ML MAJOR PROJECT

debashish mohapatra
Debashishmohapatra44@gmail.com

Table of Contents

1.Abstract

In the last few years, use of social networking sites has increased tremendously. People use social media platforms to share their views on almost all subjects. These views are in various forms like, blogs, tweets, Facebook posts, online discussion boards, Instagram posts, etc. Sentiment analysis deals with the process of computationally defining and classifying the views expressed in the comment, post or document. Typically, the aim of sentiment analysis is to find out the customer’s attitude towards a product or service. Customers’ feedback is vital for businesses, and social media being a powerful platform, can be used to improve and enhance business opportunities if the feedback on social media can be analyzed timely. Therefore, the focus of this project paper is to analyze the customer reviews.

In this Paper First, it performs sentiment analysis and classifies each comment as positive, negative. Second, by using Countvectorizer techniques, comments are automatically classified according to feedback about food taste, ambiance, service, and value for money. A dataset of around 1000 records was used for training and testing. Two algorithms were used for classification, including Naive Bayes Classifier and Support Vector Machine (SVM) also using direct method and pipelining method. The performance comparison of these algorithms is presented. The best results, that is 79.2% accuracy, were achieved by using pipelining method on SVC and CountVectorizer.

Keywords— Sentiment Analysis, Category-Classification, Naïve Bayes Classifier, CountVectorizer Support Vector Machine, Restaurant Reviews Classification, and Machine Learning.

2.Introduction

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. he second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.

"It is a quite boring restaurant…….. but the foods were good enough. "

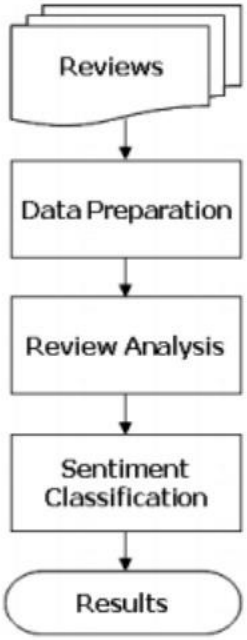
The given line is a restaurant review that states that "it" (the restaurant) is quite boring but the foods were good. Understanding such sentiments require multiple tasks.

Hence, SENTIMENTAL ANALYSIS is a kind of text classification based on Sentimental Orientation (SO) of opinion they contain.

- Firstly, evaluative terms expressing opinions must be extracted from the review.
- Secondly, the SO, or the polarity, of the opinions must be determined.
- Thirdly, the opinion strength, or the intensity, of an opinion should also be determined.
- Finally, the review is classified with respect to sentiment classes, such as Positive and Negative, based on the SO of the opinions it contains.

3.Objective of the Project

- Scrapping product reviews on various restaurants featuring various foods.
- Analyze and categorize review data.
- Analyze sentiment on dataset from document level (review level).
- Categorization or classification of opinion sentiment into-
  - Positive
  - Negative



A typical sentiment analysis model

4.Methodology for Implementation

DATACOLLECTION:

In this study, we utilized the reference dataset "Restaurant Reviews.tsv" for analyzing restaurant reviews. These evaluations are written in simple language and contain some slang and colloquial language. It includes both good and negative evaluations that are unique from one another. Classification models may be taught to understand the user's sentiment. We gathered data from 1,000 reviews of a restaurant and their associated feelings. There are two columns in this dataset. The first column contains text data denoted by the term "Review," while the second column has binary values denoted by the term "Liked." For example, if a review is favorable to the restaurant, as in a positive review, the associated mood is specified as "1". On the other hand, if a review is unfavorable to restaurants, like in a negative review, it is classified as "0."

A.DATA PRE-PROCESSING

Different pre-processing techniques were applied to remove the noise from out data set. It helped to reduce the dimension of our data set, and hence building more accurate classifier, in less time.

The main steps involved are :-

- document pre-processing.
- feature extraction / selection.
- model selection.
- training and testing the classifier.

