CV Classification using Natural Language Processing (NLP) and Machine Learning (ML)

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Gaurav Sahu(161004)
Arav Sood(151356)

Under the supervision of

Dr. Kapil Sharma

to

Department of Computer Science & Engineering and Information Technology **Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh**

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled "Securing data using Steganography in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science and Information Technology, Jaypee University of Information Technology Waknaghat is an record of my own work and carried out over a period from August 2019 to December 2019 under the supervision of **Dr. Kapil Sharma**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sood.

Arav Sood (151356)

Sahu

Gaurav Sahu(161004)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Kapil

Dr. Kapil Sharma

Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Dated:

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and indebtedness to **Dr.Kapil Sharma,** Department of Computer Science &Engineering and Information Technology, my supervisors for this work, for there guidance, support, motivation and encouragement throughout the completion of this work. Their eagerness to listen to my problems, their educative comments, their advices for the timely and successful completion of this work has been exemplary.

I am grateful to 0**Prof. Dr. Satya Prakash Ghrera**,Head of Dept. of CSE and IT for his excellent support during our work. We would also like to thank all the professors & other supporting members of the department of Computer Science & Engineering and Information Technology for their generous help in various ways for the completion of this work.

Gaurav Sahu (161004)
Arav Sood (151356)

TABLE OF CONTENT

ABSTRACT

In this day and age it very well may be unmistakably observed that online training has gotten one of the most mainstream advanced education choices. With the advantages of having assortment of projects and courses on the web, from understudies having the option to successfully deal with their chance to get familiar with the materials, and complete the assignments all alone, lower expenses of courses, online courses are picking up fame. An ever increasing number of understudies are getting themselves took on such courses to build their insight and overhaul their abilities. There are numerous online stages that give such great courses. One such notable stage is coursera. Coursera is an overall web based learning stage that offers huge open online courses (MOOC), specializations, and degrees. Coursera works with colleges and different associations to offer online courses, specializations, and degrees in an assortment of subjects, for example, building, information science, AI, arithmetic, business, software engineering, computerized advertising, humanities, medication, science, sociologies, and others.

There are a huge number of surveys given by understudies. At the point when another understudy chooses taking up another course, first thing he does is to experience the audits to know whether this course is directly for him. The objective of the task is to assist understudies with finding the best courses accessible on this stage. We have made a model utilizing Natural Language Processing and Machine Learning based on surveys given by the current understudies which will assist understudies with knowing the ubiquity of the course and will assist the understudy with choosing the correct way for his vocation. It utilizes different AI strategies so as to accomplish its objectives, for example, SVM and Naive Bayes. Assumption examination is likewise performed on the printed surveys we get from the different seminars on Coursera to dissect the conclusions and sentiments present in the audits. The general reason for existing is to examine the surveys of different courses through estimation investigation, learn a wide range of AI calculations, figure out which ones would be exact and proficient, and afterward contrast their outcomes with get the most ideal arrangement.

INTRODUCTION

1.1 INTRODUCTION

Characterization is a significant idea in the field of AI and AI today. It enables PC clients to be increasingly secure from spam assaults, it can enable an AI to decide, and it can drive a vehicle, or even assistance fix malignant growth and numerous different maladies. Grouping is commonly a regulated AI method, which utilizes an extremely enormous informational collection to anticipate where the new example would fall inside the gave informational collection. On account of this undertaking the classifiers see whether an audit is negative or positive or nonpartisan, yet in the event that fitted with various information the classifiers could do some other sort of expectation. We have executed different models, for example, gullible bayes, choice tree calculation, SVM, Random backwoods and thought about the various outcomes.

Technologies Used

- Python - used to program the project.
- Jupyter noytebook : platform to implement our code
- NumPy: Python library used for working with mathematical expressions such as arrays, matrices etc.
- Pandas: python package providing fast and expressive data structures that help import, present and work with data in a structured and organized form.
- Scikit & SKLearn - Library Used to implement algorithms, such as the SVM, Random Forests etc.
- TextBlob: library for processing textual data, and providing a simple API for common NLP tasks such as sentiment analysis, classification, etc.
- Seaborn and Matplotlib : to analyze and visualize the data efficiently and in a better way
- Sentiment Intensity Analyzer: imported from the NLTK toolkit itself
- NLTK: platform for building Python programs to work with human language data.

Steps for Text Classification:

1. Get the data

Examining the information we will be dealing with is significant advance before starting the task .It is essential to get a dataset that satisfies our examination necessity on the specific issue. Before starting it is critical to make a workspace. Need to ensure if our framework has enough stockpiling, need to expel any delicate data before dealing with our task.

2. Explore the Data

Before we begin chipping away at the information it is fundamental to concentrate each property and its qualities.

In the event of regulated learning (on which we are working in this undertaking) it is critical to distinguish the objective traits. We have to expel the clamor in the information. Picturing the information will help us hugely in investigating our dataset.

3. Preparing Data

This is the most significant advance. Information cleaning includes fixing or evacuating anomalies. To improve the presentation it is imperative to drop the traits that give no valuable data to the errand. If there should arise an occurrence of content expelling stop words, tokenization, POS is a portion of the strategies utilized in examining the information. Highlight scaling is another significant advance in setting up the correct information. These are a portion of the information preprocessing strategies we have utilized in our task.

4. Identifying algorithms

Subsequent stage is to break down the suitable calculations which will assist us with arriving at the outcome. A portion of the calculations that we have utilized in our usage are: Gaussian Naïve Bayes, Decision Tree Algorithm, SVM, Random woods and so forth.

Model training and evaluation

One misstep that we as a whole make is the point at which we train and test our model on the equivalent dataset .This is certifiably not a decent practice as we won't have the option to assess how well our model will function in genuine conditions and concealed information and models that over fit to the information will appear to perform better.

It is acceptable practice to part the information in two sections:

1. Training set that the model will be trained on.

2. A test set to evaluate the model’s performance.

Other applications of Text Classification:

- Classification news stories into various kinds, for example, sports news, diversion and so on, or arranging books based on subject in libraries are for the most part uses of content arrangement.

- It can totally be utilized at whatever point there are sure labels to guide to a lot of word-based information. Particularly in advertising, as it has moved from Search Engines to Social Media level supporting surfaces where genuine correspondence among brands and clients occur**.**

- Tagging substance or items utilizing classes is one of the a method to disentangle taking a gander at or to recognize related substance on your site. Raised Platforms, for example, E-business, news organizations, content ace authorities, shared internet composing pages, indexes, and likes can utilize mechanized innovations to arrange and label substance and items.

- Text Classification of substance on the site utilizing labels assists Google with creeping your site effectively which at long last aides in SEO. Likewise, robotizing the substance labels on site and application can make client experience better and assists with making something look or work a similar way every time . There is one more use case for the advertisers which help to investigate and break down labels and watchwords utilized by contenders. We can utilize Text order to robotize and accelerate this procedure.

- Another application is a quicker crisis reaction framework that can be made by grouping alarm discussion via web-based networking media. Individuals in control can screen and characterize crisis circumstance to make a brisk reaction if any such circumstance comes up. This is an instance of exceptionally particular order.

- We watch these days promoting is turning out to be more focused on consistently; mechanized grouping of clients into partners can make advertiser's life basic. Advertisers can screen and characterize clients dependent on how they talk about an item or brand on the web. The classifier can be prepared to recognize advertisers or depreciators. Suggests this application can make brands to serve the accomplices better.

- Other offices/fields like Academia, law experts, social analysts, government, and non-benefit association can likewise utilize content grouping the explanation being, these associations manage a great deal of unstructured content, taking care of the information would be a lot simpler in the event that it gets normalized.

1.2 PROBLEM STATEMENT

There are a huge number of audits given by understudies on each course. At the point when another understudy joins first thing he does is to experience the audits to know whether this course is directly for him. It gets hard to experience each survey. The objective of the task is to assist understudies with finding the best courses accessible on this stage. Understudies can without much of a stretch select their ideal courses dependent on their advantage and fitness by looking at evaluations. The suppositions of the audits can likewise be distinguished to show signs of improvement and faster thought regarding the course. We have made a model utilizing Natural Language Processing and Machine Learning based on audits given by the current understudies which will assist understudies with knowing the prominence of the course and will assist the understudy with choosing the correct way for his vocation. It utilizes different AI calculations to get the outcome, for example, SVM and Naive Bayes, Decision Tree Algorithm, Random backwoods and a couple of others. Literary and Sentiment Analysis is additionally performed to get a more clear thought regarding the content. Our principle reason for existing is to break down the surveys through representation strategies, learn a wide range of AI methods and discover which strategy furnished us with better outcome.

The problem at hand consists of the following task:

To remove the significant data from the dataset, that encourages our model to arrange without any problem.

Natural Language Processing procedures are utilized to help make the extraction of highlights simpler by expelling superfluous and redundant data.

Further, select a fitting Machine Learning model for order and afterward train and test the classifier to robotize the procedure of survey arrangement. We have additionally performed Sentiment Analysis to show the extremity of surveys and dissect the printed data present in it.

1.3 OBJECTIVES

Benefits of process automation:

- Helps understudy pick if the specific course is beneficial for him.
- One can know the most well known courses accessible which will assist understudy with understanding the pattern and will pick admirably.
- Also it will help the course makers to realize which courses are not progressing nicely and will get urged to improve the nature of courses.
- This will ensure that the issues looked by understudies in the past don't re-happen in future.

Machine learning and NLP can be used to achieve the above benefits:

- We will utilize Machine Learning and Natural Language Processing so as to remove extremely significant and required data from the surveys.

- Once the data is removed, we train it on various models and imagine the present patterns.

- Data cleaning is likewise done before utilizing the content for additional examination.

- Used in both the hunt and determination period of the course, recognizing well known courses and furthermore detecting the best course in that specific field.

- Data Visualization can be utilized to show signs of improvement judgment.

Textual and Sentiment Analysis:

- Textual examination is done to portray and decipher the qualities present in the content.

- It gives us a reasonable translation or judgment about the content before we use it for additional examination.

- Perform opinion examination and anticipate whether our content contains positive, negative or impartial conclusion dependent on their extremity scores.

- Perform an average administered order learning task where given a string, we need to classify the content string into predefined classifications.

- Use natural libraries, for example, 'seaborn' and 'matplotlib' to picture our outcomes.

- Compare results dependent on assumptions to choose the best alternative and furthermore select the best AI calculation to get results after arrangement.

