

Sentiment Analysis of the inDrive Application Reviews on Google Play Store Using the Naive Bayes Method

Ramadhan Pratama

Department of Computer Science and Information Technology, Universitas Muhammadiyah Sumatra Utara

The development of application-based technology on Android smartphones has transformed daily activities, including the transportation sector. One popular online transportation application is inDrive, offering features such as fare negotiation and driver selection criteria. With over 100 million downloads and 8 million reviews on the Google Play Store, sentiment analysis of user reviews is crucial for evaluating user satisfaction and providing insights for application developers. This study aims to analyze user sentiment on inDrive app reviews from the Google Play Store using the Naïve Bayes algorithm. The research adopts a quantitative approach with a dataset of 1,000 reviews collected through web scraping using Google Colab. The methodology includes sentiment labeling (positive and negative), data preprocessing (case folding, tokenizing, stopword removal, and stemming), weighting using Term Frequency-Inverse Document Frequency (TF-IDF), classification with the Naïve Bayes algorithm, model evaluation using a confusion matrix, and data visualization through word clouds. The results show that most user reviews are positive. The Naïve Bayes algorithm demonstrated good performance, achieving an accuracy of 80%, precision of 84%, and recall of 77%. Data visualization revealed frequently used words in positive reviews such as "excellent," "great," and "helpful," while negative reviews highlighted words like "please" and "driver." This study contributes to leveraging sentiment analysis to evaluate online transportation applications. Future research should explore alternative algorithms, such as K-Nearest Neighbor (KNN) or Support Vector Machine (SVM), and expand datasets with larger and more diverse data sources. These approaches are expected to enhance sentiment analysis model performance and provide deeper insights into user sentiment patterns.

Keywords: inDrive, Naive Bayes, Negative Reviews, Positive Reviews, Web Scrapping

INTRODUCTION

The development of application-based technology on Android smartphones continues to grow, transforming the way people perform daily activities, including in the transportation sector (Abrilia, 2020). Online transportation services, such as motorcycle taxis (*ojek* online), have become an innovative solution that facilitates mobility without requiring the use of personal vehicles (Edo & Fariadi, 2024).

One popular online transportation application is inDrive, established in 2012 in Yakutsk, Russia. With affordable pricing and unique features, such as driver selection criteria and trip distance options, inDrive has gained significant user interest in various major cities across Indonesia, including Medan, Jakarta, and Bandung.

User reviews on Google PlayStore play a crucial role in evaluating applications, including inDrive, which has garnered over 100 million downloads and 8 million reviews with a 4.5 rating. Sentiment analysis of these reviews is essential for providing

OPEN ACCESS
ISSN XXX-XXX (Online)

*Correspondence:
Ramdhan Pratama
ramadhanpratama185@gmail.com

Received: 24th December 2024
Accepted: 22nd January 2025
Published: 25th February 2025

Citation:
Pratama. R. (2025). *Sentiment Analysis of the inDrive Application Reviews on Google Play Store Using the Naive Bayes Method*. JACIIS (Journal of Advance in Computational Intelligence and Information Systems), 1(1), 10-23.
doi: xxxxxxxxxxxx

feedback to application developers in improving their features and services. In this study, the Naïve Bayes algorithm was chosen to classify positive and negative sentiments from the reviews, as this method is known for its high accuracy and low error rate, particularly when handling large datasets (Susana, 2022; Widiawati et al., 2023).

This study aims to classify sentiments in inDrive application reviews on Google Play Store using the Naïve Bayes method and Google Colab. Furthermore, it seeks to identify user satisfaction levels regarding inDrive's features and provide recommendations for developers to enhance the application's features in the future.

InDrive

InDrive was established in 2012 in Yakutsk, Russia, as a response to rising taxi fares during extreme weather conditions. Its initial concept involved direct negotiations between drivers and passengers through social media. Officially launched in 2013, the service has since expanded to various regions, including Latin America, Africa, Southwest Asia, and Indonesia in 2021, with its headquarters now located in Mountain View, California. InDrive offers a range of services such as intercity transportation, parcel delivery, courier services, and job search assistance. Its key advantages include a low application fee of 10% and flexible, negotiable rates. However, its drawbacks include the absence of physical offices, which complicates the process of handling complaints, and the infrequent application updates, which are often deemed impractical. Despite these limitations, inDrive remains one of the most popular transportation applications globally, ranking fourth in total downloads.

