VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE4022 – NATURAL LANGUAGE PROCESSING

IMPLEMENTATION OF SENTIMENTAL ANALYSIS OF ONLINE MOOCS REVIEWS USING ML ALGORITHMS

PROJECT REPORT

Submitted By:

RISHIKESH KUMAR(17BCE0677)
NAITIK TAILOR (17BCE2292)
DEVESH GOYAL (17BIT0103)
SHIV NARAYAN SINGH (17BIT0341)

SLOT: A1+TA1

Under the guidance of

Dr. Mangayarkarasi R , SITE

VIT, Vellore

WINTER SEMESTER 2019-20

ABSTRACT

Sentimental analysis has aroused the interest of many researchers in recent years, since subjective texts are useful for many applications. In particular, sentiment analysis on online reviews has become a hot research field. Studies on sentimental analysis mainly focus on framework and lexicon construction, feature extraction, and polarity determination. It has a wide scope and thus it can be applied over varied fields which feature subjective texts, one such implementation can be done with the social media, dealing with the public behavior and opinion trends.

In this paper we take into account the analysis of students' trending behavior towards a course by mining collective sentiment from forum posts or reviews. We hence witness a Correlation between sentiment analysis obtained from student reviews and concerned student activities with the course. This paper takes into account the coursera review dataset for implementation and testing. And will make an conclusion, whether to remove or improve the course based on the results obtained.

INTRODUCTION

In this paper, we perform sentiment analysis on feedback provided by students on educational websites which enables organizations to monitor the quality of courses and mentors, which helps them in improving their standards. We incorporate the popular machine-learning techniques such as **Naïve Bayes Classifier, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Perceptron Model** which is one of the most prominent supervised learning rule incorporated into the field of mining reviews from educational websites.

Because of easy and faster implementation design of Machine Learning Algorithms, it is taken as baseline for developing classifier in any application. The Underlying concept of one of the famous classifier is Naive Bayes technique is to calculate the probabilities of classes for a given dataset by making use of joint probabilities of terms and classes. And it makes an assumption that each term is independent to make the classification more efficient. But these assumptions adversely affect the quality of results.

Sentiment analysis on moocs is done in order to increase quality of moocs by monitoring student's reviews. In addition to taking the feedback from students at the end of the course, we even take feedback from students who are currently pursuing the course. This analysis is done to identify the polarity of the reviews, to know if the student is feeling comfortable with the course or not. Feedback is collected from the comments section on the online course page. After collecting these feedback, we perform data cleansing to remove stop words etc. And we apply machine learning techniques for classify and index the data. Then we perform analysis on this indexed data and provide results

COURSERA DATASET:

Student's review is an important The dataset on which the analysis has been made is taken from a course hosted by the Coursera.org. The dataset extracted from this course offers a wide range of topics. The extracted dataset consists of reviews from the course "Neural Networks and Deep Learning". The results obtained upon analysing these reviews and applying Machine learning algorithms are presented in results section.

LITERATURE SURVEY

S.No.	Journal Name	Authors and Publisher	Methodology	Challenges	Outcomes
1.	An Effective Method of Predicting the Polarity of Airline Tweets using sentimental Analysis	IEEE, 2018 Adarsh M J, Dr. Pushpa Ravikumar	Simple approach of sentiment ananlysis	The problem with the proposed methodology of this paper is that if some customer has done a sarcastic tweet then this methodology will give wrong score which means that there will be a wrong prediction of whether the tweet is positive or negative.	The total of positive and negative words in each tweets are calculated and the difference of positive and negative words are calculated for each tweet. The difference is termed as a score. If score > 0 then positive sentiment If score < 0 then negative sentiment If score = 0 then neutral sentiment
2.	Sentimental Analysis of Book Reviews using Unsupervise d Semantic Orientation and Supervised Machine Learning Approaches	Vipin Deep Kaur IEEE,2018	We have applied both unsupervised (Semantic Orientation – Point wise Mutual Information – Information Retrieval) and supervised (Support Vector Machine and Naïve Bayes) machine learning approaches	So for the proper and deep analysis of the book reviews we have to analyze words with adjectives as well as their adverbs.	The comparative analysis of the approaches on the datasets indicates that unsupervised approach performs better on Good Reads dataset with an accuracy of 73.23% whereas supervised approach gives better results on Amazon dataset with Naïve Bayes giving the maximum accuracy which ranges from 73.72% to 74.73% in the case of 5-folds and 10-folds respectively
3.	Stock Price Prediction	IEEE,2019 Jaeyoon Kim,	we performed sentimental	Due to the nature of Korean language, it is	y. The positive index has a value between 0

