_senti_text = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/sentistrength_id/separated_senti_text.csv", header=None)
```

```
separated_senti_score = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/sentistrength_id/separated_senti_score.csv", header=None)
```

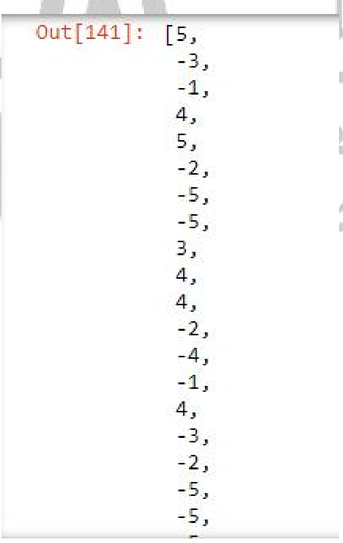
After that, the inputted lexicon dictionary would be converted into .txt format for both the separated keywords and scores. The results are shown in Figures 4.14 and 4.15 below:

```
separated_senti_text_list = separated_senti_text[0].tolist()
separated_senti_score_list = separated_senti_score[0].tolist()
```



```
Out[140]: ['abadi',
           'absen',
           'abu-abu',
           'acuh',
           'adil',
           'aduh',
           'agresi',
           'aib',
           'ajaib',
           'ajar',
           'akal',
           'alasan',
           'alibi',
           'alien',
           'aman',
           'ambigu',
           'ambisi',
           'amoral',
           'anarki',
```

Figure 4.14 Separated Lexicon Dictionary Keywords



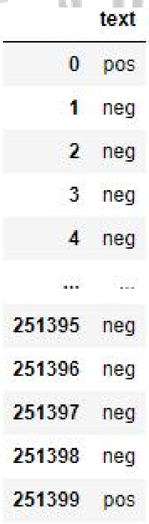
```
Out[141]: [5,
           -3,
           -1,
           4,
           5,
           -2,
           -5,
           -5,
           3,
           4,
           4,
           -2,
           -4,
           -1,
           4,
           -3,
           -2,
           -5,
           -5,
```

Figure 4.15 Separated Lexicon Dictionary Score

After the lexicon dictionary has been inputted, we can do the lexicon dictionary replacement. The lexicon labeling can be done afterward. First, the compound score would be calculated. Then, based on the compound score, we can classify the sentiment as either positive or negative sentiment.

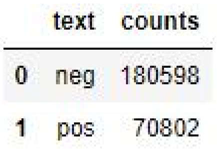
```
sia2.lexicon.update(zip(separated_senti_text_list, separated_senti_score_list))
print(reprlib.repr(sia2.lexicon))

def is_positive_senti(tweet: str) -> bool:
    """True if tweet has positive compound sentiment, False otherwise."""
    return sia2.polarity_scores(tweet)["compound"] > 0

tweets = Corpus["text"]

# Classifying sentiment based on the compound value of sentiwords_id
output = r'C:\Users\Dell\Documents\Thesis\sentiment_smartfren\df_nonchurn_cut-slang-stop-lb-senti.txt'
with open(output, 'w') as f:
    for tweet in tweets:
        if is_positive_senti(tweet) == True:
            label = "pos"
        else:
            label = "neg"
        f.write(str(label+'\n'))
```

The result below shows that from the non-churn-related tweets, there are 180,598 tweets with negative sentiment, while there are 70,802 tweets with positive sentiment. This means, about 72% of all non-churn-related tweets have negative sentiment, and 28% have positive sentiment. The results are shown in Figures 4.16 and 4.17 below:



```
        text
     0   pos
     1   neg
     2   neg
     3   neg
     4   neg
   ...   ...
251395   neg
251396   neg
251397   neg
251398   neg
251399   pos
```

Figure 4.16 Non-churn-related Tweets Sentiment Labels



	text	counts
0	neg	180598
1	pos	70802

Figure 4.17 Non-churn-related Tweets Sentiment Result

4.3.2 Churn-related Tweets Sentiment Labelling

Rule-based lexicon labeling would be implemented for dealing with churn-related tweets. The rule would count the total of words' score for each tweet text, based on each keyword's score contained in the modified sentistrength_id lexicon dictionary. Churn-related lexicon dictionary is different than the Non-churn lexicon dictionary because each keyword has 2-4 words. As the researcher has explained previously, to detect churn-related tweets' sentiment, we need at least 2 words for each keyword within the lexicon dictionary. In total, there are 42,210 churn-related keywords inside the churn-related tweets dictionary. The churn-related tweets dictionary has a lot higher number of keywords because it has many possibilities for word combinations. For example, as the table shows below, the keyword that contains 2 words like "Telkomsel bagus", needs to have another variation of words to deal with the tweet text's possibility. The modified sentistrength_id has been built to handle as many variations of words as possible as shown in Table 4.3 below:

Table 4.3 Lexicon Dictionary Keywords to Handle Churn-related Tweet Text Variations

Tweet Sentence	Keyword Text	Keyword Score
"Sinyal nya Telkomsel bagus ya"	Telkomsel bagus	-5
"Bagus Telkomsel daripada Smartfren"	Bagus Telkomsel	-5
"Bagusnya Telkomsel daripada Smartfren"	Bagusnya Telkomsel	-5

To process the tweet text, the churn-related tweets need to be inputted in .txt format, in which the process is the same as the previous non-churn-related tweets input process. After that, we need to input the negation keywords which consist of Indonesian negative words. Negation keywords need to be added for the rules of the sentiment labeling that later would be explained. And then, the lexicon words are also converted into the list as well. Here's the code for the negation words and lexicon list conversion:

```
negasi = ["tidak", "tidaklah", "tdk", "tak", "tdak", "tidk",
"bukan", "bukanlah", "bukannya","ngga", "nggak", "enggak",
 "nggaknya", "kagak", "gak", "anti", "kurang", "ga", "g",
 "kgk", "not", "no"]

lexicon_word = lexicon['text'].to_list()
#lexicon_num_words = lexicon['number_of_words']
```

Next, the labeling is conducted using the sentiment rules. There are several logical steps of how the rules worked:

1. The rules are contained in the function called "sentiment_scoring". The sentiment_scoring function iterates the rules per text and would continue to examine the next row when the lexicon score for a particular row has been calculated. First, a Sentiment list is created to store the sentiment score result for all the tweet text.

```
def sentiment_scoring(df_tweet):
    sentiment_list = []
```

2. The first level loop is created (for the tweet in df_tweet) to tokenize or separate text sentences into each word using the word_tokenize function. The tokenized words would later be stored in the list called (list_tweet). And the number of words input is counted, to be used for identifying the number of words.

```
    for tweet in df_tweet:
        #print(tweet)
        list_tweet = word_tokenize(tweet)

        #adding 2-4 words to list tweet to be compared with list
         lexicon later
        word_count = len(list_tweet)
```

3. The first second-level loop is then storing the combination of 2-4 words to the list_tweet from the tweet text one by one within the text sentence in a particular row. All the word combinations are then appended to the list_tweet.

```
for i in range(word_count):
    if(word_count - i == 1):
        break;

    text_2_word = str(list_tweet[i] +' '+ list_tweet[i+1])
    list_tweet.append(text_2_word)

    if(word_count - i <= 2):
        continue;

    text_3_word = str(list_tweet[i] +' '+ list_tweet[i+1] +
    ' '+ list_tweet[i+2])
    list_tweet.append(text_3_word)

    if(word_count - i <= 3):
        continue;

    text_4_word = str(list_tweet[i] +' '+ list_tweet[i+1] +
    ' '+ list_tweet[i+2] + ' '+list_tweet[i+3])
    list_tweet.append(text_4_word)
```

4. Next, the listed word combinations within a particular row are compared with the lexicon dictionary. The corresponding keyword's score is stored in "sentiment_temp". But first, the rule would check if the detected keyword score has the negation word before, the sentiment score would be the opposite. For example, "Mahal Smartfren" keyword has a negative sentiment score. But, if that keyword has negation right before it, such as "Enggak mahal Smartfren", then it would later turn the keyword score to a positive sentiment score. When all the word combinations have been checked in a particular row, the rule would accumulate the total score of all the keyword scores that were found on the lexicon score and store the total score in the "sentiment_list". The rule would continue to do the sentiment calculation for the next rows.

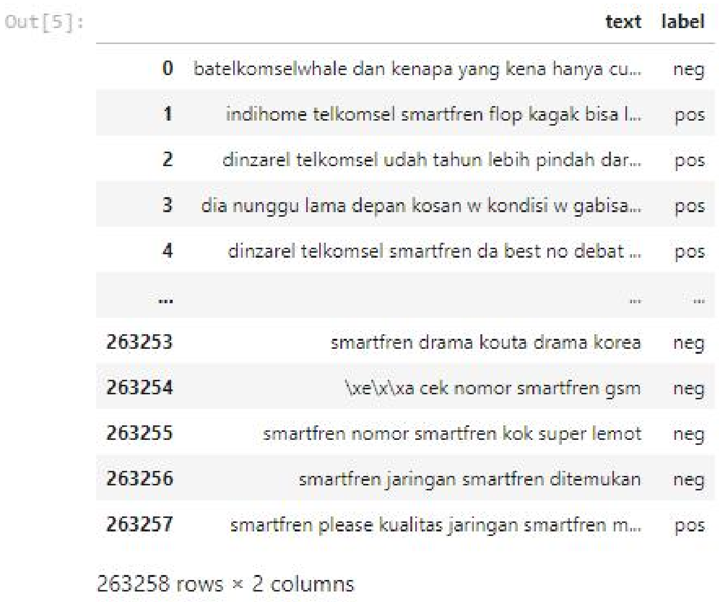
```
    # Sentiment Scoring by lexicon
    sentiment_temp = 0


    for word_counter in range(len(list_tweet)):
        #print(list_tweet[word_counter])

        word_to_search = list_tweet[word_counter]
        word_before = None
        if word_counter > 0:
            word_before = list_tweet[word_counter-1]

        if word_before != None:
            if word_before in negasi:
                sentiment_temp += -
                (lex_dict.get(word_to_search, 0))
                continue;

        sentiment_temp += lex_dict.get(word_to_search,0)


    sentiment_list.append(sentiment_temp)



return sentiment_list
```

4.4 Sentiment Classification and Evaluation

This study uses SVM as a classification algorithm (classifier). The training data and test data were obtained from each feature extraction result (term presence, BoW, and TF-IDF) and lexicon labeling results (InSet and sentistrength_id). The algorithm used for sentiment classification utilizes the svm, accuracy_score, model_selection, and LabelEncoder modules from scikit-learn (Pedregosa et al., 2011). SVM with a linear kernel was chosen as a classifier because the classification carried out in this study was in the form of binary or linear classification. For the record, the random_state parameter in the classifier is used to produce measurable randomization. For reasons of simplicity, the regularization parameter in the classifier is not configured, it is left at default (C=1.0).

First, the SVM classifier should be defined. We also need to set the random state to keep the same result every time we run the classification. Next, the tweet text with its sentiment label data should be inputted. In total, there are 263,258 rows of tweet text that are going to be processed. Below are the codes and the data that is inputted, as shown in Figure 4.18:

```
clf = svm.SVC(kernel='linear', random_state=42)
.chdir(r'C:\Users\Freekek\Documents\Thesis\sentiment_smartfren')
 base = r'C:/Users/Freekek/Documents/Thesis/sentiment_smartfren/for_classification.txt'
Corpus = pd.read_csv(base, encoding='latin-1', header=None, sep='\t', names=['text', 'label'], dtype=str)
```



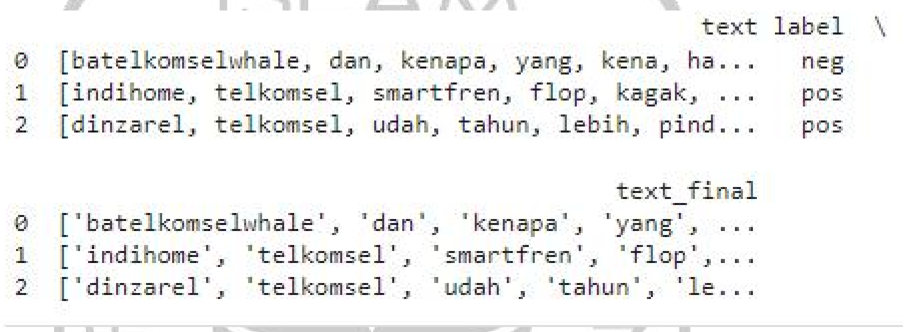
```
Out[5]:
                                                    text  label
     0  batelkomselwhale dan kenapa yang kena hanya cu...   neg
     1     indihome telkomsel smartfren flop kagak bisa l...   pos
     2    dinzarel telkomsel udah tahun lebih pindah dar...   pos
     3   dia nunggu lama depan kosan w kondisi w gabisa...   pos
     4     dinzarel telkomsel smartfren da best no debat ...   pos
   ...                                                 ...   ...
263253                     smartfren drama kouta drama korea   neg
263254                        \xe\x\xa cek nomor smartfren gsm   neg
263255                 smartfren nomor smartfren kok super lemot   neg
263256                    smartfren jaringan smartfren ditemukan   neg
263257     smartfren please kualitas jaringan smartfren m...   pos

263258 rows × 2 columns
```

Figure 4.18 Data Input for Classification

Next, the data input would be split into a list of words. But before that, we need to define that the data training and testing would be tested using the same dataset. Then, every sentence can be broken down into a list of words (words tokenization). Here are the codes and the result of the tokenization, as shown in Figure 4.19 below:

```
# Define that we're going to use the inputted data for both the training and testing process
LL = Corpus[['label']]
LLmark = 0

# Step - a : Delete the blank rows (if exist)
Corpus['text'].dropna(inplace=True)
# # Step - b : lower case all the words
# Corpus['text'] = [entry.lower() for entry in Corpus['text']] # we've done this in '[1] text cleaning.ipynb'
# Step - c : Tokenisasi : Breaking the sentence into list of words
```

```
Corpus['text']= [word_tokenize(entry) for entry in Corpus['text']]

for index,entry in enumerate(Corpus['text']):
    # Define list to save the list of words according to the rule
    Final_words = []
    for word in entry:
        # Consider only the alphabet
        if word.isalpha():
            word_Final = word
            Final_words.append(word_Final)
    Corpus.loc[index,'text_final'] = str(Final_words)
```

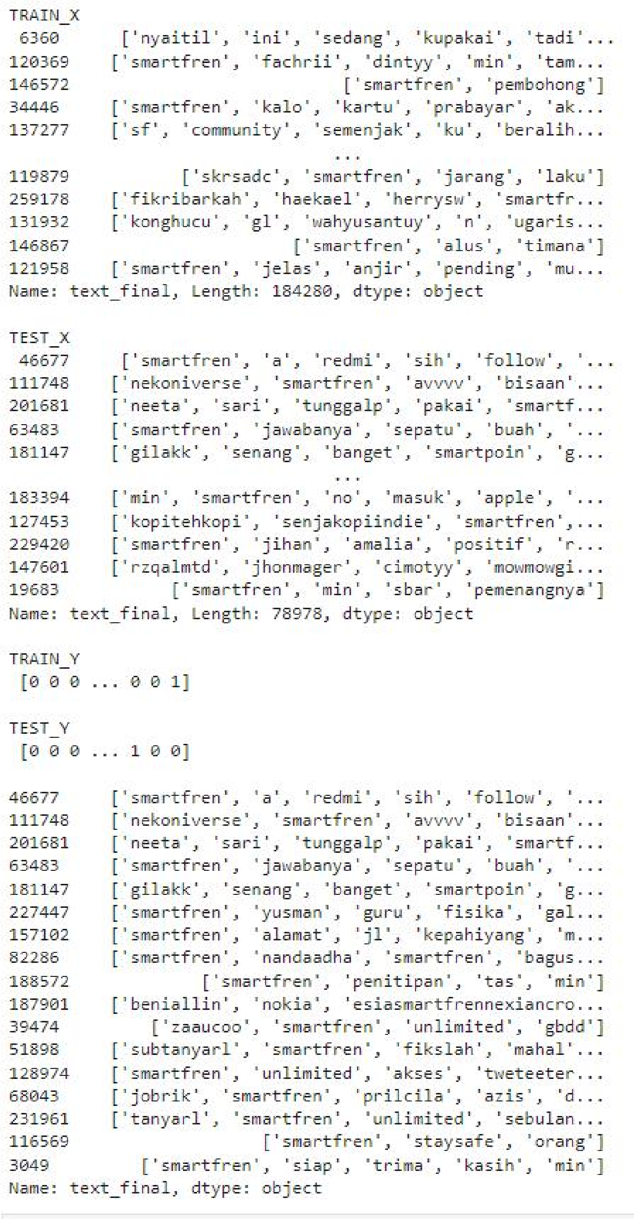


```
                                                text label  \
0  [batelkomselwhale, dan, kenapa, yang, kena, ha...   neg
1  [indihome, telkomsel, smartfren, flop, kagak, ...   pos
2  [dinzarel, telkomsel, udah, tahun, lebih, pind...   pos

                                          text_final
0  ['batelkomselwhale', 'dan', 'kenapa', 'yang', ...
1  ['indihome', 'telkomsel', 'smartfren', 'flop',...
2  ['dinzarel', 'telkomsel', 'udah', 'tahun', 'le...
```

Figure 4.19 Word Tokenization for Classification

Next, the data input would be split into test and train data. The data is split with a 70:30 ratio. It means, 70% of the data would be used for training and 30% of the data would be used for testing. During the process, the sentiment label (positive and negative) is also encoded into 0 and 1. Here are the codes and the output as shown in Figure 4.20 below:

```
#Split the data with 70:30 ratio
Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(Corpus['text_final'],
LL['label'],test_size=0.3, random_state=42)
Train_Y_Actual, Test_Y_Actual = model_selection.train_test_split(Corpus['label'],
test_size=0.3, random_state=42)

# Encode the sentiment label into numbers between 0 and class_n-1
Encoder = LabelEncoder()
Train_Y = Encoder.fit_transform(Train_Y)
Test_Y = Encoder.fit_transform(Test_Y)
Train_Y_Actual = Encoder.fit_transform(Train_Y_Actual)
Test_Y_Actual = Encoder.fit_transform(Test_Y_Actual)
```



```
TRAIN_X
 6360       ['nyaitil', 'ini', 'sedang', 'kupakai', 'tadi'...
120369     ['smartfren', 'fachrii', 'dintyy', 'min', 'tam...
146572                          ['smartfren', 'pembohong']
34446      ['smartfren', 'kalo', 'kartu', 'prabayar', 'ak...
137277     ['sf', 'community', 'semenjak', 'ku', 'beralih...
                                  ...
119879            ['skrsadc', 'smartfren', 'jarang', 'laku']
259178     ['fikribarkah', 'haekael', 'herrysw', 'smartfr...
131932     ['konghucu', 'gl', 'wahyusantuy', 'n', 'ugaris...
146867                     ['smartfren', 'alus', 'timana']
121958     ['smartfren', 'jelas', 'anjir', 'pending', 'mu...
Name: text_final, Length: 184280, dtype: object

TEST_X
 46677      ['smartfren', 'a', 'redmi', 'sih', 'follow', '...
111748     ['nekoniverse', 'smartfren', 'avvvv', 'bisaan'...
201681     ['neeta', 'sari', 'tunggalp', 'pakai', 'smartf...
63483      ['smartfren', 'jawabanya', 'sepatu', 'buah', '...
181147     ['gilakk', 'senang', 'banget', 'smartpoin', 'g...
                                  ...
183394     ['min', 'smartfren', 'no', 'masuk', 'apple', '...
127453     ['kopitehkopi', 'senjakopiindie', 'smartfren',...
229420     ['smartfren', 'jihan', 'amalia', 'positif', 'r...
147601     ['rzqalmtd', 'jhonmager', 'cimotyy', 'mowmowgi...
19683           ['smartfren', 'min', 'sbar', 'pemenangnya']
Name: text_final, Length: 78978, dtype: object

TRAIN_Y
 [0 0 0 ... 0 0 1]

TEST_Y
 [0 0 0 ... 1 0 0]

46677      ['smartfren', 'a', 'redmi', 'sih', 'follow', '...
111748     ['nekoniverse', 'smartfren', 'avvvv', 'bisaan'...
201681     ['neeta', 'sari', 'tunggalp', 'pakai', 'smartf...
63483      ['smartfren', 'jawabanya', 'sepatu', 'buah', '...
181147     ['gilakk', 'senang', 'banget', 'smartpoin', 'g...
227447     ['smartfren', 'yusman', 'guru', 'fisika', 'gal...
157102     ['smartfren', 'alamat', 'jl', 'kepahiyang', 'm...
82286      ['smartfren', 'nandaadha', 'smartfren', 'bagus...
188572              ['smartfren', 'penitipan', 'tas', 'min']
187901     ['beniallin', 'nokia', 'esiasmartfrennexiancro...
39474          ['zaaucoo', 'smartfren', 'unlimited', 'gbdd']
51898      ['subtanyarl', 'smartfren', 'fikslah', 'mahal'...
128974     ['smartfren', 'unlimited', 'akses', 'tweteeter...
68043      ['jobrik', 'smartfren', 'prilcila', 'azis', 'd...
231961     ['tanyarl', 'smartfren', 'unlimited', 'sebulan...
116569                    ['smartfren', 'staysafe', 'orang']
3049          ['smartfren', 'siap', 'trima', 'kasih', 'min']
Name: text_final, dtype: object
```

Figure 4.20 Data Split and Label Encoding

After doing so, the data is ready for the classification process using SVM. The classification would be separated into 3 ways using different feature extractions, which are Count Vectorizer, Bags of Words (BoW), and Term Frequency and Inverse Document Frequency (TF-IDF). As mentioned in the previous part, the result of classification using each feature extraction would be compared afterward. Here are the codes for Count Vectorizer SVM Classification:

```
from sklearn.feature_extraction.text import CountVectorizer

# binary=True means the frequency is not considered
vectorizerTP = CountVectorizer(binary=True)
```

```
X = vectorizerTP.fit_transform(Corpus['text_final'])

# Transform Train_X and Test_X to Count Vectorizer vector
Train_X_TP = vectorizerTP.transform(Train_X)
Test_X_TP = vectorizerTP.transform(Test_X)

# fitting training data with classifier
clf.fit(Train_X_TP,Train_Y)
# predicting trained data using testing data
predictions_SVM_TP = clf.predict(Test_X_TP)

# Using accuracy function to get the accuracy number
accuracy_tp = accuracy_score(Test_Y_Actual, predictions_SVM_TP)*100
print('SVM Accuracy Score -> ', accuracy_tp)

# Comparing Lexicon value with prediction value
df = pd.DataFrame({'Lexicon Values':Test_Y, 'Predicted Values':predictions_SVM_TP})
```

Here are the codes for Bags of Words (BoW) SVM Classification:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(Corpus['text_final'])

# Transform Train_X dan Test_X to BoW vector
Train_X_BoW = vectorizer.transform(Train_X)
Test_X_BoW = vectorizer.transform(Test_X)

# fitting training data on classifier
clf.fit(Train_X_BoW,Train_Y)
# predicting trained data using testing data
predictions_SVM_BoW = clf.predict(Test_X_BoW)

# Using accuracy function to get the accuracy number
accuracy_bow = accuracy_score(Test_Y_Actual, predictions_SVM_BoW)*100
print('SVM Accuracy Score -> ',accuracy_bow)
```

Here are the codes for TF-IDF SVM Classification:

```
Tfidf_vect = TfidfVectorizer()
Tfidf_vect.fit(Corpus['text_final'])

X = Tfidf_vect.fit_transform(Corpus['text_final'])
```

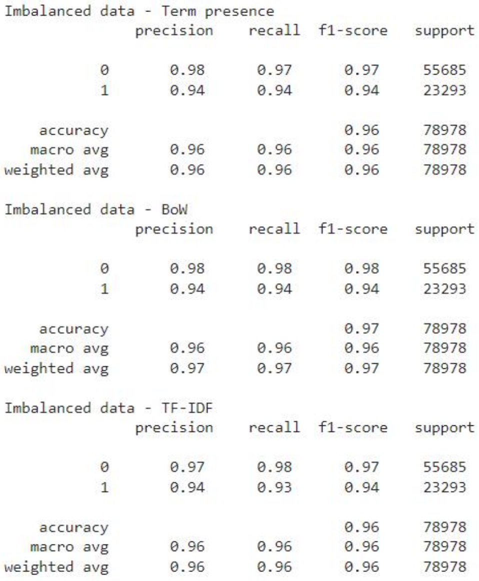
```

# Transform Train_X dan Test_X to TF-IDF vector
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)

# fitting training data on classifier
clf.fit(Train_X_Tfidf,Train_Y)
# predicting trained data using testing data
predictions_SVM_Tfidf = clf.predict(Test_X_Tfidf)

# Using accuracy function to get the accuracy number
accuracy_tfidf = accuracy_score(Test_Y_Actual, predictions_SVM_Tfidf)*100
print('SVM Accuracy Score -> ',accuracy_tfidf)
```

To evaluate the result of the prediction, we need to use a confusion matrix to get the precision, accuracy, and f1-score value for each of the feature extraction results. The evaluation result would be compared, and the result with the higher evaluation value would be chosen for the most optimal result of SVM classification. Here are the codes to evaluate the prediction.

```
# Create confusion matrix from prediction label againts actual label
from sklearn.metrics import confusion_matrix


y_true = Test_Y_Actual

## Term presence ##
print('Confusion Matrix - Term presence')
y_pred = predictions_SVM_TP
conf_matrix = confusion_matrix(y_true, y_pred, labels=[1,0])
print(conf_matrix, conf_matrix.sum())

## BoW ##
print('\nConfusion Matrix - BoW')
y_pred = predictions_SVM_BoW
conf_matrix = confusion_matrix(y_true, y_pred, labels=[1,0])
print(conf_matrix, conf_matrix.sum())

## TF-IDF ##
print('\nConfusion Matrix - TF-IDF')
```

```
y_pred = predictions_SVM_Tfidf
conf_matrix = confusion_matrix(y_true, y_pred, labels=[1,0])
print(conf_matrix, conf_matrix.sum())
```

Here's the result of the precision, accuracy, and f1-score for each feature extraction classification, as shown in Figure 4.21 below:



```
Imbalanced data - Term presence
              precision    recall  f1-score   support

           0       0.98      0.97      0.97     55685
           1       0.94      0.94      0.94     23293

    accuracy                           0.96     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.96      0.96      0.96     78978

Imbalanced data - BoW
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     55685
           1       0.94      0.94      0.94     23293

    accuracy                           0.97     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.97      0.97      0.97     78978

Imbalanced data - TF-IDF
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     55685
           1       0.94      0.93      0.94     23293

    accuracy                           0.96     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.96      0.96      0.96     78978
```

Figure 4.21 Evaluation Result

4.5 Topic Categorization

In the topic categorization stage, the labeled data would be categorized based on its text topic which is related to the telecommunication company topic. The categorization is conducted using keywords inside the text. In this case, the text could have 1 or more than 1 topic, because sometimes Twitter users are not only talking about 1 topic. For example, in the tweet text like this, "Kemarin koneksi jaringan Smartfren lemot banget, udah mahal lagi harganya", the tweet can be categorized for both network and product (price) topic. During the visualization later, we will be able to select multiple tweet topics, because the researcher would like to use binary values (0 and 1) to define a particular topic. Each topic would have its own column, where 1 means that the tweet text belongs to a particular category, and 0 means that the tweet text does not belong to a particular category. Below is a simple example of the expected output for topic categorization:

Table 4.4 Topic Categorization Labelling Result

Tweet examples	Network	Product	Gaming	Streaming

	Experience	Experience	Experience	Experience
“Smartfren murah banget”	0	1	0	0
“Keren smartfren udah murah, lancar juga jaringan nya”	1	1	0	0
“Buat nge game sama nonton film Smartfren lancar banget”	1	0	1	1

First, the keywords which defined each topic should be listed. There are dozens of keywords, including any possible variations of keywords that are listed. Therefore the details of these keywords are attached in the Appendix. After the keywords are listed, the rules are created to search any keywords contained within each tweet text. Below are the codes for detecting the topics:

```
# network_keyword_level_1
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(network_keyword_level_1)), 'network_category_level_1'] = 1
sf_tweet['network_category_level_1'] = sf_tweet['network_category_level_1'].fillna(value=0)

# product_keyword_level_1
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(product_keyword_level_1)), 'product_category_level_1'] = 1
sf_tweet['product_category_level_1'] = sf_tweet['product_category_level_1'].fillna(value=0)

# customer_service_keyword_level_1
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(customer_service_keyword_level_1)), 'customer_service_category_level_1'] = 1
sf_tweet['customer_service_category_level_1'] = sf_tweet['customer_service_category_level_1'].fillna(value=0)

# video_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(video_keyword_level_2)), 'video_category_level_2'] = 1
sf_tweet['video_category_level_2'] = sf_tweet['video_category_level_2'].fillna(value=0)

# games_keyword_level_2
```

```
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(games_keyword_level_2)), 'games_category_level_2'] = 1
sf_tweet['games_category_level_2'] = sf_tweet['games_category_level_2'].fillna(value=0)

# social_media_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(social_media_keyword_level_2)), 'social_media_category_level_2'] = 1
sf_tweet['social_media_category_level_2'] = sf_tweet['social_media_category_level_2'].fillna(value=0)

# video_call_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(video_call_experience_keyword_level_2)), 'video_call_experience_category_level_2'] = 1
sf_tweet['video_call_experience_category_level_2'] = sf_tweet['video_call_experience_category_level_2'].fillna(value=0)

# voice_call_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(voice_call_experience_keyword_level_2)), 'voice_call_experience_category_level_2'] = 1
sf_tweet['voice_call_experience_category_level_2'] = sf_tweet['voice_call_experience_category_level_2'].fillna(value=0)

# download_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(download_experience_keyword_level_2)), 'download_experience_category_level_2'] = 1
sf_tweet['download_experience_category_level_2'] = sf_tweet['download_experience_category_level_2'].fillna(value=0)

# upload_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(upload_experience_keyword_level_2)), 'upload_experience_category_level_2'] = 1
sf_tweet['upload_experience_category_level_2'] = sf_tweet['upload_experience_category_level_2'].fillna(value=0)

# coverage_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(coverage_keyword_level_2)), 'coverage_category_level_2'] = 1
sf_tweet['coverage_category_level_2'] = sf_tweet['coverage_category_level_2'].fillna(value=0)
```

4.6 Sudden Sentiment Change Tagging

In this stage, the tweets are going to be categorized based on their sentiment change, whether a particular tweet from a Twitter user has a higher or lower sentiment score from his/her previous tweet which mentions Smartfren. The rule is created to tag the tweet sentiment change. However, the sentiment change tagging would be applied for the tweets that have a sentiment score change from a positive score to a negative score, and vice versa.

```
def numbering_tweet_by_username_and_sentiment_change(x):
    # Sort the tweets based on the upload dates of the tweets, for each
      username of the Twitter account
    # Then, add the value for number of tweets by username
    df_numbering = sf_tweet[sf_tweet['username'] == x].sort_values(by='date')
    for i in range(df_numbering.shape[0]):
        temp = int(df_numbering[i:i+1].tweet_id)
        sf_tweet.loc[sf_tweet.tweet_id == temp, 'number_of_tweet_by_username']
        = int(i+1)

    # Compare the score between 2 tweets, then the polarity of those 2 tweets
    if df_numbering.shape[0] > 1:
        for i in range(1, df_numbering.shape[0]+1):
            if i == df_numbering.shape[0]:
                break;

            i = int(i)

            id_a = df_numbering[i-1:i].tweet_id.values[0]
            id_b = df_numbering[i:i+1].tweet_id.values[0]

            score_a = df_numbering.loc[df_numbering.tweet_id==id_a,
                     'sentiment_score'].values[0]
            score_b = df_numbering.loc[df_numbering.tweet_id==id_b,
                     'sentiment_score'].values[0]

            polarity_a = df_numbering.loc[df_numbering.tweet_id==id_a,
                       'sentiment'].values[0]
            polarity_b = df_numbering.loc[df_numbering.tweet_id==id_b,
                       'sentiment'].values[0]

            if score_b > score_a:
                sf_tweet.loc[sf_tweet.tweet_id == id_b,
            'Sentiment_changing_before_by_username'] = 'Sentiment_Up'
        elif score_b < score_a:
            sf_tweet.loc[sf_tweet.tweet_id == id_b,
            'Sentiment_changing_before_by_username'] = 'Sentiment_Down'
        else:
            sf_tweet.loc[sf_tweet.tweet_id==id_b,
            'Sentiment_changing_before_by_username'] = 'Sentiment_stable'

        if polarity_a=='neg' > polarity_b=='pos':
            sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sudden_Changes'] =
            'to_positive'
        elif polarity_a=='pos' > polarity_b=='neg':
            sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sudden_Changes'] =
            'to_negative'
        else:
            sf_tweet.loc[sf_tweet.tweet_id==id_b, 'Sudden_Changes'] =
            'no_change'
```

CHAPTER V

Discussions

This chapter will present the results of the data processing, which includes the data insights and sentiment accuracy explanations. The research questions that have been described at the beginning of the research document are going to be answered in this chapter. Data insights elaboration is being done by using data visualizations that have been processed using Power BI software, while the sentiment accuracy explanation would be based on the result of sentiment classification and evaluation that has been gathered from Chapter 4.

5.1 Data Insights

The data insights would be discussed in 4 main parts which would also answer the research questions. All main parts are the result has been adjusted to Smartfren business questions regarding social media analysis back in 2021. It would be visualized in the form of a data dashboard, as it would be more convenient to read and understand the data. The dashboard is consisting of 7 pages in total.

5.1.1 The Total of Positive and Negative Tweets and The Most Discussed Customer Experience Aspects

There are 2 pages of data dashboard which would elaborate the total of positive and negative tweets, as well as the most discussed aspects of customer experience. On the first page, we can see several parts of data visualization:

1. Total tweet sentiment by month and Percentage of the tweet
2. Total percentage of level 1 experience and the total amount of level 2 network experience

Below is the result of the page 1 data dashboard that represents the total tweet sentiment from January 2019 to September 2021 as shown in Figure 5.1:

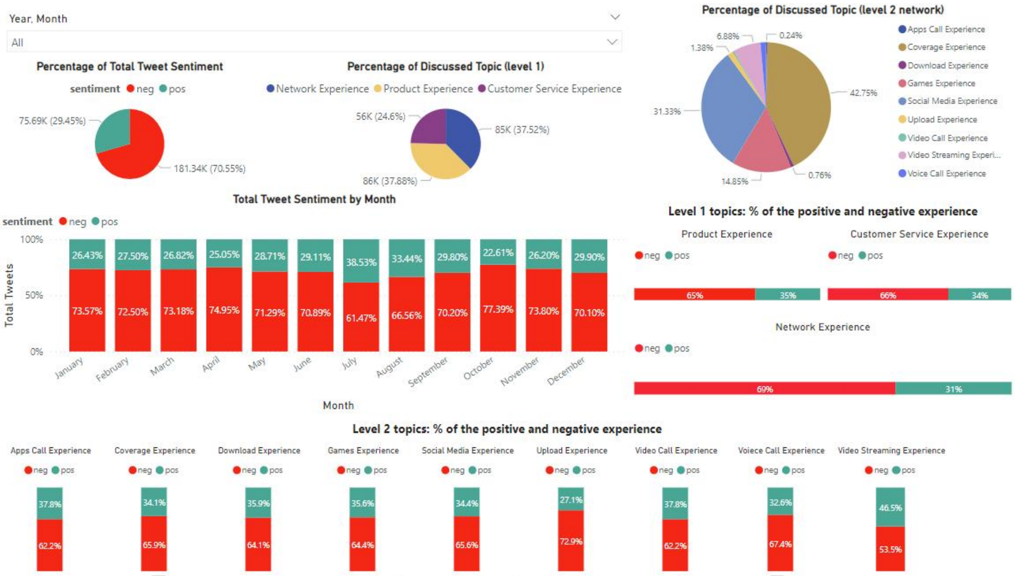


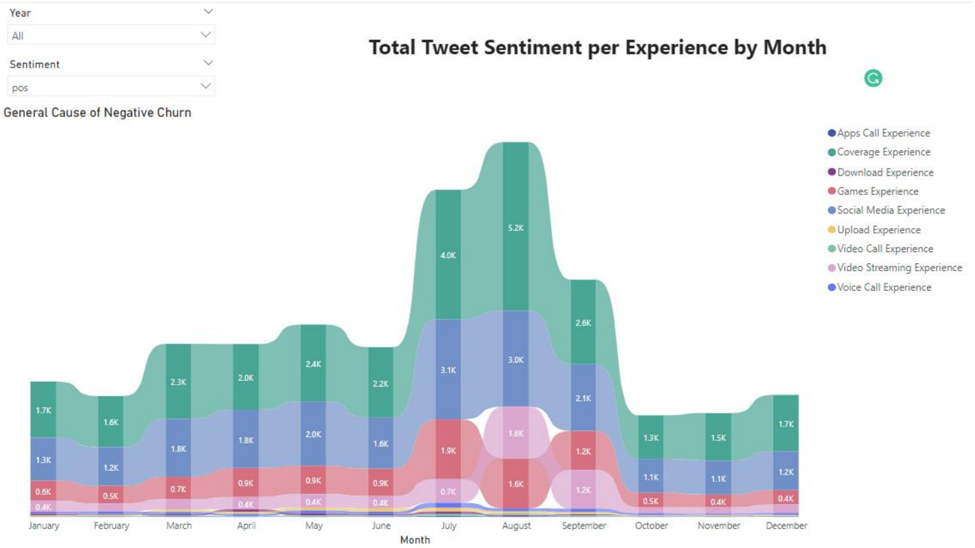
Figure 5.1 The Total of Positive and Negative Tweets and The Most Discussed Customer Experience Aspects

From figure 30 above, we can see that in total there are more negative tweets compared to positive ones, with 70.5% or 181k of negative tweets and only 29.5% or 75.7k of positive tweets, which come from Twitter users who mentioned Smartfren. Over 2.5 years, the data also shows that every month, the total of negative tweets is also higher than positive ones, October with the highest percentage of negative tweets (77.4%), while July has the lowest percentage of negative tweets (61.5%).

Moving on to the most discussed aspects of customer experience, both product experience and network experience are the most discussed topics which represent 37.8% and 37.5% of all tweets level 1 topics being discussed respectively. If we break down the most discussed network experience based on level 2 topics, coverage experience is the most discussed topic (42.7%), followed by social media experience (31.3%) and games experience (14.8%).

Going into the sentiment of the experience aspects itself, all the level 1 and level 2 topics have a higher number of negative tweets. In terms of level 1 topics, network experience has the highest negative tweets (69%), followed by customer service experience (66%), and product experience (65%). Level 2 network-related topics breakdown shows that the upload experience has the highest proportion of negative tweets (72.9%), while the video streaming experience has the lowest proportion of negative tweets (53.4%).

Figures below are page 2 of the data dashboard which shows the amount of positive and negative tweets for all network-related level 2 topics in each month. The visualization below shows the accumulation of tweets over 2.5 years for each month because there might be changes in the most discussed level 2 topics for each month as shown in Figures 5.2 and 5.3 below:

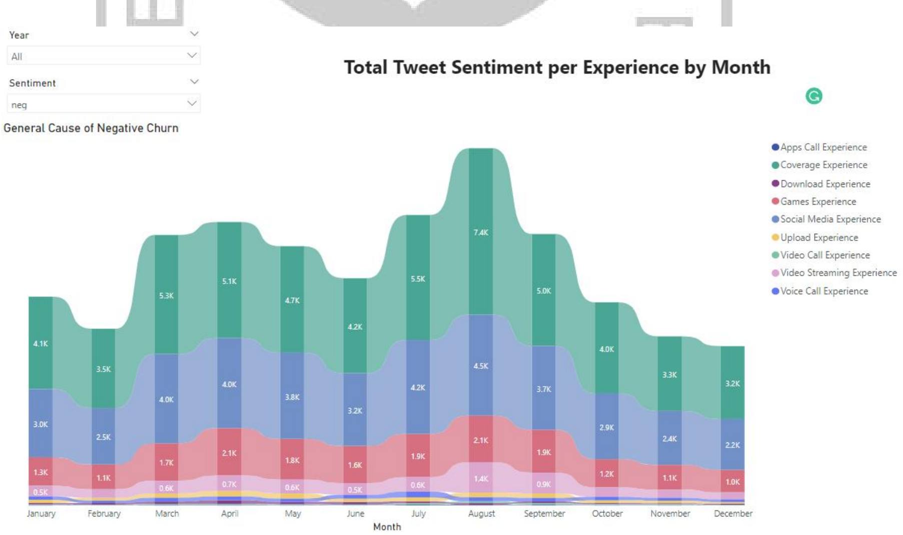


Year: All

Sentiment: pos

General Cause of Negative Churn

Figure 5.2 Total of Positive Tweets per Experience by Month



Year: All

Sentiment: neg

General Cause of Negative Churn

Figure 5.3 Total of Negative Tweets per Experience by Month

Figures 5.2 and 5.3 above show that the highest amount of positive and negative tweets throughout the months are mainly consisting of coverage experience, social media experience, and games experience. Although in August, video streaming experience jumped into the top 3 most discussed topics for positive sentiment.

