MACHINE LEARNING REPORT

SENTIMENTAL ANALYSIS ON RESTAURANT REVIEWS

PROJECT BY:

KAJA LAKSHMI SAI PRIYA (1001843857)

ADITYA DHEERAJ GUNDA (1001960773)

MOHAN VIKAS REDDY PULAGAM (1001916601)

INTRODUCTION

Sentimental Analysis also referred as Opinion Mining, is a Natural Language Processing technique which identified the emotion/sentiment behind the text. Sentimental Analysis analyzes a bunch of text to determine if it is positive, negative or neutral. Sentiment analysis tools extracts opinions which helps to understand what users think and react to improve the user experience. Sentimental analysis is widely applied that ranges from healthcare to marketing.

APPLICATION OF SENTIMENTAL ANALYSIS ON RESTAURANT REVIEWS

A lot of data is generated who is not being valued properly. Now a days, restaurants wanted to understand know opinions of people about their food and service. It helps them to understand what customers are thinking about their restaurant? Restaurants can quantify the customer's reviews and analyze them with good accuracy using sentimental analysis.

We might be unaware of the restaurant how does the food and quality tastes like? These reviews helps customers to take decision for choosing the restaurant. This is going to create better user experience.

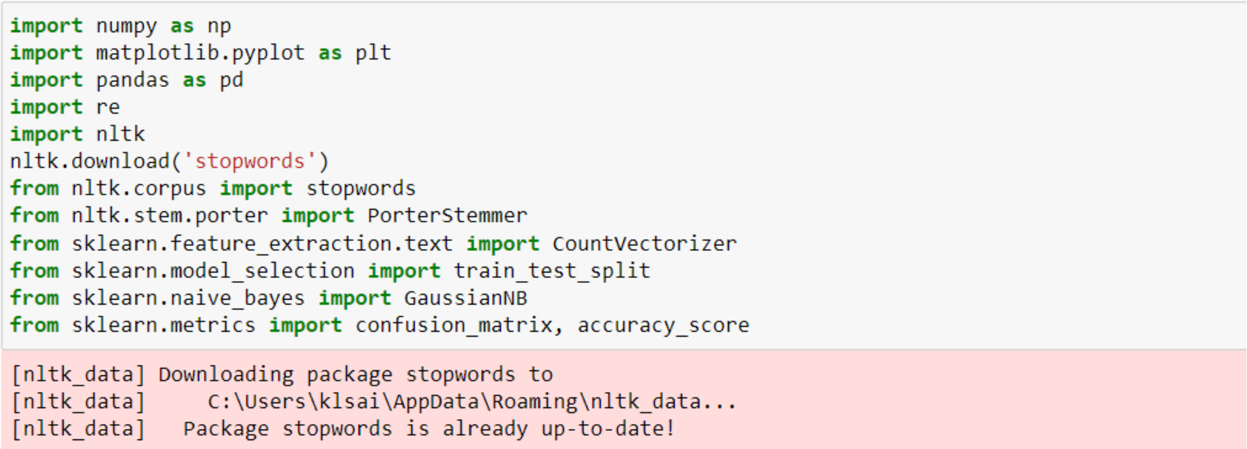
TECHNOLOGIES USED:

Programming Languages Used: PYTHON

IDE : JUPYTER NOTEBOOK

GETTING STARTED

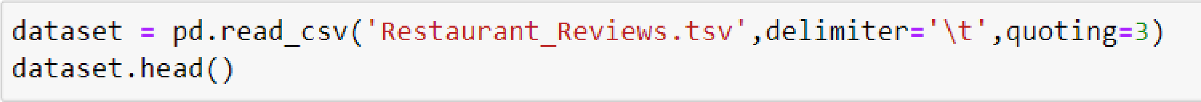
Project requires to import the libraries to implement the functionality.



