UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SENTIMENT ANALYSIS OF INDONESIA'S NEW CAPITAL CITY ON TWITTER USING NAIVE BAYES

HIBBAN KAMIL HIZBUTTAHRIR (A20EC5022)

Innovating Solutions

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

RESEARCH OUTLINE

1. INTRODUCTION
2. PROBLEM STATEMENT
3. OBJECTIVES
4. SCOPE OF RESEARCH
5. LITERATURE REVIEW
6. METHODOLOGY
7. OUTCOME RESULT

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

INTRODUCTION

In the rapidly growing social media usage era such as X (Twitter), the analysis of sentiment expressed by individuals plays a crucial role in understanding public opinion. With the expansion of online platforms, people can share their opinions with others. Those opinions can be invaluable insight into public opinion, helping policymakers to make informed decisions.

Sentiment analysis is a part of Natural Language Processing (NLP) that has come as a tool for drawing analysis. With the significance of sentiment analysis has been its ability to automatically classify text as positive, negative or neutral, allowing for analysis of public opinion. This capability is essential to better understand public sentiment and identify trends over time (Kaharudin et al., 2023).

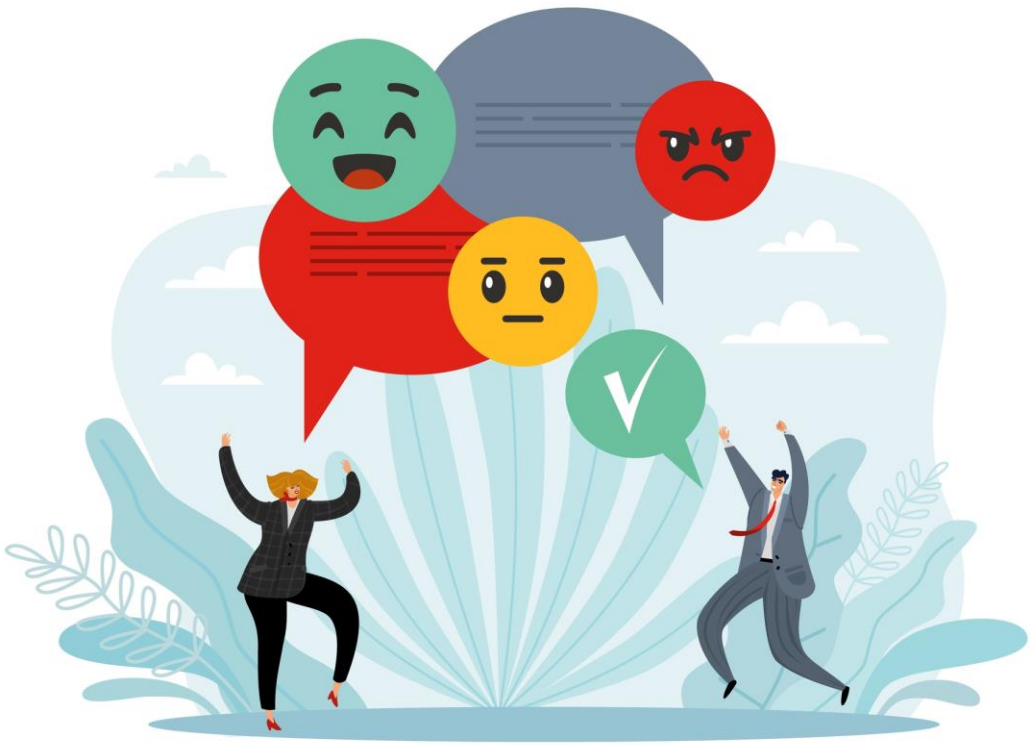
The text of sarcasm, irony and vague language will require powerful NLP techniques and Machine Learning (ML) to accurately extract sentiment. This research aims to address these challenges by developing a robust sentiment analysis system using the Naïve Bayes Classification Algorithm. The system will categorize sentiments into positive, neutral, and negative, and compare the results to evaluate accuracy, precision, and recall (Wandani et al., 2021).