B. ALGORITHM USED: -

Naïve Bayesian classifier:

The Naïve Bayesian classifier works as follows: Suppose that there exist a set of training data, D, in which each tuple is represented by an n-dimensional feature vector, $X=x_1,x_2,..,x_n$, indicating n measurements made on the tuple from n attributes or features. Assume that there are m classes, C_1,C_2,...,C_m . Given a tuple X, the classifier will predict that X belongs to C_i if and only if: $P(C_i|X)>P(C_j|X)$, where $i,j\in[1,m]$a n d $i\neq j$. $P(C_i |X)$ is computed as:

$$P(C_i|X) = \prod_{k=1}^{n} P(x_k|C_i)$$

Support Vector Machine:

Support vector machine (SVM) is a method for the classification of both linear and nonlinear data. If the data is linearly separable, the SVM searches for the linear optimal separating hyperplane (the linear kernel), which is a decision boundary that separates data of one class from another. Mathematically, a separating hyper plane can be written as: W·X+b=0, where W is a weight vector and W=w1,w2,...,w n. X is a training tuple. b is a scalar. In order to optimize the hyperplane, the problem essentially transforms to the minimization of ‖W‖, which is eventually computed as:

$$\sum_{i=1}^{n} \alpha_i y_i x_i,$$

where α_i are numeric parameters, and y_i are labels based on support vectors, X_i.

That is: if y_i =1 then

$$\sum_{i=1}^{n} w_i x_i \geq 1;$$

if y_i=−1 then

$$\sum_{i=1}^{n} w_i x_i \geq -1.$$

PIPELINE:

A machine learning pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative.

OBJECTIVE:-

1. The main objective of having a proper pipeline for any ML model is to exercise control over it. A well-organised pipeline makes the implementation more flexible. It is like having an exploded view of a computer where you can pick the faulty pieces and replace it- in our case, replacing a chunk of code.

2. The term ML model refers to the model that is created by the training process.

3. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer to be predicted), and it outputs an ML model that captures these patterns.

4. A model can have many dependencies and to store all the components to make sure all features available both offline and online for deployment, all the information is stored in a central repository.

5. A pipeline consists of a sequence of components which are a compilation of computations. Data is sent through these components and is manipulated with the help of computation.

CHALLENGES ASSOCIATED WITH ML PIPELINES:

A typical machine learning pipeline would consist of the following processes:

- Data collection
- Data cleaning
- Feature extraction (labelling and dimensionality reduction)
- Model validation
- Visualisation

*The goal for ML is simple: "**Make faster and better predictions**"*

5.IMPLEMENTATION DETAILS

SPITTING DATASET:

Splitting the data set into two halves is a critical component of the Machine Learning model.

1. Training Set

2.Evaluation Set

Machine Learning's primary objective is to generalize beyond the data examples used to train models. We wish to test the model to determine the quality of its pattern generalization on un-trained data. However, because future instances will have unknown target values and we will be unable to verify the accuracy of our predictions for future instances at this time, we will need to use some of the data for which we already know the answer as a proxy for future data, which will be referred to as our Test Set. When dealing with big datasets, the most common method is to divide them into training and test subsets, often with a ratio of 70-80% for training and 20-30% for testing. The Train test split function, which is loaded from the scikit-learn package, does this splitting randomly.

1.Training Set :

We've incorporated 80% of the data from 1000 reviews into our train set. Both the independent variable (x_train) and the dependent variable (y_train) are known in the training set.

2.Testing Set :

The test set contains 20% of the data from 1000 reviews, where the dependent variable is denoted by (x_test) and the independent variable is denoted by (y_test).

Fitting Algorithm to Training Dataset.

Classification is a type of supervised learning, which occurs when a training set of properly recognized observations is provided. A classifier is an algorithm that performs classification, particularly in a concrete implementation. Occasionally, the word "classifier" refers to the mathematical function implemented by a classification algorithm that categorizes incoming data. It might be challenging to identify an excellent, or even a well-performing, machine learning method for a given dataset. We used trial and error to determine which algorithm produces the best results. We will demonstrate and discuss the effectiveness of MultiBinomial Naive Bayes classifier and Support Vector Classifier(Direct/Pipeline) in predicting whether restaurant reviews are positive or negative.

Prediction of the Result:-

The Machine Learning system utilizes the training data to train models to recognize patterns, and the test data to assess the trained model's prediction ability. The machine learning system measures predictive performance by comparing predicted values on the evaluation data set to real values using a number of criteria. We will use MultiBinomial Naive Bayes and SVM method(Direct/Pipeline) to forecast the test outcome in terms of the value of y_pred.