Sentiment Analysis

Sentiment analysis is a field of study aimed at analyzing opinions, sentiments, evaluations, attitudes, and emotions expressed in written text about entities and their attributes (Liu, 2020). This method is used to understand, extract, and process opinion data automatically to identify the sentiment contained within an opinion. Sentiments in this analysis can be classified as positive, negative, or neutral, enabling companies or organizations to comprehend public responses to a product or service through feedback (Evelina & Safitri, 2019; Hafidz & Liliana, 2021). Sentiment also depends on the specific focus of a topic, where statements with similar meanings might differ in context when applied to different subjects. Hence, in application review studies, sentiment analysis often begins with determining the specific elements of the application being discussed.

Naïve Bayes

Naïve Bayes is a machine learning algorithm classified under supervised learning, which requires labeled training data samples. Supervised learning is divided into classification and regression, with classification used for categorical variables, such as colors (red or yellow) or conditions (diseased or not) (Prabowo & Wiguna, 2021). This algorithm is part of the Bayes method used for text classification under the assumption that attribute values are conditionally independent if the output value is given. Furthermore, this algorithm is employed in text mining for sentiment analysis by predicting probabilities based on prior data but is limited to detecting text and numerical data, excluding images (Prabowo & Wiguna, 2021).

Bayes' theorem was discovered by Thomas Bayes, an English clergyman, in 1763 and later refined by Laplace. This theorem introduces pattern recognition through a statistical approach and can depict the probability of relationships between two events by considering additional data. In modern medicine, the theorem is often used to refine probability estimates using supplementary information (Retnosari, 2021).

Google Colab

Google Colab is a cloud-based service provided by Google that allows users to run and manage Jupyter notebooks for free. This service is utilized for writing and executing Python code without requiring additional software installations, as users only need a Google account to log in and save files to Google Drive. Google Colab offers a user-friendly notebook interface and was developed as part of Google Internal Research (Zain et al., 2023).

Python

Python is a high-level programming language that is interpretive, multipurpose, and adopts an Object-Oriented Programming approach with dynamic semantics to enhance syntax readability. Python features automatic memory management and is easy to learn, making it particularly accessible for

beginners. As an open-source programming language, Python is free to use and supports collaborative development. Its functions include website development, data analysis, scripting, and game development (Simanjuntak & Anita, 2022).

METHOD

Research Design

This study employs a quantitative approach using the Naïve Bayes algorithm to analyze user sentiment regarding the inDrive application based on reviews from the Google Play Store. The research process involves several stages, starting from data collection to visualization of the results.

Research Flow

The research flow is systematically structured as follows:

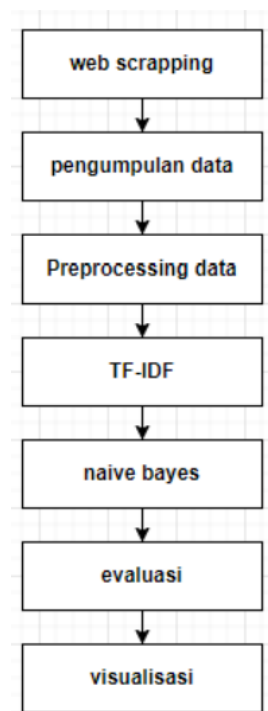


Figure 1. Research Flow

1. Web Scraping

Reviews of the inDrive application are extracted from the Google Play Store using the web scraping method implemented through the Google Colab platform. The scraped data is then exported to a Microsoft Excel format for analysis.

2. Data Collection

The dataset obtained from scraping consists of 1,000 user reviews. This data includes text reviews that will be further processed.

3. Data Labeling

The data labeling process categorizes the sentiment into positive or negative. The labeled data serves as a reference for training and validating the Naïve Bayes model.

4. Data Preprocessing

The reviews collected through Microsoft Excel undergo preprocessing with the following steps:

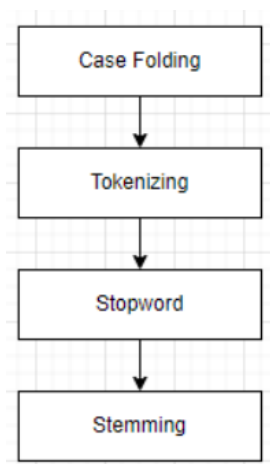


Figure 2. Data Preprocessing Steps

- a. Case Folding: Converts all text to lowercase.
- b. Tokenizing: Splits sentences into individual words.
- c. Stopword Removal: Removes insignificant words, such as conjunctions.
- d. Stemming: Converts affixed words to their root form.

5. TF-IDF Weighting

After preprocessing, weighting is conducted using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This process assigns a weight to each word in the dataset to determine its importance in sentiment analysis.

6. Classification with Naïve Bayes

The Naïve Bayes algorithm is used to classify sentiment based on word probabilities in the dataset. The classification produces two sentiment categories: positive and negative.

