JURNAL INFORMATIKA, Vol.11 No.2 October 2024, Page 55-64
ISSN (Print) 2355-6579 | ISSN (Online) 2528-2247
DOI: https://doi.org/10.31294/inf.v11i2.22066

Sentiment Analysis of Sirekap Application Review Using Logistic Regression Algorithm

Audi Hagi[1*], Dionisia Bhisetya Rarasati[2]

[1,2] Faculty of Technology and Design, Bunda Mulia University
Jl. Jalur Sutera Barat kav 7-9, Alam Sutera, Tangerang 15143, Banten, Indonesia

Correspondence e-mail: ahagi010902@gmail.com

Submission:	Revision:	Acceptance:	Available Online:
15-05-2024	28-06-2024	11-07-2024	20-08-2024

Abstract

The General Election Commission (KPU) launched the Sirekap application as a tool in the election process. This research aims to implement and measure the accuracy of the Logistic Regression algorithm in sentiment analysis of Sirekap application reviews. After data preprocessing and z-score normalization, the dataset is divided into 80% training data and 20% test data. Words are weighted with TF-IDF method, and the model is trained using Logistic Regression, then tested with confusion matrix. The analysis results show that the Logistic Regression algorithm is effective in classifying the sentiment of Sirekap app reviews with an accuracy of 91%. The precision scores for positive and negative classes are 90% and 92%, respectively. The recall scores for positive and negative classes are 94% and 87%, respectively. The f1-score for positive and negative classes are 92% and 90% respectively. From these results, it can be concluded that the majority of users gave negative reviews before data balancing, with only 9% positive reviews, indicating a user dissatisfaction rate of 91%. However, after data balancing, positive reviews increased to 55%, while negative reviews dropped to 45%. This shows that the user dissatisfaction rate decreased to 45%. The results of this study can be used by the KPU to understand and improve the quality of the Sirekap application ahead of the 2024 Regional Head Election (Pilkada). Suggestions for future research are to use a larger and more balanced dataset, as well as enrich the stopwords dictionary and word merge dictionary to anticipate new word variations.

Keywords: Logistic Regression, Sentiment Analysis, Sirekap

1. Introduction

Elections are a series of procedures designed to form a government, recruit politicians through political parties so that people can choose leaders who have a vision and mission in accordance with the wishes of the community, and limit government actions or policies so that they are not arbitrary which can be done through the rotation of power (Izzaty & Nugraha, 2019). In the electoral process, the number of votes collected becomes the deciding factor in the election of a leader to be appointed as the head of a region or country. Therefore, the vote counting process becomes very important and sensitive. Thus, the General Election Commission (KPU) is the official institution responsible for counting and recapitulating votes (Fauzi et al., 2022). In the 2024 elections, the KPU will use a recapitulation system, namely SiRekap.

Based on the General Election Commission Decree Number 66 of 2024 concerning Technical Guidelines for the Implementation of Voting and Vote Counting in General Elections, the Electronic Recapitulation Information System, hereinafter referred to as Sirekap, is an information technology-based application device as a means of publishing the results of the vote count and the process of recapitulating the results of the vote count as well as a tool in the implementation of the recapitulation of the results of the election vote count (KPU, 2024).

Source: https://www.kpu.go.id

Figure 1. KPU Logo

The role of SiRekap is considered very important because it is able to increase the level of transparency and accountability in elections organized by the KPU. By using Sirekap, the vote counting process becomes more effective and accurate, reducing the potential for error and minimizing data manipulation. In addition, Sirekap allows information on election results to be disseminated quickly to the public, increasing confidence in the integrity of election results (Gauru et al., 2022).

SiRekap was released on Google Play Store on January 22, 2024. Google Play Store is an online platform for distributing digital applications, managed and created by Google. This app offers various features to help users, including a feedback feature. Users can use this feedback feature to evaluate apps by giving star ratings and writing reviews. To get a clearer feedback, it is recommended to refer to users' textual reviews. Textual reviews have the potential to influence new users to download the app (Azizah et al., 2023).

Seeing the large number of reviews given by users through the Google Play Store for Sirekap, a system is needed to be able to assess the sentiment of the users' reviews. This research uses the Logistic Regression algorithm to perform sentiment analysis of reviews sent by users of the SiRekap application on the Google Play Store.

Sentiment analysis is a text analysis process to determine sentiment and categorize it as positive, negative, or neutral. Sentiment analysis involves text mining to evaluate, process, and extract textual data by involving data preprocessing which includes tokenization, removing stopwords, stemming, and classifying sentiment (Samsir et al., 2021).

Text mining is a computer science methodology that addresses various information challenges by integrating methods from data mining, machine learning, and natural language processing. Text mining aims to derive important information from data by identifying and examining significant patterns (Nofenky & Rarasati, 2022).