5.1.2 The Total of Sudden Sentiment Changes and Cause of Sentiment Change

There are 3 pages of data dashboards that would elaborate on the total of positive and negative sudden sentiment change, as well as the causes of sentiment change. Positive sudden sentiment change means that the tweet from a particular user who mentioned Smartfren suddenly changed from negative to positive sentiment, and vice versa. On the first page, we can see several parts of data visualization:

1. Total of sudden sentiment change and the effect of engagement rate
2. General causes of sudden sentiment change
3. Detailed causes of sudden sentiment change

Below is the result of the page 1 data dashboard that represents the total tweet sentiment from January 2019 to September 2021, as shown in Figure 5.4:

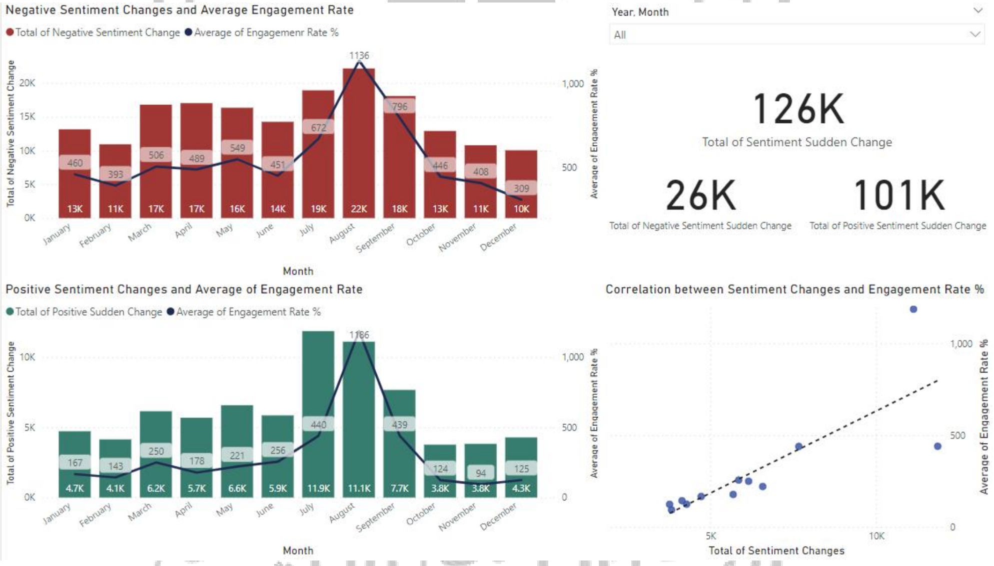


Figure 5.4 Total of sudden sentiment change and the effect of engagement rate

According to Figure 5.4 above, the total of all sudden sentiment change over 2.5 years is around 126k. The total of positive sudden sentiment change is 101k, which is 4 times higher than the total of negative sudden sentiment change (only 26k). Throughout the months, August has the highest total amount of negative sentiment sudden change, while July has the highest total amount of positive sentiment sudden change. It can be seen, that based on the correlation line, the impact of engagement rate is significant towards the total amount of sudden sentiment change. The fluctuations of every month's total sentiment change are proportional to the increase or decrease in the average engagement rate. When the average engagement rate is higher, then the sudden sentiment change total would also be higher in a certain month. It means, the engagement rate which defines how impactful the tweet opinions are, does affect the other users' perception or sentiment. The details are shown in Figures 5.5 and 5.6 below:

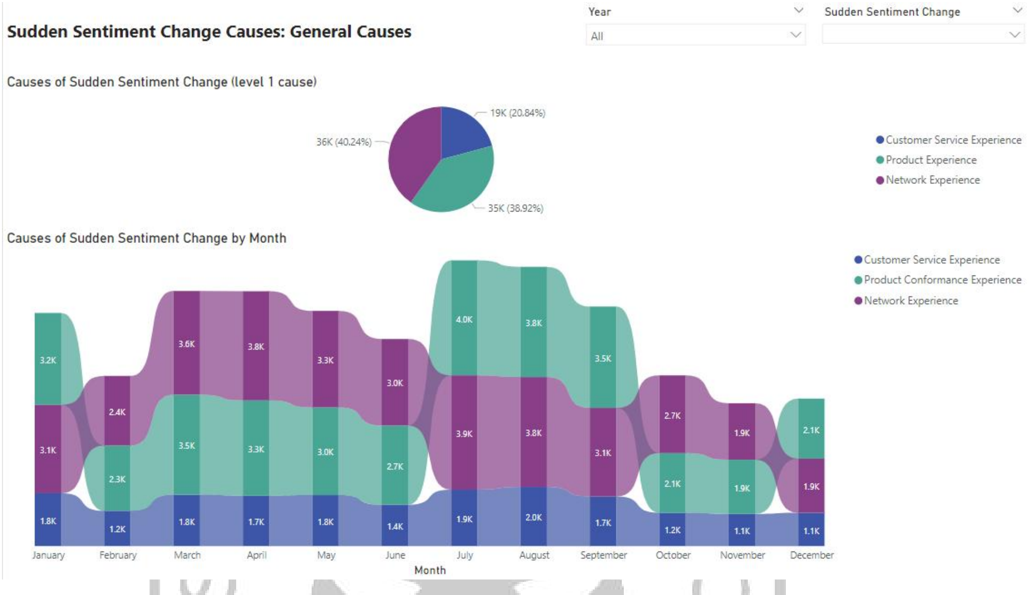


Figure 5.5 Total of Positive Sudden Sentiment Change Level 1 Cause by Month

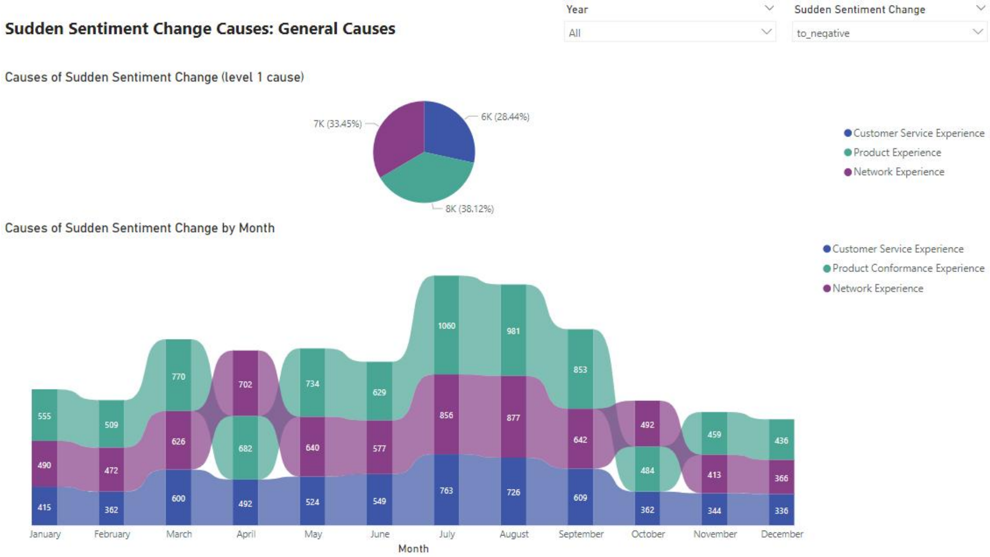
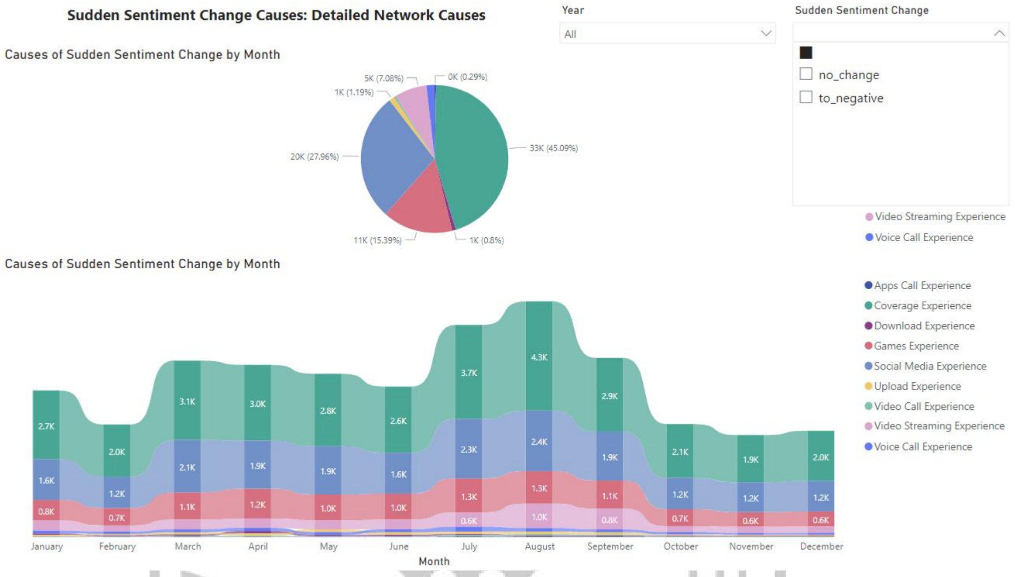


Figure 5.6 Total of Negative Sudden Sentiment Change Level 1 Cause by Month

Based on figures 5.5 and 5.6 above, most of the sudden sentiment changes are caused by product experience and network experience. About more than 30-40% of all positive and negative sudden sentiment changes are caused by those 2 experiences. Based on this finding, the worthiness of the products (product prices and irrelevant product packages) is the most prominent caused by the sentiment change. In January, July-September, and December, the highest cause of positive sudden sentiment change is product experience, while network experience is the highest cause for the other months. For negative sudden sentiment change, all months have product experience as the highest cause of sudden sentiment change, except for April and October where network experience is the main issue. The details are shown in Figures 5.7 and 5.8 below:

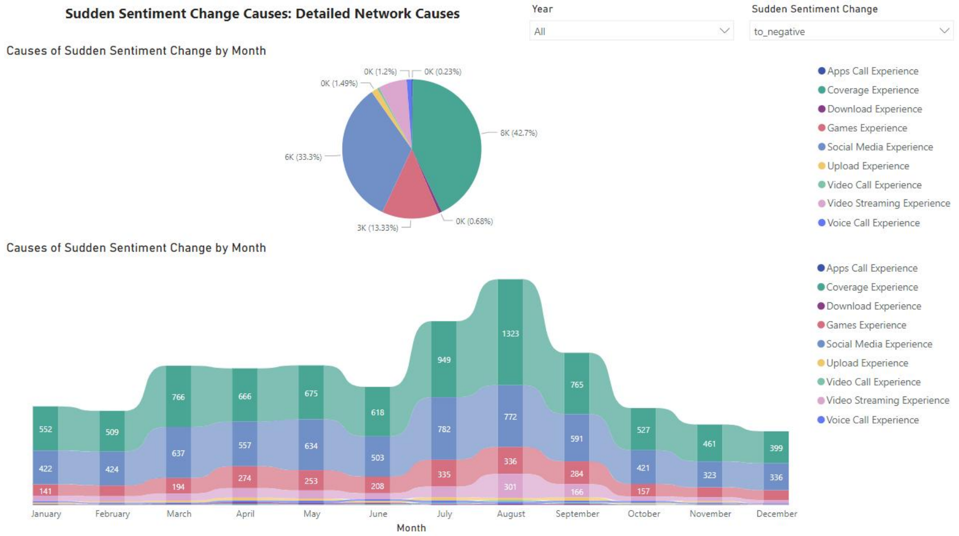


Sudden Sentiment Change Causes: Detailed Network Causes

Year: All

Sudden Sentiment Change: ■ □ no_change □ to_negative

Figure 5.7 Total of Positive Sudden Sentiment Change per Level 2 Network Cause by Month



Sudden Sentiment Change Causes: Detailed Network Causes

Year: All

Sudden Sentiment Change: to_negative

Figure 5.8 Total of Negative Sudden Sentiment Change per Level 2 Network Cause by Month

Figures 5.7 and 5.8 above shows that for both positive and negative sudden sentiment change, the highest causes are coverage experience (around 42-45% of total sudden sentiment change), social media experience (around 28-33% of total sudden sentiment change), and games experience (around 15-33% of total sudden sentiment change). Over several months, the top 3 highest causes are still the same as the total ones for each month.

5.1.3 The Total of Churn-related Tweets and Cause of Churn

There are 3 pages of data dashboard which would elaborate the total of positive and negative churn, as well as the causes of churn. Positive churn means that there's an indication of users converting from another telecommunication provider to use Smartfren products, and vice versa. On the first page, we can see several parts of data visualization:

1. Total churn and the effect of engagement rate
2. General causes of churn
3. Detailed causes of churn

Below is the result of the page 1 data dashboard that represents the total tweet sentiment from January 2019 to September 2021, as shown in Figure 5.9:

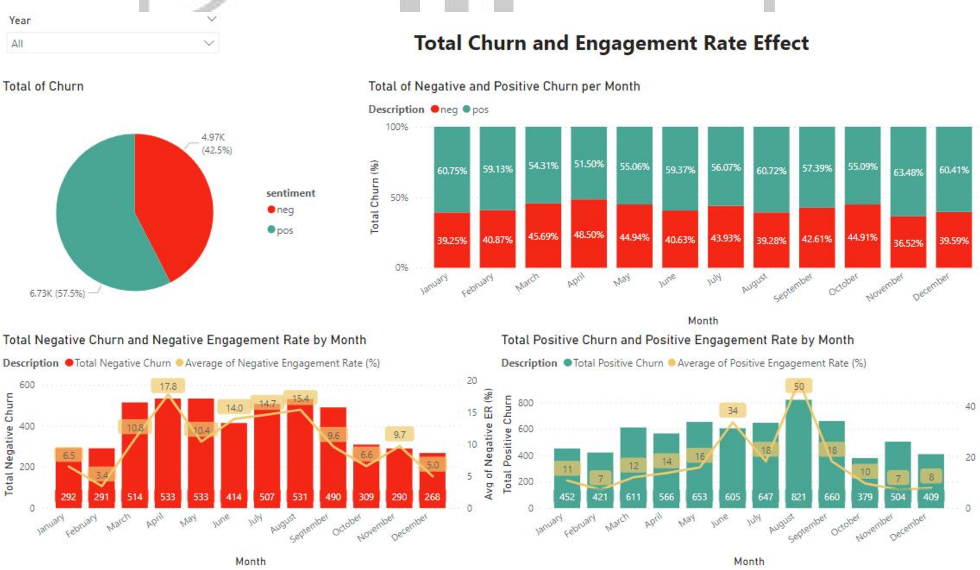


Figure 5.9 Total Churn and The Effect of Engagement Rate

According to Figure 5.9 above, over 2.5 years, there are more tweets, which indicated positive churn compared to negative churn, with 57.5% (6.7k) of churn-related tweets considered positive churn, and 42.5% (4.9k) considered negative churn. Throughout the entire month, positive churn always has a higher proportion than negative churn. The highest proportion of positive churn happened in November (63.5%), while the lowest proportion happened in April (51.5%).

The impact of positive and negative tweets' engagement rates is significant towards the amount of churn that happened each month. If we look at the total of negative churn and negative engagement rate by month, we can see that with the higher engagement rate of negative tweets in a particular month, the amount of negative churn will also be higher in the next month, and vice versa. The same effect happened with the positive churn as well. It's because, if the average engagement rate of negative tweets is higher, the likeliness of other users' perceptions who saw the tweets to be affected by the tweets' sentiment would be higher. It means that the engagement rate represents how impactful the tweets are to the other users' perspective towards Smartfren, as shown in Figure 5.10 below:

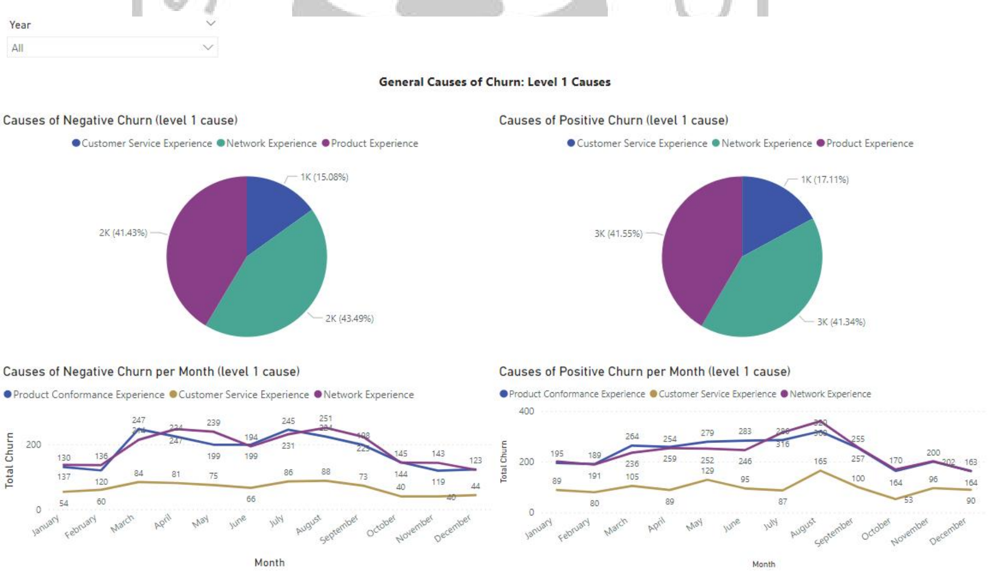
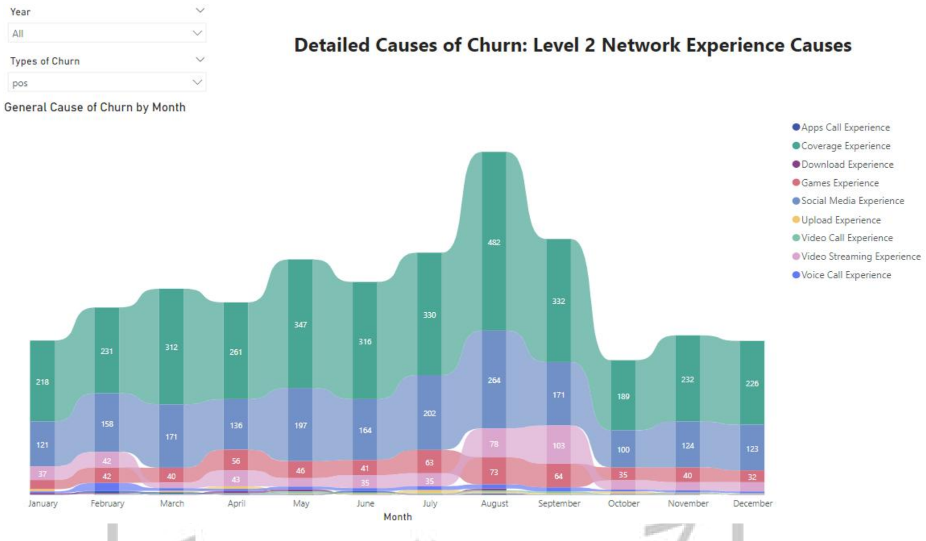


Figure 5.10 Level 1 Causes of Churn

Figure 5.10 above shows the causes of negative and positive churn, based on level 1 causes. It can be seen, that both negative and positive churn is caused by mostly network experience and product experience. Each cause represents around more than 40% of all churn causes. It means network stability for certain uses and product worthiness (price and package) are the most prominent causes of churn. For each month, the highest cause of churn was dominated by either network issues or product issues, as shown in Figures 5.11 and 5.12 below:

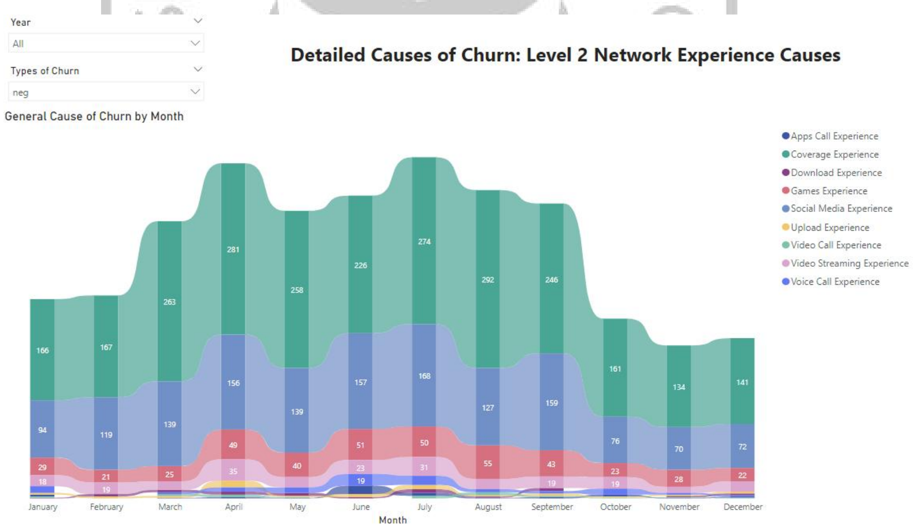


Year: All

Types of Churn: pos

Detailed Causes of Churn: Level 2 Network Experience Causes

Figure 5.11 Total of Positive Churn per Level 2 Network Causes by Month



Year: All

Types of Churn: neg

Detailed Causes of Churn: Level 2 Network Experience Causes

Figure 5.12 Total of Negative Churn per Level 2 Network Causes by Month

Figures 5.11 and 5.12 above show that the highest amount of positive and negative churn throughout the months is mainly caused by coverage experience, social media experience, and games experience. Although during August and September, video streaming experience jumped into the top 3 highest causes of positive churn.

