SOLUTION DOCUMENT

Sentiment Analysis on Newly Released Movie Study Case “Dune 2”

By :

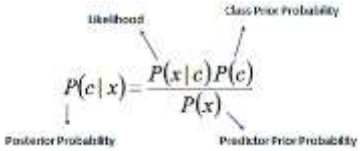
I Nengah Dharma Pradnyandita

1. Workflow of Sentiment Analysis Solution

Collect Data from Verified Movie Review

Data Preprocessing and convert data to numerical data

Split data into training and test data



Apply on Naïve Bayes method

Evaluating model based on accuracy

1.1 Collect Data from verified movie review

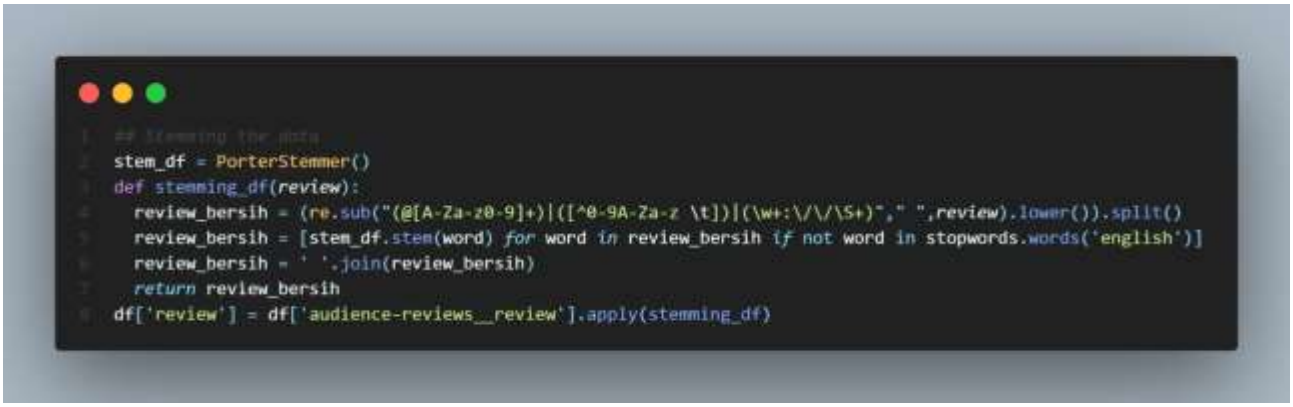
Data was obtained from Scraping the Rotten Tomatoes website in the Audience Review section. The scraped data was 510 rows. The data consists of six columns as seen in Figure 1.

review-data href	audience-reviews__name	audience-reviews__name href	audience-reviews__duration	audience-reviews__review	Sentiment
https://www.rotter	Scott B	https://www.rottentomatoes.cc	Apr 25, 2024	This movie is a stunning, visu	1
https://www.rotter	alan t	https://www.rottentomatoes.cc	Apr 25, 2024	Seen all this before. Star War	0
https://www.rotter	Drew F	https://www.rottentomatoes.cc	Apr 25, 2024	Zpoiler Alert: Wonka helps sa	1
https://www.rotter	Campolongo C	https://www.rottentomatoes.cc	Apr 25, 2024	Question: Is Denis Villeneuve	0
https://www.rotter	Nicholas S	https://www.rottentomatoes.cc	Apr 24, 2024	Honestly I enjoyed the first m	1
https://www.rotter	Jamey S	https://www.rottentomatoes.cc	Apr 24, 2024	If you actually read the books	0

Gambar 1 Table of Data from Dune 2 Movie reviews Rotten Tomatos

The Sentiment column functions as a label for each review with the information "1" meaning positive, "0" meaning negative and "2" meaning neutral.

1.2 Data Preprocessing converts data to numerical data



```
## Stemming the data
stem_df = PorterStemmer()
def stemming_df(review):
  review_bersih = (re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",review).lower()).split()
  review_bersih = [stem_df.stem(word) for word in review_bersih if not word in stopwords.words('english')]
  review_bersih = ' '.join(review_bersih)
  return review_bersih
df['review'] = df['audience-reviews__review'].apply(stemming_df)
```

Gambar 2 Data Preprocessing to cleaning the text

In data preprocessing, several steps are carried out

1. Remove unnecessary characters such as symbols.
2. Change all words to lowercase, this is done to reduce variations in the text.
3. Tokenization of each review, where the text is divided into smaller units, namely words.
4. Eliminate stop words that often appear in the text and do not have much influence on processing,
5. Stemming each word to reduce variations in words.
6. Rearrange each word into a sentence.

1.3 Convert review text into numerical data

```
## Convert the text review to numerical values
#Converting into numerical data
vectorizer = TfidfVectorizer()
Xtrain = vectorizer.fit_transform(Xtrain)
Xtest = vectorizer.transform(Xtest)
```

Gambar 3 Convert Review Text into Numerical Data

In the Vectorizer Process, the data for each review is converted into numeric using TfidfVectorizer. This method is used because the data to be implemented are sentences, so this method is used to convert sentences into a vector of numbers that can be processed by the model. This method is also used because each word will be identified based on its level of importance, making it easier to identify sentiment. The results of the implementation of data are shown in Figure 4.

```
(0, 519)      0.1486252548638906
(0, 1513)     0.17049244489902232
(0, 2291)     0.2601638907180995
(0, 1965)     0.2475024982888377
(0, 274)      0.18660781159600848
```

Gambar 4 Result of TFidVectorizer

1.4 Split data into training and testing data

Gambar 5 Spliting data into Training and testing

In this process, the data will be divided into two parts, namely Training data and Testing data. Training data is used to train the model and testing data is used to test the model. The percentage of Training data is 80% of the total data while Testing data is 20% of the total data.

1.5 Implement the Naïve Bayes Model to predict the data

Gambar 6 Implement the data into Naïve Bayes Model

The Machine Learning model used to implement sentiment data is Naïve Bayes. Model selection is based on accuracy and the degree of fit of the data to the model. The Naïve Bayes model classifies data based on the probability of a match between new data and training data. Compared to other Machine Learning models, Naïve Bayes is very suitable for Categorical data types, therefore this model can prevent overfitting. In this case, Multinominal Naïve Bayes is used, because the data used is in vector form.

1.6 Evaluate model based on Accuracy

```
start_ts = time.time()
print('Accuracy Score on the training data :', accuracy_score(ytrain,modelnb.predict(Xtrain)))
print('Accuracy Score on the test data :',accuracy_score(ytest,modelnb.predict(Xtest)))
end_ts = time.time()
print(f"Prediction speed [s]: {(end_ts-start_ts):.3f}")
```

Gambar 7 Evaluate Model Based On Accuracy

Accuracy Measures are used to evaluate models that have been trained with Sentiment data. The model accuracy results obtained were 75%, this proves that the model works quite well on movie review data. Meanwhile, the speed of data processing is 0.069 seconds. A small amount of data affects the processing speed of the model, the larger the data the longer the process.

```
Accuracy Score on the training data : 0.8088235294117647
Accuracy Score on the test data : 0.7549019607843137
Prediction speed [s]: 0.069
```

Gambar 8 The result of Model Accuracy

2. Next Steps to Improve the Solution

The next step that can be taken to improve the solution is to select more appropriate parameters using Grid Search. Apart from that, the data used can be increased so that the model has more references in predicting data sentiment. The model used can be varied using an Optimization Algorithm.