1.4 METHODOLOGY

This Coursera’s Course Reviews Classification method follows this framework:

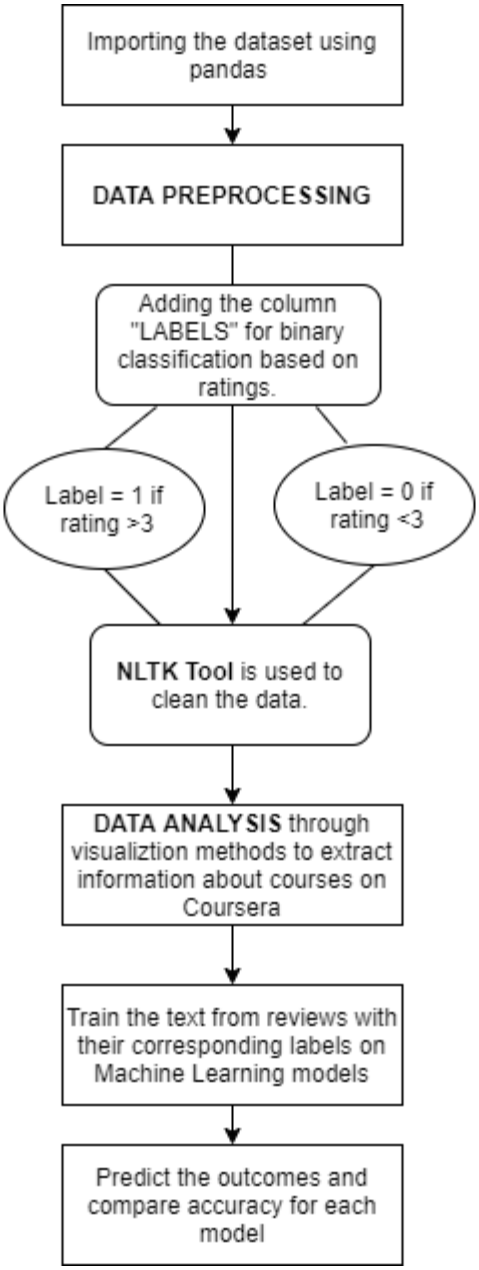

Importing the dataset using pandas
DATA PREPROCESSING
Adding the column "LABELS" for binary classification based on ratings.
Label = 1 if rating >3
Label = 0 if rating <3
NLTK Tool is used to clean the data.
DATA ANALYSIS through visualiztion methods to extract information about courses on Coursera
Train the text from reviews with their corresponding labels on Machine Learning models
Predict the outcomes and compare accuracy for each model

Figure1. Framework used in the project

The main modules are:

a) Textual and Sentiment Analysis

b) Classification of Reviews using supervised Machine Learning algorithms

a) TEXTUAL AND SENTIMENT ANALYSIS:

Natural Language Processing or NLP is a field of Artificial Intelligence that enables the machines to peruse, comprehend and get importance from human dialects. It is a method that manages the collaborations among PCs and people, specifically it encourages us to program PCs to process and break down a lot of characteristic language information. NLP is the part of AI which helps in breaking down any content and furthermore helps in dealing with prescient examination.

- NLP empowers the PC to collaborate with people in a characteristic way.
- It causes the PC to comprehend the human language and get significance from it.
- NLP is relevant in a few tricky from discourse acknowledgment, language interpretation, ordering reports to data extraction

In our undertaking we have utilized NLP utilizing Natural Language Toolkit (NLTK) as follows:

NLTK is an amazing Python bundle that gives a lot of assorted common dialects calculations. It is free, open source, simple to utilize, huge network, and very much archived. NLTK comprises of the most well-known calculations, for example, tokenizing, grammatical form labeling, stemming, slant investigation, theme division, and named element acknowledgment. NLTK encourages the PC to examination, preprocess, and comprehend the composed content.

1. **Remove all the stopwords:** expel all accentuations, commas, and numbers as they don't help much in preparing, they simply increment the length of pack of words.
2. **Stemming:** fundamentally intends to expel the postfix from a word and decrease it to its root word.

3. **Convert each word into its lower case:** There is no point of having same words in various cases.
4. **Tokenization**: It fundamentally intends to diminish a word or character into pieces called token and afterward to dispose of not all that significant characters, for example, comma.
5. **Making the bag of words via sparse matrix:**
    - Without rehashing any word, we have taken all the words from the survey.
    - There is one segment for each word.
    - Rows have audits.
    - If word is there in line of dataset of surveys, at that point the tally of word will be there in line of pack of words under the segment of the word.

The surveys are exposed to Natural Language Processing procedures so as to acquire just important and significant subtleties.

Figure 2 gives a detailed overview of the entire process of NLP used in our project

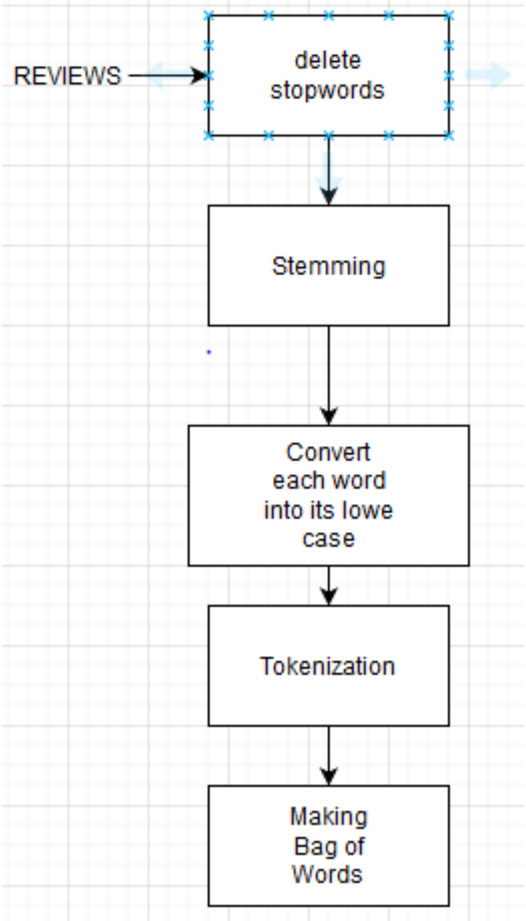


Figure 2 : NLP processes used

Further, sentiment analysis and plots are plotted to get a more intuitive result.

b) CLASSIFICATION USING MACHINE LEARNING ALGORITHMS:

AI calculations have been actualized on the corpus of the words which have been gotten after appropriate content preprocessing and cleaning. Exactnesses have been thought about and as well as can be expected are picked for survey grouping.

LITERATURE SURVEY

As on April 24, 2020, as indicated by [1], Coursera has included around 10 million clients in just the previous month that demonstrates the quick increment of instruction through online stages. Coursera likewise gives budgetary guide to its understudies where an individual is simply required to fill in the motivation behind why he needs money related guide in the fulfillment of the course and Coursera then offers the ideal course to that individual.

In [2], P.Kalaivani et.al have grouped film surveys dependent on estimations and afterward by directed AI calculations. In this paper, they have contemplated film surveys accessible web based utilizing different notion examination draws near. These are then applied to the film audits to recognize the feelings of individuals and judge their feelings in the wake of viewing a specific film. They have analyzed three AI draws near, in particular SVM, Navie Bayes and kNN for the arrangement of surveys dependent on assumptions. Their results show that SVM approach outflanked the Navie Bayes and kNN procedures, and the preparation dataset had an enormous number of audits. With a dataset of around 1000 positive and 1000 negative prepared audits, they had the option to accomplish exactness of about 80% utilizing the SVM classifier.

In [3],Gayatree et al. have improved Rating Predictions utilizing Review Text Content. Their primary commitment is the evaluation of the effect of content determined data in foreseeing the rating of an audit in a proposal framework. They have show that both point and feeling data at the sentence level are valuable data to use in a survey. Their outcomes show that utilizing literary data brings about better broad or customized survey score expectations than those got from the numerical star appraisals given by the clients.

In [4], Sakhare et al. have grouped the content documentation. They have demonstrated that k – closest neighbors is probably the best classifier for content grouping. They have additionally assisted with proposing an Automatic Text Classification System which will include appointing a book archive to a lot of pre-characterized classes. Highlights would then be able to be separated.

SYSTEM DEVELOPMENT

3.1 DATASET DESCRIPTION:

The dataset utilized in the venture has been taken from [5]. This dataset contains 3 columns (or includes) and 140320 rows.

Portrayal of Columns:

Table 1: Dataset Description

COLUMN	DESCRIPTION
CourseraID	The ID of the course accessible on courser
Review	Review as content got by the client for a specific course
Label (Ratings)	Rating given to a specific course in the scope of 1 to 5

For usage of Binary Classification an additional segment to be specific "Marks" has been included:

Label = 1 if Rating > 3, else Label = 0.

3.2 STEPS USED IN THE DEVELOPMENT OF PROJECT:

1. '100k Coursera's Course Reviews Dataset' is downloaded from [1].
2. Once the dataset has been imported, the subsequent stage is to investigate the content.
3. Reviews from Coursera are broke down to discover bits of knowledge and importance from the content.
4. The after plots and diagrams are plotted utilizing matplotlib (Python plotting library) and Seaborn (Python Data Visualization Library):
   - 15 Most Popular Courses on Coursera dependent on CoursersaID
   - Ratings of a specific seminar on Coursera
   - Plot of Labels (what number of Labels of 1 and of 0 are there in the dataset)
   - Listing the most widely recognized words present in the audits.
   - Sentiment Analysis to show the extremity of audits.