6.RESULTS

DATASET	CLASSIFIERS	ACCURACY
RESTAURANT REVIEW DATA	SVC (DIRECT METHOD)	72.00%
	SVC (PIPELINE METHOD)	79.20%
	NAÏVE BAYES (DIRECT METHOD)	73.60%
	NAÏVE BAYES (PIPELINE METHOD)	78.40%

The data set described is being used to test the performance of base classifiers. In the proposed approach, first the base classifiers Naïve Bayes, and SVM are constructed individually to obtain a very good generalization performance. According to Table the proposed pipeline model shows significantly larger improvement of classification accuracy than the base classifiers and the results are found to be statistically significant. The proposed ensemble of Naïve Bayes and SVM are shown to be superior to individual approaches for Restaurant review data in terms of Classification accuracy.

7.Performance Analysis

Sentiment Analysis is a new growing field which brings together the topic of natural language processing and machine learning as it is in essence a two class classification of natural language texts. An important feature of sentiment analysis is that it is a cost-sensitive classification.In order to be compared, the methods applied to this field should be all evaluated with the same corpus and within the same cost-sensitive framework. In this paper, the performances of Support Vector Machines (SVM), and Naïve Bayes (NB) techniques are compared using direct and pipeline method for different cost scenarios. The training time complexities of the methods are also evaluated. The results show that SVM has significantly better performance than NB, having acceptable training times. NB gives better results than SVM when the cost is extremely high while in all other cases SVM outperforms NB.

The main difference between the two classification algorithm is that in supervised classification, labeled sample data is needed to train the system whereas, in unsupervised classification, labeled sample data are not required for training the system. The main purpose of these classification techniques is to predict the classes of given data points and on the basis of these data points, specifically, every classifier such as KNN, SVM and logistic regression achieves an accurate result in a specific area with some limitations.

We have created a comparison table above by applying certain classification algorithms and from above we can see that svc with pipelining method gives us the best accuracy that is 79.20%.After that we have NB with 78.4% accuracy with pipeline method and after that we have NB and SVM direct method with accuracy of 73.6 and 72%.

8.CONCLUSION

After analysing a huge corpus of reviews, we conclude that SCV pipeline model outperforms competing methods on nearly every assessment criteria. Fitting and predicting the output takes relatively little time, which is why it may be utilised for real-time classification systems. We propose MultiBinomial Naive Bayes Classifier and SVC Model for sentiment analysis in this thesis. These model may be used to analyse the sentiment of any type of text data, including tweets, brand/product reviews, and vacation destination reviews. This model was tested on a dataset of 1000 restaurant reviews.

Sentiment analysis is critical for customers and service providers alike. Now, in the modern era of the internet and globalisation, both customers and service providers are curious about the general public's view of a certain brand/product/location, etc. It benefits the service provider since it includes a business component, but it also benefits customers because it assists them in selecting the finest product. We have concluded from our project work that the SVC and NB Classifier is an excellent machine learning model for sentiment analysis. It improves sentiment analysis prediction. It is a significant difficulty in the field of sentiment analysis to analyse a sarcastic review/text. A machine might be capable of detecting sarcasm. Future studies might concentrate on sarcastic expressions, which are notoriously difficult to comprehend, both for humans and computers. Another difficult issue is detecting spam content in user reviews. Finally, a review is designated as one of several kinds.

9.FUTURE SCOPE: -

Because the reviews are created with a mixture of real-life review data and sarcastic phrases, review mining is a difficult process. Furthermore, sentiment classification may be done at three levels. Only document-level sentiment classification is performed by this system. Following that, this system uses purely statistical approaches to get an edge. The accuracy of the results might be enhanced by combining the use of semantic resources such as WordNet and SentiWordNet with a statistical method. Furthermore, because this method only examines the sentiment classification of subjective comments, a subjectivity function that can identify whether a statement is an objective or subjective may be added.There are also many ML techniques which we can apply to improve our accuracy such as NN(Neural Network).

10.REFERENCES

[1] Danescu-Niculescu-Mizil, C.; Kossinets, G.; Kleinberg, J.; Lee, L. How opinions are received by online communities: A case study on amazon.com helpfulness votes. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 141–150.

[2] Guo, Y.; Wang, Y.; Wang, C. Exploring the Salient Attributes of Short-Term Rental Experience: An Analysis of Online Reviews from Chinese Guests. Sustainability 2019, 11, 4290.

[3] Pan, Y.; Zhang, J.Q. Born unequal: A study of the helpfulness of user-generated product reviews. J. Retail. 2011, 87, 598–612.