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score
```

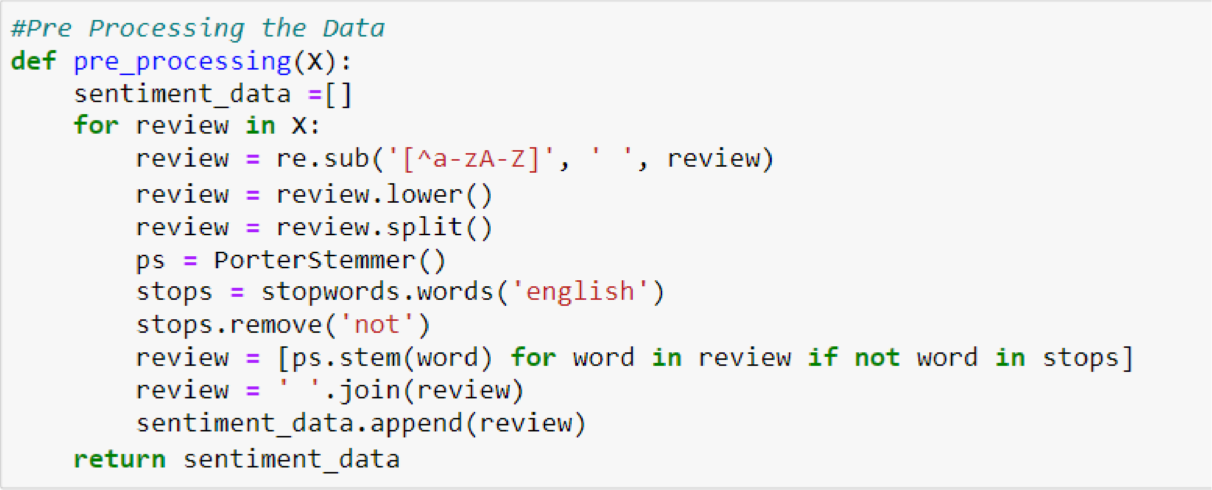
```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\klsai\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Importing the Dataset

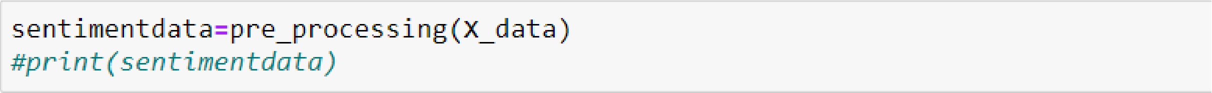


```
dataset = pd.read_csv('Restaurant_Reviews.tsv',delimiter='\t',quoting=3)
dataset.head()
```

Preprocessing the Data

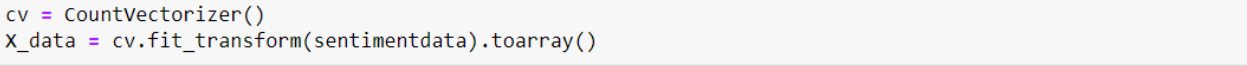


```
#Pre Processing the Data
def pre_processing(X):
    sentiment_data =[]
    for review in X:
        review = re.sub('[^a-zA-Z]', ' ', review)
        review = review.lower()
        review = review.split()
        ps = PorterStemmer()
        stops = stopwords.words('english')
        stops.remove('not')
        review = [ps.stem(word) for word in review if not word in stops]
        review = ' '.join(review)
        sentiment_data.append(review)
    return sentiment_data
```

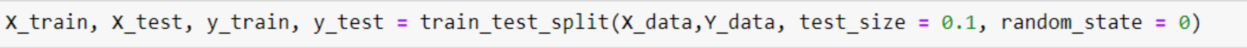


```
sentimentdata=pre_processing(X_data)
#print(sentimentdata)
```

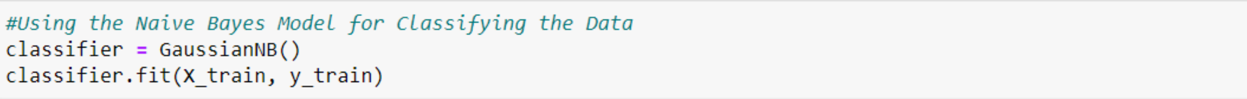
Applying Naïve Bayes Algorithms for classification



```
cv = CountVectorizer()
X_data = cv.fit_transform(sentimentdata).toarray()
```