- Plot Sentiment versus tally of audits to the comparing supposition
- Plot of Sentimetn versus Number of Reviews
- Plot of Ratings versus Number of Reviews

5. Add the section 'Name' to the dataframe and allocate names dependent on evaluations with the end goal that Label = 1 if Rating > 3, else Label = 0. This will help in the parallel grouping of information.
6. Preprocess the content utilizing the NLTK toolbox.
7. Remove accentuations, stopwords and different characters from the Review segment of the dataframe and make a corpus of content.
8. Use the corpus to make a Bag of Words model with the assistance of Count Vectorizer.
9. Create an inadequate grid to isolate information for preparing on the Machine Learning Model.
10. The information is part arbitrarily into preparing and testing information in the proportion 2:1
11. Features will be separated from the preparation and testing information as per the recurrence and pertinence of the words in the audits.
12. Following Supervised Learning Algorithms are utilized on the information to get results:
    - Decision Tree Classifier
    - SVM
    - Naive Bayes
    - Logistic Regression
    - K-NN (k- Nearest Neighbors)

    A portion of the outfit classifiers utilized are:

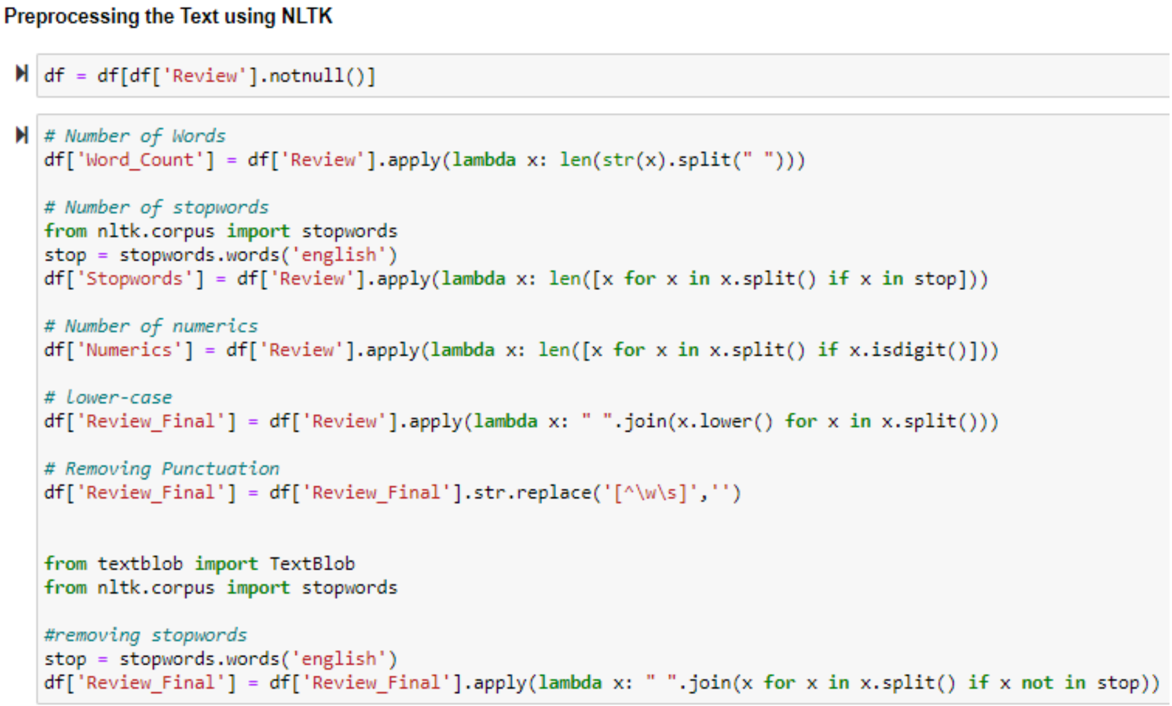
    - Random Forest Classifier
    - XBG Classifier

- Bagging Classifier

13. Different models are prepared, and everyone is thoroughly assessed.
14. The prepared model is deciphered.
15. Classifiers are then utilized on the information to acquire results.
16. ROC plots are plotted to think about the exhibition of these classifiers.

3.3 STEPS IMPLEMENTED IN TEXTUAL AND SENTIMENT ANALYSIS:

1. Preprocessing of text using NLTK Toolkit:



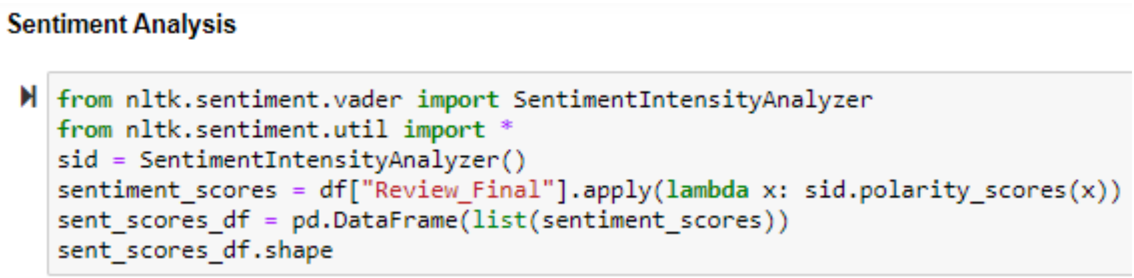
Preprocessing the Text using NLTK

```
df = df[df['Review'].notnull()]
```

```
# Number of Words
df['Word_Count'] = df['Review'].apply(lambda x: len(str(x).split(" ")))

# Number of stopwords
from nltk.corpus import stopwords
stop = stopwords.words('english')
df['Stopwords'] = df['Review'].apply(lambda x: len([x for x in x.split() if x in stop]))

# Number of numerics
df['Numerics'] = df['Review'].apply(lambda x: len([x for x in x.split() if x.isdigit()]))

# Lower-case
df['Review_Final'] = df['Review'].apply(lambda x: " ".join(x.lower() for x in x.split()))

# Removing Punctuation
df['Review_Final'] = df['Review_Final'].str.replace('[^\w\s]','')


from textblob import TextBlob
from nltk.corpus import stopwords

#removing stopwords
stop = stopwords.words('english')
df['Review_Final'] = df['Review_Final'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
```

Figure 3: Text is preprocessed using NLTK toolkit

2. Sentiment Intensity Analyzer is used for SENTIMENT ANALYSIS:

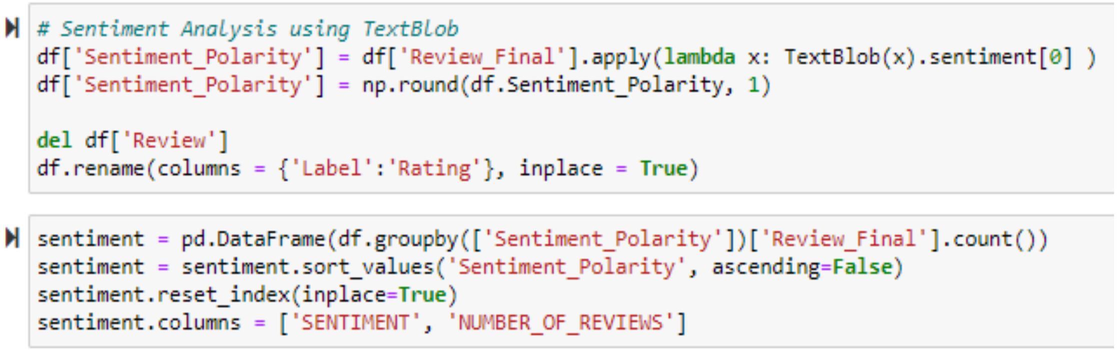


Sentiment Analysis

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.util import *
sid = SentimentIntensityAnalyzer()
sentiment_scores = df["Review_Final"].apply(lambda x: sid.polarity_scores(x))
sent_scores_df = pd.DataFrame(list(sentiment_scores))
sent_scores_df.shape
```

Figure 4: Sentiment Analysis using NLTK Toolkit

3. Sentiment Analysis using TextBlob:

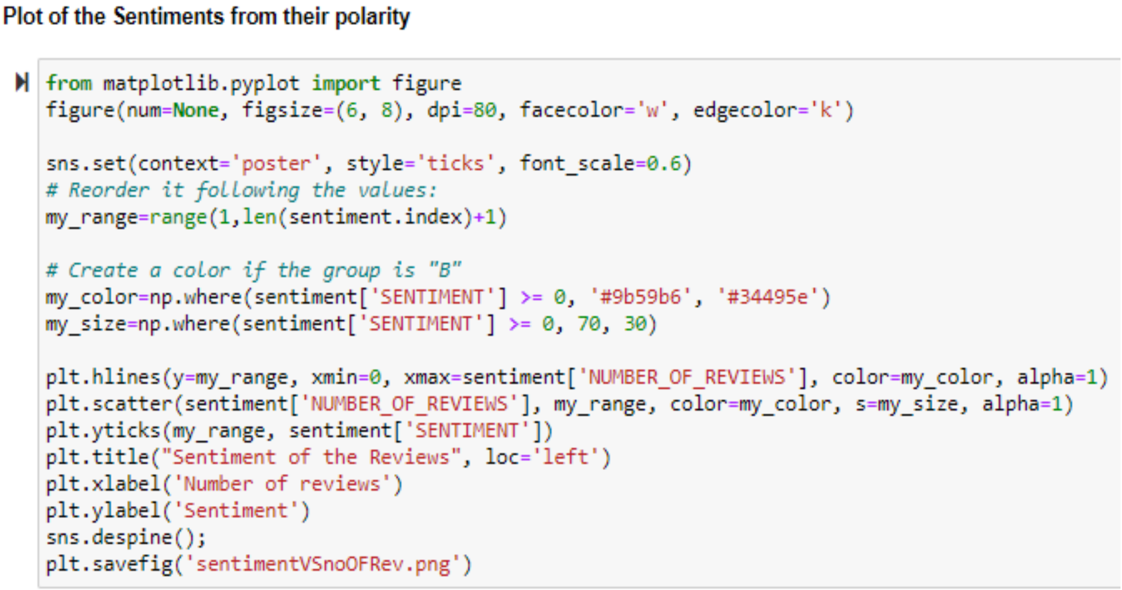


```
# Sentiment Analysis using TextBlob
df['Sentiment_Polarity'] = df['Review_Final'].apply(lambda x: TextBlob(x).sentiment[0] )
df['Sentiment_Polarity'] = np.round(df.Sentiment_Polarity, 1)

del df['Review']
df.rename(columns = {'Label':'Rating'}, inplace = True)
```

```
sentiment = pd.DataFrame(df.groupby(['Sentiment_Polarity'])['Review_Final'].count())
sentiment = sentiment.sort_values('Sentiment_Polarity', ascending=False)
sentiment.reset_index(inplace=True)
sentiment.columns = ['SENTIMENT', 'NUMBER_OF_REVIEWS']
```

Figure 5: Sentiment Analysis using TextBlob

4. Plot of the results:



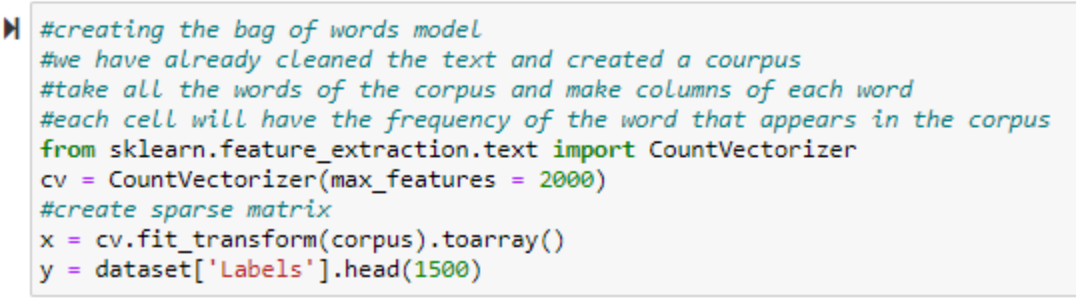
Plot of the Sentiments from their polarity