The results of this sentiment analysis can be used by the KPU to understand the level of user satisfaction and improve the quality of the SiRekap application for the 2024 Pilkada, in accordance with Article 58A of the Regulation of the General Election Commission of the Republic of Indonesia Number 18 of 2020 concerning Amendments to the General Election Commission Regulation Number 8 of 2018 concerning Voting and Vote Counting for the Election of Governors and Deputy Governors, Regents and Deputy Regents, and / or Mayors and Deputy Mayors in article 58A that the KPU uses a recapitulation information system tool in the form of SiRekap for means of publication of Vote Counting results at polling stations and recapitulation of Vote Counting results (KPU, 2020).

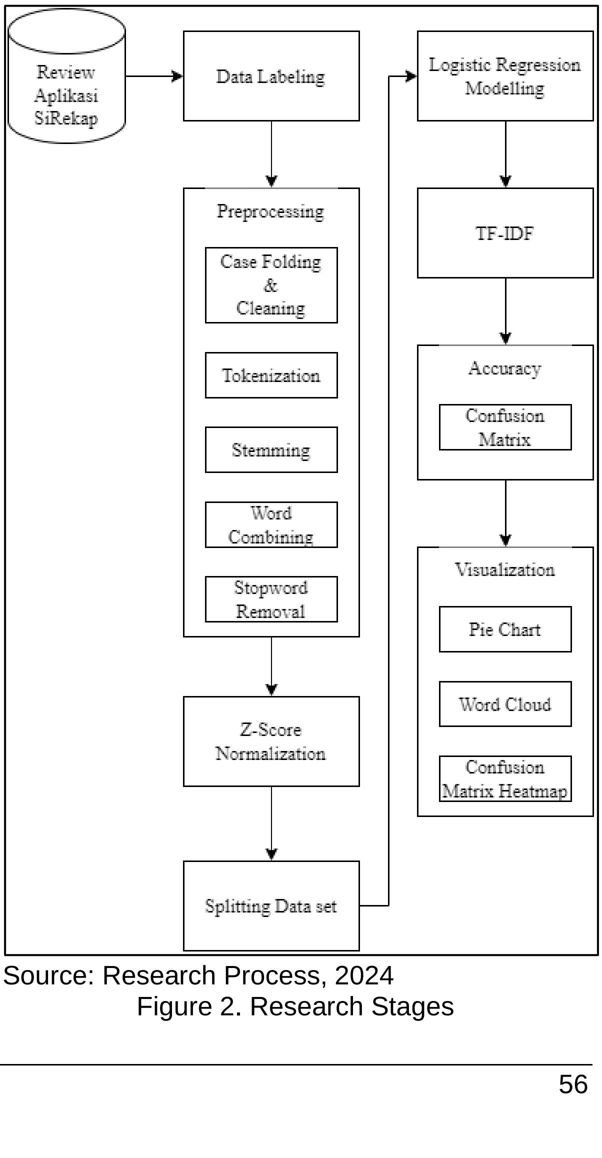
2. Research Methods

This research analyzes user sentiment towards the Sirekap application using the Logistic Regression algorithm. This research method will cover all steps from data collection (data crawling) at the beginning to visualization of sentiment analysis results at the end.

The following are the steps taken in the research process to analyze the sentiment of community reviews of the Sirekap application, as illustrated in Figure 2.

2.1. Data Crawling

In the data crawling stage, this research uses user reviews of the Sirekap application available on the Google Play Store as a data source. To access the data, the method used is to utilize the Google Play Scraper library. The data retrieved covers the time period from January 22, 2024 to February 13, 2024. After data collection is complete, the results are saved in csv format to facilitate sentiment analysis at a later stage.



Source: Research Process, 2024
Figure 2. Research Stages

2.2. Data Labelling

Data labeling is the process of assigning a specific label to each data in a dataset. The goal is to recognize or measure certain characteristics or properties that we want to study. For example, in the data labeling process to determine whether a text is considered positive, negative, or neutral (Kholilullah et al., 2024).

2.3. Text Preprocessing

Text preprocessing is a process carried out to organize text data to be more structured and reduce the noise contained in the text data (Shiddicky & Agustian, 2022). This procedure has several stages, namely:

- Text Case Folding dan Text Cleaning
  At this stage, all letters in the sentence will be converted to lowercase to maintain uniformity of writing. Furthermore, all characters other than letters will be removed, such as punctuation marks and numbers (Yutika et al., 2021).
- Text Tokenization
  At this stage, whole sentences that have passed the case folding and cleaning stages will be broken down into words (Rarasati, 2020).
- Text Stemming
  At this stage, all affixes, prefixes, greetings, suffixes, and combinations in the data will be cleaned. Thus, each word in the data only consists of basic vocabulary (Yutika et al., 2021).
- Word Combining
  Word combining (synonymizing) is a way of speaking that is equivalent to other forms that have similar meanings. This process occurs when words that are different but have the same meaning are combined to reduce the number of words in the system without reducing the frequency of the words (Rarasati, 2020).
- Text Stopword Removal
  At this stage, words that are deemed unimportant are removed, such as 'and', 'or', 'this', etc (Yutika et al., 2021).

