

CHAPTER 3

RESEARCH METHODOLOGY

3.1. Literature Study

From the journals I collected on previous chapter, there are 10 journals that relates to my study on sentiment analysis including using TextBlob library, Term Frequency–Inverse Document Frequency (TF-IDF), and Support Vector Machine (SVM) algorithm they used for their some previous work.

3.2. Datasets and Code

There are 2 datasets used in this project. The main dataset and prediction dataset. The main dataset used to make analyzing positive and negative sentiment process. The prediction dataset used to identify the sentiment of each reviews directly after making the sentiment process. The number of main dataset in this project are 67986 datas obtained from Kaggle¹ which only uses the review file. Whereas, the number on prediction dataset are 720 datas obtained from Amazon² which contains the Iphone 11 review from the web scrapper by using Python code and Docker Desktop with Splash, a Javascript rendering service from Github³ website.

This project uses Python through IPython notebook (ipynb) file using Jupyter Notebook. 2 obtained datasets used on this project are in csv format, so this project will input 2 datasets for making analyzer and predict sentiment on it.

3.3. Implementation and Analysis

3.3.1. Data Crawling and Collecting

As mentioned above, they are two datasets, the main dataset and prediction dataset. The main dataset are obtained from Kaggle dataset that contains Amazon's phone reviews and the prediction dataset that obtained from scrapping the website using Python code based on Github

¹ <https://www.kaggle.com/datasets/grikomsn/amazon-cell-phones-reviews?select=20191226-reviews.csv>

² <https://www.amazon.com/Apple-iPhone-11-64GB-Unlocked/dp/B07ZPKF8RG/?th=1>

³ <https://github.com/jhnwr/scrape-amazon-reviews>

website that using Docker Desktop with Splash Javascript. This scrapping code collects all of reviews from all review pages that all of reviews are saved through csv files although often not all reviews are completely collected because of scraper limited collecting ability. In main dataset, there are columns that are in csv such as 'asin', 'name', 'overall', 'reviewTime', 'verified', 'title', 'reviewText', and 'helpfulVotes' which are modified a bit on some column names such as 'rating' column changed into 'overall', 'date' column changed into 'reviewTime', 'body' changed into 'reviewText' and besides that, in the date column, I only change the date into only year in example like 'March 5, 2016' change only '2016'. In this project, I only use 'overall', 'reviewTime', 'reviewText', and 'helpfulVotes' columns to make sentiment analyzer. And in prediction dataset, there are columns that are in csv such as 'product', 'title', 'date', 'rating', and 'body' to predicting sentiment after analyzing using main dataset.

3.3.2. Preprocessing Data

There are some processing data before mark and average the sentiment from each datas in the main dataset. The same processing data is also used in the prediction dataset before detecting the sentiment.

1. Removing Symbols

The symbols, numbers, emoticons and other unusual symbols on each reviews are removed into only uppercases, lowercases, and spaces. For example, "I don't like this phone!" changed into "I dont like this phone".

2. Lowercasing

All of the each letters on reviews are changed into lowercase. For example, "My sister likes Samsung Galaxy S Note 8 phone" changed into "my sister likes samsung galaxy s note 8 phone"

3. Tokenization

After removing symbols and lowercasing, each word of reviews are tokenized to make an array to ease the stopwords, lemmatization, and stemming process. So, if after using post-tokenization process like stopwords process, the changed stopword process result on each reviews are retokenize again toward another process.

4. Stopwords

After tokenizing these words, the tokenized reviews are processed into stopwords removal which unimportant words in each word on reviews are removed like I, we, before, after, etc. In this project, there are some words that should be removed by stopwords removal but they are important for me to not remove it like 'no' and 'not' words because without these words, the meaning of the word will change. For example, "The phone is not good" if processed with stopwords removal, then the sentence of that changed into "phone good" which it means that the phone is good. So I make the stopwords exception of the sentence into "phone not good" that means the phone is not good.