Illustration

The focus of this study is on the public's response to the New Capital City of Indonesia, known as Ibu Kota Nusantara (IKN). The relocation of the capital from Jakarta to IKN represents strategic decisions and planning by the government of Indonesia, targeting the challenges in the country's contemporary urbanization context (Perdana, 2024)

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

PROBLEM STATEMENT



- Complexities of human language, including sarcasm, irony, and vague expressions.
- Current sentiment analysis models are not robust enough to handle these complexities accurately.
- There is a need for accurate and reliable sentiment analysis to gauge public opinion on significant like the New Capital City of Indonesia

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

OBJECTIVES

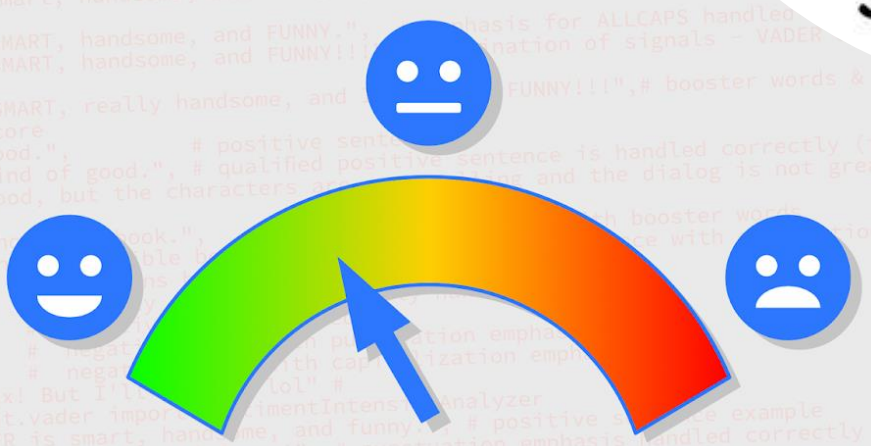
- To gather and pre-process the public sentiment data from Twitter regarding the New Capital City of Indonesia.
- To design and implement the Naive Bayes Algorithm for classifying the gathered Twitter data into positive, neutral, and negative sentiments.
- To analyze and evaluate the performance of the Naive Bayes Algorithm using accuracy metrics, confusion matrix, and comparison with another classification algorithm.

SCOPE OF RESEARCH

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCOPE

- Research study focusing on Indonesia's New Capital City and sentiment analysis as the main issue
- Utilize Python programming language and the dataset sourced from Tweet on Twitter (X) will be preprocessed for sentiment analysis.
- Activities include data collection, NLP technique and Naïve Bayes implementation, model performance assessment, refinement based on evaluation results, and real-world scenario testing to validate the developed sentiment analysis system.