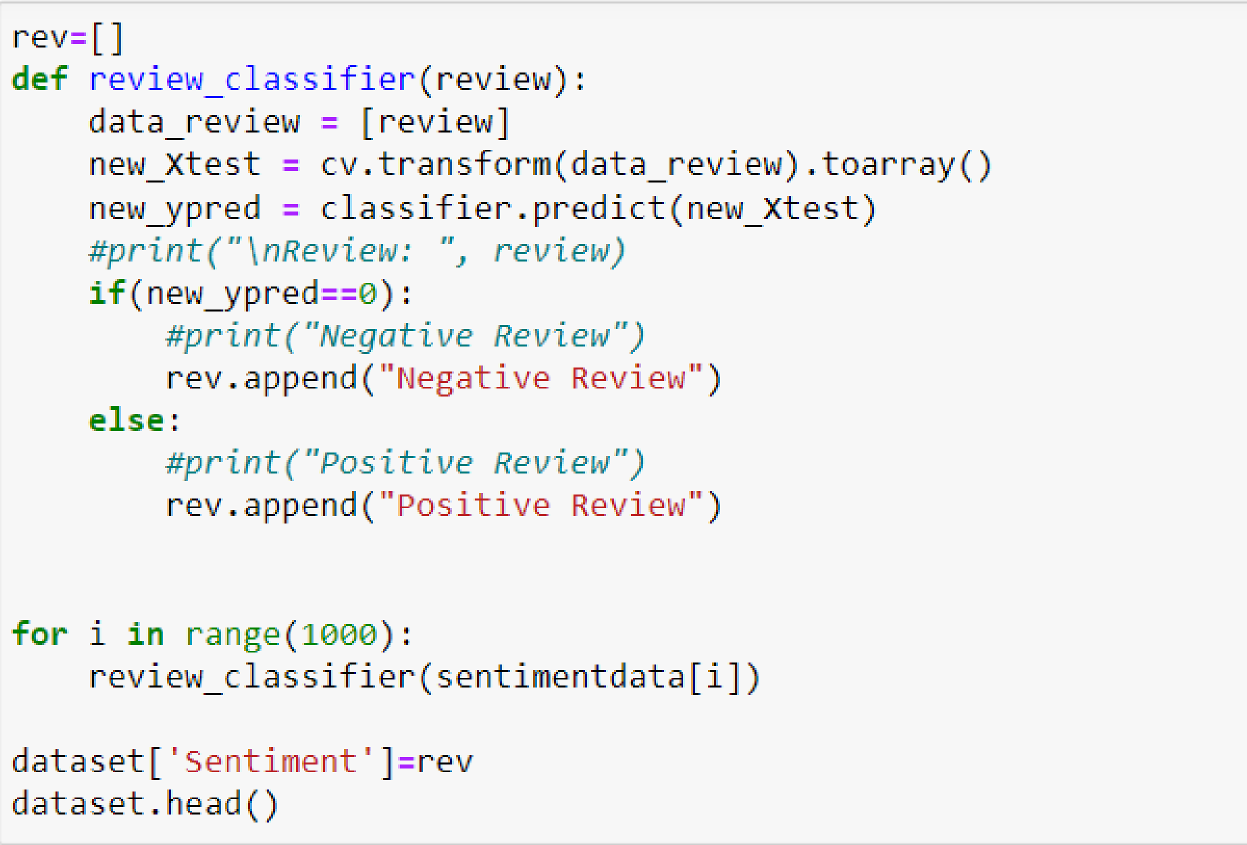


```
X_train, X_test, y_train, y_test = train_test_split(X_data,Y_data, test_size = 0.1, random_state = 0)
```