```
from matplotlib.pyplot import figure
figure(num=None, figsize=(6, 8), dpi=80, facecolor='w', edgecolor='k')

sns.set(context='poster', style='ticks', font_scale=0.6)
# Reorder it following the values:
my_range=range(1,len(sentiment.index)+1)

# Create a color if the group is "B"
my_color=np.where(sentiment['SENTIMENT'] >= 0, '#9b59b6', '#34495e')
my_size=np.where(sentiment['SENTIMENT'] >= 0, 70, 30)

plt.hlines(y=my_range, xmin=0, xmax=sentiment['NUMBER_OF_REVIEWS'], color=my_color, alpha=1)
plt.scatter(sentiment['NUMBER_OF_REVIEWS'], my_range, color=my_color, s=my_size, alpha=1)
plt.yticks(my_range, sentiment['SENTIMENT'])
plt.title("Sentiment of the Reviews", loc='left')
plt.xlabel('Number of reviews')
plt.ylabel('Sentiment')
sns.despine();
plt.savefig('sentimentVSnoOFRev.png')
```

Figure 6: Sentiment Analysis using TextBlob

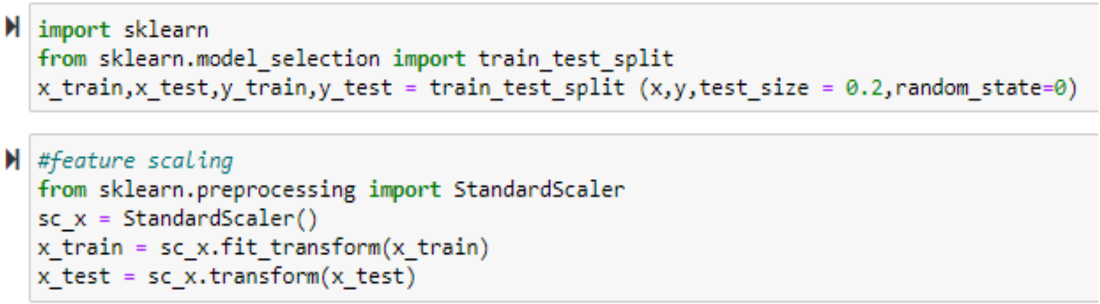
3.2 STEPS FOR IMPLEMENTATION OF ML ALGORITHMS USED:

Creation of a bag of words and Splitting Data:



```
#creating the bag of words model
#we have already cleaned the text and created a courpus
#take all the words of the corpus and make columns of each word
#each cell will have the frequency of the word that appears in the corpus
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 2000)
#create sparse matrix
x = cv.fit_transform(corpus).toarray()
y = dataset['Labels'].head(1500)
```

Figure 7: Creation of Bag of Words from the corpus

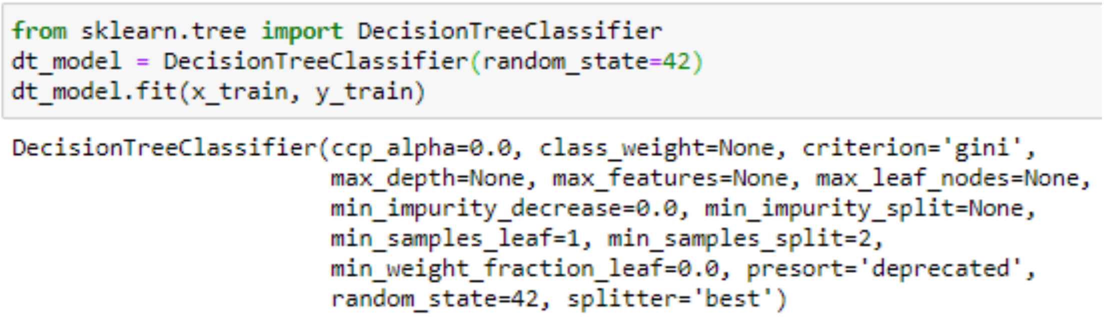


```
import sklearn
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split (x,y,test_size = 0.2,random_state=0)
```

```
#feature scaling
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x_train = sc_x.fit_transform(x_train)
x_test = sc_x.transform(x_test)
```

Figure 8: Splitting the data into training and testing sets

DECISION TREE ALGORITHM:

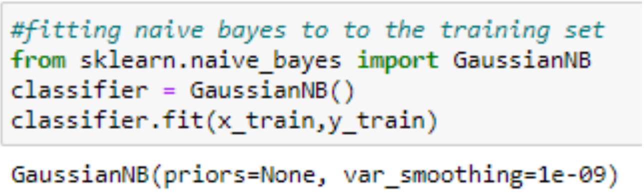


```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(x_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=42, splitter='best')
```

Figure 9: Fitting Decision Tree Classifier on data

NAÏVE BAYES CLASSIFIER:

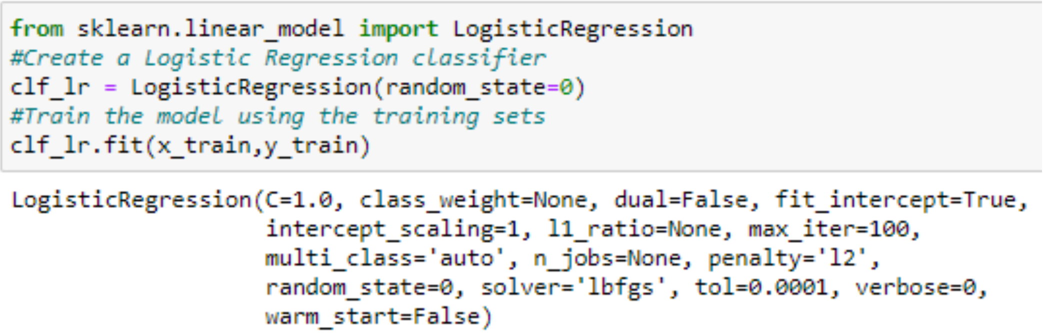


```
#fitting naive bayes to to the training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train,y_train)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

Figure 10: Fitting Naïve Bayes Classifier on data

LOGISTIC REGRESSION:

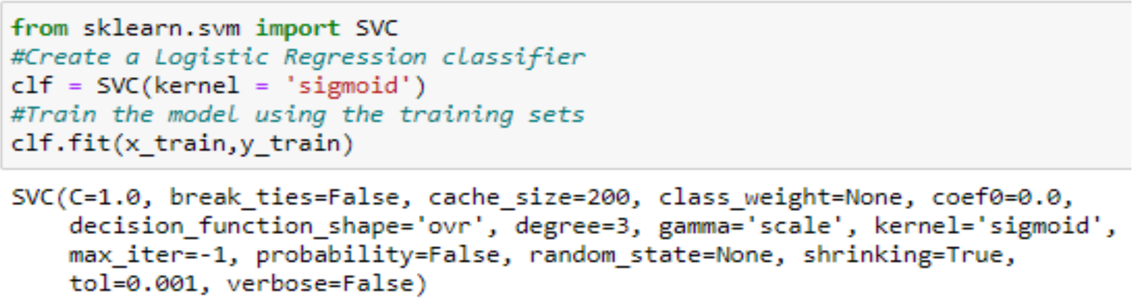


```
from sklearn.linear_model import LogisticRegression
#Create a Logistic Regression classifier
clf_lr = LogisticRegression(random_state=0)
#Train the model using the training sets
clf_lr.fit(x_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

Figure 11: Fitting Logistic Regression Classifier on data

SUPPORT VECTOR MACHINES (SVMs):

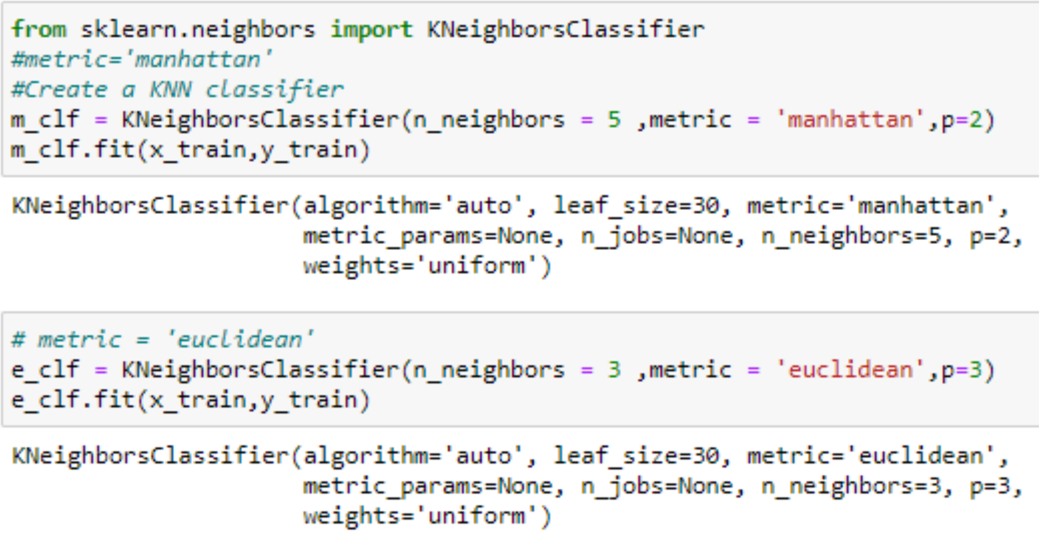


```
from sklearn.svm import SVC
#Create a Logistic Regression classifier
clf = SVC(kernel = 'sigmoid')
#Train the model using the training sets
clf.fit(x_train,y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 12: Fitting SVM Classifier on data