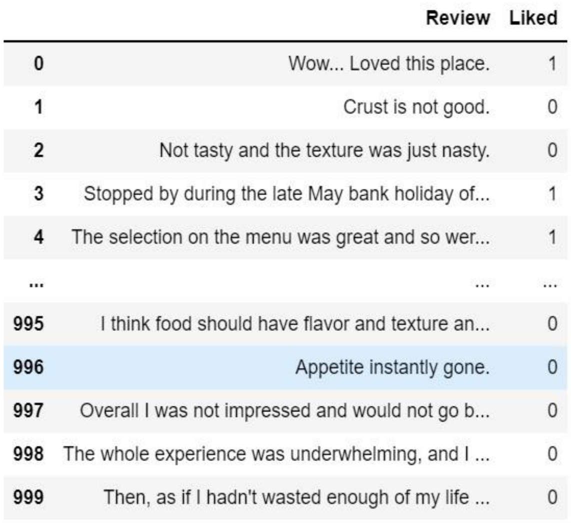
[4] Nam, S.; Ha, C.; Lee, H. Redesigning In-Flight Service with Service Blueprint Based on Text Analysis. Sustainability 2018, 10, 4492.

[5] Kim, W.G.; Li, J.J.; Brymer, R.A. The impact of social media reviews on restaurant performance: The moderating role of excellence certificate. Int. J. Hosp. Manag. 2016, 55, 41–51.

[6] Baek, H.; Ahn, J.; Choi, Y. Helpfulness of online consumer reviews: Readers' objectives and review cues. Int. J. Electron. Commer. 2012, 17, 99–126.

11.SOURCE CODE

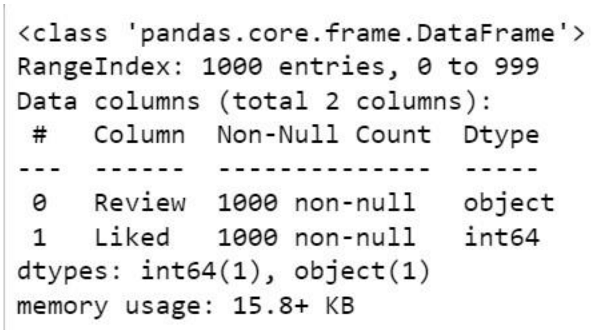
```
#We have to import some basic important libraries before working on the machine learning model.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score


‘’’Next, we have to create a data frame. Download the dataset which was shown previously. And create
using pandas’’’

df=pd.read_table("C:\\Users\\DELL\\OneDrive\\Desktop\\ML\\Restaurant_Reviews.tsv")
df
```



	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1
...	...	...
995	I think food should have flavor and texture an...	0
996	Appetite instantly gone.	0
997	Overall I was not impressed and would not go b...	0
998	The whole experience was underwhelming, and I ...	0
999	Then, as if I hadn't wasted enough of my life ...	0

1000 rows × 2 columns

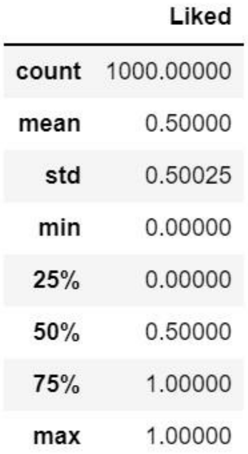
‘’’It will show the output like this. It will show the first five and last five rows and also it will show the number of rows and number of columns in the data frame’’’ df.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Review  1000 non-null   object
 1   Liked   1000 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 15.8+ KB
```

‘’’info() method gives the information about the data frame. I will give the number of columns, column labels, number of non-null entries, the data type of the column, memory usage.’’’

df.describe()

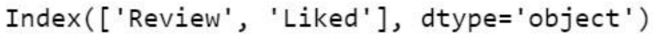


	Liked
count	1000.00000
mean	0.50000
std	0.50025
min	0.00000
25%	0.00000
50%	0.50000
75%	1.00000
max	1.00000

‘’’It will give total count, mean, standard deviation, minimum value, maximum value, 25% of data, 50% of data, 75% of data.’’’

->Let’s see the total columns in the df.

df.columns