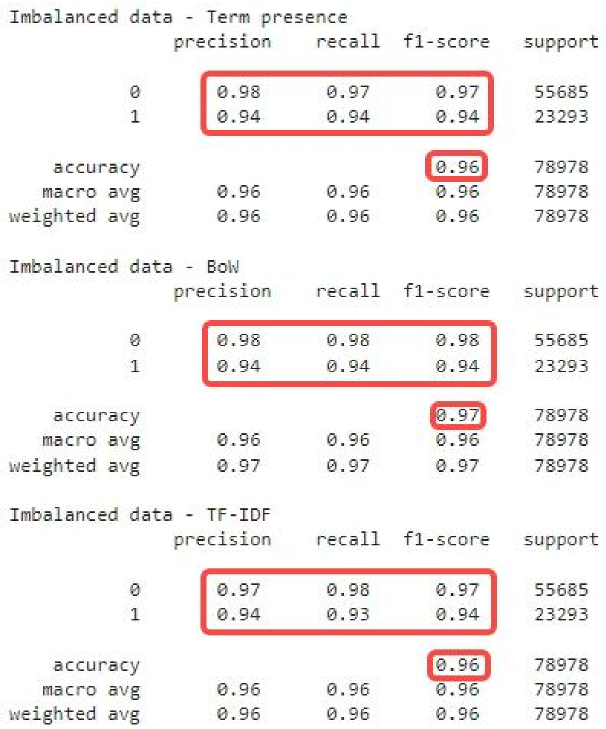
5.2 Model Evaluation

To find out the performance of the model more specifically, the classification model is then evaluated using a classification report. The data used for evaluation is training data (text and labels) from the initial test. The algorithm used utilizes the prediction module from scikit-learn. The training data used in the initial test has been divided into 80% for evaluation training data and 20% for evaluation test data.

As we recall the theory that has been explained in the previous chapter, there are four ways to determine if the predictions were accurate or not:

- True Negative (TN): the actual was negative and predicted negative
- True Positive (TP): the actual was positive and predicted positive
- False Negative (FN): the actual was positive but predicted negative
- False Positive (FP): the actual was negative but predicted positive

All four ways above would be summarized using the calculation of precision, recall, and F1 score. The classification result of 3 feature extraction methods would be compared, to see which feature extraction is the best suited for the classification, and therefore the highest classification result values would be taken. Classification result values are ranged from 0-1, with the higher value indicating the better result. For every precision, recall, and F1 score in each feature extraction, we can see that there are 2 rows of values, which are row 0 and row 1. Row 0 described the result values for negative sentiment prediction, while row 1 represents the result values for positive sentiment prediction. The values with the red marks are the result for each feature extraction method:



```
Imbalanced data - Term presence
              precision    recall  f1-score   support

           0       0.98      0.97      0.97     55685
           1       0.94      0.94      0.94     23293

    accuracy                           0.96     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.96      0.96      0.96     78978

Imbalanced data - BoW
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     55685
           1       0.94      0.94      0.94     23293

    accuracy                           0.97     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.97      0.97      0.97     78978

Imbalanced data - TF-IDF
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     55685
           1       0.94      0.93      0.94     23293

    accuracy                           0.96     78978
   macro avg       0.96      0.96      0.96     78978
weighted avg       0.96      0.96      0.96     78978
```

Figure 5.13 Classification Report Result

Precision value represents the percentage of predictions that were correct, or the ability of a classifier to not label a case that is actually negative as positive. It is described for each class as the proportion of true positives to the total of true positives and false positives, or TP/(TP+FP). In the result, we can see that for the positive sentiment prediction, all the feature extraction methods have 0.94 or 94% precision. For the negative sentiment prediction, both Term Presence (Count Vectorizer) and BoW have 0.98 or 98% precision, while TF-IDF has 0.97 or 97% precision. It means that negative sentiment prediction has a slightly higher proportion of true prediction of around 97% to 98% for each feature extraction, while there are 94 % of all the positive sentiment true predictions.

Recall value is the percentage of positive cases that can be found, or the classifier’s ability to find all the positive sentiments. It is described as the proportion of true positives to the total of true positives and false negatives for each class, or TP/(TP+FN). In the result above, for positive sentiment recall, both Term Presence (Count Vectorizer) and Bags of Words (BoW) have 0.94 or 94% of recall, while TF-IDF has 0.93 or 93% of recall. For the negative sentiment, both BoW and TF-IDF have 0.98 or 98% recall, while Term Presence

(Count Vectorizer) has 0.97 or 97% recall. It means that negative sentiment has a slightly higher proportion of negative sentiment that can be found of around 97% to 98% for each feature extraction, while there are 93% to 94 % of all the positive sentiment can be found.

The F1 score is the percentage of positive predictions that were correct. Since F1 scores incorporate precision and recall into their computation, they are typically lower than accuracy measures. It is often recommended to compare classifier models using the weighted average of F1, rather than overall accuracy. In this case, the result of the positive sentiment F1 score is 0.94 or 94% for all the feature extraction methods, which means there are 94% of positive predictions that were correctly predicted using the classifier. For negative sentiment F1 score, both Term Presence (Count Vectorizer) and TF-IDF have 0.97 or 97% of F1 score, while BoW has 0.98 or 98% of F1 recall. It means that negative sentiment has more correct predictions for about 97% to 98% of all negative sentiment predictions, while positive sentiment has 94% correct predictions.

Lastly, the accuracy value is used to summarize all the values above (precision, recall, and f1 score). It is used to describe the number of correct predictions over entire predictions. The calculation that is conducted to get the accuracy value is (TP+TN) / (Dataset Size). According to the result, BoW has the highest accuracy with 97% of correct predictions, while Term Presence (Count Vectorizer) and TF-IDF have slightly lower accuracy of 96%.

The accuracy can be increased by employing larger datasets of the lexical dictionaries to label the sentiment, as well as the training and testing datasets, as shown by the F-1 score result comparison of this research with the other previous research which used the same SVM algorithm as the classifier and feature extractions. This research has higher accuracy than several similar previous research, such as the research conducted by Vidya et al. (2015), Saragih and Girsang (2017), and Abdillah et al. (2021) due to the larger amount of data for training and/or the lexical resources for the sentiment tagging. The result of F1 score for this research is 97%, while Vidya et al. (2015) have an 89.33% of F1-score, Saragih, and Girsang (2017) have an 80.1% of F1 score and 61.1% of F1-score. This research used a total of 44,533 words for the lexical dictionary, compared to the 10,218 words used by Abdillah et al. (2021) to label the sentiment. In terms of training and testing data, this research used 292,547 tweets in total, whereas Abdillah et al. (2021) used 1,912 tweets Vidya et al. (2015) used 10,000 tweets, and Saragih and Girsang (2017) used 1,000 tweets. These significant data differences may have made the accuracy result different. In addition, the usage of several classifier algorithms can be done to make sure that the accuracy given is the best accuracy from the best algorithm, like what Vidya et al. (2015) have previously done. Due to the lack of computing resources for processing much larger data than the other research, the researcher only used 1 algorithm. Therefore, the utilization of larger datasets and lexical dictionaries, as well as utilizing several classifier algorithms can give better accuracy in the future.

CHAPTER VI
CONCLUSIONS AND SUGGESTIONS

This chapter includes recommendations and suggestions for more research based on research that has already been done, results that respond to the formulation of the problem, and prove current hypotheses.

6.1 Conclusion

1. Sentiment analysis has been used to discover the positive and negative tweets regarding Smartfren brand, according to the customer experience shared on Twitter's social media platform. Inside of the tweets, the most discussed topics can also be identified using certain keywords according to the most important customer experience aspects in the telecommunication industry. The most important details are that there are more negative tweets compared to positive ones over 2.5 years (70.5% of negative tweets), with October being the month with the highest percentage of negative tweets and July having the lowest percentage of negative tweets. The most discussed aspects of customer experience are product experience and network experience, with 37.8% and 37.5% of all tweets level 1 topics being discussed respectively. It means the stability of the network (in many apps usage) and product worthiness (prices and suitable product package content) are the most discussed topics that determined the sentiment of the tweets. Most of the network-related positive and negative tweets throughout the months are mainly discussing coverage experience, social media experience, and games experience, with video streaming experience jumping into the top 3 most discussed topics for positive sentiment in August.
2. There are many Twitter users who changed their perspective about Smartfren after they used Smartfren products or after they see what other users think about this brand. The perception of users is fluctuating according to the level of influence that the users get inside of the platform and might be caused by certain reasons behind. The total amount of sudden sentiment change over 2.5 years is 126k, with positive sudden sentiment change being 4 times higher than negative sudden sentiment change. The average engagement rate (%) for each positive and negative tweet is impacting the number of sudden sentiment changes. The higher the engagement rate, the more sudden sentiment change

happened in a particular period, and vice versa. Most of the sudden sentiment changes are caused by product experience and network experience, with product experience being the highest cause in January, July-September, and December. It means the users' cause of sentiment sudden change varied from network stability, or product prices and package worthiness. Over several months, the top 3 highest causes of network-related sudden sentiment changes are still the same as the total ones for each month, which are coverage experience, social media experience, and games experience respectively.

3. Inside of the tweets' content, the likeliness of churn can be identified based on certain keywords that may lead to churning from one provider brand to another. Over 2.5 years, there has been more positive churn than negative churn, with 57.5% (6.7k) of churn-related tweets considered positive churn. The highest proportion of positive churn happened in November (63.5%), while the lowest proportion happened in April (51.5%). The impact of positive and negative tweets' engagement rate is significant towards the amount of churn, as the higher engagement rate of negative tweets in a particular month, the amount of negative churn will also be higher in the next month. The result shows that both negative and positive churn is caused by mostly network experience and product experience, with the network-caused churn mostly being caused by coverage experience, social media experience, and games experience. For the product cause of churn, it means that the churn is caused by the appropriateness of product price and worthiness of product package or bundling.

6.2 Suggestion

1. For the related company, the result of the customer experience satisfaction from social media that has been analyzed, could be validated by doing synchronization with technical telecommunication assessment and testing. It's because the result presented using social media sentiment analysis only represents the subjective (yet real) perspective from the customer or user of Smartfren. According to Haryadi (2018), any of the customer experience surveys or analyses that come from telecommunication users can be varied according to the subjective assessment, certain psychological conditions, and influence from other media. Therefore, the real quality of customer experience needs to be further assessed and synchronized by using technical evaluation using

International Telecommunication Standard (ITU) standard. However, due to the time-sensitive actions that Smartfren needs to do immediately, in accordance with the improvement of customer experience, Smartfren could use the insights of the sentiment analysis result. Some of the key actions for the near future are:

a. Maintain a higher average engagement rate for positive tweets in social media, specifically Twitter, by continuously running the marketing campaign, inviting influencers and public figures to help to promote Smartfren products, and improving the quality and coverage of customer service on the Twitter platform. This way, the dominance of negative sentiment tweets could be polarized and shifted to the positive sentiment, by the positive campaign that might gain a lot of attention from the Twitter user.

b. Hold a special meeting with other connected departments in charge of the Smartfren network coverage and product strategy, as they are the most prominent cause of sudden sentiment change and churn.

    i. Network coverage: Evaluation related to the coverage strategy could be further conducted, in response to the dominance of coverage experience which caused negative sentiment. The improvement of network coverage can be conducted by providing a wider area of the network, especially in a more remote area and area which has the most Smartfren users. By doing this, Smartfren would be able to retain the existing users and gain more users.

    ii. Product strategy: The adjustment of product packages and product bundling should be conducted according to the needs of the customer. Further research about what customers often do on their mobile phones can be done, in order to adjust product offers according to the customer needs. The price can further be adjusted according to the user segmentation.

c. The whole workflow of the sentiment analysis could later be deployed in the cloud computing platform, to ensure the continuity of Smartfren sentiment analysis in an end-to-end and real-time manner. Therefore, the whole process from dataset scrapping to data visualization can be updated from time to time automatically.

2. For further research, several things are recommended to be conducted for better accuracy and implementation, as the researcher has discussed in Chapter 5 previously:
    a. To improve the accuracy of the sentiment labeling and topic categorization of the tweets' content, the machine learning algorithm can be expanded using other methods such as Naïve Bayes, Random Forest, CNN, etc. Thus, the accuracy of each method can be assessed and selected. For the higher accuracy of the topic categorization, such as machine learning method like LDA can be used, to find the hidden topics inside of the tweets data more accurately. Of course, these methods' recommendations could be considered if the computing resources are advanced enough, due to the massive amount of data that might take a lot of time to be processed.
    b. Enlarging the dataset by scrapping the customer experience data from other social media such as Facebook, Instagram, etc. This way, more customer experience can be gathered, and the result of the sentiment analysis would be more accurate.

REFERENCES

Abdillah, W. F., Permana, A., & Bhakti, R. H. (2021). Analisis Sentimen Penanganan COVID-19 dengan Support Vector Machine: Evaluasi Leksikon dan Metode Ekstraksi Fitur. *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, 160-170.

Afful-Dadzie, E., Nabareseh, S., Oplatková, Z. K., & Klimek, P. (2014). Enterprise Competitive Analysis and Consumer Sentiments on Social Media Insights from Telecommunication Companies. *3rd International Conference on Data Management Technologies and Application*, 22-32.

Ajayi, D., & Sodha, S. (2020, August 11). *Solving common challenges in sentiment analysis with help from Project Debater - Watson Blog*. IBM. https://www.ibm.com/blogs/watson/2020/08/solving-common-challenges-in-sentiment-analysis-with-help-from-project-debater/

Akuma, S., Lubem, T., & Adom, I. T. (2022). Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-022-01096-4

Al-Shabi, M. (2020). Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining. *International Journal of Computer Science and Network Security.*

Aliyah Salsabila, N., Ardhito Winatmoko, Y., Akbar Septiandri, A., & Jamal, A. (2018). Colloquial Indonesian Lexicon. *2018 International Conference on Asian Language Processing (IALP)*. https://doi.org/10.1109/ialp.2018.8629151

Bifet, A., & Frank, E. (2010). Sentiment Knowledge Discovery in Twitter. *Proceedings of the 13th international conference on Discovery science* (pp. 1-15). Berlin: Springer-Verlag.

Camilleri, M. A. (2018). Understanding Customer Needs and Wants. *Springer EBooks*, 29–50. https://doi.org/10.1007/978-3-319-49849-2_2

Carley, K. M., Malik, M. M., Kowalchuck, M., Pfeffer, J., & Landwehr, P. M. (2015). Twitter Usage in Indonesia. *Social Science Research Network.* https://doi.org/10.2139/ssrn.2720332

Chorana, A., & Cherroun, H. (2021). User Generated Content and Engagement Analysis in Social Media case of Algerian Brands. *The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021).*

D. Evans & j. Mackee (2010). Social Media Marketing: The Next Generation of Business Engagement, Canada: Wiley Publishing

Das, S. R., & Chen, M. Y. (2001). Yahoo! for Amazon: Sentiment Parsing from Small Talk on the Web. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.276189

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of International World Wide Web Conference (WWW'03)*, 1-3.

Deepa, D., Raaji, & Tamilarasi, A. (2019). Sentiment Analysis using Feature Extraction and Dictionary-Based Approaches. *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC).* https://doi.org/10.1109/i smac47947.2019.9032456

Diebes, H. M., & Iriqat, R. A. (2019). Social Media as a Strategic Marketing Communication Tool in Palestinian Mobile Telecom Companies - Business to Customers Relationship Perspective. *International Review of Management and Marketing*, 31-40.

eBizMBA. (2014, November 5). *Top 10 Most Popular Websites*. Retrieved from eBizMBA company websites: http://www.ebizmba.com/articles/most-popular websites

Evans, D., & Mckee, J. (2010). Social Media Marketing : The Next Generation of Business Engagement. Canada: Wiley Publishing.

GSMA. (2022). The Mobile Economy 2022. From GSMA: https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222 The-Mobile-Economy-2022.pdf

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques: third edition. Waltham: Elsevier

Haryadi, S. (2018). Chapter 1 The Concept of Telecommunication Network Performance and Quality of Service. *Institut Teknologi Bandung*. https://doi.org/10.31227/osf.io/cq8na

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics*. https://doi.org/10.3115/976909.979640

Hermansyah, R., & Sarno, R. (2021). Sentiment Analysis Based on Quality Aspects in Effort

to Improve Quality of Indihome Product and Services PT Telkom Indonesia Tbk. *Advances in Economics, Business and Management Research*, 30-39.

HubSpot & Brandwatch. (2023). Global Social Media Trends Report. In *HubSpot*.

HubSpot.

Hutto, CJ & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eigth international AAAI conference on weblogs and social media*.

Imbug, N., Ambad, S. N. A., & Bujang, I. (2018). The Influence of Customer Experience

on Customer Loyalty in Telecommunication Industry. International Journal of Academic Research in Business and Social Sciences, 8(3), 103–116.

International Trade Administration. (2022, 07 28). Indonesia - Country Commercial Guide: Information and Telecommunications Technology. From International Trade Administration: https://www.trade.gov/country-commercial-guides/indonesia-information-and-telecommunications-technology

Jayasanka, S. C., Madhushani, T., Marcus, E. R., & Aberathne, I. a. a. U. (2013).

Sentiment Analysis for Social Media. *Conference: Information Technology Research Symposium*, *4*.

Kabadayi, S., & Price, K. (2014). Consumer – brand engagement on Facebook: liking and

commenting behaviors. *Journal of Research in Interactive Marketing*, 8(3), 203–223. https://doi.org/10.1108/jrim-12-2013-0081

Kaji, N., Kitsuregawa, M. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. *Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 1075–1083. Association for Computational Linguistics.

Khatri, H. (2022, July). Indonesia Mobile Network Experience Report. From Open Signal: https://www.opensignal.com/reports/2022/07/indonesia/mobile-network-experience

Kumar, A., & Sebastian, T. M. (2012). Sentiment Analysis: A Perspective on its Past, Present and Future. *International Journal of Intelligent Systems and Applications*, 1-2.

Lamba, M., & Madhusudhan, M. (2022). Text Mining for Information Professionals. *Springer EBooks*. https://doi.org/10.1007/978-3-030-85085-2

Lemon, K. N., & Verhoef, P. (2016b). Understanding Customer Experience Throughout the Customer Journey. *Journal of Marketing*, *80*(6), 69–96. https://doi.org/10.1509/jm.15.0420

Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*.

Liu, B. (2011). Web Crawling. In Web Data Mining : Exploring Hyperlinks, Contents, and Usage Data (pp. 311-362). Chicago: Springer.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers

Othman, N. K., Hussin, M., & Raja Mahmood, R. A. (2019). Sentiment Evaluation of Public Transport in Social Media using Naïve Bayes Method. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1(9), 2305.

MacKay, M., Colangeli, T., Gosselin, S., Neumann, S., & Papadopoulos, A. (2022). Engagement Analysis of Canadian Public Health and News Media Facebook Posts and Sentiment Analysis of Corresponding Comments during COVID-19. *The Uncertain Communication during the COVID-19 Pandemic: People's Reactions and Coping Strategies*, 60-70.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). Introduction to Information

Retrieval. Cambridge : Cambridge University Press.

Melville, P., Gryc, W. & Lawrence, D, R. 2011. Sentiment analysis of blogs by combining lexical knowledge with text classification. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1275–1284. ACM.

Nguyen, H., & Zheng, R. (2014). A Data-driven Study of Influences in Twitter. *International Chamber of Commerce* (pp. 3938-3944). Canada: IEEE.

Nurfalah, A., Adiwijawa. & Suryani, A. A. 2017. Analisis Sentimen Berbahasa Indonesia dengan Pendekatan Lexicon-Based pada Media Sosial Twitter. Telkom University, Bandung. *Jurnal Masyarakat Informatika Indonesia*, Vol 2, No. 1, Januari-Maret, Hal 1-8.

Kolchyna, O., Souza T., Treleaven, T., & Aste, T. (2015). Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination. *ArXiv: Computation and Language*.

Patel, B., & Shah, D. D. (2013). Significance of stop word elimination in meta search engine. *International Conference On Intelligent Systems and Signal Processing* (ISSP, 52-55).

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al,"Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research,* 2011, 12, 2825–30.

PT. Amitra Visinet Indonesia. (2014, Desember 7). *About Us : PT. Amitra Visinet Indonesia*. Retrieved from Optima Web: http://www.optimaweb.co.id/social-media-marketing

Qualtrics. (2022). *What is sentiment analysis and how can users leverage it?*

https://www.qualtrics.com/experience-management/research/sentiment-analysis/

Ranjan, S., Sood, S., & Verma, V. (2018). Twitter Sentiment Analysis of Real-time Customer Experience Feedback for Predicting Growth of Indian Telecom Companies. *International Conference on Computing Sciences (ICCS)*, (4), 167.

Riserbato, R. (2022). *How to Measure Customer Experience (+ 8 Metrics to Help You Do It)*. HubSpot. https://blog.hubspot.com/service/measuring-customer-experience

Rofiqoh, U., Perdana, R. S., & Fauzi, M. A. (2017). Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 12 (1).

Sadia, A., et al. (2018). An Overview of Lexicon-Based Approach For Sentiment Analysis. *3rd International Electrical Engineering Conference (IEEC 2018)*

Saputra, F. T., Wijaya, S. H., Nurhadryani, Y., & Defina. (2020). Lexicon Addition Effect on Lexicon-Based of Indonesian Sentiment Analysis on Twitter. *International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS).*

Saragih, M. H. (2017). Sentiment Analysis of Customer Engagement on Social Media in Transport Online. *International Conference on Sustainable Information Engineering and Technology (SIET),* 24.