	Through the Sentimental Analysis of News Articles	Jangwon Seo, Minhyeok Lee, Junhee Seokv	analysis by building and analyzing a sentimental dictionary with news articles. Through the sentimental dictionary, we can obtain the positive index of news articles for each date. KMAA Also AI has been introduced in this project	relatively difficult to derive its high accuracy in using natural language processing methods, rather than English. Therefore, this degree of correlation in the highly complex stock data is thought to be quite meaningful	and 1. The closer the word's positive index is to 1, the more positive it means. Some words extracted from the news article, words which have very low frequency can have a high positive value. W
4.	Restaurant's Feedback Analysis System using Sentimental Analysis and Data Mining Techniques	Atharva Patil, Nishita S. Upadhyay, Karan Bheda, Rupali Sawant IEEE,2018	This paper, presents a methodology that will quell this challenge by performing sentimental analysis on the feedbacks and determine their polarity. Followed by clustering on the positive and negative feedbacks obtained from the previous process.	Here in this paper also there can be a problem of sarcastic reviews. So sarcastic tweets contains some other information but they convey some other information.	The end result of this step is a set of positive, negative and neutral sentences, out of which only the set of positive and negative sentences are the useful feedback.
5.	Convex-hull &DBSCAN clustering to predict future weather	6th International IEEE Conference and Workshop on Computing and Communicatio n, Canada, 2016	Convex-hull &DBSCAN clustering	In this paper, we mainly give our focus to predict some certain factors of weather, like temperature, humidity etc. But in future this protocol can also be used to predict rain fall, storm etc. all these important issues	This technique is suitable for the dynamic databases where the climate data are changed frequently. In this paper the accuracy of this proposed technique is calculated based on their corresponding hit and miss times.

				of weather.	In future, other incremental clustering algorithms can be used to predict the weather and can compare them with each other to detect which algorithm among them provide better accuracy
6.	Sentiment analysis in a cross-media analysis framework	Y. Woldemariam 2016 IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou	TF-IDF word level and N-Gram		Thus after doing sentiment analysis of these tweets, we founded that, TF-IDF features are giving better results (3-4%) as compared to N-Gram features. Thus we can conclude that if we are going to use machine learning algorithm for the text classification than TF-IDF is the best choice of features as compared to N-Gram
7.	Sentiment analysis on Twitter Data- set using Naive Bayes algorithm	HumaParveen and ShikhaPandey 2016 2nd IEEE Conference on Applied and Theoretical Computing and Communicatio n Technology	Sentiment Analysis and using hadoop framework for processing movie data	The existing database is not able to process the big amount of specified amunt of time.	Results of sentiment analysis on twitter data will be displayed as different sections presenting positive, negative and neutral sentiments.
8.	Sentiment Analysis in Twitter	Sayali P. Nazare, Prasad S. Nar, Akshay S. Phate, Prof.Dr. D. R. Ingle ”International Research	Sentiment analysis deals with identifying and classifying opinions	There are many approaches that analyse sentiments but hardly any work accomplished on grammatical errors. The results of sentiment analysis can be improved if these	These techniques can be applied for twitter sentiment analysis. An efficient feature vector is created by doing feature extraction in two steps after proper pre-processing. In the first step, twitter