7. Model Evaluation

The model's performance is evaluated using a confusion matrix, yielding accuracy, precision, and recall values with the following formulas:

$$\text{Accuracy} = \frac{TP + TN}{TOTAL DATA}$$

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

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

8. Data Visualization

The sentiment analysis results are visualized using a word cloud to illustrate the distribution of words in positive and negative sentiments. This visualization facilitates data interpretation.

Time and Place of Research

Research Time

This research is designed to be carried out from February to October, covering phases such as preparation, implementation, data collection, and the processing and reporting of research results. In the preparation phase, a series of activities are conducted, including submitting the title, proposing the

advisor's Decree (SK), initial observation, drafting the research proposal, and holding a proposal seminar planned to take place from February to March. The implementation and data collection phase includes the system development process and data collection, conducted from April to May. Subsequently, in the data processing and reporting phase, the research results are validated, and the final report is compiled, taking place from June to July.

Research Location

This research is conducted online, utilizing Google Play Store as the primary data source, processed using Google Colab for data scraping activities.

Research Equipment

The research equipment includes hardware, specifically a laptop with an AMD Ryzen 5 processor, 8GB RAM, and an HDD, as well as supporting software, including the Windows 11 operating system and the Google Colab application.

Classification System Flowchart

To facilitate visualization of the research process, a flowchart is created to represent the stages of the research systematically and structurally.

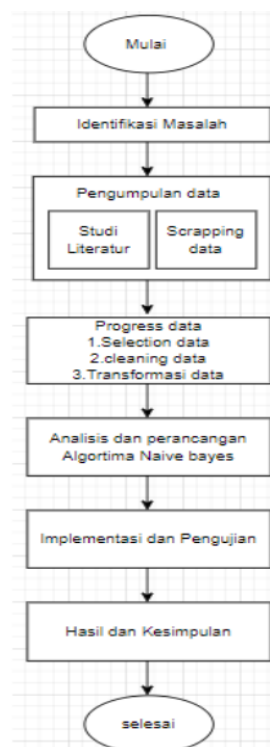


Figure 3. Classification System Flowchart

RESULTS AND DISCUSSION

1. Web Scraping Results

Data collection was conducted using Google Colaboratory by installing the google-play-scraper library. The data collected from Google Play Store reviews includes review ID, username, review, rating, and review date.

```

+ Kode + Teks
✓ 04
from google_play_scraper import Sort, reviews
import pandas as pd
from tqdm import tqdm

# Fungsi untuk mengambil ulasan dari Google Play Store
def get_reviews(app_id, lang='id', country='id', max_reviews=1000):
    all_reviews = []
    result = reviews(
        app_id=app_id,
        lang=lang,
        country=country,
        sort=Sort.NEWEST,
        count=max_reviews,
        filter_score_with=None
    )
    for review in result:
        all_reviews.append({
            'review_id': review['reviewId'],
            'user_name': review['userName'],
            'rating': review['score'],
            'review': review['content'],
            'at': review['at']
        })
    return pd.DataFrame(all_reviews)

# ID aplikasi InDrive di Google Play Store
app_id = 'sinet.startup.inDriver'

# Mengambil ulasan
data = get_reviews(app_id, max_reviews=1000)
data.head()
    
```

Figure 4. Web Scraping in Python

#	A	B	C	D	E
1	review_id	user_name	rating	review	at
2	0bb4caa5	Ulfah Hmytl	5	Min mau tanya apakah keluar	24/09/2024 19:28
3	471b3fe3	dimas arief	5	Ok	24/09/2024 18:59
4	38d7b786	Krismanto Aris	5	aplikasi yg sangat membantu	24/09/2024 17:19
5	3e810d07	Zildjian Idris	5	Mantap	24/09/2024 17:14
6	a9eaf254	Seftian	2	Uang saya nyangkut diaplikas	24/09/2024 17:05
7	20248fd2	Op. Choki Siallagan	2	Suka hoaks...tidak bisa konek	24/09/2024 17:05
8	1876fe86	Rolly Pupun	5	Klu bisa aplikasi nya di perbai	24/09/2024 16:55
9	bc8ab174	Arif Gunawan	1	makin sini makin susah dapet	24/09/2024 16:03
10	926061b7	Fadly Syahputra.	1	hilangkan nego argo dong ma	24/09/2024 15:57
11	1510ba4c	Ali Rahman	5	Istimewa	24/09/2024 15:57
12	37fb9ea1	Adit ae	1	Teman saya habis kenak romp	24/09/2024 15:23
13	5bab8452	Miftachul Mufid	1	Butuh waktu berapa lama bag	24/09/2024 15:18
14	8538f236	Dodol Garut	5	Baik.. Bagus..Biy'ceBiy'	24/09/2024 15:01
15	805c0a04	Yuanta	1	Pengemudi nya sangat mengu	24/09/2024 14:38
16	28e0ce44	Hendarto Sachro	5	good service and excellent	24/09/2024 14:21
17	d3e49ae4	Eka adila Adila	1	Autobitnya kurang support,m	24/09/2024 14:19
18	0576c8e4	á'á'ceá'á'ce á'á'É'á'á'	3	Sungguu saya sangat kesal &	24/09/2024 14:10
19	afc27dd3	joko Sugiarto	1	Aplikasi parah buruk...autobii	24/09/2024 14:04
20	364d10f7	Ikhsan Jhon	5	aku mendukung semua apk p	24/09/2024 13:50