2.4. Z-Score Normalization

Z-score normalization is one of the statistical techniques often used in big data analysis. Also known as standardized scores, z-scores are used to identify data that is significantly different from the rest of the data, often referred to as outliers. In z-scores, the data is transformed by maintaining the distribution relative to the mean value and standard deviation of the data set. Thus, the resulting value of the z-score reflects the difference between a particular value and the mean value as well as the variation in the data set. The z-score normalization method has the advantage of handling outliers or values that are far from extreme values in the data (Whendasmoro & Joseph, 2022). The z-scores can be obtained from the equation:

$$Z = \frac{x - \mu}{\sigma}$$

Based on the equation above, x represents the raw score, μ represents the mean, and σ represents the standard deviation.

2.5. Splitting Data set

In this step, the dataset will be separated into two parts: the training dataset (train data) and the testing dataset (test data). The training dataset will be used to train the Logistic Regression classification model, while the test dataset will be used to test the model.

2.6. Term Frequency–Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a technique used in text analysis and information mining to assess the significance of words in a set of documents. This method is useful for identifying key words in documents, measuring inter-document similarity, ranking search results, and so on (Kim & Gil, 2019).

Term Frequency (TF) is the number of occurrences of specific words in a document. Words with high TF values are considered very important in the document (Kim & Gil, 2019). The TF value can be obtained from the equation:

$$\mathrm{TF} = \frac{\text{number of times the word appears in the document}}{\text{total number of words in the document}}$$

Document Frequency (DF) indicates the frequency of occurrence of a particular word in the entire set of documents. DF will determine the frequency of a word in many documents, not just in one document. Words with a high DF value are considered unimportant because they often appear in most documents (Kim & Gil, 2019). The DF value can be obtained from the equation:

$$\mathrm{DF} = \frac{\text{the number of documents the word appears}}{\text{total number of documents}}$$

Inverse Document Frequency (IDF) is used to assess the importance of a word across documents. A high IDF value indicates a word that appears infrequently in the document, thus increasing the importance of the word (Kim & Gil, 2019). The IDF value can be obtained from the following equation:

$$\mathrm{IDF} = \log\left(\frac{\text{total number of documents}}{\text{the number of documents the word appears}}\right)$$

The TF-IDF value can be obtained from the equation:

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{TF} \times \mathrm{IDF}$$

The TF-IDF value increases when a keyword has a high frequency in a document and the frequency of documents containing that keyword among all documents is low. This principle can be used to find keywords that appear frequently in documents (Kim & Gil, 2019).

2.7. Logistic Regression

Logistic regression is a supervised machine learning method used to analyze data and describe the correlation between one or more variables used to predict and one response variable. In logistic regression, the response variable only has a value between 0 and 1, allowing the classification of sentiment as positive or negative with a decision boundary at 0.5 (Assaidi & Amin, 2022).
The initial equation of logistic regression can be seen in the equation:

$$z = b + (w_1 \times x_1) + \cdots (w_n \times x_n)$$

In the above equation, b represents the bias value s, w is a representation of the data in vector form, and x is a representation of the weight of the data. The function will calculate the weight for each feature in the x vector by multiplying it by the weight (Assaidi & Amin, 2022).
After getting the value of z and to be able to get a probability number that has a value between 0 and 1, the z value will be entered into the sigmoid function which has an equation:

$$\mathrm{Sigmoid}\ (z) = \frac{1}{1 + e^{-z}}$$

After the above equation is used, the results will then be compared to determine the category or classification with the following criteria:
1. If the result > 0.5 then the result of the prediction is positive.
2. If the result < 0.5 then the result of the prediction is negative.

2.8. Evaluating

In this step, the confusion matrix is used to evaluate the performance of the classification model. Confusion matrix is a table that shows the number of correctly and incorrectly classified test data (Normawati & Prayogi, 2021). An example of a confusion matrix for two classifications, positive and negative, is shown in the figure below.

Valid Classifications	Classifications Action	
	Positif	Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

Source: (Normawati & Prayogi, 2021)

Figure 3. Confusion Matrix

Based on Figure 3 above, it is known:
1. True positive (TP) refers to the amount of data that is labeled positive by the model and is actually positive (Normawati & Prayogi, 2021).
2. False positive (FP) is the amount of data that is misclassified as positive by the model when it should be negative (Normawati & Prayogi, 2021).
3. True negative (FN) includes data that is correctly classified as negative by the model (Normawati & Prayogi, 2021).
4. False negative (FN) is the amount of data that is misclassified as negative by the model that should be positive (Normawati & Prayogi, 2021).