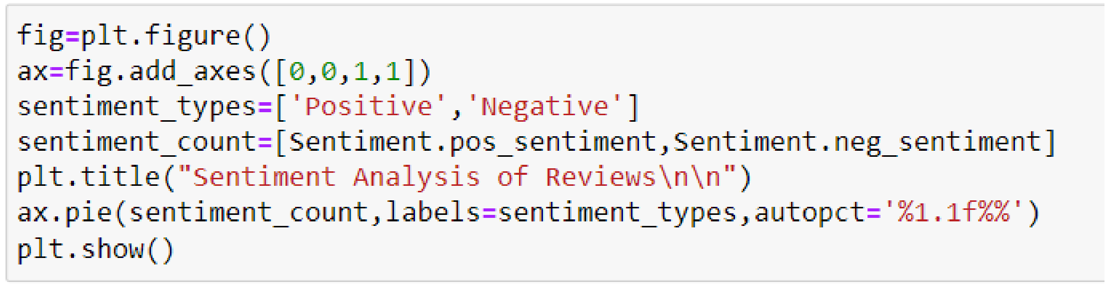
```
#Using the Naive Bayes Model for Classifying the Data
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Classifying sentiments to the review data

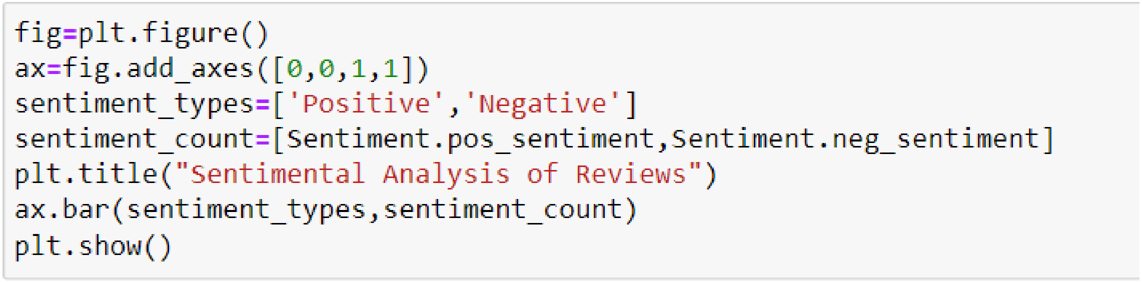


```
rev=[]
def review_classifier(review):
    data_review = [review]
    new_Xtest = cv.transform(data_review).toarray()
    new_ypred = classifier.predict(new_Xtest)
    #print("\nReview: ", review)
    if(new_ypred==0):
        #print("Negative Review")
        rev.append("Negative Review")
    else:
        #print("Positive Review")
        rev.append("Positive Review")


for i in range(1000):
    review_classifier(sentimentdata[i])

dataset['Sentiment']=rev
dataset.head()
```

Visualizing the Reviews

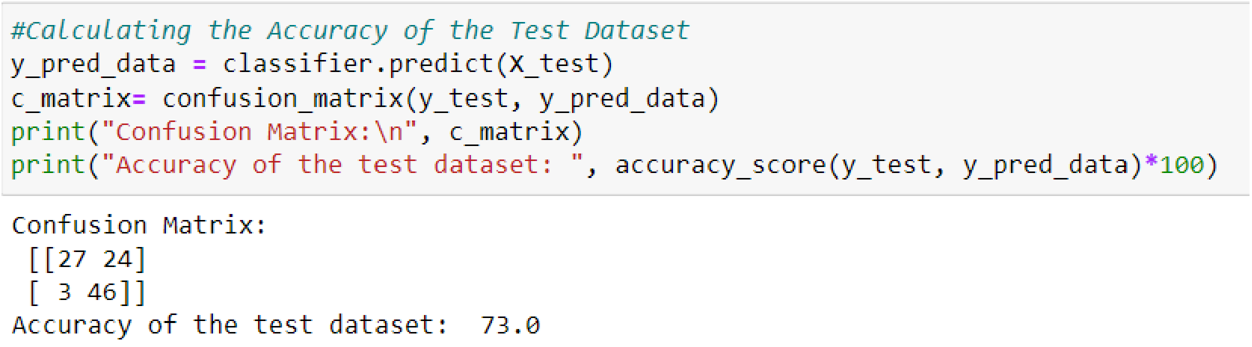


```
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
sentiment_types=['Positive','Negative']
sentiment_count=[Sentiment.pos_sentiment,Sentiment.neg_sentiment]
plt.title("Sentiment Analysis of Reviews\n\n")
ax.pie(sentiment_count,labels=sentiment_types,autopct='%1.1f%%')
plt.show()
```