Sari, Syandra, & Adriani., M. (2008). Developing Part of Speech Tagger for Bahasa Indonesia Using Brill Tagger. *The International Second Malindo*, 1.

Schneider, K.-M. (2005). Techniques for Improving the Performance. *In Proceedings of CICLing*, 5-7.

Sharawneh, T. (2020). Social media marketing activities and brand loyalty in telecommunication industry: The mediating role of brand affect. *Journal of Innovations in Digital Marketing*, 1-7. doi:https://doi.org/10.51300/jidm-2020-11

Shepherd, J. (2022). *22 Essential Twitter Statistics You Need to Know in 2022*. The Social Shepherd. https://thesocialshepherd.com/blog/twitter-statistics

Sprinklr. (2018). *How Brands Track Customer Experience Through Social Media*. https://www.sprinklr.com/blog/brands-track-customer-experience-social-media/

Sundjaja, A. M., Gaol, F. L., Abdinagoro, S. B., & Abbas, B. S. (2017). The Behavior of

Online Museum Visitors on Facebook Fan Page of the Museum in Indonesia. *Binus Business Review*, 8(3), 237-243. http://dx.doi.org/10.21512/bbr.v8i3.3742

Susanti, A. R., Djatna, T., & Kusuma, W. A. (2017). Twitter's Sentiment Analysis on

Gsm Services using Multinomial Naïve Bayes. *Telkomnika*, 3(15), 1354.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. & Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*. Volume 37, No. 2, pp. 267–307, MIT Press.

Tsiaras, C., Sehgal, A., & Seeber, S. (2014). Towards Evaluating Type of Service Related Quality-of-Experience on Mobile Networks. *Conference: 7th IFIP Wireless and Mobile*.

Vidya, N. A., Fanany, M. I., & Budi, I. (2015). Twitter Sentiment to Analyze Net Brand

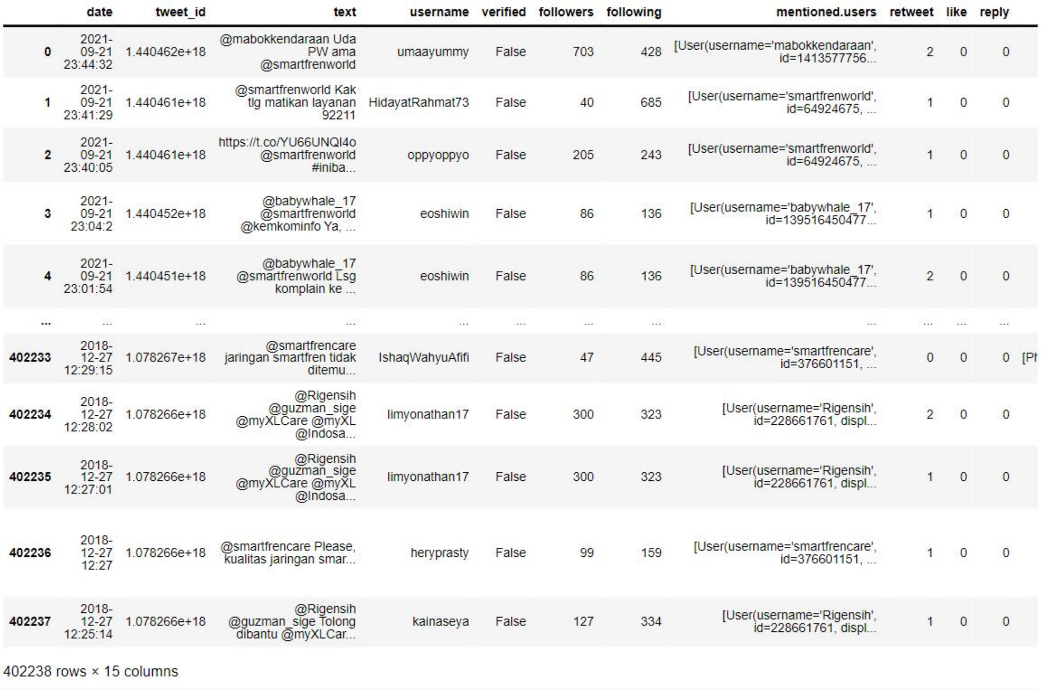
Reputation of Mobile Phone Providers. Procedia Computer Science, (72), 519

Wahid, D. H., & Azhari, S. (2016). Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF- IDFdan Cosine Similarity. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*.

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques Third Edition. Burlington: Morgan Kaufmann.

APPENDIX

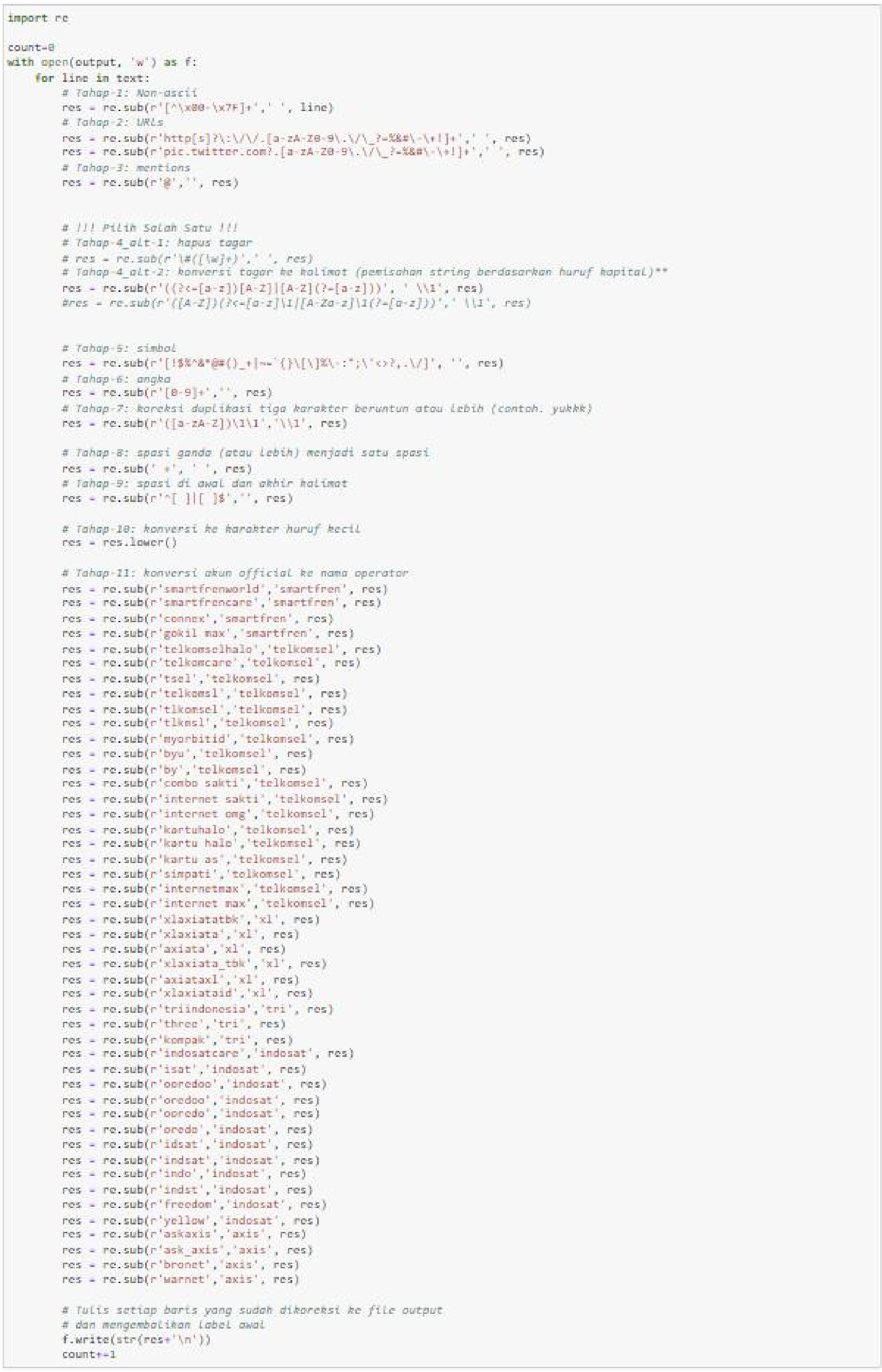
A. Raw Tweets Data



	date	tweet_id	text	username	verified	followers	following	mentioned.users	retweet	like	reply	
0	2021-09-21 23:44:32	1.440462e+18	@mabokkendaraan Uda PW ama @smartfrenworld	umaayummy	False	703	428	[User(username='mabokkendaraan', id=1413577756...	2	0	0	
1	2021-09-21 23:41:29	1.440461e+18	@smartfrenworld Kak tlg matikan layanan 92211	HidayatRahmat73	False	40	685	[User(username='smartfrenworld', id=64924675, ...	1	0	0	
2	2021-09-21 23:40:05	1.440461e+18	https://t.co/YU66UNQI4o @smartfrenworld #iniba...	oppyoppyo	False	205	243	[User(username='smartfrenworld', id=64924675, ...	1	0	0	
3	2021-09-21 23:04:2	1.440452e+18	@babywhale_17 @smartfrenworld @kemkominfo Ya, ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	1	0	0	
4	2021-09-21 23:01:54	1.440451e+18	@babywhale_17 @smartfrenworld Lsg komplain ke ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	2	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
402233	2018-12-27 12:29:15	1.078267e+18	@smartfrencare jaringan smartfren tidak ditemu...	IshaqWahyuAfifi	False	47	445	[User(username='smartfrencare', id=376601151, ...	0	0	0	[Pl
402234	2018-12-27 12:28:02	1.078266e+18	@Rigensih @guzman_sige @myXLCare @myXL @Indosa...	limyonathan17	False	300	323	[User(username='Rigensih', id=228661761, displ...	2	0	0	
402235	2018-12-27 12:27:01	1.078266e+18	@Rigensih @guzman_sige @myXLCare @myXL @Indosa...	limyonathan17	False	300	323	[User(username='Rigensih', id=228661761, displ...	1	0	0	
402236	2018-12-27 12:27	1.078266e+18	@smartfrencare Please, kualitas jaringan smar...	heryprasty	False	99	159	[User(username='smartfrencare', id=376601151, ...	1	0	0	
402237	2018-12-27 12:25:14	1.078266e+18	@Rigensih @guzman_sige Tolong dibantu @myXLCar...	kainaseya	False	127	334	[User(username='Rigensih', id=228661761, displ...	1	0	0	

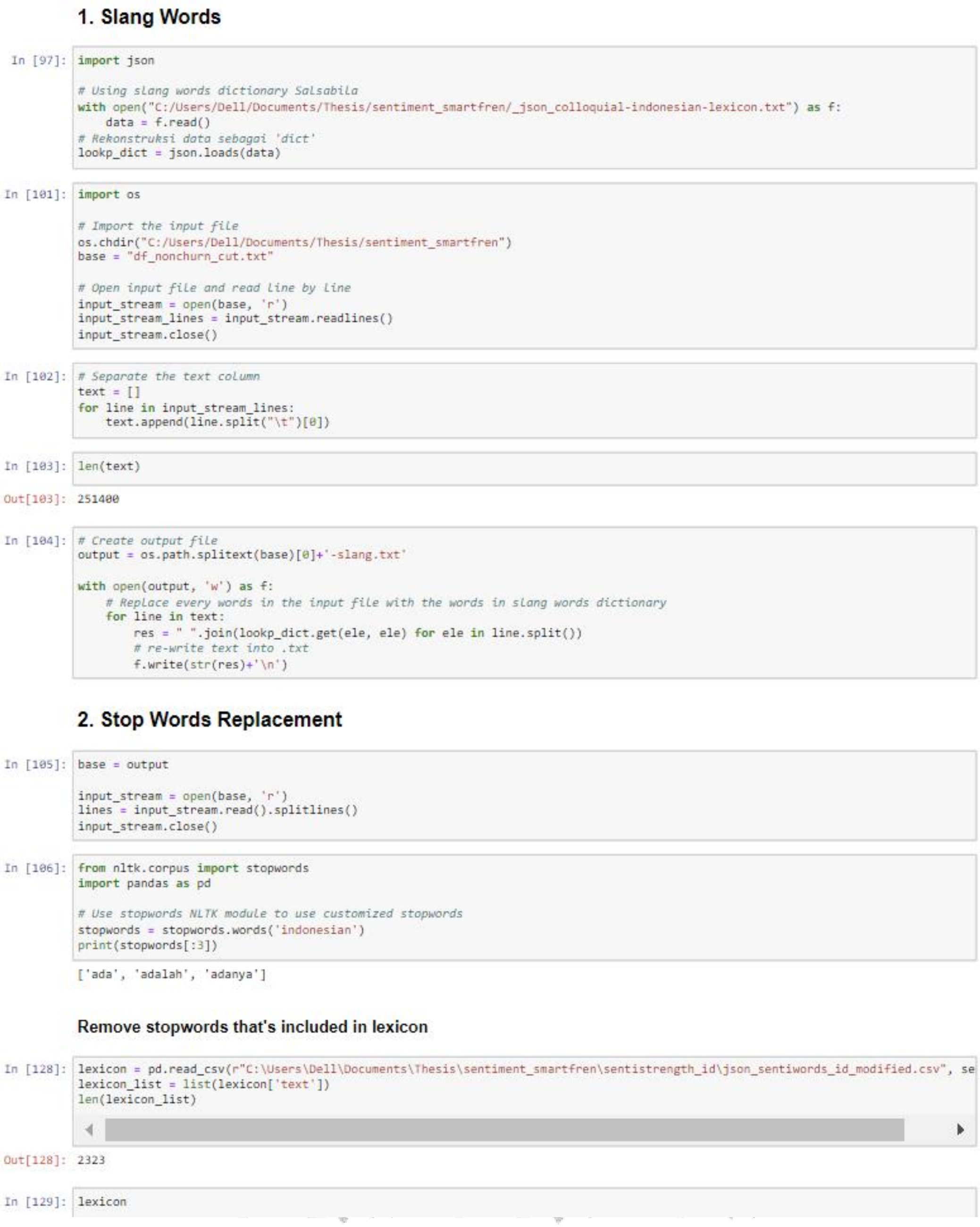
402238 rows × 15 columns

B. Cleaning Tweet Text Codes



```
import re

count=0
with open(output, 'w') as f:
    for line in text:
        # Tahap-1: Non-ascii
        res = re.sub(r'[^\x00-\x7F]+',' ', line)
        # Tahap-2: URLs
        res = re.sub(r'http[s]?\:\/\/.[a-zA-Z0-9\.\/\_?=%&#\-\+!]+',' ', res)
        res = re.sub(r'pic.twitter.com?.[a-zA-Z0-9\.\/\_?=%&#\-\+!]+',' ', res)
        # Tahap-3: mentions
        res = re.sub(r'@','', res)


        # !!! Pilih Salah Satu !!!
        # Tahap-4_alt-1: hapus tagar
        # res = re.sub(r'\#([\w]+)',' ', res)
        # Tahap-4_alt-2: konversi tagar ke kalimat (pemisahan string berdasarkan huruf kapital)**
        res = re.sub(r'((?<=[a-z])[A-Z]|[A-Z](?=[a-z]))', ' \\1', res)
        #res = re.sub(r'([A-Z])(?<=[a-z]\1|[A-Za-z]\1(?=[a-z]))',' \\1', res)


        # Tahap-5: simbol
        res = re.sub(r'[!$%^&*@#()_+|~=`{}\[\]%\-:";\'<>?,.\/]', '', res)
        # Tahap-6: angka
        res = re.sub(r'[0-9]+','', res)
        # Tahap-7: koreksi duplikasi tiga karakter beruntun atau lebih (contoh. yukkk)
        res = re.sub(r'([a-zA-Z])\1\1','\\1', res)

        # Tahap-8: spasi ganda (atau lebih) menjadi satu spasi
        res = re.sub(' +', ' ', res)
        # Tahap-9: spasi di awal dan akhir kalimat
        res = re.sub(r'^[ ]|[ ]$','', res)

        # Tahap-10: konversi ke karakter huruf kecil
        res = res.lower()

        # Tahap-11: konversi akun official ke nama operator
        res = re.sub(r'smartfrenworld','smartfren', res)
        res = re.sub(r'smartfrencare','smartfren', res)
        res = re.sub(r'connex','smartfren', res)
        res = re.sub(r'gokil max','smartfren', res)
        res = re.sub(r'telkomselhalo','telkomsel', res)
        res = re.sub(r'telkomcare','telkomsel', res)
        res = re.sub(r'tsel','telkomsel', res)
        res = re.sub(r'telkomsl','telkomsel', res)
        res = re.sub(r'tlkomsel','telkomsel', res)
        res = re.sub(r'tlkmsl','telkomsel', res)
        res = re.sub(r'myorbitid','telkomsel', res)
        res = re.sub(r'byu','telkomsel', res)
        res = re.sub(r'by','telkomsel', res)
        res = re.sub(r'combo sakti','telkomsel', res)
        res = re.sub(r'internet sakti','telkomsel', res)
        res = re.sub(r'internet omg','telkomsel', res)
        res = re.sub(r'kartuhalo','telkomsel', res)
        res = re.sub(r'kartu halo','telkomsel', res)
        res = re.sub(r'kartu as','telkomsel', res)
        res = re.sub(r'simpati','telkomsel', res)
        res = re.sub(r'internetmax','telkomsel', res)
        res = re.sub(r'internet max','telkomsel', res)
        res = re.sub(r'xlaxiatatbk','xl', res)
        res = re.sub(r'xlaxiata','xl', res)
        res = re.sub(r'axiata','xl', res)
        res = re.sub(r'xlaxiata_tbk','xl', res)
        res = re.sub(r'axiataxl','xl', res)
        res = re.sub(r'xlaxiataid','xl', res)
        res = re.sub(r'triindonesia','tri', res)
        res = re.sub(r'three','tri', res)
        res = re.sub(r'kompak','tri', res)
        res = re.sub(r'indosatcare','indosat', res)
        res = re.sub(r'isat','indosat', res)
        res = re.sub(r'ooredoo','indosat', res)
        res = re.sub(r'oredoo','indosat', res)
        res = re.sub(r'ooredo','indosat', res)
        res = re.sub(r'oredo','indosat', res)
        res = re.sub(r'idsat','indosat', res)
        res = re.sub(r'indsat','indosat', res)
        res = re.sub(r'indo','indosat', res)
        res = re.sub(r'indst','indosat', res)
        res = re.sub(r'freedom','indosat', res)
        res = re.sub(r'yellow','indosat', res)
        res = re.sub(r'askaxis','axis', res)
        res = re.sub(r'ask_axis','axis', res)
        res = re.sub(r'bronet','axis', res)
        res = re.sub(r'warnet','axis', res)

        # Tulis setiap baris yang sudah dikoreksi ke file output
        # dan mengembalikan label awal
        f.write(str(res+'\n'))
        count+=1
```

C. Slang Words and Stopwords



1. Slang Words

```
In [97]: import json

         # Using slang words dictionary Salsabila
         with open("C:/Users/Dell/Documents/Thesis/sentiment_smartfren/_json_colloquial-indonesian-lexicon.txt") as f:
             data = f.read()
         # Rekonstruksi data sebagai 'dict'
         lookp_dict = json.loads(data)
```

```
In [101]: import os

          # Import the input file
          os.chdir("C:/Users/Dell/Documents/Thesis/sentiment_smartfren")
          base = "df_nonchurn_cut.txt"

          # Open input file and read line by line
          input_stream = open(base, 'r')
          input_stream_lines = input_stream.readlines()
          input_stream.close()
```

```
In [102]: # Separate the text column
          text = []
          for line in input_stream_lines:
              text.append(line.split("\t")[0])
```

```
In [103]: len(text)

Out[103]: 251400
```

```
In [104]: # Create output file
          output = os.path.splitext(base)[0]+'-slang.txt'

          with open(output, 'w') as f:
              # Replace every words in the input file with the words in slang words dictionary
              for line in text:
                  res = " ".join(lookp_dict.get(ele, ele) for ele in line.split())
                  # re-write text into .txt
                  f.write(str(res)+'\n')
```

2. Stop Words Replacement

```
In [105]: base = output

          input_stream = open(base, 'r')
          lines = input_stream.read().splitlines()
          input_stream.close()
```

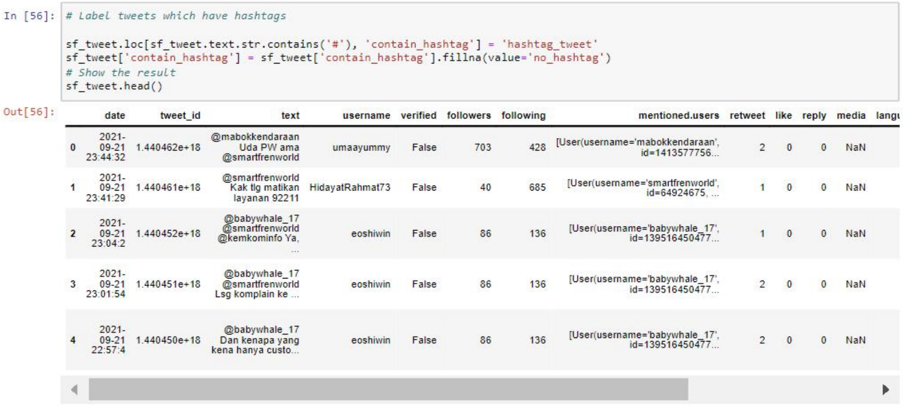
```
In [106]: from nltk.corpus import stopwords
          import pandas as pd

          # Use stopwords NLTK module to use customized stopwords
          stopwords = stopwords.words('indonesian')
          print(stopwords[:3])

          ['ada', 'adalah', 'adanya']
```