All of these data can be utilized to obtain other values that are useful for testing the performance of the model that has been created, namely:

1. Accuracy is the ratio between the number of correct predictions (both positive and negative) and the total amount of data available (Rahayu et al., 2021). The accuracy value can be obtained from the equation:

$$\mathrm{ACCURACY} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

2. Precision is the ratio of the number of correct predictions (positive or negative) to the total number of predictions (positive or negative) (Rahayu et al., 2021). The precision value can be obtained from the equation:

$$\mathrm{PRECISION} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

3. Recall is the ratio between the number of correct predictions (positive or negative) and the total amount of data that actually belongs to that category (positive or negative) (Rahayu et al., 2021). The recall value can be obtained from the equation:

$$\mathrm{RECALL} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

4. F1-score is the combined result of precision and recall, which shows the overall performance of a model (Assaidi & Amin,

2022). The f1-score value can be obtained from the equation:

$$F1 - SCORE = 2 \times \frac{RECALL \times PRECISION}{RECALL + PRECISION}$$

2.9. Visualization

This visualization stage will describe the visualization of the data set. The author will visualize the data set using word cloud and pie chart.

Word cloud is a text data visualization method that is widely used in text mining because of its clarity of concept. By using this technique, information about how often words appear can be presented in an attractive yet informative way. The size of the word in the word cloud is adjusted to how often the word is used, so the more often the word appears, the larger the size in the word cloud visualization. (Pradana, 2020).

3. Results and Discussion

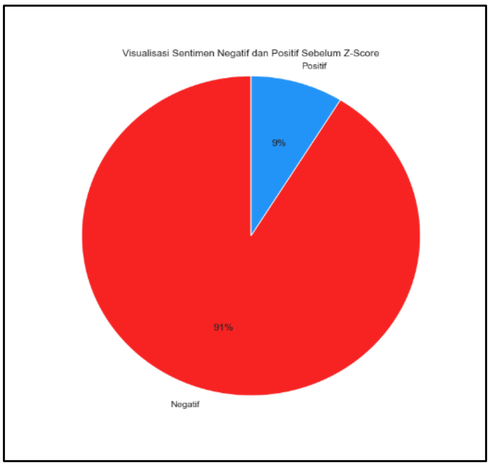
The third section covers the documentation process of each step in the research conducted. Python will be the main programming language used to conduct all stages of the research, from data collection to model evaluation.

3.1 Data Crawling

The process of collecting review data sets created by users using the Google Play Scraper library. The collected data set will be made into a csv file consisting of review_id, username, review, rating_review, and date_review columns. The data that was successfully collected amounted to 15,928 review data.

3.2 Data Labelling

After the process of collecting the review data set is complete, then the review data set will be labeled manually. Reviews with positive sentiment will be labeled 1 and reviews with negative sentiment will be labeled 0. Table 1 shows the stages of manual data labeling. The following is the percentage of the label composition of the review data:



Sumber: Research Process, 2024

Figure 4. Label Composition Data Review

3.3 Text Preprocessing

After the review data labeling process is complete, the next step is to perform text preprocessing on the user review data set. The stages of the text preprocessing process carried out, namely:

1. Text Case Folding dan Text Cleaning
2. Text Tokenization
3. Text Stemming
4. Word Combining
5. Text Stopword Removal

Table 2 shows the preprocessing steps that have undergone some changes in accordance with the functions contained in Python.

Table 1. Data Labeling

Review	Label
Aplikasi payah gada guna banyak kurangnya daripada manfaatnya	Negatif
Lebih ditingkatkan lagi, kalau nanti masih dipake di pilkada. Servernya ditambah biar gk lemot wkwkw	Positif
Aplikasi yg sangat berguna untuk mempersulit dan memperlambat pekerjaan	Negatif
Tolong tambahkan sistem upload file,jadi lebih mudah,jangan hanya sistem scaning aja	Positif
Aplikasinya selalu down dan log out sendiri,ini memperlama kerja kpps	Negatif