Sentiment Analysis
with Python

LITERATURE REVIEW

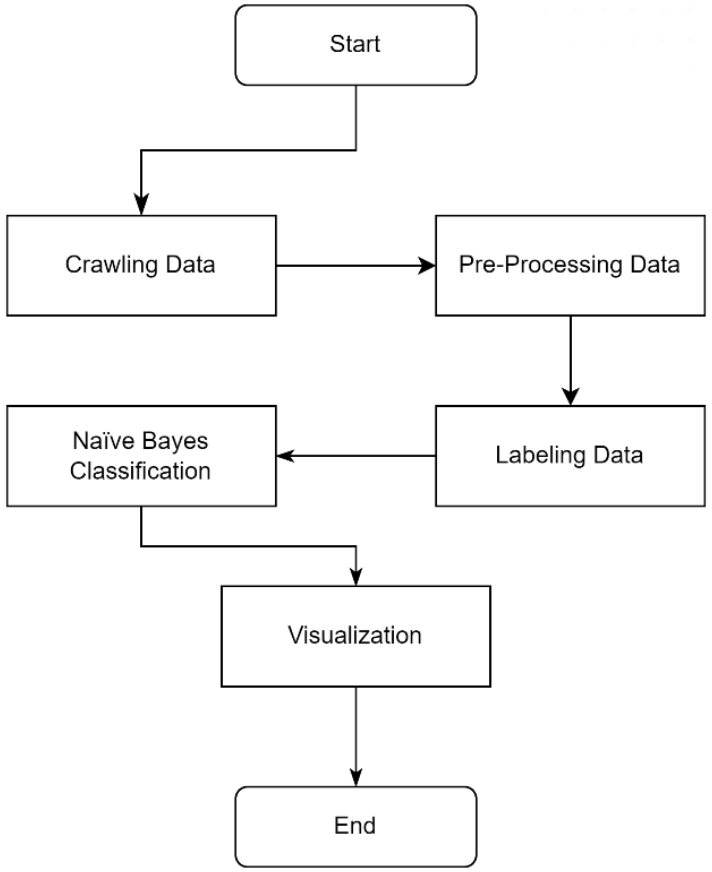
Purbayanto, B. and Suharsono, T.N. (2023) 'Sentiment Analysis of X Users against Chatgpt with Naive Bayes Algorithm', Jurnal Telematika vol.18, no.2	**Oktavia, I. and Isnain, A. R. (2024) 'Opinion Sentiment Analysis on Artificial Intelligence (AI) Tools Based on Twitter Using Naïve Bayes Algorithm', Jurnal Media Informatika Budidarma vol.8, no.2, pp. 777-787**	**Putra, S. A. and Wijaya, A. (2023) 'Artificial Intelligence (AI) Sentiment Analysis on Social Media Twitter Using Lexicon Based Method', Jurnal Sistem & Teknologi Informasi Komunikasi vol.7, no.1**
Method/Algorithm: Naive Bayes classifier Dataset: Tweet from Twitter (X) Keyword: Chatgpt Result: positive (1,543 data), negative (318 data), and neutral (1,023 data) sentiments. The Naive Bayes algorithm provides an accuracy of 87.175043%.	Method/Algorithm: Naive Bayes classifier Dataset: Tweet from Twitter (X) Keyword: AI / Artificial Intelligence Result: 58.41% positive data and 12.43% negative data. In the analysis experiment, the Naïve Bayes accuracy value reached 79.41%, with a precision of 88% and a recall of 88%	Method/Algorithm: Lexicon Based Method Dataset: Tweet from Twitter (X) Keyword: Artificial Intelligence / AI Result: 64% with positive sentiment, 25% with neutral sentiment and 11% with negative sentiment.