Figure 5. Web Scraping Results

2. Data Labeling Results

The data labeling result is in the form of a CSV file containing the categorization of reviews based on sentiment (positive or negative). This table serves as the main input for further analysis.

#	A	B	C	D	E	F	G	H	I
1	review_id	user_nam	rating	review	at	clean_rev	sentimen	score	sentimen
2	0bb4caa5	Ulfah Hmy	5	Min mau t	24/09/2024 19:28	min mau t	1 star	0.4082152	Negatif
3	471b3fe3	dimas arie	5	Ok	24/09/2024 18:59	ok	3 stars	0.4951144	Positif
4	38d7b786	Krismantc	5	aplikasi yg	24/09/2024 17:19	aplikasi yg	5 stars	0.6137180	Positif
5	3e810d07	Zildjian id	5	Mantap	24/09/2024 17:14	mantap	3 stars	0.2485850	Positif
6	a9eaf254	Seftian	2	Uang saya	24/09/2024 17:05	uang saya	5 stars	0.2480363	Positif
7	20248fd2	Op. Choki	2	Suka hoak	24/09/2024 17:05	suka hoak	4 stars	0.4092351	Positif
8	1876fe86	Rolly Pup	5	Klu bisa aj	24/09/2024 16:55	klu bisa aj	3 stars	0.4416875	Positif
9	bc8ab174	Arif Gunan	1	makin sini	24/09/2024 16:03	makin sini	2 stars	0.3235483	Negatif
10	926061b7	Fadly Sya	1	hilangkan	24/09/2024 15:57	hilangkan	1 star	0.6488752	Negatif
11	1510ba4c	Ali Rahma	5	Istimewa	24/09/2024 15:57	istimewa	5 stars	0.4993018	Positif
12	37fb9ea1	Adit ae	1	Teman san	24/09/2024 15:23	teman say	1 star	0.7554755	Negatif
13	5bab8452	Miftachul	1	Butuh wal	24/09/2024 15:18	butuh wal	2 stars	0.3931011	Negatif
14	8538f236	Dodol Gar	5	Baik.. Bag	24/09/2024 15:01	baik bagu	5 stars	0.4305090	Positif
15	805c0a04	Yuanta	1	Pengemu	24/09/2024 14:38	pengemu	1 star	0.4448235	Negatif
16	28e0ce44	Hendarto	5	good serv	24/09/2024 14:21	good serv	5 stars	0.7624075	Positif
17	d3e49ae4	Eka adila	1	Autobitny	24/09/2024 14:19	autobitny	2 stars	0.4807372	Negatif
18	0576c8e4	á'á'ceá'á'	3	Sungguu s	24/09/2024 14:10	sungguu s	2 stars	0.5156812	Negatif
19	afc27dd3	joko Sugi	1	Aplikasi p	24/09/2024 14:04	aplikasi pi	1 star	0.5804567	Negatif
20	364d10f7	Ikhsan Jhc	5	aku mend	24/09/2024 13:50	aku mend	5 stars	0.5652607	Positif

Figure 6. Data Labeling Results

3. Data Preprocessing Results

Preprocessing is carried out to standardize the format and structure of the data for further processing. The preprocessing steps include:

a. Case Folding

It is to convert all capital letters to lowercase. For example, in Table 4.3, line 7 shows the case folding result where all capital letters are converted to lowercase.