Source: Research Process

ISSN (Print) 2355-6579 | ISSN (Online) 2528-2247

Table 2. Data Preprocessing

Text Case Folding & Cleaning	Text Tokenization	Text Stemming	Text Combination	Text Stopword Removal
aplikasi payah gada guna banyak kurangnya daripada manfaatnya	['aplikasi', 'payah', 'gada', 'guna', 'banyak', 'kurangnya', 'daripada', 'manfaatnya']	aplikasi payah gada guna banyak kurang daripada manfaat	aplikasi payah tidak ada guna banyak kurang daripada manfaat	payah tidak manfaat
lebih ditingkatkan lagi kalau nanti masih dipake di pilkada servernya ditambah biar gk lemot wkwkw	['lebih', 'ditingkatkan', 'lagi', 'kalau', 'nanti', 'masih', 'dipake', 'di', 'pilkada', 'servernya', 'ditambah', 'biar', 'gk', 'lemot', 'wkwkw']	lebih tingkat lagi kalau nanti masih dipake di pilkada servernya tambah biar gk lemot wkwkw	lebih baik lagi kalau nanti masih guna di pilih kepala daerah server tambah biar tidak lambat tawa	baik pilih kepala daerah server biar tidak lambat tawa
aplikasi yg sangat berguna untuk mempersulit dan memperlambat pekerjaan	['aplikasi', 'yg', 'sangat', 'berguna', 'untuk', 'mempersulit', 'dan', 'memperlambat', 'pekerjaan']	aplikasi yg sangat guna untuk sulit dan lambat kerja	aplikasi yang sangat guna untuk sulit dan lambat kerja	sangat sulit lambat kerja
tolong tambahkan sistem upload file jadi lebih mudah jangan hanya sistem scaning aja	['tolong', 'tambahkan', 'sistem', 'upload', 'file', 'jadi', 'lebih', 'mudah', 'jangan', 'hanya', 'sistem', 'scaning', 'aja']	tolong tambah sistem upload file jadi lebih mudah jangan hanya sistem scaning aja	tolong tambah sistem unggah berkas jadi lebih mudah jangan hanya sistem pindai saja	sistem unggah berkas mudah sistem pindai
aplikasinya selalu down dan log out sendiri ini memperlama kerja kpps	['aplikasinya', 'selalu', 'down', 'dan', 'log', 'out', 'sendiri', 'ini', 'memperlama', 'kerja', 'kpps']	aplikasi selalu down dan log out sendiri ini lama kerja kpps	aplikasi selalu tidak sedia dan catat keluar sendiri ini lama kerja kelompok selenggara pungut suara	tidak sedia catat kerja kelompok selenggara pungut suara

Source: Research Process

3.4 Z-Score Normalization

Data sets that have passed the preprocessing stage will continue with the z-score normalization process. The z-score normalization process aims to balance the number of samples between negative and positive sentiments in the sentiment analysis dataset by adding additional samples from underrepresented classes.

This process involves calculating the total sentiment, calculating the z-score values for negative and positive sentiments, setting a threshold limit for the z-score, determining the number of additional samples required if the z-score is less than the threshold limit, randomly selecting additional samples, and merging the additional data set with the original data set to create a new data set that is balanced in terms of the number of negative and positive sentiments. This process is done using the zscore() function available from the scipy library.

Based on table 3, the researcher decided to use 0.005 as the threshold value. This determination is based on the observation that using this threshold value is able to create a balance between positive and negative data. The result of this process will change the amount of data from 15,928 data to 21,060 data. The following is the percentage of label composition of the review data after the normalization process with z-scores:

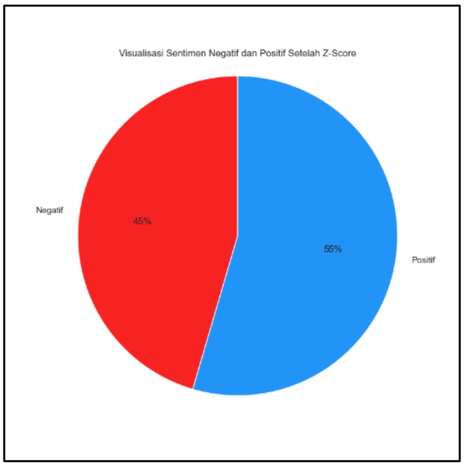
Table 3. Threshold Value Experiment

Threshold Value	Positive Data	Negative Data
0.5	63.537%	36.463%
0.05	55.533%	44.467%
0.005	54.533%	45.467%

Source: Research Process

3.5 Splitting Data Set

This splitting dataset process aims to separate the dataset used in the research into two parts, namely the training data set (train) and the test data set (test). The training data set will be used to train the Logistic Regression classification model, while the test data set will be used to test the performance of the Logistic Regression classification model. This process is done using the train_test_split() function available from the scikit-learn library.