		Journal of Engineering and Technology (IRJET)Volume: 05, Jan-2018."		types of errors can be mapped to correct words	specific features are extracted and added to the feature vector. After that, these features are removed from tweets and again feature extraction is done as if it is done on normal text.
9.	Twitter sentiment analysis using distant analysis	Go, A., Bhayani, R., & Huang, L. (2009). IEEE conference 2017.	SENTIMENT ANALYSISI USIN NAÏVE BASIS	They approach it to use different machine learning classifier and feature extractors Machine learning algorithm lead to similar performance to classify tweet sentiment	. Machine learning algorithm lead to similar performance to classify tweet sentiment
10.	Sentiment Analysis: A perspective past, present and future	Kumar, Akshi, and Teeja Mary Sebastian. "Sentiment analysis: A perspective on its past, present and future." IEEE CONFRENCE (2016): 1-14.	SENTIMENT ANALYSIS	Tackling the fuzzy definition of sentiment and the complexity of its expression in text brings up new questions providing abundant opportunity for qualitative and quantitative analysis	A vital part of the information era has been to find out the opinions of other people. In the pre-web era, it was customary for an individual to ask his or her friends and relatives for opinions before making a decision. Organizations conducted opinion polls, surveys to understand the sentiment and opinion of the general public towards its products or services. In the past few years, web documents are receiving great attention as a new medium that describes individual experiences and opinions.
11.	Lexicon based analysis on Arabic text	Abdulla, Nawaf A., et al. "Arabic sentiment analysis: Lexicon-based and corpus-	LEXICON , STEMMING	Suggest book is preferred over movie whereas result is opposite	The emergence of the Web 2.0 technology generated a massive amount of raw data by enabling Internet users to post their opinions, reviews, comments on

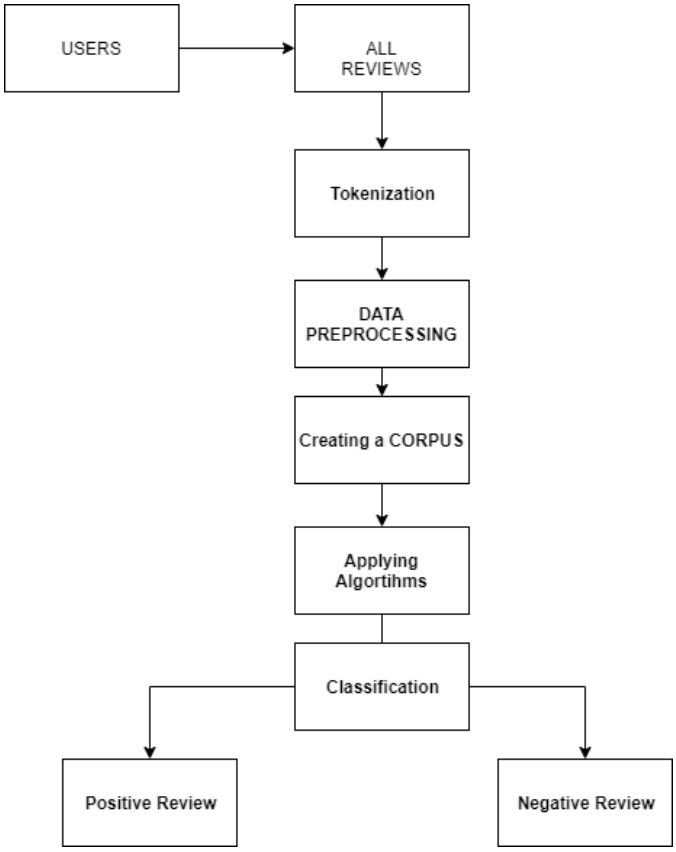
		based." 2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT). IEEE, 2017.			the web. Processing this raw data to extract useful information can be a very challenging task. An example of important information that can be automatically extracted from the users' posts and comments is their opinions on different issues, events, services, products, etc. This problem of Sentiment Analysis (SA) has been studied well on the English language and two main approaches have been devised: corpus-based and lexicon-based. This paper addresses both approaches to SA for the Arabic language
12.	A survey on sentiment analysis	challengeHussein, Doaa Mohey El-Din Mohamed. "A survey on sentiment analysis challenges." 2018	Using soft computing	Coming up with right set of keyword	This survey discusses the importance and effects of sentiment analysis challenges in sentiment evaluation based on two comparisons among forty-seven papers. The first comparison is based on the relationship between the sentiment review structure and the sentiment analysis challenges. The result of this comparison reveals another essential factor to recognize the sentiment challenges which is domain-dependence.