5. Lemmatization

Lemmatization is the changing of certain words, mostly plural words into singular words like "phones", "likes", "days" into their root like "phone", "like", "day". Because in lemmatization always change plural words into singular words because of the context of the sentence, there are some words like -ing, -ed, and more inflected words are not processed with lemmatization.

6. Stemming

More like lemmatization, but stemming is not only change the plural words, but also all of inflected words like "buying", "complicated", "friendship" into their root like "buy", "complicate", and "friend" without looking at the sentence context by reducing the inflected words. But because stemming is a library that only reduces the inflected words without context, there are some words that it returns a word without -e, -ized, -ent, -ed or changed -y into -i and so on like "charge" changed into "charg", "unauthorized" changed into "unauthor", "different" changed into "differ", "certified" changed into "certifi", and "battery" into "batteri". The stemming used in this project is Snowball Stemmer which is a developed form of Porter Stemmer.

When I only used Lemmatization for preprocessing data, some of sentiment mean result of them are higher or lower than I used Lemmatization and Stemming at once. But the same sentiment mean value on Lemmatization and Stemming preprocessed text are same value on only using Stemming which can conclude that Stemming is the another advanced version of Lemmatization in terms of word changing. But in this project, I decided to use Lemmatization and Stemming at once on balanced sentiment mean result so the words I used to preprocessing are not look higher or lower because of influence of some inflected words in them and it make sentiment looks more accurate to interpret them.

3.3.3. TextBlob

TextBlob is a Python library that used to processing the sentence into polarity and subjectivity. TextBlob uses Application Programming Interface (API) with Nature Language Processing (NLP) principle. Many features like noun phrase extraction, Part-of-speech tagging (POS Tagging), N-grams, and certainly sentiment analysis and others are in TextBlob. In this project, I only focused on using sentiment analysis on TextBlob which it returns a float number by using polarity on it.

The polarity on TextBlob library is used for sentiment analysis that ranged -1 to 1. Although there is a journal used 0 range as a neutral sentiment, but because of analyzing sentiment on product reviews, the 0 range polarity used as a negative sentiment so there are no neutral on this sentiment. That because in the e-commerce product review, people tend to review how good and how bad the product after they bought it.

So, example if the sentence “The phone is good, i like it!” are processed with TextBlob library to know the polarity of it (`TextBlob("The phone is good, i like it!").sentiment.polarity`), then it will return 0,875 as a float. And if using the sentence “Too bad. So disappointed with the phone screen” (`TextBlob("Too bad. So disappointed with the phone screen").sentiment.polarity`), then it will return -0.725 as a float.

3.3.4. Sentiment Mean

After through preprocessing process step, all of reviews are processed with sentiment polarity from TextBlob as I mentioned before with range -1 until 1 and overall mapping from 1, 2, 3, 4, and 5 into -1, -0.5, 0, 0.5, and 1 fitted to TextBlob sentiment polarity range. If there are any

helpful votes in a review, then mapped overall are multiplied by 2. Then all of numbers are there are counted and averaged into a sentiment mean. If the number >0 , then sentiment is positive and if the number ≤ 0 , then sentiment is negative.

If only use TextBlob sentiment, comments are in 1 and 2 rating overall which are tend to negative got >0 so it wrongly classified as Positive, and comments are in 4 and 5 rating overall which are tend to positive got ≤ 0 so it wrongly classified as Negative although for reviews with 4 and 5 rating in this case are rarely found.

The helpful votes calculation on Amazon dataset is used to increase the sentiment mean or fixing the error sentiment labelling for indicate that there are people who support this review from a user. So, it means that the people are agreeing this review and in this calculation, an mapped overall are added for value addition.

- Without helpful votes

$$\text{Sentiment mean} = \frac{\text{Mapped overall} + \text{TextBlob polarity}}{2}$$

Example, if there are a review that have 0,7 on Textblob polarity and have 5 star rating and no helpful vote, then it computed like this :

$$\text{Sentiment mean} = \frac{1 + 0,7}{2}$$

$$\text{Sentiment mean} = 0,85$$

Remember that the rating stars on review are mapped as I mentioned before, so 5 star polarity have changed into 1. After that, it can seen that the number of it is >0 , so the sentiment result is Positive.