Source: Research Process, 2024

Figure 5. Label Composition of Review Data After Z-Score Normalization

Table 4. Data Set Splitting Experiment

Train Data	Test Data	Accuracy
60%	40%	91.377%
70%	30%	91.929%
80%	20%	92.257%

Source: Research Process

Based on Table 4, researchers decided to divide the research data into two parts, namely 80% training data and 20% test data. This decision was made because researchers observed that the amount of test data used affects the accuracy rate obtained. The more test data used, the accuracy rate tends to decrease. This happens because of the differences between test data and training data that can affect the performance of the system. By using more test data that is similar to the training data, the accuracy rate is likely to increase (Wijaya & Hakim, 2023).

3.6 TF-IDF

In the TF-IDF weighting stage, after the data set has passed the normalization process with z-scores and splitting the data set, the next step is to calculate the weight of the words in each review using the TF-IDF method.

The TF-IDF weighting process is done to convert the review data into a numerical representation that can be processed by the system. This process is implemented through the use of the TfidfVectorizer() class found in the sklearn library.

The weighting process on the training dataset is implemented through the use of the fit_tranform() function, while the test dataset only applies the transform() function. The fit_tranform() function summarizes the functionality of fit() and transform(), where fit() serves to obtain parameters such as vocabulary and idf from the training dataset, while transform() plays a role in transforming documents into matrix form.

Table 5. TF-IDF

Term	TF-IDF
hormat	0.583961
tahap	0.479130
pilih	0.420262
komisi	0.305553
eror	0.284842

Source: Research Process

3.7 Logistic Regression Modelling

After weighting the training dataset and the test dataset, the next step is to train the Logistic Regression classification model. This process is implemented through the use of the LogisticRegression() class found in the sklearn library.

The model training process will use the fit() function to train the model with 2 input data, namely the feature matrix that has been converted into a vector with the tf-idf representation of the training data, as well as the label corresponding to the training data. After that, to predict the test data, we will use the predict() function with the input data of the feature matrix that has been converted into a vector with TF-IDF representation from the test data.

3.8 Visualisasi

Word cloud visualization is used to identify the words that appear most in a set of documents. Review data sets that have passed the z-score normalization stage will be separated based on labels, namely positive and negative, then the frequency of word occurrence in each sentiment will be calculated. The following is an example of word cloud visualization:

Source: Research Process, 2024

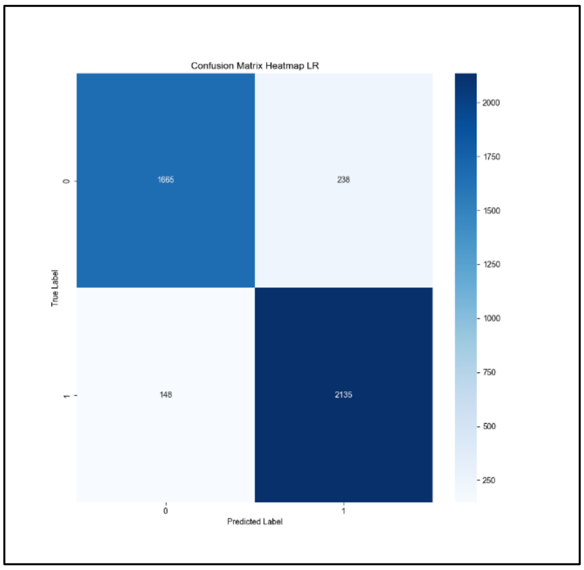
Figure 6. Positive Word Cloud

Source: Research Process, 2024

Figure 7. Negative Word Cloud

3.9 Evaluasi Model

Model evaluation is used to assess the effectiveness of the system that has been created. This test includes analyzing the performance of the model in classifying reviews on the test data set. This process is implemented through the use of the confusion_matrix() function found in the sklearn library. The model performance analysis will use confusion matrix to evaluate the accuracy, precision, recall, and f1-score of each class (positive and negative).



Source: Research Process, 2024

Figure 8. Confusion Matrix Heatmap

Valid Classifications	Classifications Action	
	Positif	Negatif
Positif	2135	148
Negatif	238	1665

Source: Research Process, 2024

Figure 9. Confusion Matrix

Based on the results of the confusion matrix above, it can be seen that true positives totaled 2135, true negatives totaled 1665, false positives totaled 238, and false negatives totaled 148. So that the 4 values that have been obtained can be used to calculate several metrics, namely:

1. Accuracy

To calculate the accuracy of the model, we use the formula:

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

$$\text{Accuracy} = \frac{2135 + 1665}{2135 + 1665 + 238 + 148} = \frac{1900}{2093}$$

$$\text{Accuracy} = 0.908 \times 100 = 90.8\% = 91\%$$

Thus, the accuracy of the model is 91%.

2. Precision

To calculate the positive precision of the model, we use the formula:

$$\text{Precision}_{\text{pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{2135}{2135 + 238} = \frac{305}{339}$$

$$\text{Precision}_{\text{pos}} = 0.900 \times 100 = 90.0\% = 90\%$$

Thus, the positive precision of the model is 90%.