4	38d7b786-Krismanto Aris	5 aplikasi yg sangat membantu
5	3e810d07-Zildjian Idris	5 Mantap
6	a9edf254-Septian	2.Uang saya nyangkut diaplikasi orderan pertama,
7	20248fd2-Op.Choki Siallagan	2Suka hoaks...tidak bisa koneksi
8	1876fe86-Rolly Pupun	5 Klu bisa aplikasi nya di perbaiki lagi.karena titik

Figure 7. Case Folding Data

```

import re

# Fungsi case folding untuk teks
def casefolding(text):
    if isinstance(text, str): # Pastikan 'text' adalah string
        text = text.lower() # Mengubah teks menjadi huruf kecil
        text = re.sub(r'https?://[a-z0-9.]+', '', text) # Menghilangkan URL
        text = re.sub(r'[0-9]+', '', text) # Menghilangkan angka
        text = re.sub(r'[^\w\s]', '', text) # Menghilangkan tanda baca
        return text
    else:
        return text # Jika bukan teks, kembalikan apa adanya

# Ambil sampel dari kolom teks, misalnya 'content' atau 'label'
raw_sample = data['review'].iloc[5] # Asumsi 'content' adalah kolom yang berisi teks tweet
case_folding = casefolding(raw_sample)

print('Raw data: ', raw_sample)
print('Case folding: ', case_folding)

```

Raw data : Suka hoaks...tidak bisa koneksi
Case folding : suka hoakstidak bisa koneksi

Figure 8. Case Folding Results

b. Tokenizing

Breaking sentences into individual words using several methods such as:

1. sent_tokenize: Splits the text into separate sentences.
2. word_tokenize: Splits the text into individual words.
3. gensim_tokenize: Tokenization used in the context of text modeling.

```

sent_tokenize \
0 Min mau tanya apakah keluar kota aman pake ind...
1 Ok
2 aplikasi yg sangat membantu
3 Mantap
4 Uang saya nyangkut diaplikasi orderan pertama ...

word_tokenize \
0 [Min, mau, tanya, apakah, keluar, kota, aman, ...
1 [ok]
2 [aplikasi, yg, sangat, membantu]
3 [Mantap]
4 [Uang, saya, nyangkut, diaplikasi, orderan, pe...

spanish_tokenize \
0 Min mau tanya apakah keluar kota aman pake ind...
1 Ok
2 aplikasi yg sangat membantu
3 Mantap
4 Uang saya nyangkut diaplikasi orderan pertama ...

gensim_tokenize
0 Min mau tanya apakah keluar kota aman pake ind...
1 Ok
2 aplikasi yg sangat membantu
3 Mantap
4 Uang saya nyangkut diaplikasi orderan pertama ...

```

Figure 9. Tokenizing Results

c. Stopword Removal

Removing irrelevant words to leave only essential words for further analysis.

```

[87] stopwords_ind = stopwords.words('Indonesian')
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

[88] from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords

stopwords_ind = stopwords.words('Indonesian')

[89] len(stopwords_ind)
758

[90] stopwords_ind
'sebetunya',
'sebisanya',
'sebuah',
'sebut',
'sebutlah',
'sebutnya',
'secara',
'secukupnya',
'sedang',
'cadanean'

```

Gambar 4.7 Stopword Pada google colab

```

Raw data      : Suka hoaks...tidak bisa koneksi
Case folding  : suka hoakstidak bisa koneksi
Stopword removal : suka hoakstidak koneksi

```

Figure 10. Stopwords in Google Colab

d. Stemming

Converting inflected words to their root form using the Sastrawi library.

```

raw_sample = data['review'].iloc[5]
case_folding = str(casefolding(raw_sample))
stopword_removal = remove_stop_words(case_folding)
text_stemming = stemming(stopword_removal)

print('Raw data\t\t: ', raw_sample)
print('Case folding\t\t: ', case_folding)
print('Stopword removal\t: ', stopword_removal)
print('Stemming\t\t\t: ', text_stemming)

```

```

Raw data      : Suka hoaks...tidak bisa koneksi
Case folding  : suka hoakstidak bisa koneksi
Stopword removal : suka hoakstidak koneksi
Stemming      : suka hoakstidak koneksi

```

Figure 11. Stemming Results

4. TF-IDF Weighting Results

The TF-IDF (Term Frequency-Inverse Document Frequency) method is used to assess the importance of a word in a particular document compared to the entire collection of documents. The weighting results serve as the basis for analysis using the classification algorithm.

```

x_tf_idf = vec_TF_IDF.transform(X)

# Menyimpan vocabulary dari TF-IDF menggunakan pickle
pickle.dump(vec_TF_IDF.vocabulary_, open("feature_tf-idf.pkl", "wb"))

# Melihat vocabulary dari TF-IDF
vec_TF_IDF.vocabulary_

```

```

{'min': 1593,
'mau': 1419,
'tanya': 2475,
'apakah': 95,
'keluar': 1113,
'kota': 1218,
'aman': 57,
'pake': 1871,
'indrive': 919,
'contohnya': 469,
'dari': 515,
'bandung': 189,
'ke': 1876,
'jakartasoalnya': 963,
'takut': 2453,
'penipuan': 1980,
'malah': 1352,
'rampok': 2118,
'atau': 127,