K-NN ALGORITHM:



```
from sklearn.neighbors import KNeighborsClassifier
#metric='manhattan'
#Create a KNN classifier
m_clf = KNeighborsClassifier(n_neighbors = 5 ,metric = 'manhattan',p=2)
m_clf.fit(x_train,y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='manhattan',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

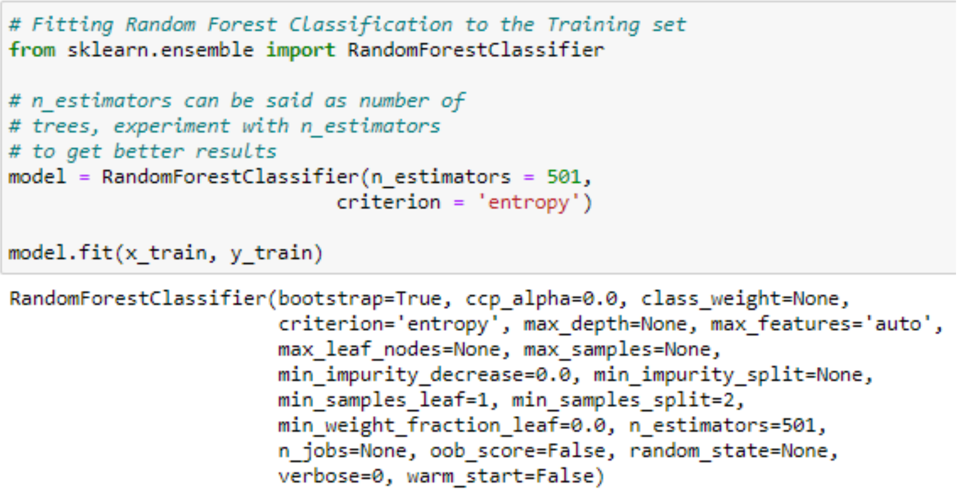
```
# metric = 'euclidean'
e_clf = KNeighborsClassifier(n_neighbors = 3 ,metric = 'euclidean',p=3)
e_clf.fit(x_train,y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=3,
                     weights='uniform')
```

Figure 13: Fitting k-NN Classifier on data

ENSEMBLE ALGORITHMS FOR CLASSICIFATION:

RANDOM FOREST:

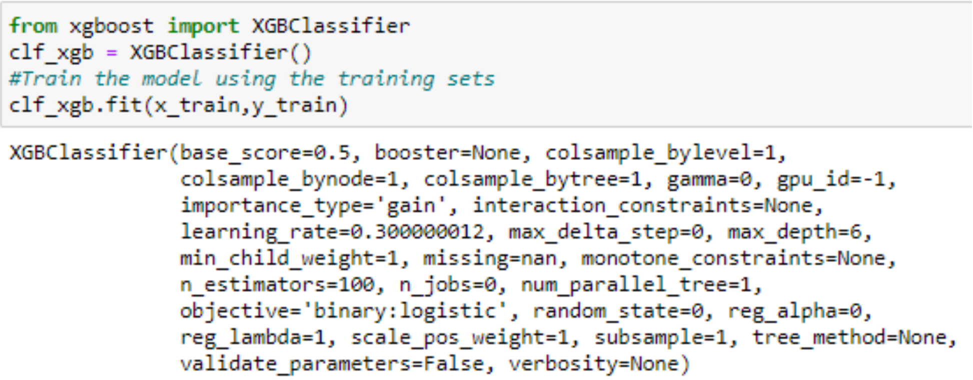


```
# Fitting Random Forest Classification to the Training set
from sklearn.ensemble import RandomForestClassifier

# n_estimators can be said as number of
# trees, experiment with n_estimators
# to get better results
model = RandomForestClassifier(n_estimators = 501,
                            criterion = 'entropy')

model.fit(x_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=501,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figure 14: Fitting Random Forest Classifier on data

XGB CLASSIFIER:

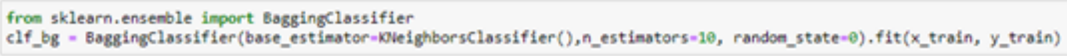


```
from xgboost import XGBClassifier
clf_xgb = XGBClassifier()
#Train the model using the training sets
clf_xgb.fit(x_train,y_train)
```

```
XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None,
              validate_parameters=False, verbosity=None)
```

Figure 13: Fitting XGB Classifier on data

BAGGING CLASSIFIER:



```
from sklearn.ensemble import BaggingClassifier
clf_bg = BaggingClassifier(base_estimator=KNeighborsClassifier(),n_estimators=10, random_state=0).fit(x_train, y_train)
```

Figure 14: Fitting Bagging Classifier on data

PERFORMANCE ANALYSIS

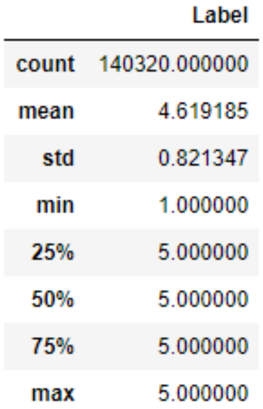
4.1 DATA ANALYSIS:

Information present in our dataset is investigated to get legitimate experiences and results in the wake of assessing it. The content present in the audits of different courses is utilized to perform assumption investigation to get a thought regarding the slant of these writings and order them as needs be. The information is likewise broke down dependent on the evaluations and the course IDs. This is finished utilizing plots which makes it simpler for us to dissect and envision the significance of the information present in this dataset.

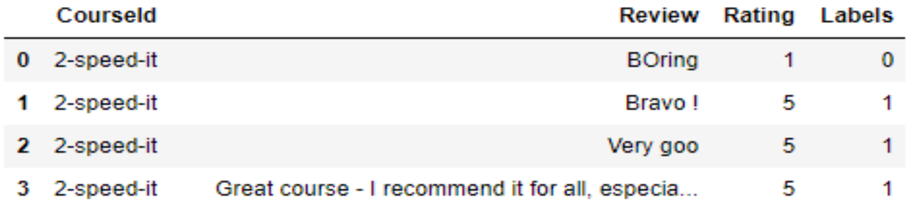
4.1.1 TEXTUAL ANALYSIS:

Following are the outputs attained at various stages of textual analysis:

4.1.1.1DESCRIPTION OF DATASET:



	Label
count	140320.000000
mean	4.619185
std	0.821347
min	1.000000
25%	5.000000
50%	5.000000
75%	5.000000
max	5.000000



	CourseId	Review	Rating	Labels
0	2-speed-it	BOring	1	0
1	2-speed-it	Bravo !	5	1
2	2-speed-it	Very goo	5	1
3	2-speed-it	Great course - I recommend it for all, especia...	5	1

Figure 15,16 : Figures presenting the descriptions of dataset

4.1.1.2 PLOT FOR 15 MOST POPULAR COURSES ON COUSERA:

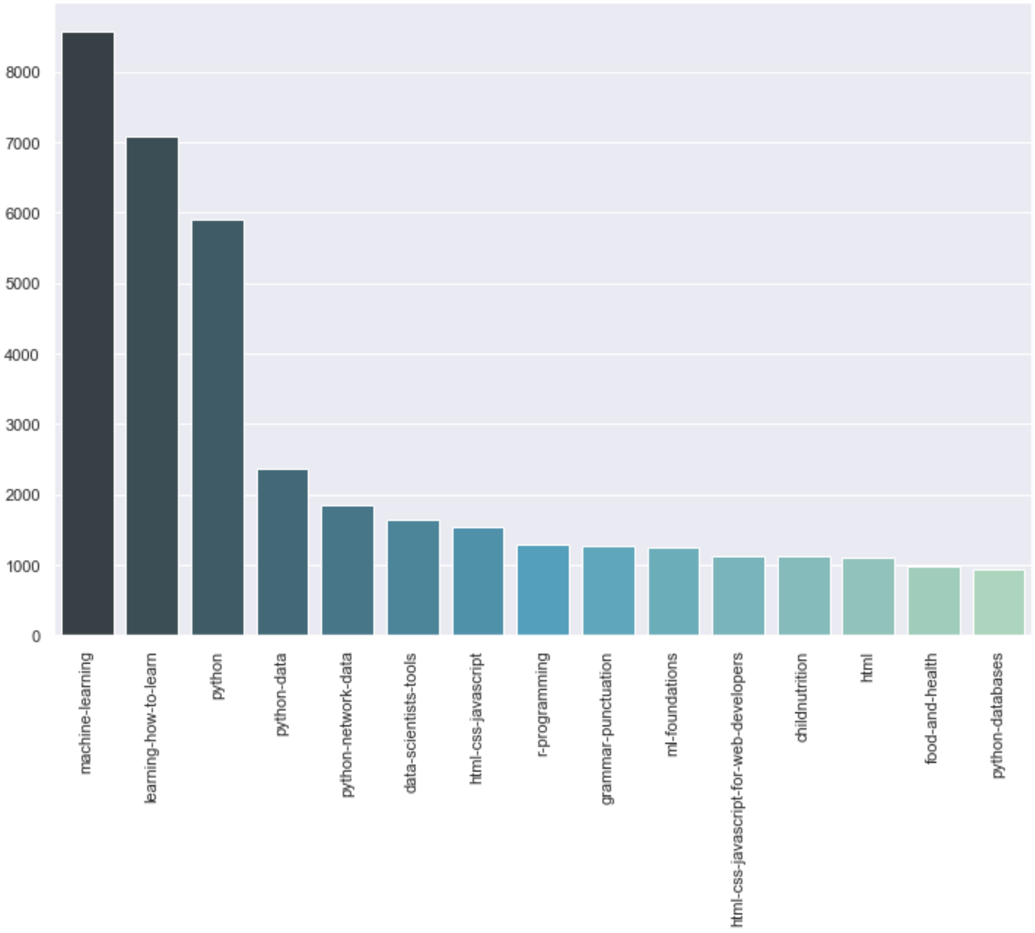
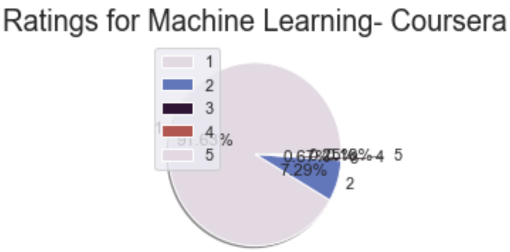
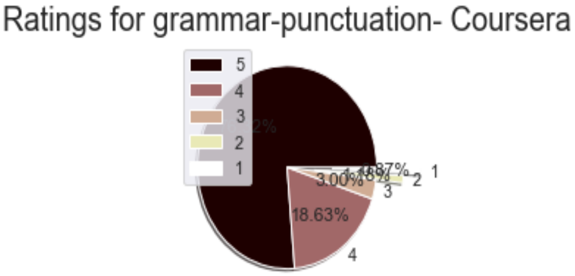