13.	Sentiment Analysis of Movie Review Data Using SentiLexicon Algorithm	: Deebha Mumtaza, Bindiya Ahujab IEEE, 2016	Sent-Lexicon Algorithm	Challenges such as the presence of sarcasm, blind negation, complex sentences, spam detection, forged reviews, sensitivity over time, handling hidden features could be taken up as research areas.	The result of the analysis may be acquired as positive, negative or neutral review based totally on the Score values. We can also achieve the histogram.
14.	Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews	Shilpi Chawla, Gaurav Dubey, Ajay Rana IEEE,2017	Naïve Bayes Classification technique and SVM	Unstructured data and having a large number of entries and If the dataset is unbalanced, so it is very difficult and time consuming to make a training dataset.	The overall accuracy of the classifier thus trained using Naïve Bayes Classification technique was around 40%. SVM approach gives 90% accuracy.
15.	Sentiment Analysis in MOOCs: A case study	Pedro Manuel Moreno-Marcos, Carlos Alario-Hoyos, Pedro J. Muñoz-Merino, Iria Estévez-Ayres, and Carlos Delgado Kloos IEEE,2018	Lexicon approaches(unsupervised) and Supervised approaches	There were no labeled data for training the algorithms and for their evaluation, and this required a manual labeling process. All messages used for training and evaluation were taken from the same MOOC.	The best value for AUC was between 0.71 and 0.85 and the best value for kappa was between 0.38 and 0.61, depending on the data sample used for evaluation. One possible future work would be collecting data from other MOOCs to train with a wider variety of messages.
16.	A Study of Sentiment Analysis Task and It's Challenges	Shubham V. Pandey , A. V. Deorankar IEEE,2019	Data extraction and preprocessing, tokenization, Stopping, Stemming	A system designed for a particular domain may not produce the desired result for other domain. : Co-reference resolution is a major challenge in aspect based sentiment analysis, because at this level of analysis we need to find the opinion target for each	Helps to answer the queries like what is sentiment analysis, how to perform it, and what challenges one has to face while developing a sentiment analysis system.

				sentiment phrase expressed in sentences.	
17.	Text and Image Based Spam Email Classification using KNN, Naive Bayes and Reverse DBSCAN Algorithm	Anirudh Harisinghaney, Arnan Dixit, Saurabh Gupta, Anuja Arora, IEEE	KNN algorithm, Naive Bayes algorithm and Reverse DBSCAN algorithm	Used text filtering which is time consuming. The OCR based detection also has disadvantages like, the recognition is not always perfect, and works for certain fonts only, cannot predict for CAPTCHA images and obviously are expensive.	KNN with pre-processing data getting 83% accuracy in text and image based spam filtering as compared to 45% which was without pre-processed data. Reverse DBSCAN, attaining 74% accurate result using preprocessed data as compared to 48% accuracy without preprocessed data. Naive Bayes algorithm which is 87% accurate result which was just 47% without pre-processed data.
18.	Personality Classification Based on Twitter Text Using Naive Bayes, KNN and SVM	Bayu Yudha Pratama, Riyanarto Sarno, Department of Informatics Institut Teknologi Sepuluh Nopember Surabaya, 60111, Indonesia IEEE Conference Publication	Naive Bayes K-Nearest Neighbours Support Vector Machine	Experiment fails to improve accuracy . System has 65% accuracy compared to questionnaires based test. Further improvement can be done by using more accurate dataset to improve the accuracy and using native Indonesian language for Indonesia classification (not translated).	MNB got the best accuracy in three methods tested with average accuracy 60%. SVM and KNN performed similarly. Combined method got the best results on respondent testing with an accuracy of 65%.
19.	Improving e-learning with sentiment analysis of users'	Zeid Kechaou, Adel Alimi, Mahmoud bouhajja ben ammar, IEEE	HMM and SVM-based hybrid learning method.	paving the way for further issues to be thoroughly addressed in future research works, namely : combining some of	The experimental results in this paper indicates that IG performs the best for