```

Figure 12. TF-IDF Vocabulary

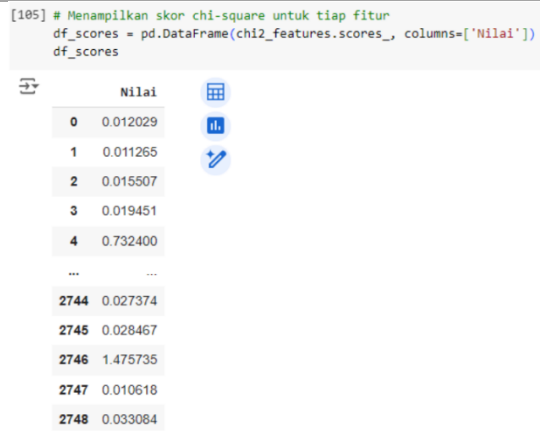


Figure 13. Chi-Square Score for Each TF-IDF Feature

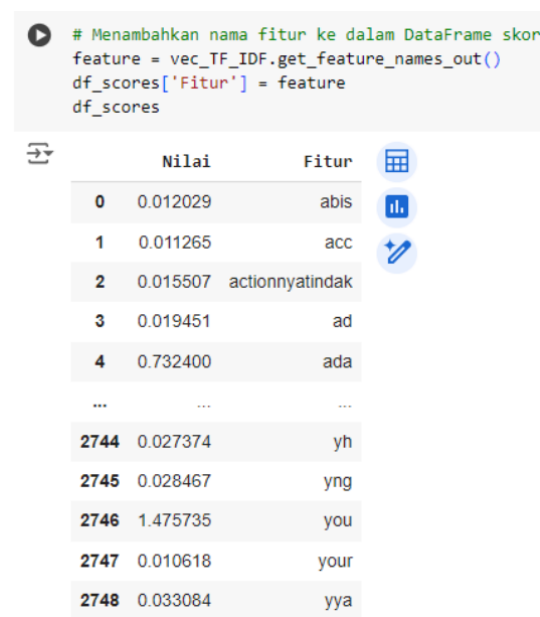


Figure 14. Features in Dataframe Score

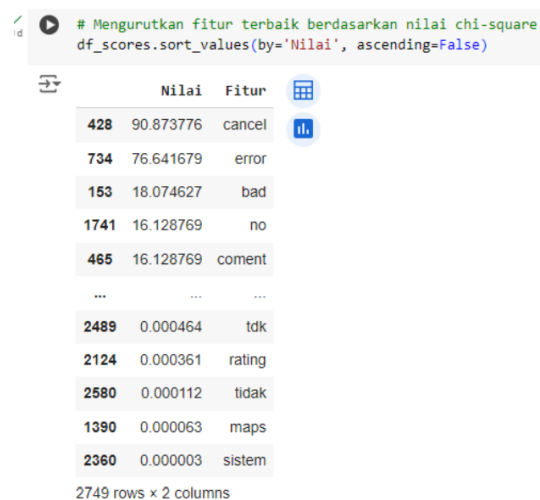


Figure 15. Best Features Based on Chi-Square Value

```

✓ id ▶ # Melihat jumlah fitur yang ada
print(len(vec_TF_IDF.vocabulary_))

↔ 2749
    
```

Figure 16. Number of Existing Features

5. Naive Bayes Results

Review data is divided into training and testing data to train the Naive Bayes model. This process is conducted in Google Colaboratory.

```

✓ 04 ▶ # It seems like the variable you meant to use was X_kbest_features
selected_x = X_kbest_features
selected_x

↔ array([[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.]])
    
```

```

✓ [148] #print
print("Banyak data x_train :",len(x_train))
print("Banyak data x_test :",len(x_test))
print("Banyak data y_train :",len(y_train))
print("Banyak data y_test :",len(y_test))

↔ Banyak data x_train : 800
    Banyak data x_test : 200
    Banyak data y_train : 800
    Banyak data y_test : 200
    
```

Figure 17. Training and Testing Data

6. Evaluation Results

Model evaluation is conducted using a confusion matrix. The evaluation results are as follows:

	Predicted Positive	Predicted Negative	Total
Actual Positive	84 (TP)	24 (FN)	108
Actual Negative	16 (FP)	76 (TN)	92
Total	100	100	200

$$\text{Accuracy} = \frac{84 + 76}{200} = 0.8 = 80\%$$

$$\text{Precision} = \frac{84}{84 + 16} = 0.84 = 84\%$$

$$\text{Recall} = \frac{84}{84 + 24} = 0.777 = 77.7\%$$

- **Negative Reviews: 450**

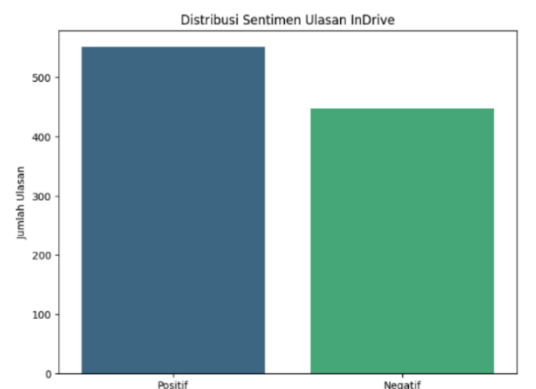


Figure 21. Test Results Chart

The chart shows the distribution of review numbers and frequently occurring words for each sentiment category.

CONCLUSION AND SUGGESTION

Based on the research results regarding sentiment analysis of user reviews for the inDrive application on Google Play Store, with a total of 1,000 reviews that have undergone text preprocessing (including transformation, tokenization, stopword removal, and stemming), and classification using the Naïve Bayes algorithm with data evaluation through a Confusion Matrix on Google Colaboratory, it can be concluded that the majority of responses or reviews towards the inDrive application are positive. The Naïve Bayes model demonstrated fairly good performance, with an accuracy of 80%, precision of 84%, and recall of 77%. These results indicate that the Naïve Bayes algorithm is quite effective in classifying review data from the Google Play Store.

For future research, it is recommended to try other algorithms, such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), or other algorithms, to compare performance in sentiment analysis. Additionally, it is advisable to expand the dataset to include a larger and more diverse sample, incorporating data from various sources, industry sectors, or different languages. This approach is expected to enhance the performance of the sentiment analysis model and provide deeper insights into user sentiment patterns.

ACKNOWLEDGMENT

The researcher expresses gratitude to Universitas Muhammadiyah Sumatera Utara for the facilities and guidance provided during this research, to the supervising lecturer for valuable direction and input, to fellow students for their support and technical assistance, and to family and friends for their prayers and moral support throughout the research process. Special thanks are also extended to everyone who has helped, both directly and indirectly, who cannot be mentioned individually. It is hoped that this research will be beneficial for the development of knowledge and serve as a reference for future research.

REFERENCES

- Abrilia, N. D. (2020). Pengaruh Persepsi Kemudahan dan Fitur Layanan terhadap Minat Menggunakan E-Wallet pada Aplikasi Dana di Surabaya. *Jurnal Pendidikan Tata Niaga (JPTN)*, 8(3), 1006–1012. <https://doi.org/10.26740/jptn.v8n3.p1006-1012>
- Afgiansyah, A. (2023). Artificial intelligence neutrality: Framing analysis of GPT-powered-Bing Chat and Google Bard. *Jurnal Riset Komunikasi*, 6(2), 179–193. <https://jurnalrisetkomunikasi.org/index.php/jrk/index>
- Amrullah, A. Z., Anas, A. S., & Hidayat, M. A. J. (2020). Analisis sentimen *movie review* menggunakan *Naive Bayes Classifier* dengan seleksi fitur *Chi Square*. *Jurnal Bumigora Information Technology (BITe)*, 2(1), 40–44.
- Annisa, Z., & Ulama, B. S. S. (2023). Analisis Sentimen Data Ulasan Pengguna Aplikasi “Pedulilindungi” pada Google Play Store Menggunakan Metode Naïve Bayes Classifier