- With helpful votes

$$\text{Sentiment mean} = \frac{(2 \times \text{Mapped overall}) + \text{TextBlob polarity}}{3}$$

Example, if there are a review that have 0,274 on Textblob polarity and have 4 star rating and 4 helpful votes, then it computed like this :

$$\textit{Sentiment mean} = \frac{(2 \times 0,5) + 0,274}{3}$$

$$\textit{Sentiment mean} = 0,425$$

Remember that the rating stars on review are mapped as I mentioned before, so 4 star polarity have changed into 0,5. Because of any helpful vote here, so the mapped overall is multiplied by 2. After that, it can seen that the number of it is >0, so the sentiment result is Positive.

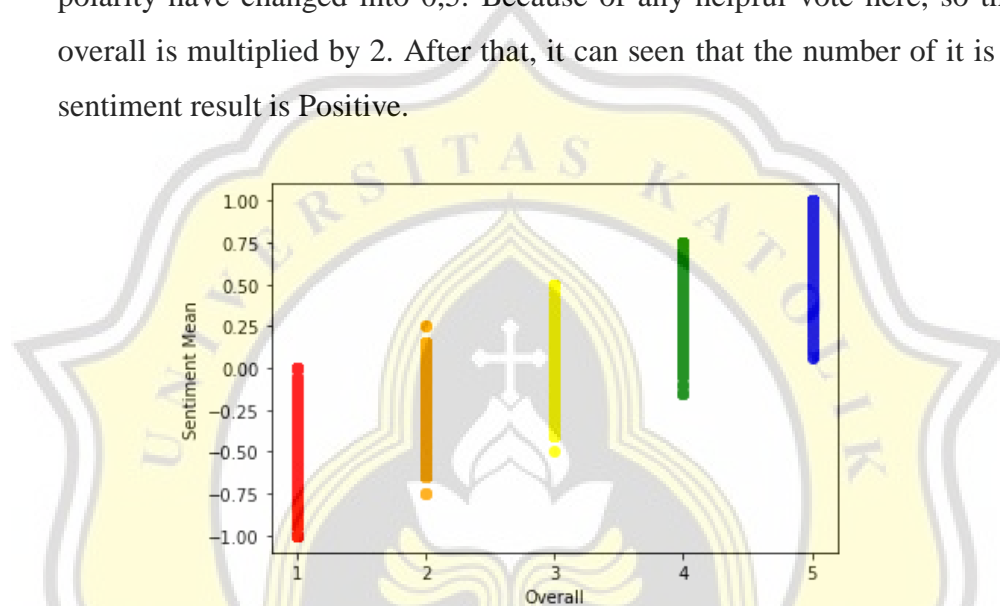


Figure 3.1 The result of all sentiment calculation with overalls in diagram

In the Figure 3.1, using all of reviews to compute sentiment mean, the result of them are tend to linear towards the rating with all of 1 rating reviews are in negative sentiment and all 5 rating reviews are in positive reviews. But there are some positive sentiment in 2 rating reviews and some negative sentiment in 4 rating reviews although of few in them.

In helpful votes calculation when I used helpful votes calculation if a review data have helpful votes on it, the false sentiment labelling are proved decreased. And it can be seen that there are some data with 2 rating with any helpful votes which incorrectly labelled as positive sentiment from sentiment calculation without helpful votes calculation are changed into negative sentiment because of using helpful votes sentiment calculation.