```
Index(['Review', 'Liked'], dtype='object')
```

->nunique() method gives the number of unique values in the particular column

df['Liked'].nunique()

2

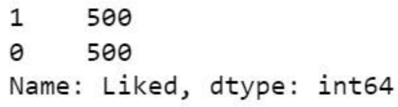
->unique() method gives unique values in the particular column.

print(df['Liked'].unique())

[1 0]

->value_counts() method gives the number of times the particular value repeated in that column through the data frame.

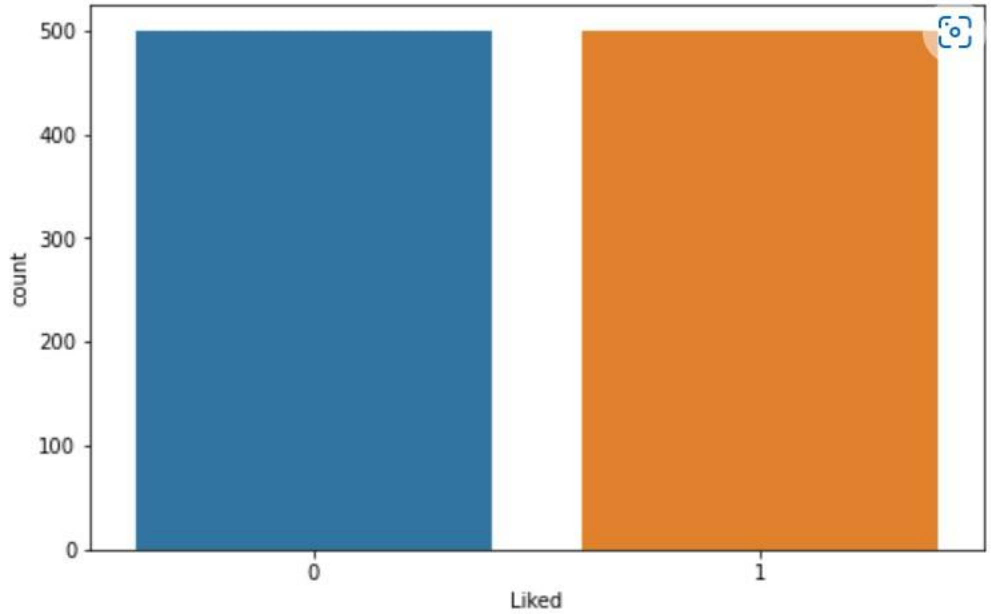
df['Liked'].value_counts()



```
1    500
0    500
Name: Liked, dtype: int64
```

Visualizations

plt.figure(figsize=(8,5))
sns.countplot(x=df.Liked)



“’Here we used the seaborn library to visualize the data frame. This is a count plot where it counts the entries of the column and plots it.’’’

x=df['Review'].values
y=df['Liked'].values

“’Here, X is the input feature that we give to the model, and Y is the output that the model should predict. And coming to our dataset, the Review column is the input that we give, and Liked is going to be predicted by the model.’’’

```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
x_train.shape
```

```
750,
```

```
X_test.shape
```

```
250,
```

‘’’For this, we have to import train_test_split from the scikit learn library. And then whole data frame is divided into four data sets. They are, x_train, x_test, y_train, y_test. Bot x and y are divided into training and test datasets.’’’

Import CountVectorizer

‘’’from the sci-kit learn library we have to import CountVectorizer. And then store it in a variable something like vect with setting stop_wors as “English”.
This count vectorizer transforms the text into a vector based on the count of the words like the number of times the word is repeated in the sentence.’’’

```
vect=CountVectorizer(stop_words='english')
x_train_vect=vect.fit_transform(x_train)
x_test_vect=vect.transform(x_test)
```

‘’’Import Support Vector Classifier(SVC) from Support Vector Machine (SVM) library and assign it to a variable called a model.
The fit method is used to train the model and we have to pass training datasets as arguments in it to train the model.’’’

```
model=SVC()
model.fit(x_train_vect,y_train)
```

‘’’Use predict method to predict the test results. Pass the x variables of the testing dataset in it.
For machine learning models to evaluate it, we use variable methods and all these are in the metrics library and here for support vector classifier(svc), we use accuracy score to evaluate it.
Import accuracy_score from scikit learn metrics library and then pass two arguments to which we have to compare and evaluate. Here predicted dataset and test dataset are taken to evaluate.’’’