```
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
sentiment_types=['Positive','Negative']
sentiment_count=[Sentiment.pos_sentiment,Sentiment.neg_sentiment]
plt.title("Sentimental Analysis of Reviews")
ax.bar(sentiment_types,sentiment_count)
plt.show()
```

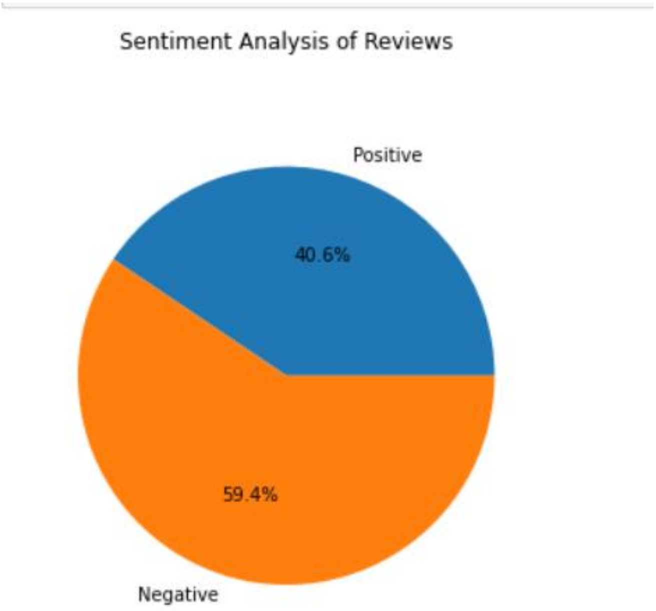
Finding Accuracy of the Model

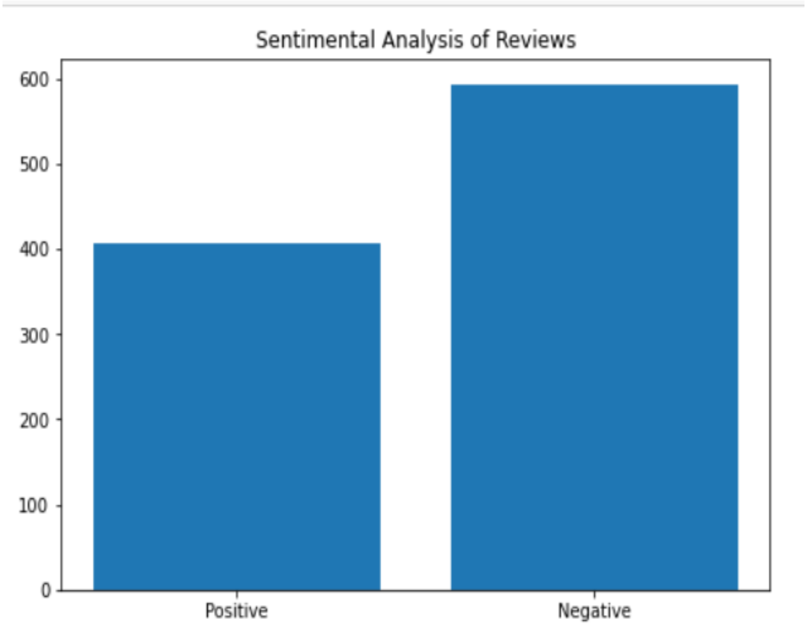


```
#Calculating the Accuracy of the Test Dataset
y_pred_data = classifier.predict(X_test)
c_matrix= confusion_matrix(y_test, y_pred_data)
print("Confusion Matrix:\n", c_matrix)
print("Accuracy of the test dataset: ", accuracy_score(y_test, y_pred_data)*100)
```

```
Confusion Matrix:
 [[27 24]
 [ 3 46]]
Accuracy of the test dataset:  73.0
```

RESULTS


Sentiment Analysis of Reviews
Positive
40.6%
59.4%
Negative


Sentimental Analysis of Reviews
600
500
400
300
200
100
0
Positive
Negative

CONCLUSION

This type of Sentimental Analysis will help to extract opinions and classify sentiments to the reviews that will help to understand the reviews of a restaurant. Although multiple methods can be used for classification, the reason behind choosing Naïve Bayes Classifier inspite of less accuracy is that it showed better performance with the examples we have tried compared to others.

REFERENCES

https://machinelearninggeek.com/analyzing-sentiment-of-restaurant-reviews/

https://www.datacamp.com/tutorial/simplifying-sentiment-analysis-python