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

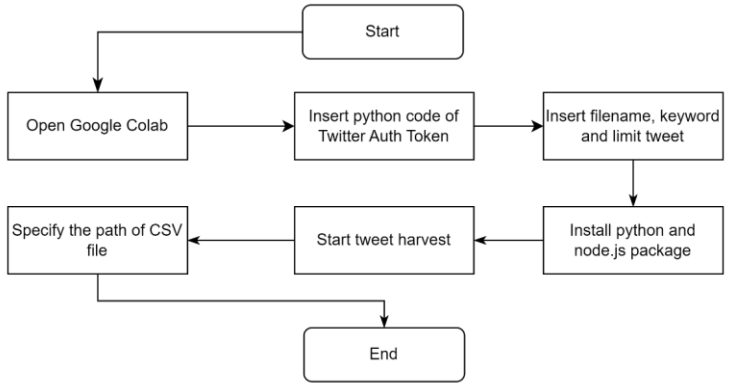
METHODOLOGY

NAÏVE BAYES CLASSIFIER

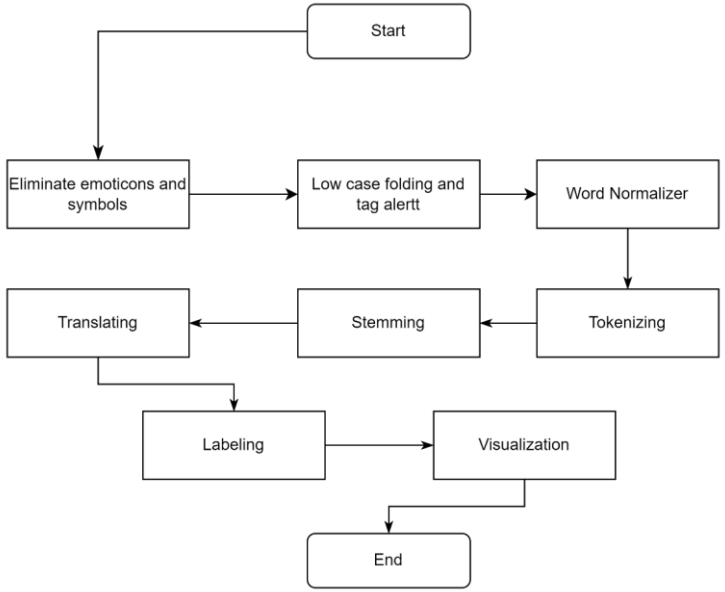
- **Tools Used:** Python, Google Colab.
- **Data Collection:** Using Tweet Harvest to scrape tweets from Twitter.
- **Data Preprocessing:** Cleaning and preparing data for analysis through tokenization, removal of stop words, and stemming.
- **Model Implementation:** Training and testing the Naïve Bayes model for sentiment classification



Methodology workflow



Crawling Dataset workflow



Data pre-processing workflow

OUTCOME
RESULT

```
# Crawl Data

filename = 'IKNafter.csv'
search_keyword = 'ikn until:2022-12-15 since:2022-01-22 lang:id'
limit = 2000

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```

```
Tweet Harvest [v2.6.1]

Research by Helmi Satria
Use it for Educational Purposes only!

This script uses Chromium Browser to crawl data from Twitter with your Twitter auth token.
Please enter your Twitter auth token when prompted.

Note: Keep your access token secret! Don't share it with anyone else.
Note: This script only runs on your local device.


Opening twitter search page...


Found existing file ./tweets-data/IKNafter.csv, renaming to ./tweets-data/IKNafter.old.csv