Meanwhile, to calculate the negative precision of the model, we use the formula:

$$\text{Precision}_{\text{neg}} = \frac{\text{TN}}{\text{TN} + \text{FN}} = \frac{1665}{1665 + 148} = \frac{45}{49}$$

$$\text{Precision}_{\text{neg}} = 0.918 \times 100 = 91.8\% = 92\%$$

Thus, the negative precision of the model is 92%.

3. Recall

To calculate the positive recall of the model, we use the formula:

$$\text{Recall}_{\text{pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{2135}{2135 + 148} = \frac{2135}{2283}$$

$$\text{Recall}_{\text{pos}} = 0.935 \times 100 = 93.5\% = 94\%$$

Thus, the positive recall of the model is 94%.

Meanwhile, to calculate the negative recall of the model, we use the formula:

$$\text{Recall}_{\text{neg}} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{1665}{1665 + 238} = \frac{1665}{1903}$$

$$\text{Recall}_{\text{neg}} = 0.875 \times 100 = 87.5\% = 87\%$$

Therefore, the negative recall of the model is 87%.

4. F1-Score

To calculate the positive f1-score of the model, we use the formula:

$$\mathrm{F1Score_{pos}} = 2 \times \frac{\mathrm{Recall_{pos}} \times \mathrm{Precision_{pos}}}{\mathrm{Recall_{pos}} + \mathrm{Precision_{pos}}}$$

$$\mathrm{F1Score_{pos}} = 2 \times \frac{0.935 \times 0.900}{0.935 + 0.900} = \frac{1683}{1835}$$

$$\mathrm{F1Score_{pos}} = 0.917 \times 100 = 91.7\% = 92\%$$

Therefore, the positive f1-score of the model is 92%.

Meanwhile, to calculate the negative f1-score of the model, we use the formula:

$$\mathrm{F1Score_{neg}} = 2 \times \frac{\mathrm{Recall_{neg}} \times \mathrm{Precision_{neg}}}{\mathrm{Recall_{neg}} + \mathrm{Precision_{neg}}}$$

$$\mathrm{F1Score_{neg}} = 2 \times \frac{0.875 \times 0.918}{0.875 + 0.918} = \frac{3213}{3686}$$

$$\mathrm{F1Score_{neg}} = 0.896 \times 100 = 89.6\% = 90\%$$

Therefore, the negative f1-score of the model is 90%.

4. Conclusion

Based on sentiment analysis of the SiRekap application review on the Google Play Store, it shows that the Logistic Regression algorithm is very effective in classifying review sentiment. This process involves several important steps, from data preprocessing to model testing, which results in several high evaluation metrics.

Specifically, the system's accuracy rate reached 91%. This means that out of the entire review data analyzed, 91% of the sentiment predictions (both positive and negative) were correct. This accuracy rate indicates that the model is able to recognize patterns in the data well and provide results that are consistent with the actual sentiments expressed by users.

Furthermore, the precision for the positive class is 90%, which indicates that of all the reviews predicted to be positive, 90% are actually positive reviews. Conversely, the precision for the negative class is 92%, indicating that of all the reviews predicted to be negative, 92% are actually negative reviews. This figure illustrates the model's ability to minimize false positive and false negative prediction errors.

For recall, the value of 94% for the positive class indicates that out of all the reviews that were actually positive, 94% were correctly identified by the model. In contrast, the recall for the negative class is 87%, which means that out of all the reviews that were actually negative, 87% were correctly identified. The high recall value for the positive class indicates that the model is very effective in detecting positive reviews, although there is still room for improvement in detecting negative reviews.

F1-Score, which is a combination of precision and recall, was 92% for the positive class and 90% for the negative class. F1-Score provides an overall picture of the model's performance, showing that the model is balanced in terms of precision and recall, and is able to handle variations in review data well.

For further research development, it is recommended to use a larger dataset to ensure the model can handle more variations in reviews. In addition, it is necessary to balance the amount of data in each class to improve model performance. Adding data management methods for other languages is also important, given the presence of English and Javanese reviews in the SiRekap review column. Adding words to the stopwords dictionary and word merge dictionary is also recommended to handle new word variations that are not yet covered by the system. With these steps, future research can improve and expand the methodology and scope of sentiment analysis on the SiRekap application, thus providing more comprehensive and accurate results.

Reference

Assaidi, S. A., & Amin, F. (2022). Analisis Sentimen Evaluasi Pembelajaran Tatap Muka 100 Persen pada Pengguna Twitter menggunakan Metode Logistic Regression. *Jurnal Pendidikan Tambusai*, *6*(2), 13217–13227.

