

|  |
| --- |
| **Twitter Sentimental Analysis** |

|  |  |
| --- | --- |
| ***Class and Section*** | **BS Mathematics Semester(A)** |
| ***Session*** | **2021-2025** |
| ***Submitted to*** | **Dr Nimra Tariq** |
| ***Submission Deadline*** | **14-05-2024** |

*Submitted by:*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Sr No*** | ***Name*** | ***Roll No*** | ***Signature*** |
| ***1*** | ***Ushna Fatima*** | ***BSMM-F21-018*** |  |
| ***2*** | ***Anum Nawaz*** | ***BSMM-F21-024*** |  |
| ***3*** | ***Umm e Farwa*** | ***BSMM-S22-003*** |  |
| ***4*** | ***Zohaib*** | ***BSMM-S22-001*** |  |

Marks Obtained: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of Project Advisor: \_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Abstract:**

Sentiment analysis on Twitter data has become increasingly important in understanding public opinion and sentiment trends. In this project, we perform sentiment analysis on Twitter data using machine learning techniques. Our approach involves preprocessing the Twitter text data, including cleaning, tokenization, and vectorization. We then build a sentiment classification model using Multinomial Naive Bayes and evaluate its performance on a test dataset.

1. **Introduction:**

With the proliferation of social media platforms like Twitter, understanding the sentiment of users has become crucial for various applications such as brand monitoring, market research, and public opinion analysis. Sentiment analysis, also known as opinion mining, involves extracting and analyzing sentiment information from text data. In this project, we focus on sentiment analysis of Twitter data. We aim to develop a machine learning model capable of classifying the sentiment of tweets as positive or negative.

1. **Implementation Details:**
2. **Library Import**

import pandas as pd

import numpy as np

import nltk

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from nltk.stem import WordNetLemmatizer

import re

from sklearn.model\_selection import train\_test\_split

This section imports necessary libraries for data manipulation, natural language processing (NLP), and machine learning.

1. **Dataset Import and Exploration**

dataset = pd.read\_csv('C:/Users/Hp/Desktop/datascience PROJECTS/Sentimental Analysis/Twitter analysis/Sentiment.csv')

print(dataset)

dataset = dataset[['text','sentiment']]

print(dataset)

dataset = dataset[dataset['sentiment']!= 'Neutral']

print(dataset)

Here, the dataset is imported from a CSV file and explored. It is then filtered to remove tweets with neutral sentiment.

1. **Data Preprocessing**

train,test = train\_test\_split(dataset,test\_size=0.1)

print(train)

print(test)

print(train['text'][9720])

The dataset is split into training and testing sets. A sample tweet is printed to ensure data integrity.

1. **Text Cleaning and Preprocessing**

pattern = r"(#\w+)|(RT\s@\w+:)|(http.\*)|(@\w+)"

ps = PorterStemmer()

lemmatizer = WordNetLemmatizer()

def Clean\_text(data):

tweets = []

sentiments = []

for index,row in data.iterrows():

sentence = re.sub(pattern,'',row.text)

words = [e.lower() for e in sentence.split()]

words = [lemmatizer.lemmatize(word) for word in words if word not in stopwords.words('english')]

words = ' '.join(words)

tweets.append(words)

sentiments.append(row.sentiment)

return tweets,sentiments

This section defines a function to clean and preprocess the text data, including removing special characters, stopwords, and performing lemmatization.

1. **Label Encoding**

from sklearn.preprocessing import LabelEncoder

labelencoder = LabelEncoder()

processed\_data['sentiments'] = labelencoder.fit\_transform(processed\_data['sentiments'])

print(processed\_data)

The sentiment labels are encoded into numerical values using LabelEncoder for model compatibility.

1. **Vectorization**

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer(ngram\_range=(1,3))

cv.fit(processed\_data['tweets'])

X\_train = cv.transform(processed\_data['tweets'])

print(X\_train.shape)

print(X\_train.toarray())

The text data is converted into numerical vectors using CountVectorizer, which creates a matrix of word frequencies.

**7. Model Building (Naive Bayes)**

from sklearn.naive\_bayes import MultinomialNB

classifier = MultinomialNB()

print(classifier.fit(X\_train.toarray(),target))

A Multinomial Naive Bayes classifier is trained using the vectorized training data.

**8. Testing and Evaluation**

test\_tweets,test\_sentiments = Clean\_text(test)

data\_test = {'tweets':test\_tweets,'sentiments':test\_sentiments}

final\_test\_data = pd.DataFrame(data\_test)

X\_test = cv.transform(final\_test\_data['tweets'])

print(X\_test.shape)

y\_pred = classifier.predict(X\_test.toarray())

The trained model is tested on the test dataset, and predictions are made on the test data.

**9. Results Analysis**

final\_test\_data['sentiments'] = labelencoder.fit\_transform(final\_test\_data['sentiments'])

print(final\_test\_data)

actual\_values = final\_test\_data['sentiments'].values

print(actual\_values)

from sklearn.metrics import accuracy\_score

print(accuracy\_score(y\_pred,actual\_values))

The predicted sentiments are compared against the actual sentiments from the test data, and accuracy is calculated using the accuracy\_score metric.

**10. Checking**

print(final\_test\_data)

print(X\_test.toarray()[0])

print("Predicted Sentiment:",classifier.predict(X\_test.toarray()[[0]]))

The final test data and the sentiment prediction for a sample tweet are printed, concluding the sentiment analysis process.

1. **Result:**

The Multinomial Naive Bayes classifier achieved an accuracy of 80+ on the test dataset, demonstrating the effectiveness of the sentiment analysis model in classifying Twitter sentiment.

1. **Conclusion:**

In this project, we successfully conducted sentiment analysis on Twitter data using machine learning techniques. By preprocessing the text data and building a classification model, we were able to accurately classify the sentiment of tweets as positive or negative. This study highlights the importance of sentiment analysis in understanding public opinion and sentiment trends on social media platforms like Twitter. Further improvements could be made by exploring advanced machine learning algorithms and incorporating more sophisticated text preprocessing techniques.