-- Scrolling... (1) (2)
Filling in keywords: ikn until:2022-12-15 since:2022-01-22 lang:id
```

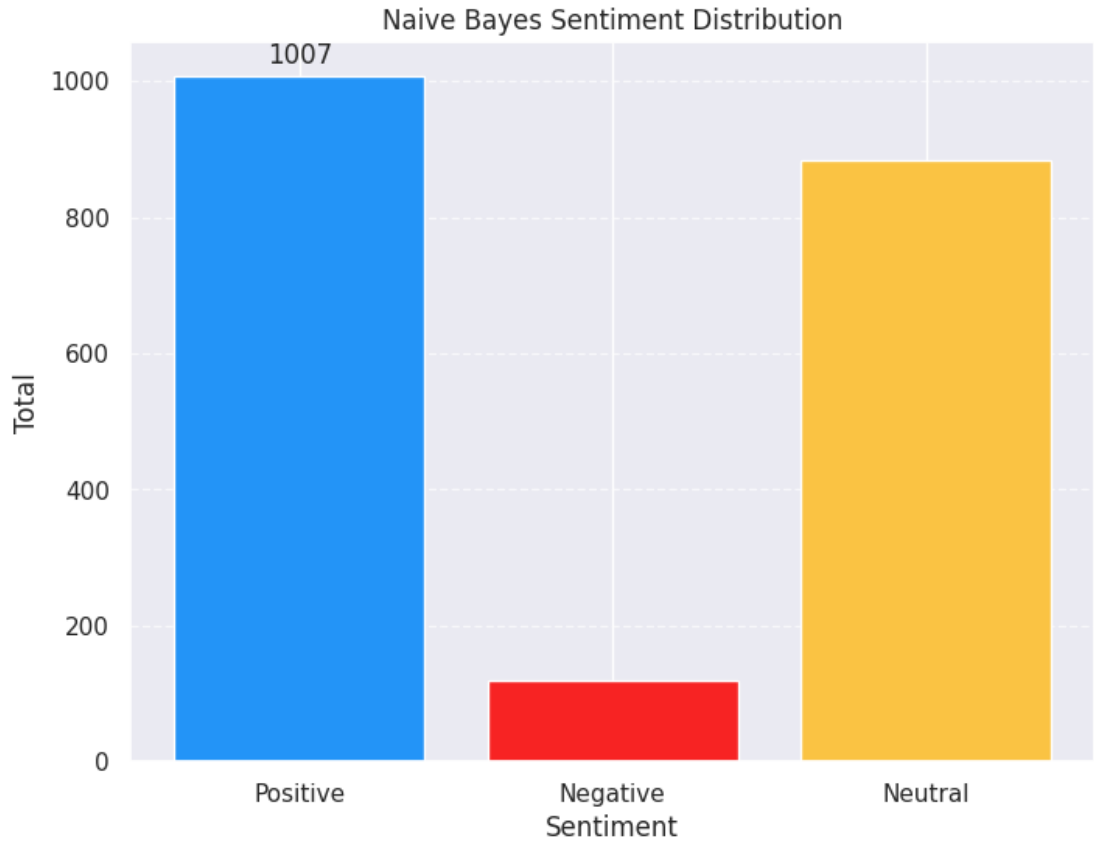
```
[ ] # Crawl Data

    filename = 'IKNbefore.csv'
    search_keyword = 'ikn until:2021-12-30 since:2021-07-01 lang:id'
    limit = 2000

    !npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```

```
    Tweet Harvest [v2.6.1]

    Research by Helmi Satria
    Use it for Educational Purposes only!

    This script uses Chromium Browser to crawl data from Twitter with your Twitter auth token.
    Please enter your Twitter auth token when prompted.

    Note: Keep your access token secret! Don't share it with anyone else.
    Note: This script only runs on your local device.


    Opening twitter search page...


    -- Scrolling... (1)
    Filling in keywords: ikn until:2021-12-30 since:2021-07-01 lang:id
```


ikn
pindah
negara
baru
kota
ibu
bangun
kalimantan
banjir
perintah
pansus
buat
jadi
ada
gak
mau
rakyat
kaltim
sama
apa
indonesia
dpr
jangan
banyak
rencana
anggota
aja
paser
calon
tak
akan
tdk
ikut
ekonomi
cepat
semua
bukan
orang
atur
undang
penting
lebih
bahas
lokasi
jalan
utara
proyek
presiden
najam
timur
dgn
sdh
kait
kalo
yang
utk
khusus
kerja
jd
amp
baik
uu
tetap
jakarta
tidak
masuk
ibukota
daerah
laku
tahun
sebut
lahan
kata
makin
punya
nya
atas
pak
pk
selamatkannegeridariazab
infrastruktur
kepala
tanah
lu
soal
minta
jual
hukum
siap
bebas
masyarakat
kawasan
bikin
kereta
kaji
gesa
bakal
satu
tolak
kok
ini
jelas
jokowi
hari
mana
air
pakai
paku
tinggal
kabupaten
dampak
anak
mungkin
ubah
pimpin
tambah
butuh
segera
selesai
dasar
bilang
ppu
sah
benar
tahu
terus
lihat
dulu
pasti
dki
aku
partai
bencana
moga
perlu
milik
anies
dalam
kali
dana
nyata
tempat
udah
mulai
siapa
hutan
tata
kan
dapat
tangan
bentuk
tatib
masalah
tuju
status
sangat
tol
klo
utang
nama
biaya
pusat
dukung
sekarang
kuasa
bisa
besar
itu
salah
malah
krn
gubernur
wilayah
lanjut
sudah
mati
warga
investasi
cuma
dr
dlm
hutang
org
waktu
akhir
harus
ala
lg
ni
diri
tau
ri
ketua
target
wak