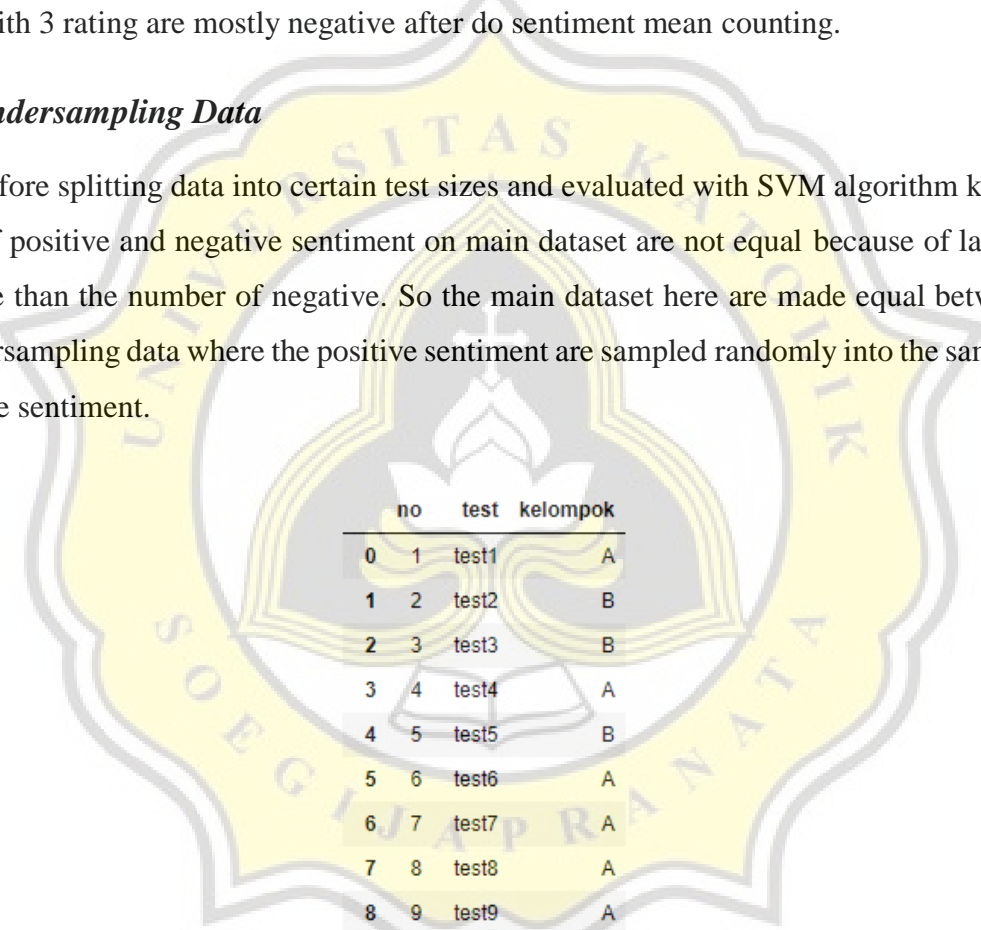
In example, the sentiment mean result of a review with 2 rating is 0,15 which this number show that the sentiment is Positive because of >0 value. Because of any helpful votes in this

review, the sentiment mean calculation is changed to sentiment mean calculation with helpful votes which the ranged overall multiplied by 2. The result will changed the value and the sentiment mean from 0,15 into -0,067 that means the sentiment is Negative because of ≤ 0 value.

Although on some journals that 3 rating reviews are not used in another experiment, in this project I decided to not remove the reviews with 3 rating for using sentiment mean based on reviews. So, the 3 rating reviews can be in positive or negative reviews because there are some reviewers that tend on bad reviews despite on good things on product and vice versa which the reviews with 3 rating are mostly negative after do sentiment mean counting.

3.3.5. Undersampling Data

Before splitting data into certain test sizes and evaluated with SVM algorithm kernels, the number of positive and negative sentiment on main dataset are not equal because of larger number of positive than the number of negative. So the main dataset here are made equal between them with undersampling data where the positive sentiment are sampled randomly into the same number of negative sentiment.



no	test	kelompok	
0	1	test1	A
1	2	test2	B
2	3	test3	B
3	4	test4	A
4	5	test5	B
5	6	test6	A
6	7	test7	A
7	8	test8	A
8	9	test9	A
9	10	test10	A
10	11	test11	B
11	12	test12	A
12	13	test13	A
13	14	test14	B
14	15	test15	A