- Model Multinomial. *Jurnal Sains dan Seni ITS*, 11(6), D464–D471. <http://dx.doi.org/10.12962/j23373520.v11i6.94064>
- Edo, M. P., & Fariadi, H. (2024). Development Strategy and Marketing System of Maxim Kelurahan Tanah Patah Kota Bengkulu, Bengkulu Province. *Jurnal Pengabdian Mandiri*, 1(1), 5-8. <https://doi.org/10.37676/mandiri.v1i1.339>
- Evelina, L. W., & Safitri, Y. (2019). Customer experience Bali natural beauty care through social media. *2019 International Conference on Information Management and Technology (ICIMTech)*, 1, 82-86. <https://doi.org/10.1109/ICIMTech.2019.8843757>
- Hafidz, N., & Liliana, D. Y. (2021). Klasifikasi Sentimen pada Twitter Terhadap WHO terkait Covid-19 Menggunakan SVM, N-Gram, PSO. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(2), 213 - 219. <https://doi.org/10.29207/resti.v5i2.2960>
- Indrayanto, C. G., Ratnawati, D. E., & Rahayudi, B. (2023). Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia Pada Google Play Store Menggunakan Metode Random Forest. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(3), 1131–1139. <http://j-ptiik.ub.ac.id>
- Indrayuni, E., Nurhadi, A., & Kristiyanti, D. A. (2021). Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc. *Faktor Exacta*, 14(2), 64–71. https://journal.lppmunindra.ac.id/index.php/Faktor_Exacta/index
- Kholilullah, M., Martanto, M., & Hayati, U. (2024). Analisis Sentimen Pengguna Twitter (X) tentang Piala Dunia Usia 17 Menggunakan Metode Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 392–398. <https://ejournal.itn.ac.id/index.php/jati/index>
- Kusnia, U., & Kurniawan, F. (2022). Analisis Sentimen Review Aplikasi Media Berita Online pada Google Play Menggunakan Metode Algoritma Support Vector Machines (SVM) dan Naive Bayes. *Explore IT: Jurnal Keilmuan dan Aplikasi Teknik Informatika*, 14(1), 24–28.
- Liu, B. (2020). Text sentiment analysis based on CBOW model and deep learning in big data environment. *Journal of ambient intelligence and humanized computing*, 11(2), 451-458.
- Prabowo, W. A., & Wiguna, C. (2021). Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode SCRUM. *Jurnal Media Informatika Budidarma*, 5(1), 149-156. <http://dx.doi.org/10.30865/mib.v5i1.2604>
- Putri, A., Hardiana, C. S., Novfujia, E., Siregar, F. T. P., Rahmadden, R., Fatma, Y., & Wahyuni, R. (2023). Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir: Comparison of K-NN, Naive Bayes and SVM Algorithms for Final-Year Student Graduation Prediction. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(1), 20-26. <https://doi.org/10.57152/malcom.v3i1.610>
- Retnosari, R. (2021). Analisa Kelayakan Kredit Usaha Mikro Berjalan pada Perbankan dengan Metode Naive Bayes. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 8(1), 53-59. <https://doi.org/10.30656/prosisko.v8i1.2848>
- Sandi, D., Utami, E., & Kusnawi, K. (2023). Analisis Sentimen Publik terhadap Elektabilitas Ganjar Pranowo di Tahun Politik 2024 di Twitter dengan Algoritma KNN dan Naive Bayes. *Jurnal Media Informatika Budidarma*, 7(3), 1097-1108. <http://dx.doi.org/10.30865/mib.v7i3.6298>
- Sari, F. V., & Wibowo, A. (2019). Analisis sentimen pelanggan toko online Jd. Id menggunakan metode Naive Bayes Classifier berbasis konversi ikon emosi. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 10(2), 681-686. <https://doi.org/10.24176/simet.v10i2.3487>
- Simanjuntak, A. Y., & Anita, A. (2022). Implementasi Data Mining Menggunakan Metode Naive Bayes Classifier untuk Data Kenaikan Pangkat Dinas Ketenagakerjaan Kota Medan. *Journal of Science and Social Research*, 5(1), 85-91. <https://doi.org/10.54314/jssr.v5i1.804>
- Susana, H. (2022). Penerapan Model Klasifikasi Metode Naive Bayes terhadap Penggunaan Akses Internet. *Jurnal Riset Sistem Informasi dan Teknologi Informasi (JURSISTEKNI)*, 4(1), 1-8. <https://doi.org/10.52005/jursistekni.v4i1.96>
- Tanggaraeni, A. I., & Sitokdana, M. N. (2022). Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naive Bayes. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 9(2), 785-795. <https://doi.org/10.35957/jatisi.v9i2.1835>
- Verawati, I., & Audit, B. S. (2022). Algoritma Naive Bayes Classifier Untuk Analisis Sentiment Pengguna Twitter Terhadap Provider By. u. *Jurnal Media Informatika Budidarma*, 6(3), 1411-1417. <http://dx.doi.org/10.30865/mib.v6i3.4132>
- Wahyusetyawati, E. (2017). Dilema pengaturan transportasi online. *Jurnal RechtsVinding*, 2089–9009.
- Widiawati, F., Kurniawan, R., & Suprapti, T. (2023). Klasifikasi Data Tingkat Kualitas Udara di

-
- Tangerang Selatan Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3739-3745. <https://doi.org/10.36040/jati.v7i6.8261>
- Zain, H. H., Awangga, R. M., & Rahayu, W. I. (2023). Perbandingan Model SVM, KNN dan Naïve Bayes untuk Analisis Sentiment pada Data Twitter: Studi Kasus Calon Presiden 2024. *JIM: Jurnal Ilmiah Mahasiswa Pendidikan Sejarah*, 8(3), 2083-2093. <https://doi.org/10.24815/jimps.v8i3.25342>