Figure 17: Plot for 15 most popular courses

4.1.1.3 RATINGS OF PARTICULAR COURSES:


Ratings for Machine Learning- Coursera


Ratings for grammar-punctuation- Coursera

Figures 18,19 : Pie chart showing ratings of particular courses

4.1.1.4 PLOT OF LABELS vs COUNT:

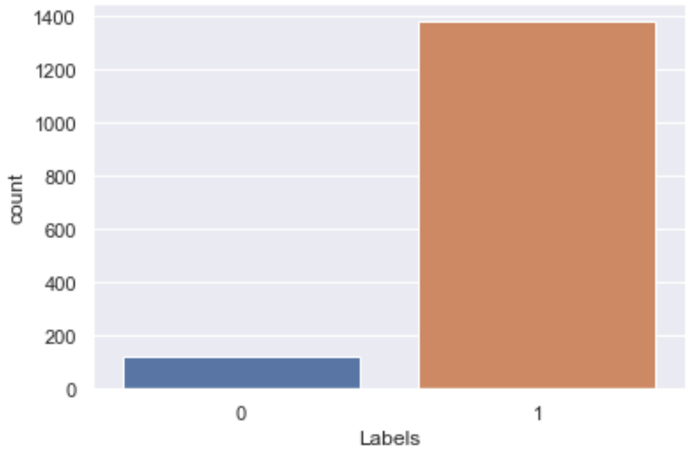
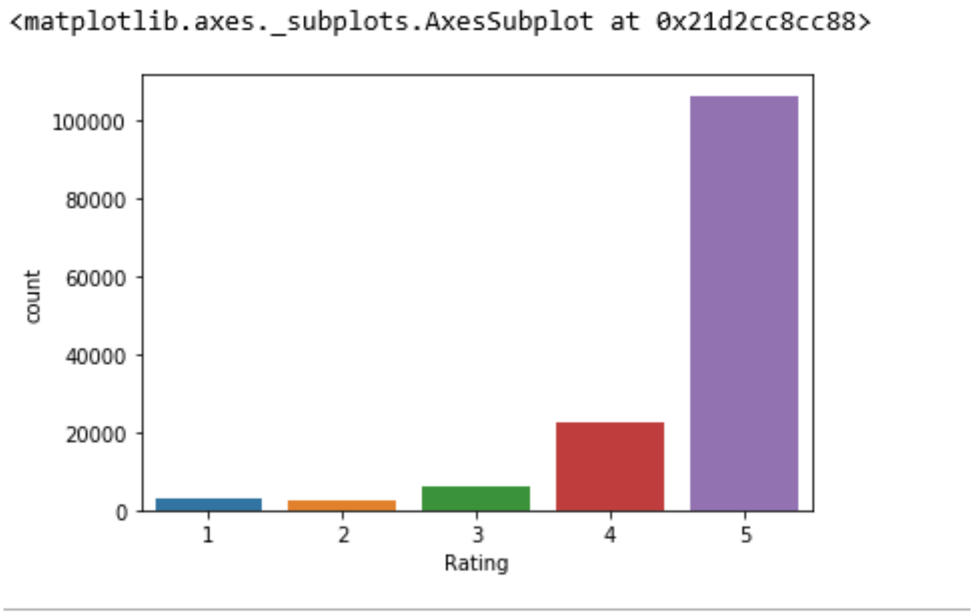


Figure 20: Plot of labels vs count

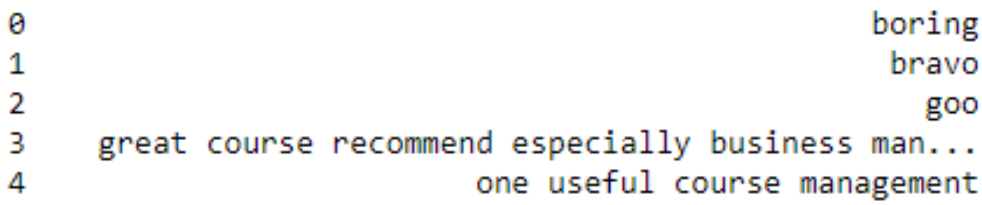
4.1.1.5 PLOT OF RATING vs COUNT:



```
<matplotlib.axes._subplots.AxesSubplot at 0x21d2cc8cc88>
```

Figure 21: Plot of Count vs Ratings

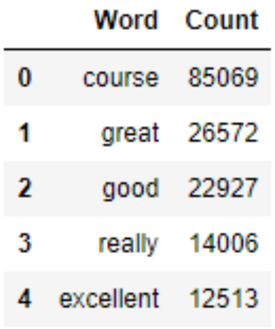
4.1.1.6 TEXT AFTER PREPROCESSING:



```
0                                                boring
1                                                 bravo
2                                                   goo
3    great course recommend especially business man...
4                          one useful course management
```

Figure 22: Text after being preprocessed

4.1.1.7 MOST COMMON WORDS PRESENT:

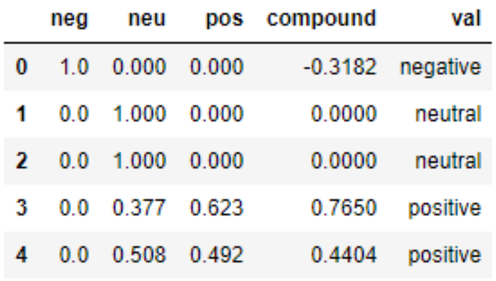


	Word	Count
0	course	85069
1	great	26572
2	good	22927
3	really	14006
4	excellent	12513

Figure 23: Count of most common words

4.1.2 SENTIMENT ANALYSIS:

4.1.2.1 SENTIMENT SCORES:



	neg	neu	pos	compound	val
0	1.0	0.000	0.000	-0.3182	negative
1	0.0	1.000	0.000	0.0000	neutral
2	0.0	1.000	0.000	0.0000	neutral
3	0.0	0.377	0.623	0.7650	positive
4	0.0	0.508	0.492	0.4404	positive

Figure 24: Sentiment scores and their corresponding values

4.1.2.2 PLOT OF COUNT vs SENTIMENT:

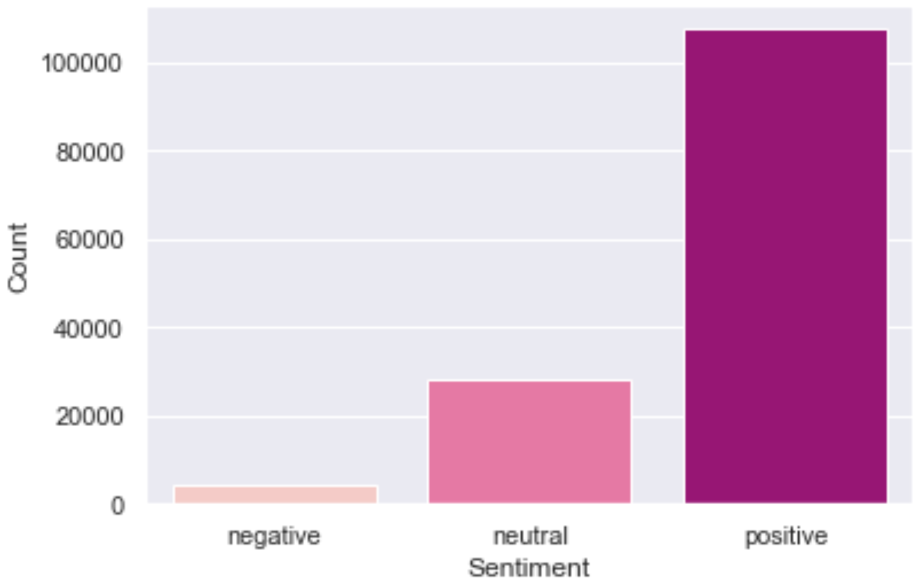


Figure 25: Plot of count vs sentiment

4.1.2.3 PIE CHART of SENTIMENTS:

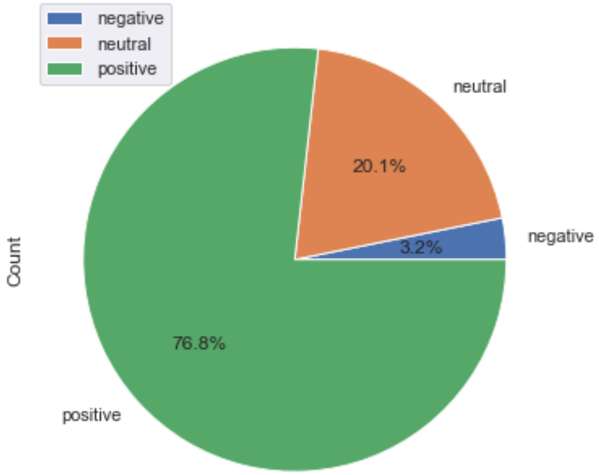
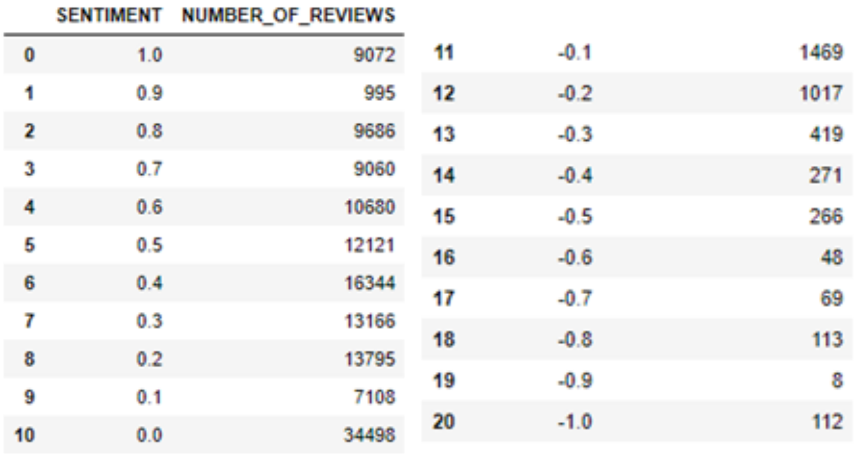