Figure 3.1 Example from a dataset that contains A and B group

Example, there is a dataset that contains 15 datas that contain A and B. The A group has 10 datas and the B group has 5 datas. Then I want to make an equal between A and B group because it is not balanced. I use the least number of data that is B group to take random sample of A group into the same number of B group so the number of each A and B group are 5 datas.

no	test	kelompok	no	test	kelompok		
9	10	test10	A	5	6	test6	A
14	15	test15	A	9	10	test10	A
12	13	test13	A	12	13	test13	A
7	8	test8	A	6	7	test7	A
11	12	test12	A	11	12	test12	A

Figure 3.2 Different result after 1st sampling (left) and 2nd sampling (right)

Because of random sampling on A group, the result of this sampling are different when tried to sampling again. So, the result of sampling always different when sampling of them is going on and on. Therefore, in this sentiment sampling the number sample of positive sentiment are always random when I used another file for different test size data or if I repeat it.

3.3.6. TF-IDF

TF-IDF is a kind of calculation that calculate the frequent of word occurs that change each word into a vector. The formula of TF-IDF is :

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

Where tf is the number of x words in y document, df is the number of documents that contains x word, and N is the number of words in y document. If the word occurs more, then the vector value (w) will be smaller and if the word occurs less, then the vector value (w) will be bigger.

In this project, after splitting the main dataset into x train and x test, I use TF-IDF from Scikit Learn Library from Python to make them into a vector datas. In x train, it used to fit

transform process and in x test, it used to transform process. According from Scikit Learn website⁴, fit transform is to learn vocabulary and idf, return document-term matrix and transform is to transform documents to document-term matrix.

3.3.7. SVM Algorithm

SVM Algorithm is a kind of supervised machine learning algorithm that uses a hyperplane in many form through the kernel to segregate and group the datas based from different data kinds in the terms of classifying the data. The result of SVM Algorithm usually are used to evaluate the dataset to focus on the accuracy obtained from testing and training data. There are 4 SVM kernels which I used on this project :

1. Linear

It is the basic SVM kernels which generally used on data classifying. This SVM kernel classifying the different datas by separating each different datas with only a straight line with the formula :

$$K(x, x_i) = \text{sum}(x * x_i)$$

Where all of x and x_i variable which are the different data type are multiplied and added up into a line separation between different data types.

2. Radial Basis Function (RBF)

It is the type of SVM kernels that separate each of different datas by separating each different datas by bordering area with the majority of data type are marked in majority area with drawing borders. The formula of this kernel is :

$$K(x, x_i) = \exp(-\gamma * \text{sum}((x - x_i)^2))$$

Where γ is the number of the nearest data which it decide to determining the gap between datas (x) and another different datas (x_i).

⁴https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html?highlight=tfidfvectorizer#sklearn.feature_extraction.text.TfidfVectorizer

3. Polynomial

The Polynomial SVM kernel is the kernel that separate each of different datas by separating each different datas with curved lines. This curved lines are representing the vector similarity of each different datas. The formula of this kernel is :

$$K(x, x_i) = 1 + \text{sum}(x * x_i)^d$$

Same with Linear Kernel, but there are d variable and 1 constant where d is the curve level of data classification border which if the d variable is larger and more curved, then the accuracy of this kernel will decrease and unstable.

4. Sigmoid

This SVM kernel is the kernel which based from Neural Networks to separating each different datas that use two layers from activation function for artificial neurons to classifying each of different datas. The formula of this kernel is :

$$K(x, x_i) = \tanh(\gamma(x * x_i) + c)$$

Where γ is the number of the nearest data which it decide to determining the gap between datas (x) and another different datas (x_i). And c or cost is the SVM parameters that used as a gap between data (x) and another different data (x_i).

3.3.8. Prediction Dataset

After using main dataset to evaluating from SVM dataset, the prediction dataset used on focusing whether the sentiment on each reviews are positive or negative. Before predicting, the reviews from prediction dataset is processed through preprocessing like on using on the main dataset. After that, it transformed into using transform process from TF-IDF library and then the transformed review are predicted into a certain sentiment. If it occurs “[1]”, that mean the review predicted Positive, and if it occurs “[0]”, that mean the review predicted Negative.