Remove stopwords that's included in lexicon

```
In [128]: lexicon = pd.read_csv(r"C:\Users\Dell\Documents\Thesis\sentiment_smartfren\sentistrength_id\json_sentiwords_id_modified.csv", se
          lexicon_list = list(lexicon['text'])
          len(lexicon_list)

Out[128]: 2323
```

```
In [129]: lexicon
```

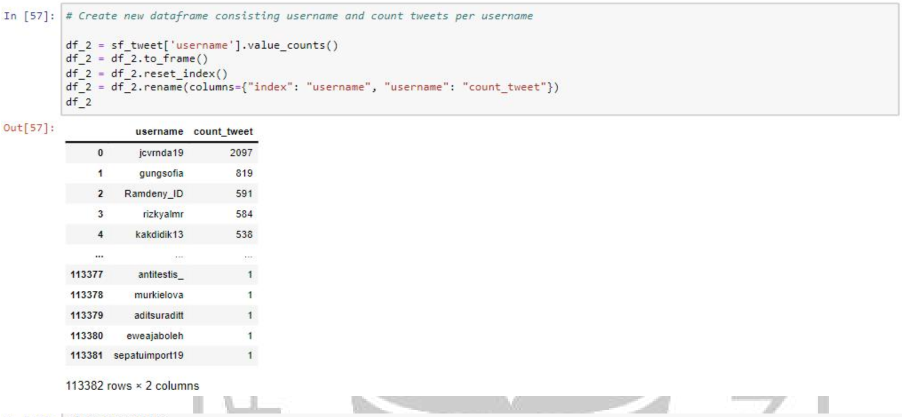
D. Removing Buzzers Codes



In [56]:
```
# Label tweets which have hashtags

sf_tweet.loc[sf_tweet.text.str.contains('#'), 'contain_hashtag'] = 'hashtag_tweet'
sf_tweet['contain_hashtag'] = sf_tweet['contain_hashtag'].fillna(value='no_hashtag')
# Show the result
sf_tweet.head()
```

Out[56]:

	date	tweet_id	text	username	verified	followers	following	mentioned.users	retweet	like	reply	media	langu
0	2021-09-21 23:44:32	1.440462e+18	@mabokkendaraan Uda PW ama @smartfrenworld	umaayummy	False	703	428	[User(username='mabokkendaraan', id=1413577756...	2	0	0	NaN	
1	2021-09-21 23:41:29	1.440461e+18	@smartfrenworld Kak tlg matikan layanan 92211	HidayatRahmat73	False	40	685	[User(username='smartfrenworld', id=64924675, ...	1	0	0	NaN	
2	2021-09-21 23:04:2	1.440452e+18	@babywhale_17 @smartfrenworld @kemkominfo Ya, ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	1	0	0	NaN	
3	2021-09-21 23:01:54	1.440451e+18	@babywhale_17 @smartfrenworld Lsg komplain ke ...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	2	0	0	NaN	
4	2021-09-21 22:57:4	1.440450e+18	@babywhale_17 Dan kenapa yang kena hanya custo...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	2	0	0	NaN	

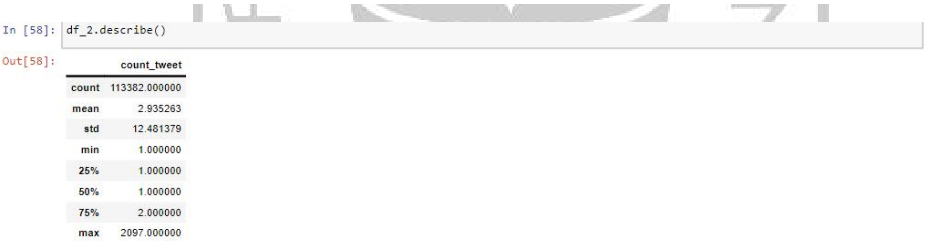


In [57]:
```
# Create new dataframe consisting username and count tweets per username

df_2 = sf_tweet['username'].value_counts()
df_2 = df_2.to_frame()
df_2 = df_2.reset_index()
df_2 = df_2.rename(columns={"index": "username", "username": "count_tweet"})
df_2
```

Out[57]:

	username	count_tweet
0	jcvrnda19	2097
1	gungsofia	819
2	Ramdeny_ID	591
3	rizkyalmr	584
4	kakdidik13	538
...	...	...
113377	antitestis_	1
113378	murkielova	1
113379	aditsuraditt	1
113380	eweajaboleh	1
113381	sepatuimport19	1

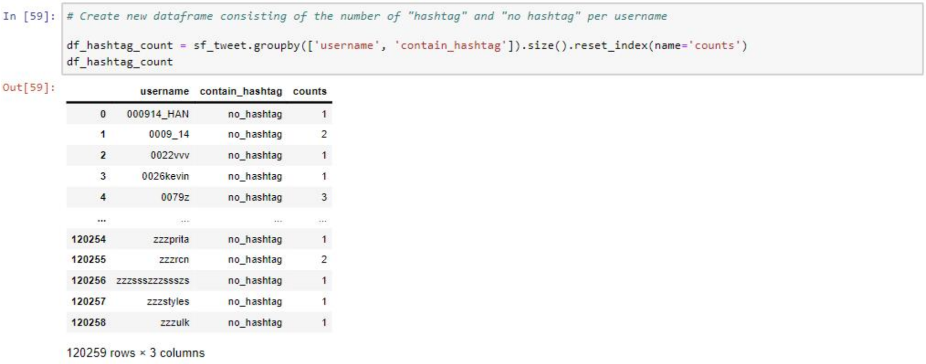
113382 rows × 2 columns



In [58]:
```
df_2.describe()
```

Out[58]:

	count_tweet
count	113382.000000
mean	2.935263
std	12.481379
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	2097.000000

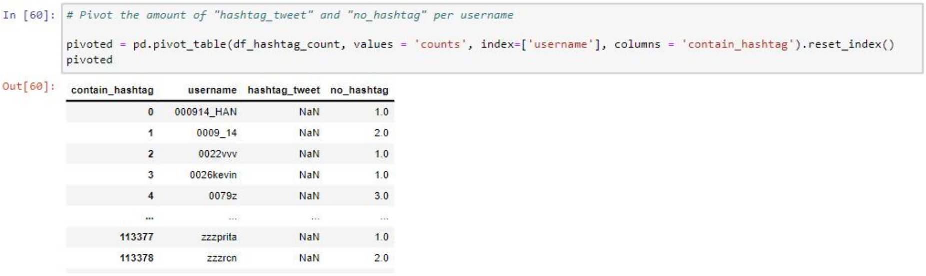


In [59]:
```
# Create new dataframe consisting of the number of "hashtag" and "no hashtag" per username

df_hashtag_count = sf_tweet.groupby(['username', 'contain_hashtag']).size().reset_index(name='counts')
df_hashtag_count
```

Out[59]:

	username	contain_hashtag	counts
0	000914_HAN	no_hashtag	1
1	0009_14	no_hashtag	2
2	0022vvv	no_hashtag	1
3	0026kevin	no_hashtag	1
4	0079z	no_hashtag	3
...	...	...	...
120254	zzzprita	no_hashtag	1
120255	zzzrcn	no_hashtag	2
120256	zzzssszzzssszs	no_hashtag	1
120257	zzzstyles	no_hashtag	1
120258	zzzulk	no_hashtag	1

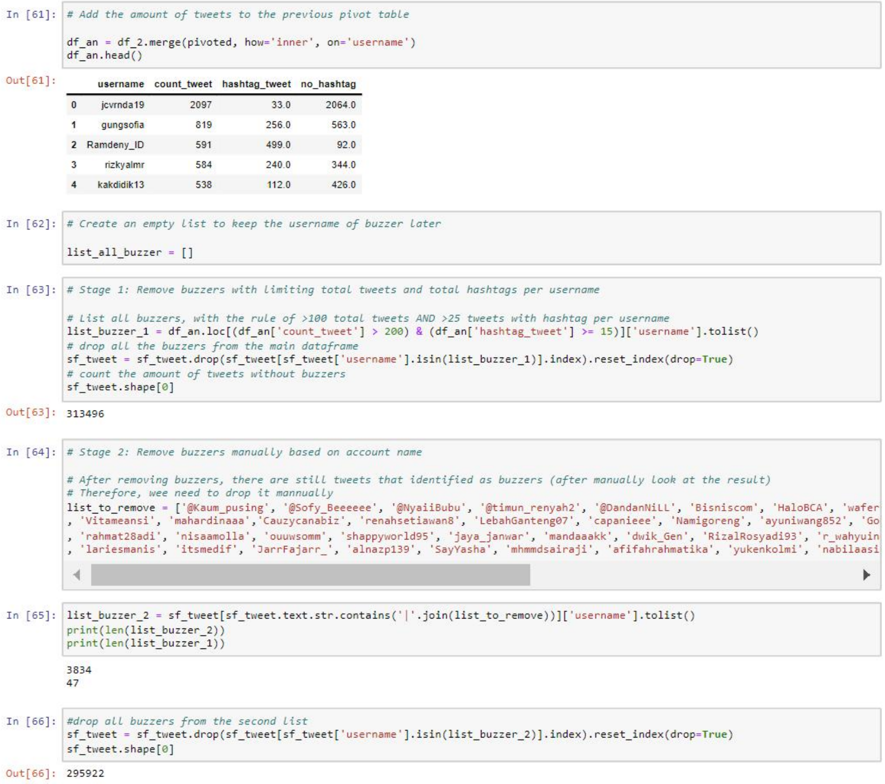
120259 rows × 3 columns



In [60]:
```
# Pivot the amount of "hashtag_tweet" and "no_hashtag" per username

pivoted = pd.pivot_table(df_hashtag_count, values = 'counts', index=['username'], columns = 'contain_hashtag').reset_index()
pivoted
```

Out[60]:

contain_hashtag	username	hashtag_tweet	no_hashtag
0	000914_HAN	NaN	1.0
1	0009_14	NaN	2.0
2	0022vvv	NaN	1.0
3	0026kevin	NaN	1.0
4	0079z	NaN	3.0
...	...	...	...
113377	zzzprita	NaN	1.0
113378	zzzrcn	NaN	2.0



In [61]:
```
# Add the amount of tweets to the previous pivot table

df_an = df_2.merge(pivoted, how='inner', on='username')
df_an.head()
```

Out[61]:

	username	count_tweet	hashtag_tweet	no_hashtag
0	jcvrnda19	2097	33.0	2064.0
1	gungsofia	819	256.0	563.0
2	Ramdeny_ID	591	499.0	92.0
3	rizkyalmr	584	240.0	344.0
4	kakdidik13	538	112.0	426.0

In [62]:
```
# Create an empty list to keep the username of buzzer later

list_all_buzzer = []
```

In [63]:
```
# Stage 1: Remove buzzers with limiting total tweets and total hashtags per username

# List all buzzers, with the rule of >100 total tweets AND >25 tweets with hashtag per username
list_buzzer_1 = df_an.loc[(df_an['count_tweet'] > 200) & (df_an['hashtag_tweet'] >= 15)]['username'].tolist()
# drop all the buzzers from the main dataframe
sf_tweet = sf_tweet.drop(sf_tweet[sf_tweet['username'].isin(list_buzzer_1)].index).reset_index(drop=True)
# count the amount of tweets without buzzers
sf_tweet.shape[0]
```

Out[63]:
```
313496
```

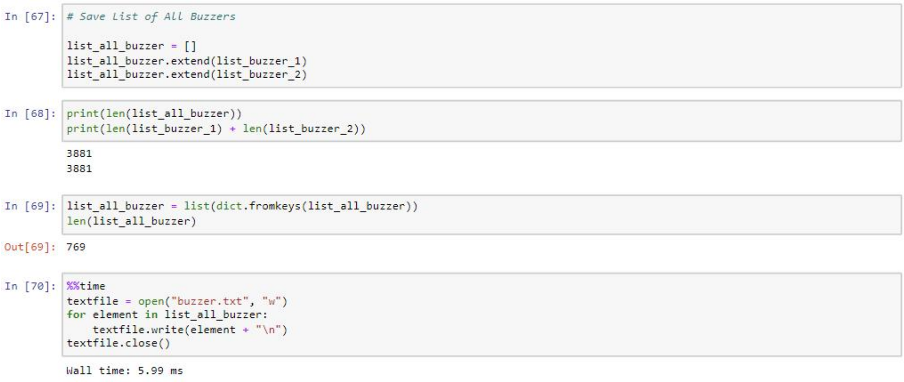
In [64]:
```
# Stage 2: Remove buzzers manually based on account name

# After removing buzzers, there are still tweets that identified as buzzers (after manually look at the result)
# Therefore, wee need to drop it mannually
list_to_remove = ['@Kaum_pusing', '@Sofy_Beeeeee', '@NyaiiBubu', '@timun_renyah2', '@DandanNiLL', 'Bisniscom', 'HaloBCA', 'wafer
, 'Vitameansi', 'mahardinaaa','Cauzycanabiz', 'renahsetiawan8', 'LebahGanteng07', 'capanieee', 'Namigoreng', 'ayuniwang852', 'Go
, 'rahmat28adi', 'nisaamolla', 'ouuwsomm', 'shappyworld95', 'jaya_janwar', 'mandaaakk', 'dwik_Gen', 'RizalRosyadi93', 'r_wahyuin
, 'lariesmanis', 'itsmedif', 'JarrFajarr_', 'alnazp139', 'SayYasha', 'mhmmdsairaji', 'afifahrahmatika', 'yukenkolmi', 'nabilaasi
```

In [65]:
```
list_buzzer_2 = sf_tweet[sf_tweet.text.str.contains('|'.join(list_to_remove))]['username'].tolist()
print(len(list_buzzer_2))
print(len(list_buzzer_1))
```

```
3834
47
```

In [66]:
```
#drop all buzzers from the second list
sf_tweet = sf_tweet.drop(sf_tweet[sf_tweet['username'].isin(list_buzzer_2)].index).reset_index(drop=True)
sf_tweet.shape[0]
```

Out[66]:
```
295922
```



In [67]:
```
# Save List of All Buzzers

list_all_buzzer = []
list_all_buzzer.extend(list_buzzer_1)
list_all_buzzer.extend(list_buzzer_2)
```

In [68]:
```
print(len(list_all_buzzer))
print(len(list_buzzer_1) + len(list_buzzer_2))
```

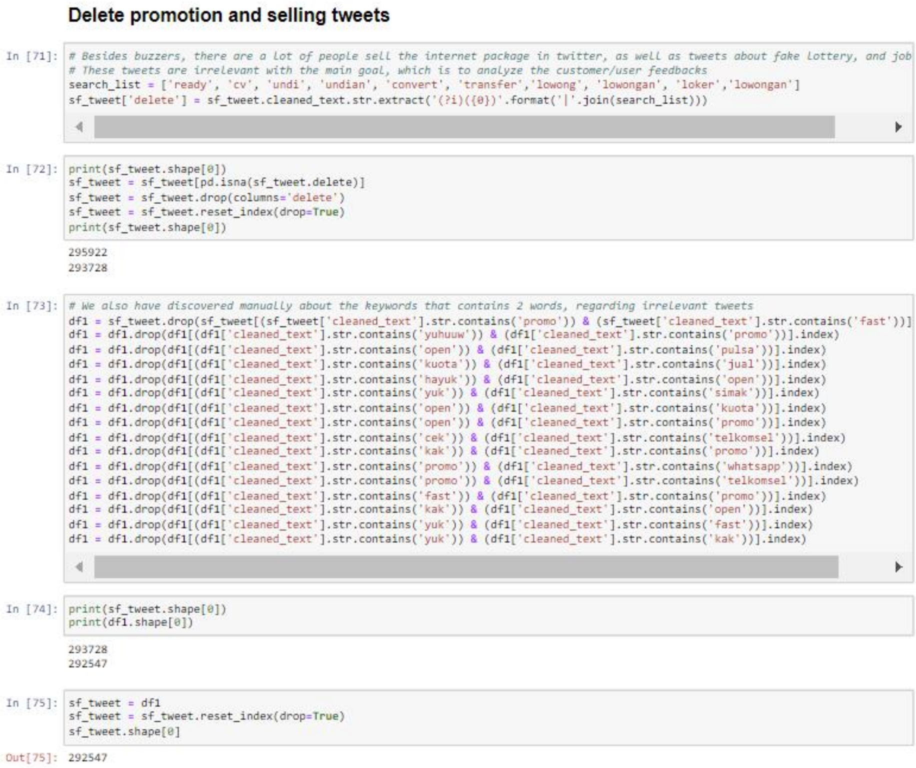
```
3881
3881
```

In [69]:
```
list_all_buzzer = list(dict.fromkeys(list_all_buzzer))
len(list_all_buzzer)
```

Out[69]:
```
769
```

In [70]:
```
%%time
textfile = open("buzzer.txt", "w")
for element in list_all_buzzer:
    textfile.write(element + "\n")
textfile.close()
```

```
Wall time: 5.99 ms
```



Delete promotion and selling tweets

```
In [71]: # Besides buzzers, there are a lot of people sell the internet package in twitter, as well as tweets about fake lottery, and job
         # These tweets are irrelevant with the main goal, which is to analyze the customer/user feedbacks
         search_list = ['ready', 'cv', 'undi', 'undian', 'convert', 'transfer','lowong', 'lowongan', 'loker','lowongan']
         sf_tweet['delete'] = sf_tweet.cleaned_text.str.extract('(?i)({0})'.format('|'.join(search_list)))
```

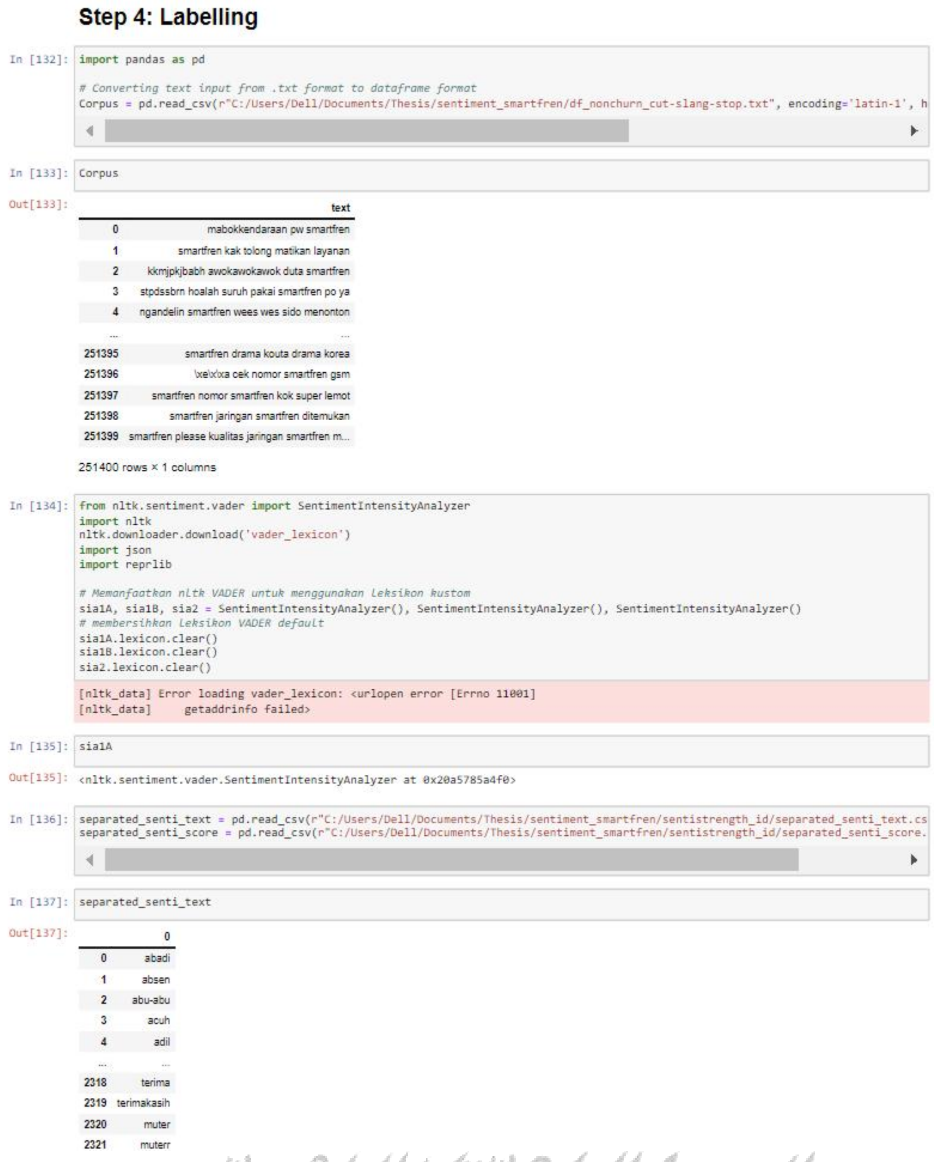
```
In [72]: print(sf_tweet.shape[0])
         sf_tweet = sf_tweet[pd.isna(sf_tweet.delete)]
         sf_tweet = sf_tweet.drop(columns='delete')
         sf_tweet = sf_tweet.reset_index(drop=True)
         print(sf_tweet.shape[0])

         295922
         293728
```

```
In [73]: # We also have discovered manually about the keywords that contains 2 words, regarding irrelevant tweets
         df1 = sf_tweet.drop(sf_tweet[(sf_tweet['cleaned_text'].str.contains('promo')) & (sf_tweet['cleaned_text'].str.contains('fast'))]
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuhuuw')) & (df1['cleaned_text'].str.contains('promo'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('pulsa'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kuota')) & (df1['cleaned_text'].str.contains('jual'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('hayuk')) & (df1['cleaned_text'].str.contains('open'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('simak'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('kuota'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('open')) & (df1['cleaned_text'].str.contains('promo'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('cek')) & (df1['cleaned_text'].str.contains('telkomsel'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kak')) & (df1['cleaned_text'].str.contains('promo'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('promo')) & (df1['cleaned_text'].str.contains('whatsapp'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('promo')) & (df1['cleaned_text'].str.contains('telkomsel'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('fast')) & (df1['cleaned_text'].str.contains('promo'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('kak')) & (df1['cleaned_text'].str.contains('open'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('fast'))].index)
         df1 = df1.drop(df1[(df1['cleaned_text'].str.contains('yuk')) & (df1['cleaned_text'].str.contains('kak'))].index)
```

```
In [74]: print(sf_tweet.shape[0])
         print(df1.shape[0])

         293728
         292547
```

```
In [75]: sf_tweet = df1
         sf_tweet = sf_tweet.reset_index(drop=True)
         sf_tweet.shape[0]

Out[75]: 292547
```

E. Non-churn Tweets Sentiment Labelling



Step 4: Labelling

In [132]:
```
import pandas as pd

# Converting text input from .txt format to dataframe format
Corpus = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/df_nonchurn_cut-slang-stop.txt", encoding='latin-1', h
```

In [133]:
```
Corpus
```

Out[133]:

	text
0	mabokkendaraan pw smartfren
1	smartfren kak tolong matikan layanan
2	kkmjpkjbabh awokawokawok duta smartfren
3	stpdssbrn hoalah suruh pakai smartfren po ya
4	ngandelin smartfren wees wes sido menonton
...	...
251395	smartfren drama kouta drama korea
251396	\xe\x\xa cek nomor smartfren gsm
251397	smartfren nomor smartfren kok super lemot
251398	smartfren jaringan smartfren ditemukan
251399	smartfren please kualitas jaringan smartfren m...