```
y_pred=model.predict(x_test_vect)
accuracy_score(y_pred,y_test)
```

```
0.72
```

‘’’Now coming to our model, let’s use the pipeline method. For that import make_pipeline from the pipeline library. And pass CountVectorizer and SVC as arguments into it.
Now again as we know the fit method is used to train the model, train our new model which is made using the pipeline.’’’

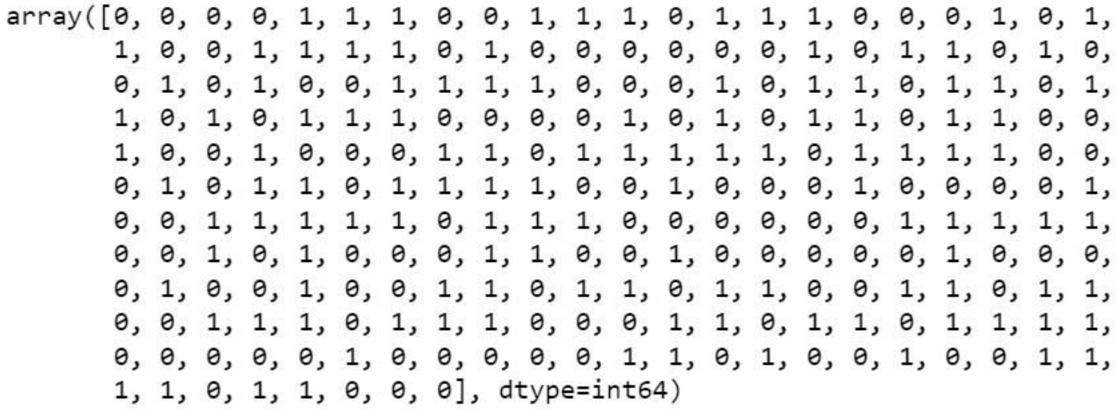
```
text_model=make_pipeline(CountVectorizer(),SVC())
text_model.fit(x_train,y_train)
```

‘’’Similarly predict the results using predict method.’’’

```
y_pred=text_model.predict(x_test)
```

Outcome

Y_pred



```
array([0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 1, 0, 0, 0], dtype=int64)
```

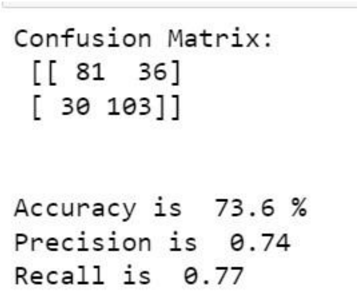
‘’’Let’s evaluate our new model using accuracy_method.’’’

```
accuracy_score(y_pred,y_test)
```

0.792

Applying Naïve Bayes

```
#Apllying Naive Bayes
classifier = MultinomialNB(alpha=0.1)
classifier.fit(x_train_vect, y_train)

# Predicting the Test set results
y_pred = classifier.predict(x_test_vect)

# Making the Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print ("Confusion Matrix:\n",cm)

score1 = accuracy_score(y_test,y_pred)
score2 = precision_score(y_test,y_pred)
score3= recall_score(y_test,y_pred)
print("\n")
print("Accuracy is ",round(score1*100,2),"%")
print("Precision is ",round(score2,2))
print("Recall is ",round(score3,2))
```



```
Confusion Matrix:
 [[ 81  36]
 [ 30 103]]


Accuracy is  73.6 %
Precision is  0.74
Recall is  0.77
```

Applying Pipelining

```
text_model1=make_pipeline(CountVectorizer(),MultinomialNB())
text_model1.fit(x_train,y_train)
y_pred=text_model1.predict(x_test)
accuracy_score(y_pred,y_test)
```

```
0.784
```

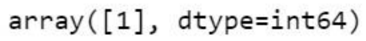
Using Joblib

```
joblib.dump(text_model,'Verzeo_Major_Project')
```

```
['Verzeo_Major_Project']
```

“’As text_model created by using pipeline method on svc gives us the highest accuracy so we consider it to predict new data.’’’

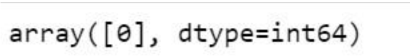
text_model.predict(['hello!!Love Your Food'])



```
array([1], dtype=int64)
```

#Here the review is a positive review and as expected our model predicted 1 for it which means positive.

text_model.predict(["omg!!it was too spice and i asked you don't add too much "])



```
array([0], dtype=int64)
```

# it gave 0 as output which means negative.

Github Link for code:-

https://github.com/debashish-datascience1/debashish-datascience1.git