Azizah, I., Cholissodin, I., & Yudistira, N. (2023). Analisis Sentimen Ulasan Pengguna Aplikasi Shopee di Google Play menggunakan Metode Word Embedding dan Long Short Term Memory (LSTM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *7*(5), 2453–2459. http://j-ptiik.ub.ac.id

Fauzi, A., Maulana, A., & Setiawan, A. (2022). Perancangan Aplikasi Rekapitulasi Hasil Pemilu Sementara Berbasis Android Mobile. *CONTEN : Computer and Network Technology*, *2*(1), 17–26. https://doi.org/10.31294/conten.v2i1.1179

Gauru, C., Martini, R., & Alfirdaus, L. K. (2022). Implementasi Sirekap Dalam Pilkada 2020 Kabupaten Semarang. *Reformasi*, *12*(2), 224–230. https://doi.org/10.33366/rfr.v12i2.3874

Izzaty, R., & Nugraha, X. (2019). Perwujudan Pemilu yang Luberjurdil melalui Validitas Daftar Pemilih Tetap. *Jurnal Suara Hukum*, *1*(2), 155.

https://doi.org/10.26740/jsh.v1n2.p155-171

Kholilullah, M., Martanto, & Hayati, U. (2024). Analisis Sentimen Pengguna Twitter(X) Tentang Piala Dunia Usia 17 Menggunakan Metode Naive Bayes. *Jurnal Mahasiswa Teknik Informatika*, *8*(1), 392–398. https://apify.com/quacker/twitter-scraper

Kim, S.-W., & Gil, J.-M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. *Human-Centric Computing and Information Sciences*, *9*(1). https://doi.org/10.1186/s13673-019-0192-7

KPU. (2020). Peraturan Komisi Pemilihan Umum Nomor 18 Tahun 2020 Tentang Perubahan Atas Peraturan Komisi Pemilihan Umum Nomor 8 Tahun 2018 Tentang Pemungutan Dan Penghitungan Suara Pemilihan Gubernur Dan Wakil Gubernur, Bupati Dan Wakil Bupati, Dan/Atau Wali Kota Dan . *Komisi Pemilihan Umum*.

KPU. (2024). Keputusan Komisi Pemilihan Umum Nomor 66 Tahun 2024 Tentang Pedoman Teknis Pelaksanaan Pemungutan dan Perhitungan Suara dalam Pemilihan Umum. *Komisi Pemilihan Umum*.

Nofenky, & Rarasati, D. B. (2022). Recommendation for Classification of News Categories Using Support Vector Machine Algorithm with SVD. *Ultimatics : Jurnal Teknik Informatika*, *13*(2), 72–80. https://doi.org/10.31937/ti.v13i2.1854

Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(2), 697–711.

Pradana, M. G. (2020). Penggunaan Fitur Wordcloud dan Document Term Matrix dalam Text Mining. *Jurnal Ilmiah Informatika*, *8*(1), 38–43.

Rahayu, W. I., Prianto, C., & Novia, E. A. (2021). Perbandingan Algoritma K-Means Dan Naïve Bayes Untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan Pada Pt. Pertamina (Persero). *Jurnal Teknik Informatika*, *13*(2), 1–8. https://ejurnal.poltekpos.ac.id/index.php/informatika/article/view/1383

Rarasati, D. B. (2020). A Grouping of Song-Lyric Themes Using K-Means Clustering. *JISA(Jurnal Informatika Dan Sains)*, *3*(2), 38–41. https://doi.org/10.31326/jisa.v3i2.658

Samsir, Ambiyar, Verawardina, U., Edi, F., & Watrianthos, R. (2021). Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes. *Jurnal Media Informatika Budidarma*, *5*(1), 149. https://doi.org/10.30865/mib.v5i1.2604

Shiddicky, A., & Agustian, S. (2022). Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression. *Jurnal CoSciTech (Computer Science and Information Technology)*, *3*(2), 99–106. https://doi.org/10.37859/coscitech.v3i2.3836

Whendasmoro, R. G., & Joseph. (2022). Analisis Penerapan Normalisasi Data Dengan Menggunakan Z-Score Pada Kinerja Algoritma K-NN. *JURIKOM (Jurnal Riset Komputer)*, *9*(4), 872. https://doi.org/10.30865/jurikom.v9i4.4526

Wijaya, H., & Hakim, L. (2023). Analisis Sentimen Kebijakan Jaminan Hari Tua (Jht) Pada Twitter Menggunakan Naïve Bayes. *Jurnal Algoritma, Logika Dan Komputasi*, *6*(1), 509–518. https://doi.org/10.30813/j-alu.v6i1.3552

Yutika, C. H., Adiwijaya, & Faraby, S. Al. (2021). Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes. *Jurnal Media Informatika Budidarma*, *5*(2), 422. https://doi.org/10.30865/mib.v5i2.2845