251400 rows × 1 columns

In [134]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.downloader.download('vader_lexicon')
import json
import reprlib

# Memanfaatkan nltk VADER untuk menggunakan Leksikon kustom
sia1A, sia1B, sia2 = SentimentIntensityAnalyzer(), SentimentIntensityAnalyzer(), SentimentIntensityAnalyzer()
# membersihkan Leksikon VADER default
sia1A.lexicon.clear()
sia1B.lexicon.clear()
sia2.lexicon.clear()
```

```
[nltk_data] Error loading vader_lexicon: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>
```

In [135]:
```
sia1A
```

Out[135]:
```
<nltk.sentiment.vader.SentimentIntensityAnalyzer at 0x20a5785a4f0>
```

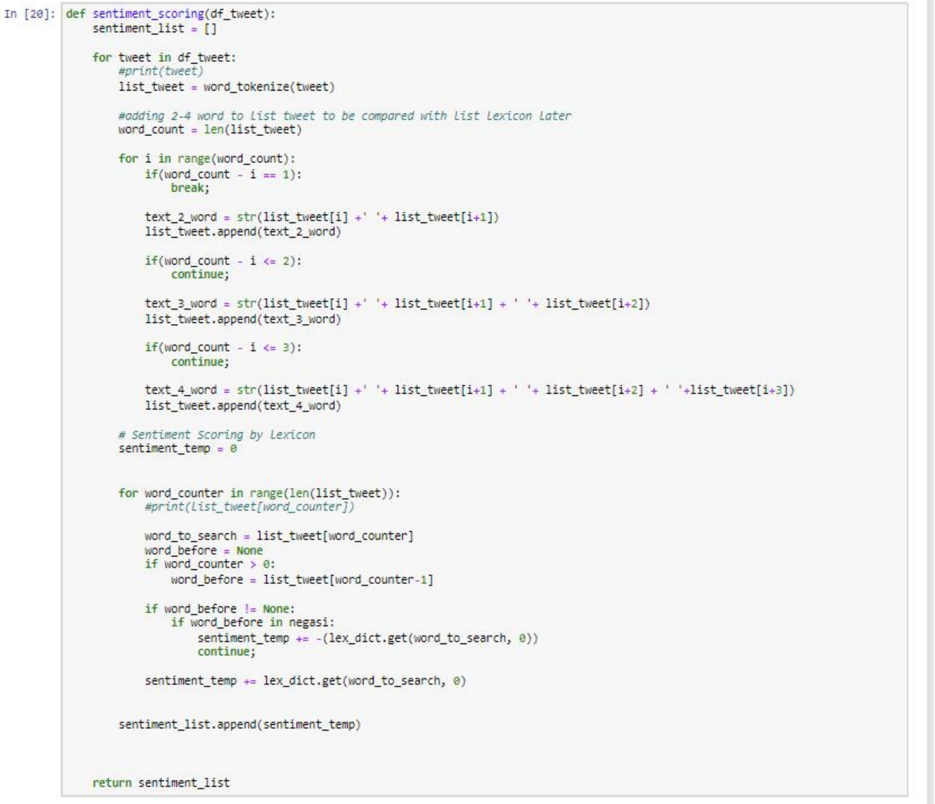
In [136]:
```
separated_senti_text = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/sentistrength_id/separated_senti_text.cs
separated_senti_score = pd.read_csv(r"C:/Users/Dell/Documents/Thesis/sentiment_smartfren/sentistrength_id/separated_senti_score.
```

In [137]:
```
separated_senti_text
```

Out[137]:

	0
0	abadi
1	absen
2	abu-abu
3	acuh
4	adil
...	...
2318	terima
2319	terimakasih
2320	muter
2321	muterr

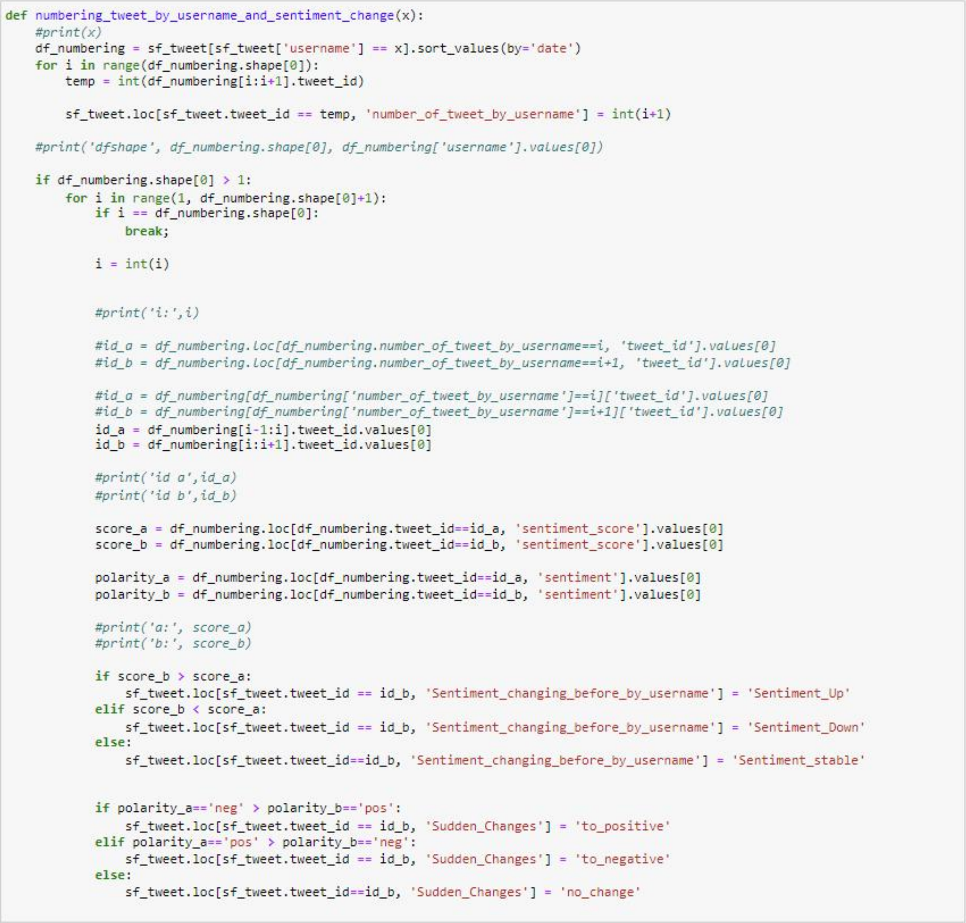
F. Churn Tweets Sentiment Labelling



```
In [20]: def sentiment_scoring(df_tweet):
             sentiment_list = []

             for tweet in df_tweet:
                 #print(tweet)
                 list_tweet = word_tokenize(tweet)

                 #adding 2-4 word to list tweet to be compared with list lexicon later
                 word_count = len(list_tweet)

                 for i in range(word_count):
                     if(word_count - i == 1):
                         break;

                     text_2_word = str(list_tweet[i] +' '+ list_tweet[i+1])
                     list_tweet.append(text_2_word)

                     if(word_count - i <= 2):
                         continue;

                     text_3_word = str(list_tweet[i] +' '+ list_tweet[i+1] + ' '+ list_tweet[i+2])
                     list_tweet.append(text_3_word)

                     if(word_count - i <= 3):
                         continue;

                     text_4_word = str(list_tweet[i] +' '+ list_tweet[i+1] + ' '+ list_tweet[i+2] + ' '+list_tweet[i+3])
                     list_tweet.append(text_4_word)

                 # Sentiment Scoring by Lexicon
                 sentiment_temp = 0


                 for word_counter in range(len(list_tweet)):
                     #print(list_tweet[word_counter])

                     word_to_search = list_tweet[word_counter]
                     word_before = None
                     if word_counter > 0:
                         word_before = list_tweet[word_counter-1]

                     if word_before != None:
                         if word_before in negasi:
                             sentiment_temp += -(lex_dict.get(word_to_search, 0))
                             continue;

                     sentiment_temp += lex_dict.get(word_to_search, 0)


                 sentiment_list.append(sentiment_temp)


             return sentiment_list
```

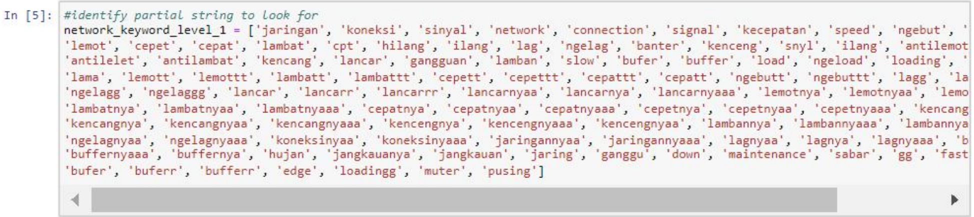
G. Sudden Sentiment Change Tagging

Sudden Change Tag

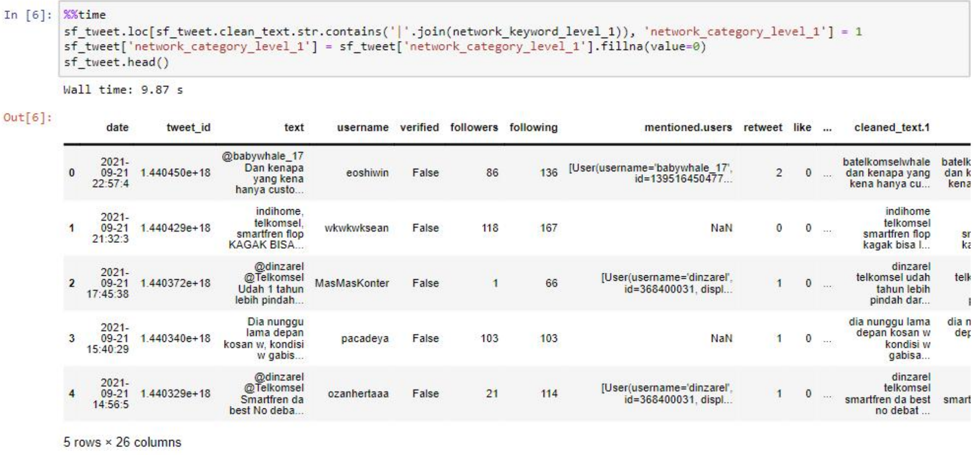


```python
def numbering_tweet_by_username_and_sentiment_change(x):
    #print(x)
    df_numbering = sf_tweet[sf_tweet['username'] == x].sort_values(by='date')
    for i in range(df_numbering.shape[0]):
        temp = int(df_numbering[i:i+1].tweet_id)

        sf_tweet.loc[sf_tweet.tweet_id == temp, 'number_of_tweet_by_username'] = int(i+1)

    #print('dfshape', df_numbering.shape[0], df_numbering['username'].values[0])

    if df_numbering.shape[0] > 1:
        for i in range(1, df_numbering.shape[0]+1):
            if i == df_numbering.shape[0]:
                break;

            i = int(i)


            #print('i:',i)

            #id_a = df_numbering.loc[df_numbering.number_of_tweet_by_username==i, 'tweet_id'].values[0]
            #id_b = df_numbering.loc[df_numbering.number_of_tweet_by_username==i+1, 'tweet_id'].values[0]

            #id_a = df_numbering[df_numbering['number_of_tweet_by_username']==i]['tweet_id'].values[0]
            #id_b = df_numbering[df_numbering['number_of_tweet_by_username']==i+1]['tweet_id'].values[0]
            id_a = df_numbering[i-1:i].tweet_id.values[0]
            id_b = df_numbering[i:i+1].tweet_id.values[0]

            #print('id a',id_a)
            #print('id b',id_b)

            score_a = df_numbering.loc[df_numbering.tweet_id==id_a, 'sentiment_score'].values[0]
            score_b = df_numbering.loc[df_numbering.tweet_id==id_b, 'sentiment_score'].values[0]

            polarity_a = df_numbering.loc[df_numbering.tweet_id==id_a, 'sentiment'].values[0]
            polarity_b = df_numbering.loc[df_numbering.tweet_id==id_b, 'sentiment'].values[0]

            #print('a:', score_a)
            #print('b:', score_b)

            if score_b > score_a:
                sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sentiment_changing_before_by_username'] = 'Sentiment_Up'
            elif score_b < score_a:
                sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sentiment_changing_before_by_username'] = 'Sentiment_Down'
            else:
                sf_tweet.loc[sf_tweet.tweet_id==id_b, 'Sentiment_changing_before_by_username'] = 'Sentiment_stable'


            if polarity_a=='neg' > polarity_b=='pos':
                sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sudden_Changes'] = 'to_positive'
            elif polarity_a=='pos' > polarity_b=='neg':
                sf_tweet.loc[sf_tweet.tweet_id == id_b, 'Sudden_Changes'] = 'to_negative'
            else:
                sf_tweet.loc[sf_tweet.tweet_id==id_b, 'Sudden_Changes'] = 'no_change'
```

H. Topic Categorization Labelling



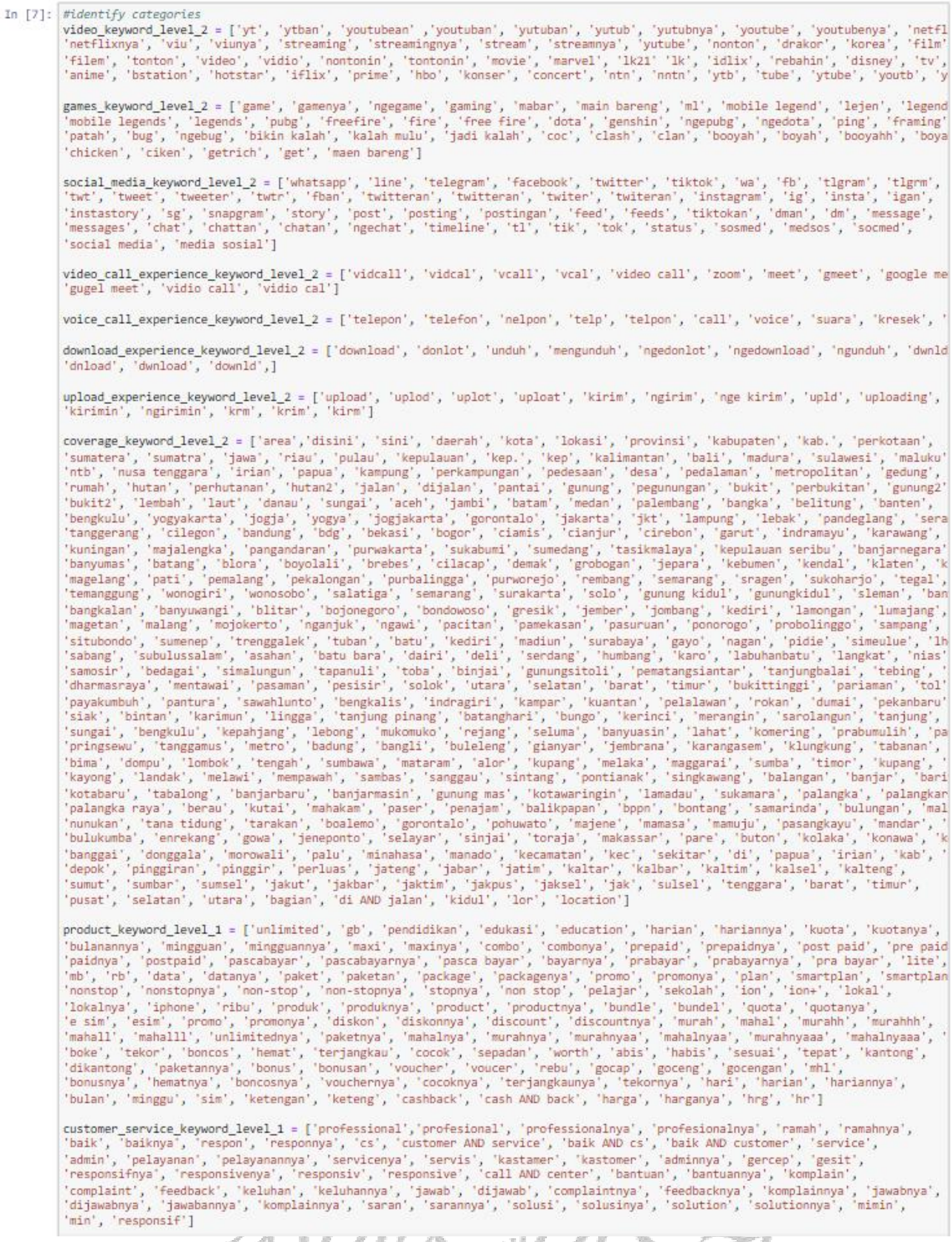
```
In [5]: #identify partial string to look for
        network_keyword_level_1 = ['jaringan', 'koneksi', 'sinyal', 'network', 'connection', 'signal', 'kecepatan', 'speed', 'ngebut', '
        'lemot', 'cepet', 'cepat', 'lambat', 'cpt', 'hilang', 'ilang', 'lag', 'ngelag', 'banter', 'kenceng', 'snyl', 'ilang', 'antilemot
        'antilelet', 'antilambat', 'kencang', 'lancar', 'gangguan', 'lamban', 'slow', 'bufer', 'buffer', 'load', 'ngeload', 'loading', '
        'lama', 'lemott', 'lemottt', 'lambatt', 'lambattt', 'cepett', 'cepettt', 'cepattt', 'cepatt', 'ngebutt', 'ngebuttt', 'lagg', 'la
        'ngelagg', 'ngelaggg', 'lancar', 'lancarr', 'lancarrr', 'lancarnyaa', 'lancarnya', 'lancarnyaaa', 'lemotnya', 'lemotnyaa', 'lemo
        'lambatnya', 'lambatnyaa', 'lambatnyaaa', 'cepatnya', 'cepatnyaa', 'cepatnyaaa', 'cepetnya', 'cepetnyaa', 'cepetnyaaa', 'kencang
        'kencangnya', 'kencangnyaa', 'kencangnyaaa', 'kencengnya', 'kencengnyaaa', 'kencengnyaa', 'lambannya', 'lambannyaaa', 'lambannya
        'ngelagnyaa', 'ngelagnyaaa', 'koneksinyaa', 'koneksinyaaa', 'jaringannyaa', 'jaringannyaaa', 'lagnyaa', 'lagnya', 'lagnyaaa', 'b
        'buffernyaaa', 'buffernya', 'hujan', 'jangkauanya', 'jangkauan', 'jaring', 'ganggu', 'down', 'maintenance', 'sabar', 'gg', 'fast
        'bufer', 'buferr', 'bufferr', 'edge', 'loadingg', 'muter', 'pusing']
```



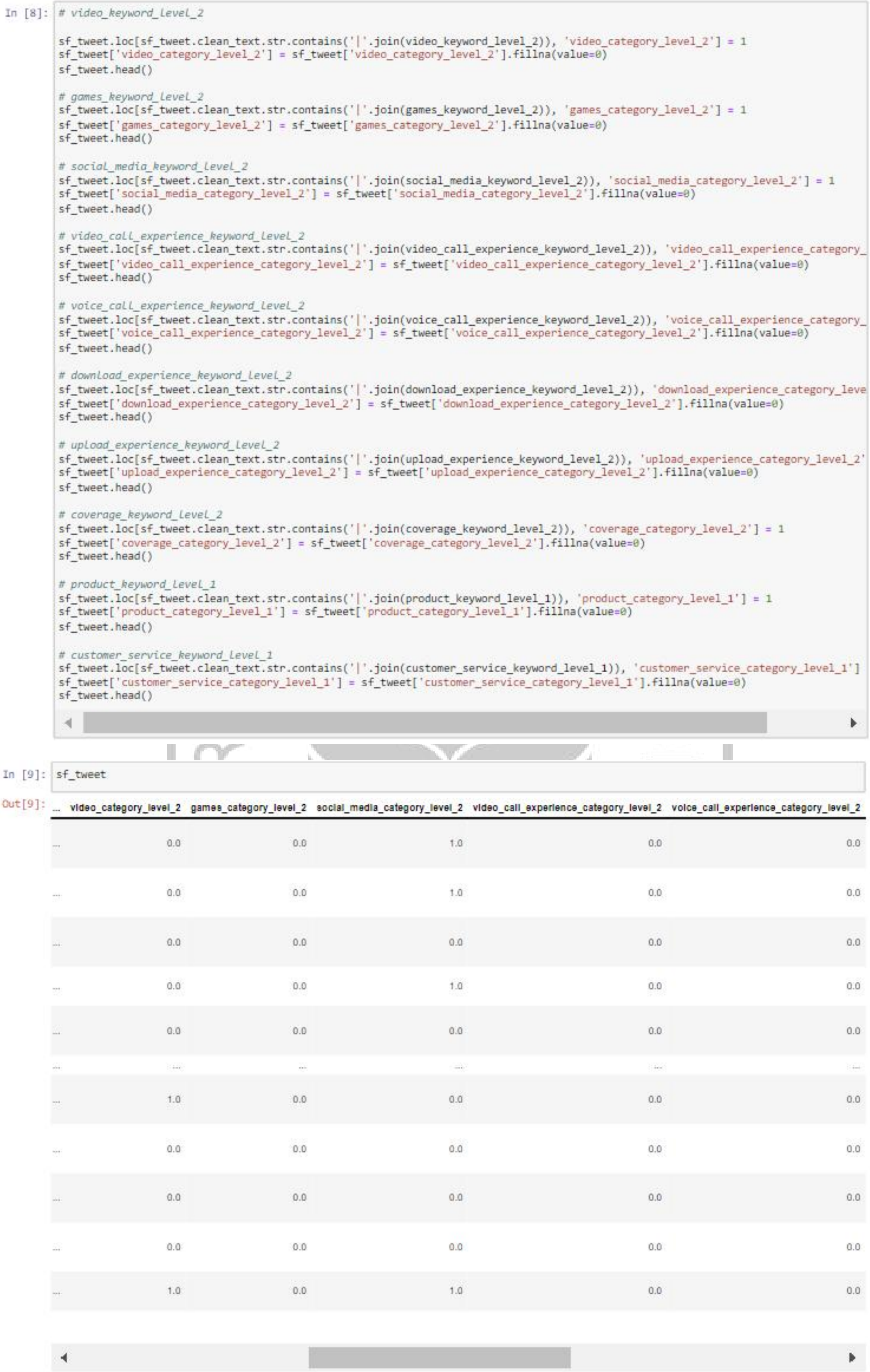
```
In [6]: %%time
        sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(network_keyword_level_1)), 'network_category_level_1'] = 1
        sf_tweet['network_category_level_1'] = sf_tweet['network_category_level_1'].fillna(value=0)
        sf_tweet.head()