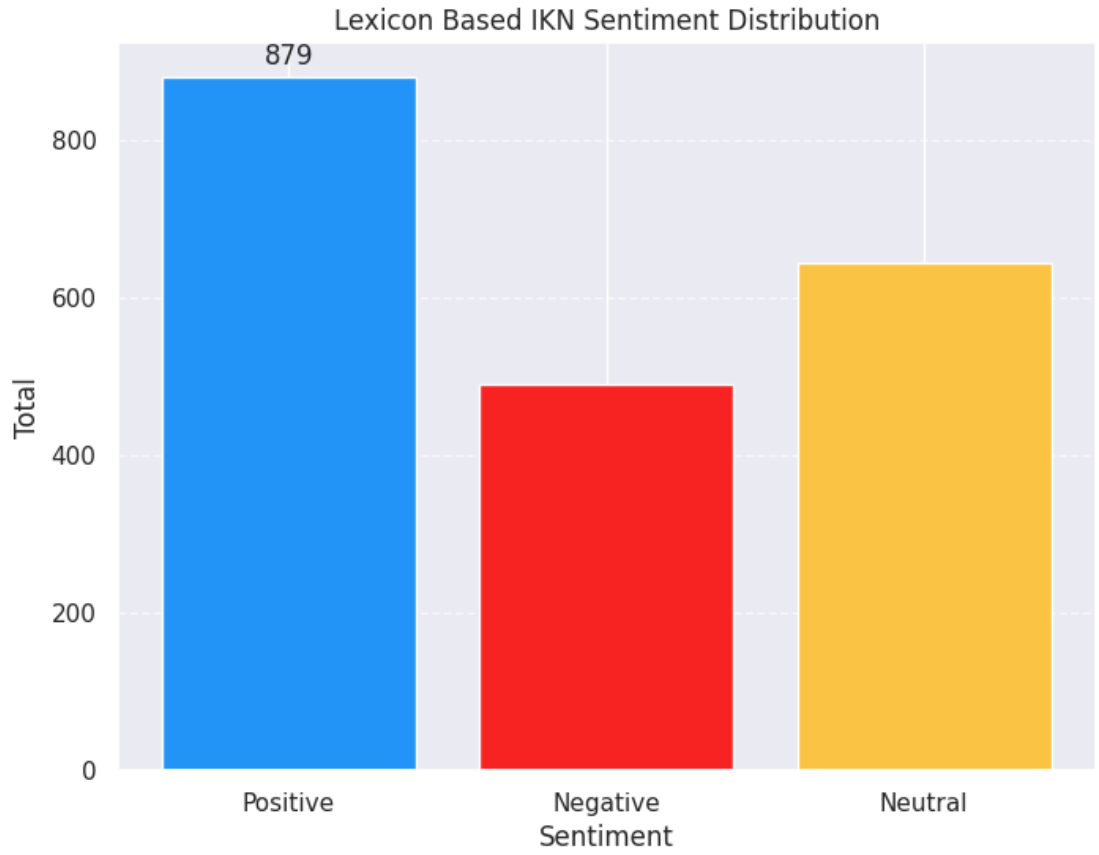
```
[125] from textblob import TextBlob

      # Assuming 'cl' is your trained classifier and 'data' is a DataFrame with 'tweet_english' column
      data_tweet = list(data['tweet_english'])

      status = []
      total_positive = total_negative = total_neutral = 0

      for i, tweet in enumerate(data_tweet):
          analysis = TextBlob(tweet, classifier=cl)
          sentiment = analysis.classify()  # Get the predicted sentiment
          status.append(sentiment)  # Store the sentiment for each tweet

          if sentiment == 'Positive':
              total_positive += 1
          elif sentiment == 'Negative':
              total_negative += 1
          else:
              total_neutral += 1

      total = total_positive + total_negative + total_neutral

      print("Naive Bayes Classifier")

      print("Total Positive: ", total_positive)
      print("Total Negative: ", total_negative)
      print("Total Neutral: ", total_neutral)
      print("Total: ", total)

      # Add the predicted sentiments to your DataFrame if needed
```

```
Naive Bayes Classifier
Total Positive:  1007
Total Negative:  119
Total Neutral:  885
Total:  2011
```



```
[145] data_tweet = list(data['tweet_english'])

      status = []
      total_positive = total_negative = total_neutral = total = 0

      for i, tweet in enumerate(data_tweet):
        # Get sentiment scores for the tweet
        scores = analyzer.polarity_scores(tweet)
        compound_score = scores['compound']

        if compound_score >= 0.05:
          total_positive += 1
          status.append('Positive')
        elif compound_score <= -0.05:
          total_negative += 1
          status.append('Negative')
        else:
          total_neutral += 1
          status.append('Neutral')

        total += 1

      print(f'Lexicon Based Classifier:\nPositive = {total_positive}\nNeutral = {total_neutral}\nNegative = {total_negative}')
      print(f'\nTotal Data: {total}')

      # Add the predicted sentiments to your DataFrame
      data['lexicon_classifier'] = status
```

```
Lexicon Based Classifier:
Positive = 879
Neutral = 643
Negative = 489

Total Data: 2011
```



```
[129] from textblob.classifiers import NaiveBayesClassifier