Figure 26: Pie chart of the percentage of sentiments present in the reviews

4.1.2.4 NUMBER OF REVIEWS CORRESPONDING TO THE POLARITY OF SENTIMENTS:



	SENTIMENT	NUMBER_OF_REVIEWS
0	1.0	9072
1	0.9	995
2	0.8	9686
3	0.7	9060
4	0.6	10680
5	0.5	12121
6	0.4	16344
7	0.3	13166
8	0.2	13795
9	0.1	7108
10	0.0	34498
11	-0.1	1469
12	-0.2	1017
13	-0.3	419
14	-0.4	271
15	-0.5	266
16	-0.6	48
17	-0.7	69
18	-0.8	113
19	-0.9	8
20	-1.0	112

Figure 27: Number of reviews corresponding to polarities

4.1.2.5 PLOT OF RATINGS CORRESPONDING TO THE NUMBER OF REVIEWS:

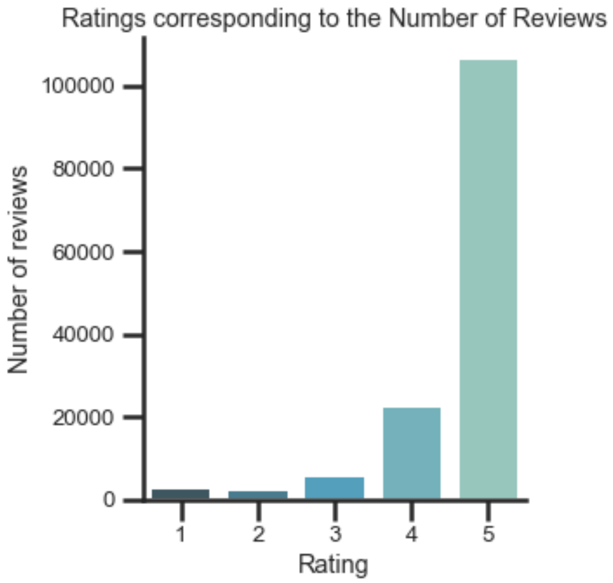


Figure 28: Plot of ratings vs number of reviews

4.1.2.6 PLOT OF SENTIMENT OF THE REVIEWS:

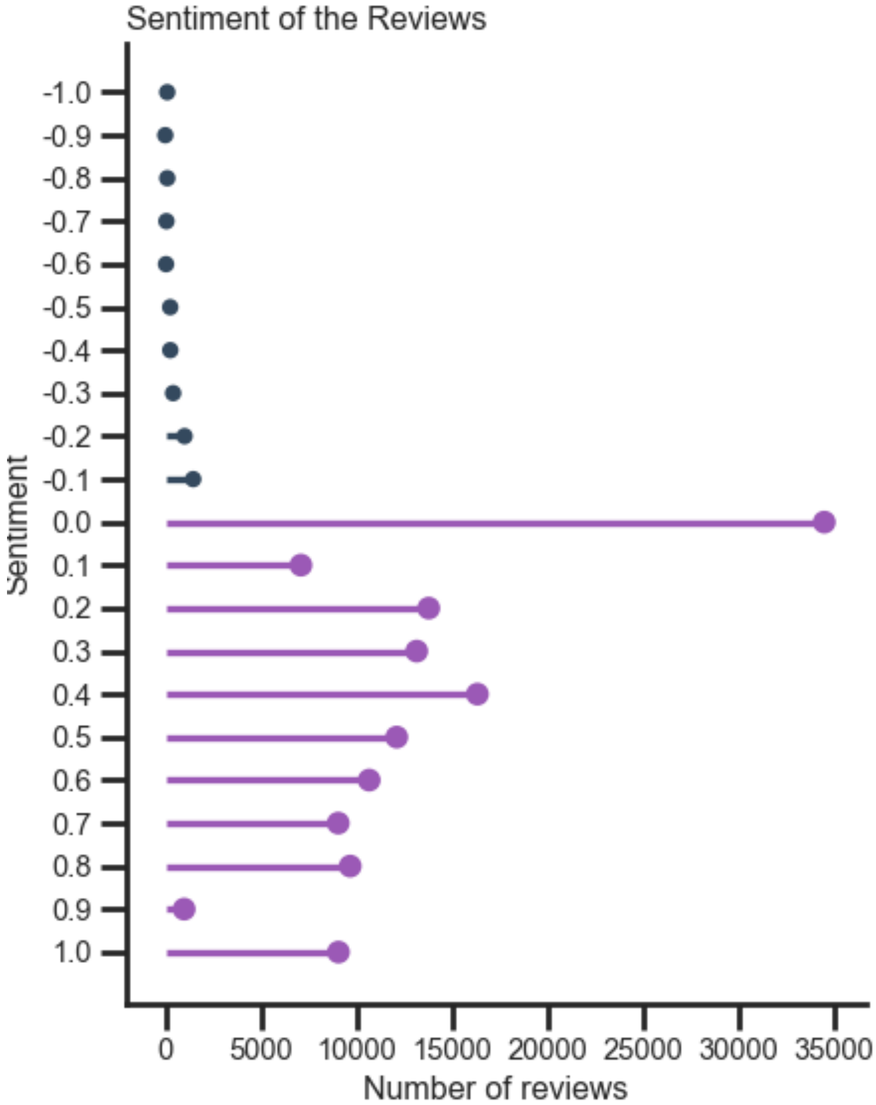
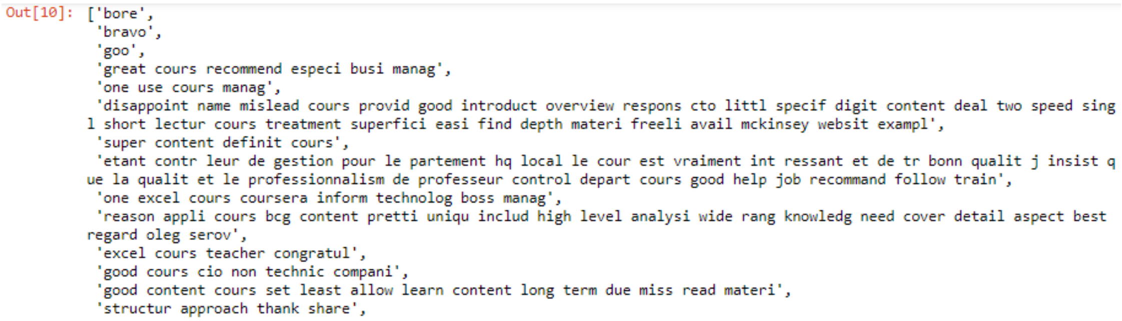


Figure 29: Plot of sentiment vs number of reviews

4.2 CLASSIFIACTION USING MACHINE LEARNING ALGORITHMS:

4.2.1 CORPUS OF WORDS AFTER TEXT PREPROCESSING:



```
Out[10]: ['bore',
          'bravo',
          'goo',
          'great cours recommend especi busi manag',
          'one use cours manag',
          'disappoint name mislead cours provid good introduct overview respons cto littl specif digit content deal two speed sing
         l short lectur cours treatment superfici easi find depth materi freeli avail mckinsey websit exampl',
          'super content definit cours',
          'etant contr leur de gestion pour le partement hq local le cour est vraiment int ressant et de tr bonn qualit j insist q
         ue la qualit et le professionnalism de professeur control depart cours good help job recommand follow train',
          'one excel cours coursera inform technolog boss manag',
          'reason appli cours bcg content pretti uniqu includ high level analysi wide rang knowledg need cover detail aspect best
         regard oleg serov',
          'excel cours teacher congratul',
          'good cours cio non technic compani',
          'good content cours set least allow learn content long term due miss read materi',
          'structur approach thank share',
```

Figure 30: Corpus of words obtained

4.2.2 RESULTS FROM VARIOUS CLASSIFICATION MODELS:

Following are the results obtained after training the models on text from first 1500 reviews:

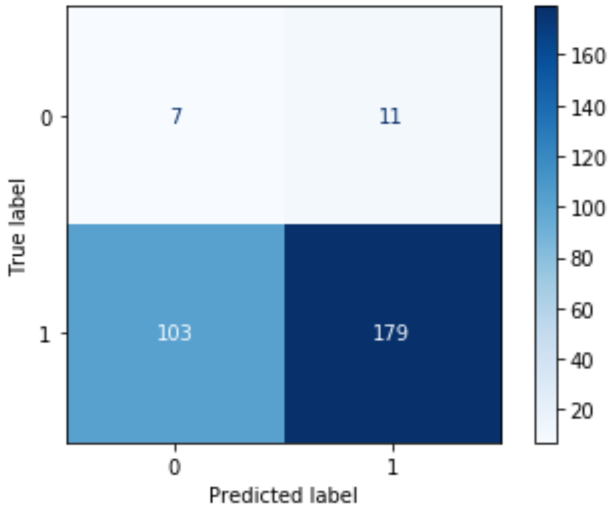
Table 2: Accuracy results from various classification models

	CLASSIFIER	**ACCURACY**
1	Naïve Bayes Classifier	62
2	SVM Classifier	93.3
3	Decision Tree Classifier	89.6
4	Logistic Regression Classifier	93.6
5	k-NN Classifier : Euclidean Distance	94.6
6	k-NN Classifier : Manhattan Distance	94.3
7	Random Forest Classifier	94
8	Bagging Classifier	93.3

9	XGB Classifier	93.3

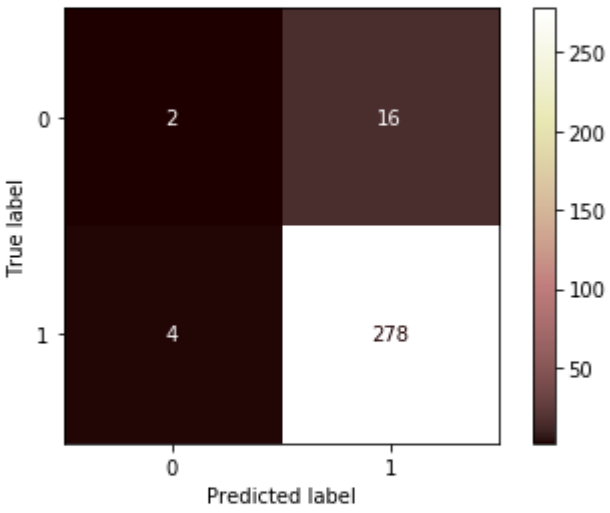
4.2.3 CONFUSION MATRICES AND ACCURACY OF VARIOUS CLASSIFIERS:

- **Naïve Bayes Classifier**



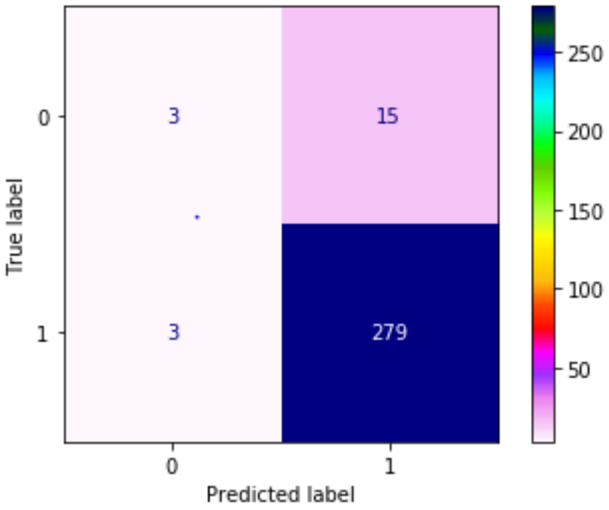
accuracy: 62.0

- **SVM Classifier**



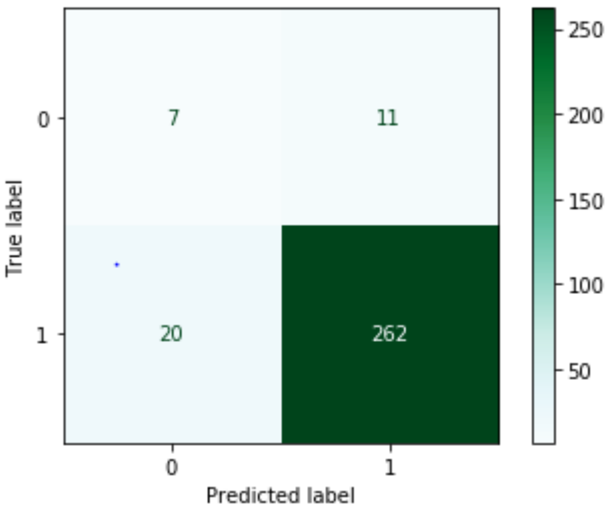
accuracy: 93.333

- **Random Forest Classifier**



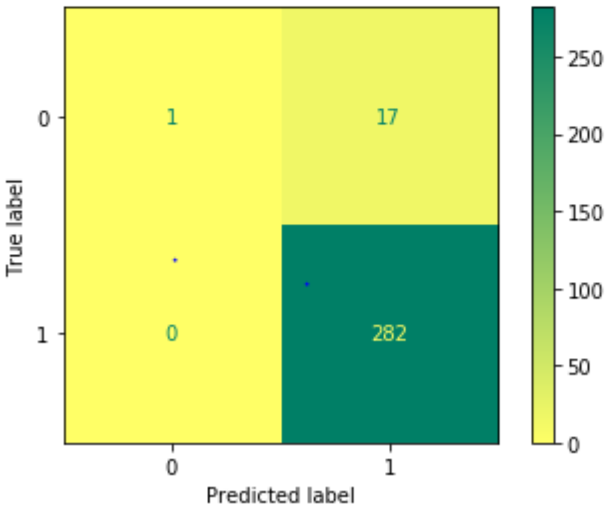
accuracy: 94.0

- **Decision Tree Classifier**



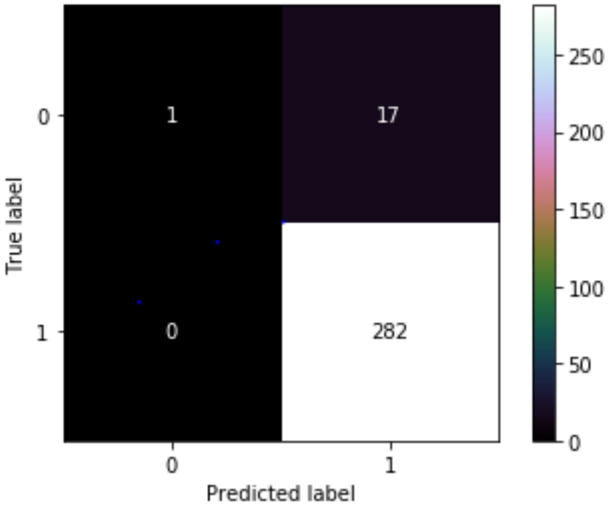
accuracy: 89.6

- **k-NN Classifier : Euclidean Distance**



accuracy =93.4

- **BaggingClassifier**



accuracy =93.3

4.2.4 ROC CURVE FOR LINEAR CLASSIFIERS:

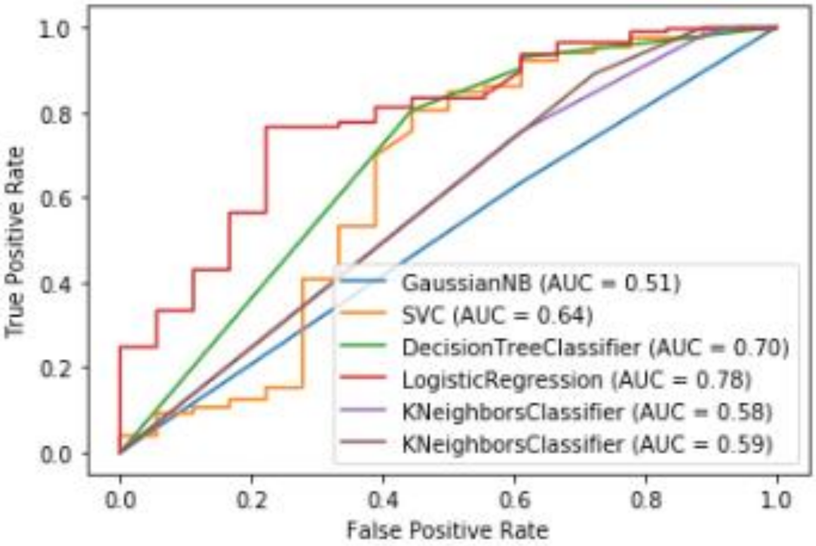


Figure 31: ROC Curve for linear classifiers

4.2.5 ROC CURVE FOR ENSEMBLE CLASSIFIERS:

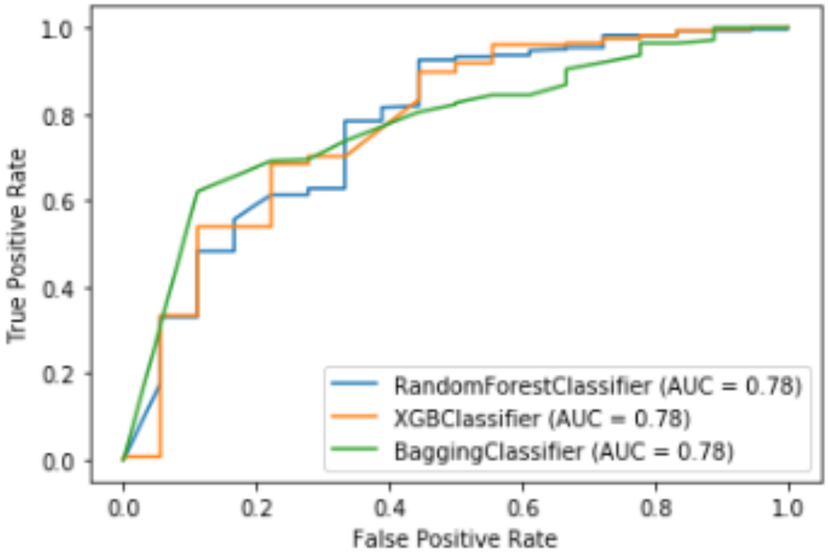


Figure 32: ROC Curve for ensemble classifiers

CONCLUSION

5.1 CONCLUSION:

We have effectively broke down and characterized surveys acquired from the different courses present on Coursera. This is done dependent on the appraisals of the audits. Estimation investigation is performed on the content to assist us with settling on better choices about the course and its audits. It can assist anybody with picking a seminar on Coursera dependent on their advantage and skill by improving judgment about the course dependent on surveys from past enlistments. Instinctive plots have been plotted by investigating the information and notions present in them.

We have additionally performed grouping of these audits utilizing managed AI calculations to arrange the surveys as positive or negative. Observational outcomes show that the most noteworthy precision of 94.6% is accomplished by utilizing k-NN Classifier (utilizing

Euclidien Distance). Irregular woodland classifier and Bagging classifier additionally show high correctnesses of about 93%.

5.2 FUTURE SCOPE:

- We can attempt other content pre preparing procedures, for example, stemming and lemmatizing that can give us better type of content to take a shot at.
- We purpose to deliver more noteworthy exactness utilizing various classifiers and train the model with a bigger dataset. We can attempt other characterization models and we can likewise apply neural systems which may create better outcomes.
- Future work may likewise include prescribing a specific course to an understudy based on his necessities.

It is vital that extra and most recent datasets ought to be thought of and made accessible for the assessment of various order issues as the data development in the ongoing innovation is stretching out to statures past presumption.

These enhancements will help to further improve the performance.

REFERENCES

[1]https://www.cnbc.com/2020/04/24/coursera-offers-unemployed-workers-thousands-of-free-online-courses.html

[2] P.Kalaivani, Dr. K.L.Shunmuganathan, “SENTIMENT CLASSIFICATION OF MOVIE REVIEWS BY SUPERVISED MACHINE LEARNING APPROACHES”, Indian Journal of Computer Science and Engineering (IJCSE) Vol. 4 No.4 Aug-Sep 2013

[3] Gayatree Ganu, Noemie Elhadad, Amelie Marian, “Beyond the Stars: Improving Rating Predictions using Review Text Content”, citeseerx.ist.psu.edu

[4] Vrusha U.Suryawanshi, Pallavi Bogawar, Pallavi Patil,Priya Meshram, Komal Yadav, Prof. Nikhil S. Sakhare,“Automatic Text Classification System”,International

Journal of Advanced Research in Computer Engineering &Technology (IJARCET) Volume 4 Issue 2, February 2015

[5] https://www.kaggle.com/septa97/100k-courseras-course-reviews-dataset