        Wall time: 9.87 s

Out[6]:
```

	date	tweet_id	text	username	verified	followers	following	mentioned.users	retweet	like	...	cleaned_text.1	
0	2021-09-21 22:57:4	1.440450e+18	@babywhale_17 Dan kenapa yang kena hanya custo...	eoshiwin	False	86	136	[User(username='babywhale_17', id=139516450477...	2	0	...	batelkomselwhale dan kenapa yang kena hanya cu...	batelk dan k kena
1	2021-09-21 21:32:3	1.440429e+18	indihome, telkomsel, smartfren flop KAGAK BISA...	wkwkwksean	False	118	167	NaN	0	0	...	indihome telkomsel smartfren flop kagak bisa l...	sr ka
2	2021-09-21 17:45:38	1.440372e+18	@dinzarel @Telkomsel Udah 1 tahun lebih pindah...	MasMasKonter	False	1	66	[User(username='dinzarel', id=368400031, displ...	1	0	...	dinzarel telkomsel udah tahun lebih pindah dar...	telk
3	2021-09-21 15:40:29	1.440340e+18	Dia nunggu lama depan kosan w, kondisi w gabis...	pacadeya	False	103	103	NaN	1	0	...	dia nunggu lama depan kosan w kondisi w gabisa...	dia n dep
4	2021-09-21 14:56:5	1.440329e+18	@dinzarel @Telkomsel Smartfren da best No deba...	ozanhertaaa	False	21	114	[User(username='dinzarel', id=368400031, displ...	1	0	...	dinzarel telkomsel smartfren da best no debat ...	smart

5 rows × 26 columns



```
In [7]: #identify categories
video_keyword_level_2 = ['yt', 'ytban', 'youtubean' ,'youtuban', 'yutuban', 'yutub', 'yutubnya', 'youtube', 'youtubenya', 'netfl
'netflixnya', 'viu', 'viunya', 'streaming', 'streamingnya', 'stream', 'streamnya', 'yutube', 'nonton', 'drakor', 'korea', 'film'
'filem', 'tonton', 'video', 'vidio', 'nontonin', 'tontonin', 'movie', 'marvel', 'lk21' 'lk', 'idlix', 'rebahin', 'disney', 'tv',
'anime', 'bstation', 'hotstar', 'iflix', 'prime', 'hbo', 'konser', 'concert', 'ntn', 'nntn', 'ytb', 'tube', 'ytube', 'youtb', 'y

games_keyword_level_2 = ['game', 'gamenya', 'ngegame', 'gaming', 'mabar', 'main bareng', 'ml', 'mobile legend', 'lejen', 'legend
'mobile legends', 'legends', 'pubg', 'freefire', 'fire', 'free fire', 'dota', 'genshin', 'ngepubg', 'ngedota', 'ping', 'framing'
'patah', 'bug', 'ngebug', 'bikin kalah', 'kalah mulu', 'jadi kalah', 'coc', 'clash', 'clan', 'booyah', 'boyah', 'booyahh', 'boya
'chicken', 'ciken', 'getrich', 'get', 'maen bareng']

social_media_keyword_level_2 = ['whatsapp', 'line', 'telegram', 'facebook', 'twitter', 'tiktok', 'wa', 'fb', 'tlgram', 'tlgrm',
'twt', 'tweet', 'tweeter', 'twtr', 'fban', 'twitteran', 'twitteran', 'twiter', 'twiteran', 'instagram', 'ig', 'insta', 'igan',
'instastory', 'sg', 'snapgram', 'story', 'post', 'posting', 'postingan', 'feed', 'feeds', 'tiktokan', 'dman', 'dm', 'message',
'messages', 'chat', 'chattan', 'chatan', 'ngechat', 'timeline', 'tl', 'tik', 'tok', 'status', 'sosmed', 'medsos', 'socmed',
'social media', 'media sosial']

video_call_experience_keyword_level_2 = ['vidcall', 'vidcal', 'vcall', 'vcal', 'video call', 'zoom', 'meet', 'gmeet', 'google me
'gugel meet', 'vidio call', 'vidio cal']

voice_call_experience_keyword_level_2 = ['telepon', 'telefon', 'nelpon', 'telp', 'telpon', 'call', 'voice', 'suara', 'kresek', '

download_experience_keyword_level_2 = ['download', 'donlot', 'unduh', 'mengunduh', 'ngedonlot', 'ngedownload', 'ngunduh', 'dwnld
'dnload', 'dwnload', 'downld',]

upload_experience_keyword_level_2 = ['upload', 'uplod', 'uplot', 'uploat', 'kirim', 'ngirim', 'nge kirim', 'upld', 'uploading',
'kirimin', 'ngirimin', 'krm', 'krim', 'kirm']

coverage_keyword_level_2 = ['area','disini', 'sini', 'daerah', 'kota', 'lokasi', 'provinsi', 'kabupaten', 'kab.', 'perkotaan',
'sumatera', 'sumatra', 'jawa', 'riau', 'pulau', 'kepulauan', 'kep.', 'kep', 'kalimantan', 'bali', 'madura', 'sulawesi', 'maluku'
'ntb', 'nusa tenggara', 'irian', 'papua', 'kampung', 'perkampungan', 'pedesaan', 'desa', 'pedalaman', 'metropolitan', 'gedung',
'rumah', 'hutan', 'perhutanan', 'hutan2', 'jalan', 'dijalan', 'pantai', 'gunung', 'pegunungan', 'bukit', 'perbukitan', 'gunung2'
'bukit2', 'lembah', 'laut', 'danau', 'sungai', 'aceh', 'jambi', 'batam', 'medan', 'palembang', 'bangka', 'belitung', 'banten',
'bengkulu', 'yogyakarta', 'jogja', 'yogya', 'jogjakarta', 'gorontalo', 'jakarta', 'jkt', 'lampung', 'lebak', 'pandeglang', 'sera
'tanggerang', 'cilegon', 'bandung', 'bdg', 'bekasi', 'bogor', 'ciamis', 'cianjur', 'cirebon', 'garut', 'indramayu', 'karawang',
'kuningan', 'majalengka', 'pangandaran', 'purwakarta', 'sukabumi', 'sumedang', 'tasikmalaya', 'kepulauan seribu', 'banjarnegara'
'banyumas', 'batang', 'blora', 'boyolali', 'brebes', 'cilacap', 'demak', 'grobogan', 'jepara', 'kebumen', 'kendal', 'klaten', 'k
'magelang', 'pati', 'pemalang', 'pekalongan', 'purbalingga', 'purworejo', 'rembang', 'semarang', 'sragen', 'sukoharjo', 'tegal',
'temanggung', 'wonogiri', 'wonosobo', 'salatiga', 'semarang', 'surakarta', 'solo', 'gunung kidul', 'gunungkidul', 'sleman', 'ban
'bangkalan', 'banyuwangi', 'blitar', 'bojonegoro', 'bondowoso', 'gresik', 'jember', 'jombang', 'kediri', 'lamongan', 'lumajang',
'magetan', 'malang', 'mojokerto', 'nganjuk', 'ngawi', 'pacitan', 'pamekasan', 'pasuruan', 'ponorogo', 'probolinggo', 'sampang',
'situbondo', 'sumenep', 'trenggalek', 'tuban', 'batu', 'kediri', 'madiun', 'surabaya', 'gayo', 'nagan', 'pidie', 'simeulue', 'lh
'sabang', 'subulussalam', 'asahan', 'batu bara', 'dairi', 'deli', 'serdang', 'humbang', 'karo', 'labuhanbatu', 'langkat', 'nias'
'samosir', 'bedagai', 'simalungun', 'tapanuli', 'toba', 'binjai', 'gunungsitoli', 'pematangsiantar', 'tanjungbalai', 'tebing', '
'dharmasraya', 'mentawai', 'pasaman', 'pesisir', 'solok', 'utara', 'selatan', 'barat', 'timur', 'bukittinggi', 'pariaman', 'tol'
'payakumbuh', 'pantura', 'sawahlunto', 'bengkalis', 'indragiri', 'kampar', 'kuantan', 'pelalawan', 'rokan', 'dumai', 'pekanbaru'
'siak', 'bintan', 'karimun', 'lingga', 'tanjung pinang', 'batanghari', 'bungo', 'kerinci', 'merangin', 'sarolangun', 'tanjung',
'sungai', 'bengkulu', 'kepahjang', 'lebong', 'mukomuko', 'rejang', 'seluma', 'banyuasin', 'lahat', 'komering', 'prabumulih', 'pa
'pringsewu', 'tanggamus', 'metro', 'badung', 'bangli', 'buleleng', 'gianyar', 'jembrana', 'karangasem', 'klungkung', 'tabanan',
'bima', 'dompu', 'lombok', 'tengah', 'sumbawa', 'mataram', 'alor', 'kupang', 'melaka', 'maggarai', 'sumba', 'timor', 'kupang', '
'kayong', 'landak', 'melawi', 'mempawah', 'sambas', 'sanggau', 'sintang', 'pontianak', 'singkawang', 'balangan', 'banjar', 'bari
'kotabaru', 'tabalong', 'banjarbaru', 'banjarmasin', 'gunung mas', 'kotawaringin', 'lamadau', 'sukamara', 'palangka', 'palangkar
'palangka raya', 'berau', 'kutai', 'mahakam', 'paser', 'penajam', 'balikpapan', 'bppn', 'bontang', 'samarinda', 'bulungan', 'mal
'nunukan', 'tana tidung', 'tarakan', 'boalemo', 'gorontalo', 'pohuwato', 'majene', 'mamasa', 'mamuju', 'pasangkayu', 'mandar', '
'bulukumba', 'enrekang', 'gowa', 'jeneponto', 'selayar', 'sinjai', 'toraja', 'makassar', 'pare', 'buton', 'kolaka', 'konawa', 'k
'banggai', 'donggala', 'morowali', 'palu', 'minahasa', 'manado', 'kecamatan', 'kec', 'sekitar', 'di', 'papua', 'irian', 'kab', '
'depok', 'pinggiran', 'pinggir', 'perluas', 'jateng', 'jabar', 'jatim', 'kaltar', 'kalbar', 'kaltim', 'kalsel', 'kalteng',
'sumut', 'sumbar', 'sumsel', 'jakut', 'jakbar', 'jaktim', 'jakpus', 'jaksel', 'jak', 'sulsel', 'tenggara', 'barat', 'timur',
'pusat', 'selatan', 'utara', 'bagian', 'di AND jalan', 'kidul', 'lor', 'location']

product_keyword_level_1 = ['unlimited', 'gb', 'pendidikan', 'edukasi', 'education', 'harian', 'hariannya', 'kuota', 'kuotanya',
'bulanannya', 'mingguan', 'mingguannya', 'maxi', 'maxinya', 'combo', 'combonya', 'prepaid', 'prepaidnya', 'post paid', 'pre paid
'paidnya', 'postpaid', 'pascabayar', 'pascabayarnya', 'pasca bayar', 'bayarnya', 'prabayar', 'prabayarnya', 'pra bayar', 'lite',
'mb', 'rb', 'data', 'datanya', 'paket', 'paketan', 'package', 'packagenya', 'promo', 'promonya', 'plan', 'smartplan', 'smartplan
'nonstop', 'nonstopnya', 'non-stop', 'non-stopnya', 'stopnya', 'non stop', 'pelajar', 'sekolah', 'ion', 'ion+', 'lokal',
'lokalnya', 'iphone', 'ribu', 'produk', 'produknya', 'product', 'productnya', 'bundle', 'bundel', 'quota', 'quotanya',
'e sim', 'esim', 'promo', 'promonya', 'diskon', 'diskonnya', 'discount', 'discountnya', 'murah', 'mahal', 'murahh', 'murahhh',
'mahall', 'mahalll', 'unlimitednya', 'paketnya', 'mahalnya', 'murahnya', 'murahnyaa', 'mahalnyaa', 'murahnyaaa', 'mahalnyaaa', '
'boke', 'tekor', 'boncos', 'hemat', 'terjangkau', 'cocok', 'sepadan', 'worth', 'abis', 'habis', 'sesuai', 'tepat', 'kantong',
'dikantong', 'paketannya', 'bonus', 'bonusan', 'voucher', 'voucer', 'rebu', 'gocap', 'goceng', 'gocengan', 'mhl',
'bonusnya', 'hematnya', 'boncosnya', 'vouchernya', 'cocoknya', 'terjangkaunya', 'tekornya', 'hari', 'harian', 'hariannya',
'bulan', 'minggu', 'sim', 'ketengan', 'keteng', 'cashback', 'cash AND back', 'harga', 'harganya', 'hrg', 'hr']

customer_service_keyword_level_1 = ['professional','profesional', 'professionalnya', 'profesionalnya', 'ramah', 'ramahnya',
'baik', 'baiknya', 'respon', 'responnya', 'cs', 'customer AND service', 'baik AND cs', 'baik AND customer', 'service',
'admin', 'pelayanan', 'pelayanannya', 'servicenya', 'servis', 'kastamer', 'kastomer', 'adminnya', 'gercep', 'gesit',
'responsifnya', 'responsivenya', 'responsiv', 'responsive', 'call AND center', 'bantuan', 'bantuannya', 'komplain',
'complaint', 'feedback', 'keluhan', 'keluhannya', 'jawab', 'dijawab', 'complaintnya', 'feedbacknya', 'komplainnya', 'jawabnya',
'dijawabnya', 'jawabannya', 'komplainnya', 'saran', 'sarannya', 'solusi', 'solusinya', 'solution', 'solutionnya', 'mimin',
'min', 'responsif']
```



In [8]:

```
# video_keyword_level_2

sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(video_keyword_level_2)), 'video_category_level_2'] = 1
sf_tweet['video_category_level_2'] = sf_tweet['video_category_level_2'].fillna(value=0)
sf_tweet.head()

# games_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(games_keyword_level_2)), 'games_category_level_2'] = 1
sf_tweet['games_category_level_2'] = sf_tweet['games_category_level_2'].fillna(value=0)
sf_tweet.head()

# social_media_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(social_media_keyword_level_2)), 'social_media_category_level_2'] = 1
sf_tweet['social_media_category_level_2'] = sf_tweet['social_media_category_level_2'].fillna(value=0)
sf_tweet.head()

# video_call_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(video_call_experience_keyword_level_2)), 'video_call_experience_category_
sf_tweet['video_call_experience_category_level_2'] = sf_tweet['video_call_experience_category_level_2'].fillna(value=0)
sf_tweet.head()

# voice_call_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(voice_call_experience_keyword_level_2)), 'voice_call_experience_category_
sf_tweet['voice_call_experience_category_level_2'] = sf_tweet['voice_call_experience_category_level_2'].fillna(value=0)
sf_tweet.head()

# download_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(download_experience_keyword_level_2)), 'download_experience_category_leve
sf_tweet['download_experience_category_level_2'] = sf_tweet['download_experience_category_level_2'].fillna(value=0)
sf_tweet.head()

# upload_experience_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(upload_experience_keyword_level_2)), 'upload_experience_category_level_2'
sf_tweet['upload_experience_category_level_2'] = sf_tweet['upload_experience_category_level_2'].fillna(value=0)
sf_tweet.head()

# coverage_keyword_level_2
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(coverage_keyword_level_2)), 'coverage_category_level_2'] = 1
sf_tweet['coverage_category_level_2'] = sf_tweet['coverage_category_level_2'].fillna(value=0)
sf_tweet.head()

# product_keyword_level_1
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(product_keyword_level_1)), 'product_category_level_1'] = 1
sf_tweet['product_category_level_1'] = sf_tweet['product_category_level_1'].fillna(value=0)
sf_tweet.head()

# customer_service_keyword_level_1
sf_tweet.loc[sf_tweet.clean_text.str.contains('|'.join(customer_service_keyword_level_1)), 'customer_service_category_level_1']
sf_tweet['customer_service_category_level_1'] = sf_tweet['customer_service_category_level_1'].fillna(value=0)
sf_tweet.head()
```

In [9]:

```
sf_tweet
```

Out[9]:

```
... video_category_level_2 games_category_level_2 social_media_category_level_2 video_call_experience_category_level_2 voice_call_experience_category_level_2
...                    0.0                    0.0                           1.0                                    0.0                                    0.0
...                    0.0                    0.0                           1.0                                    0.0                                    0.0
...                    0.0                    0.0                           0.0                                    0.0                                    0.0
...                    0.0                    0.0                           1.0                                    0.0                                    0.0
...                    0.0                    0.0                           0.0                                    0.0                                    0.0
...                    ...                    ...                           ...                                    ...                                    ...
...                    1.0                    0.0                           0.0                                    0.0                                    0.0
...                    0.0                    0.0                           0.0                                    0.0                                    0.0
...                    0.0                    0.0                           0.0                                    0.0                                    0.0
...                    0.0                    0.0                           0.0                                    0.0                                    0.0
...                    1.0                    0.0                           1.0                                    0.0                                    0.0
```