      cl = NaiveBayesClassifier(train_set)
      print("Test Accuracy: ", cl.accuracy(dataset))
```

```
Test Accuracy:  0.7747389358528095
```

```
[128] from textblob.classifiers import NaiveBayesClassifier
      from sklearn.metrics import confusion_matrix, precision_score, recall_


      # Get predictions for the test dataset
      predicted_sentiments = [cl.classify(text) for text, true_sentiment in

      # Assuming 'dataset' is a list of tuples: (text, true_sentiment)
      true_sentiments = [true_sentiment for text, true_sentiment in dataset

      # Create confusion matrix
      cm = confusion_matrix(true_sentiments, predicted_sentiments)
      print("Confusion Matrix:\n", cm)

      # Calculate precision, recall, and F1-score
      precision = precision_score(true_sentiments, predicted_sentiments, av
      recall = recall_score(true_sentiments, predicted_sentiments, average=
      f1 = f1_score(true_sentiments, predicted_sentiments, average='weighte

      print("Precision: ", precision)
      print("Recall: ", recall)
      print("F1-Score: ", f1)
```

```
Confusion Matrix:
 [[109 102 115]
 [  1 638  81]
 [  9 145 811]]
Precision:  0.7930535385571783
Recall:  0.7747389358528095
F1-Score:  0.7587477703964054
```

```
[143] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

      # 'data' is your DataFrame with 'lexicon_classifier' and 'classifier' columns
      true_labels = data['classifier']
      predicted_labels = data['lexicon_classifier']

      # Calculate metrics
      accuracy = accuracy_score(true_labels, predicted_labels)
      precision = precision_score(true_labels, predicted_labels, average='weighted', zero_division=1)
      recall = recall_score(true_labels, predicted_labels, average='weighted', zero_division=1)
      f1 = f1_score(true_labels, predicted_labels, average='weighted', zero_division=1)

      print("Accuracy:", accuracy)
      print("Precision:", precision)
      print("Recall:", recall)
      print("F1-Score:", f1)
```

```
Accuracy: 0.625559423172551
Precision: 0.6496794184673007
Recall: 0.625559423172551
F1-Score: 0.6329699890979469
```

THANK YOU

univteknologimalaysia utm.my utmofficial

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Innovating Solutions
Menginovasi Penyelesaian