	opinions			these feature selections, preprocessing refinement and taking into account emoticons and misspelled words and employing different applied a linguistics techniques in the classification processes.	sentimental terms selection and exhibits the best performance for sentiment classification at most situations in terms of precision, recall and F-measure. Besides, MI is only slightly better than CHI.
20.	An Improved K-nearest-neighbor algorithm using Genetic Algorithm for Sentiment Classification	P.Kalaivani Research Scholar Department of CSE, Sathyabama University, Chennai, India. K.L.Shunmuganathan 2014 International Conference on Circuit, Power and Computing Technologies [ICCPCT]	Hybrid genetic algorithm- an improved KNN algorithm, genetic algorithm	In this work, four evaluation measures, Accuracy, Precision, Recall and F-measure are used to test the effectiveness of opinion mining. R and P denote Recall and precision of reviews.	Here are used three classification algorithms GIKNN, KNN and NB. Among all these methods GIKNN is shown to perform better. The best Fmeasure (%) for each classification is shown in bold with statistically significant achieve over the baseline with an up arrow.

BLOCK DIAGRAM



METHODOLOGY

Data Pre-processing

Before going for any other techniques, data pre-processing is necessary for any applications.

Here we will clean our dataset and make it suitable for further analysis. In NLP, data pre-processing is the very first step to build our model to make the model more efficient and accurate. There are some steps for cleaning the review data:

1) Remove numbers and punctuations if they are not relevant to the analysis.
2) Converting text to lowercase helps to reduce the size of the vocabulary for our text data.
3) Remove non-significant words that are not relevant into predicting whether the review is positive or negative.
4) Stemming: removes affixes at the beginning (prefixes) and the end (suffixes) of the words through string operation. It means taking the root of the words.
5) Tokenization: processing of segmenting running text into sentences and words. It splits all the different reviews into different words (only relevant data)

After applying all these steps, we get cleaned dataset called as corpus . And we will go for further analysis.

Create the Bag of words Model:

We created a corpus using the original reviews. By this corpus, we are going to create the bag of words model.

Bag of Words Model: Take all the different words from all reviews and creating one column for each of these words. So we will get a table containing lot of zeroes called as sparse matrix.

Basically we are getting an independent matrix of features and dependent variable.

DIFFERENT MACHINE LEARNING ALGORITHMS

1) NAIVE BAYE'S CLASSIFIER

Naïve Bayes is a prominent technique used in text classification. Even though it is a simple algorithm, the accuracy of the results given by this algorithm is high. Naïve Bayes technique gives the results by calculating the chances of all the attributes of the dataset belonging to a particular class. Naïve Bayes technique is framed on the principles if Bayes theorem.

The Bayes theorem states that "the probability of event x given that event y has taken place is equal to the probability of event y given that event x has occurred multiplied by the probability of event x, divided by the probability of event y

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

The idea behind this technique is first we present the text document as a random collection of terms as if it is a black box consisting of words, that is, an unordered set of terms.

Naïve Bayes technique returns the class which has maximum chance to which a particular test belongs, so it is a probabilistic classifier.

In order to compute this probability, the classifier makes use of two assumptions to simplify the computation. First is as discussed above, classifier doesn't take into account the position of the terms and the algorithm gives the same results even if a term is present in 1^{st} or 2^{nd} or last position.

The second assumption is popularly known as Naïve Bayes assumption where the probabilities P(Ti/c) are considered as independent given the class c and hence can be directly multiplied as follows:

$P(T_1,T_2……..T_n/c) = P(T_1/c).P(T_2/c)…..P(T_n/c)$

Since the naïve Bayes technique uses the linear combination of the terms to classify the data, it is known as linear classifier.

2) SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature

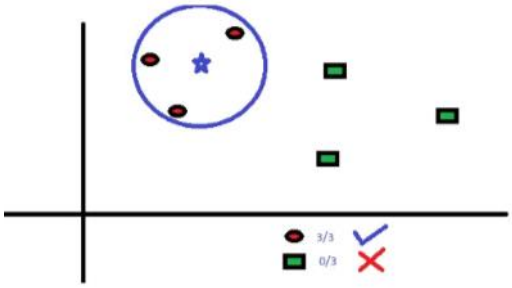
being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

In Python, scikit-learn is a widely used library for implementing machine learning algorithms. SVM is also available in the scikit-learn library and we follow the same structure for using it(Import library, object creation, fitting model and prediction).

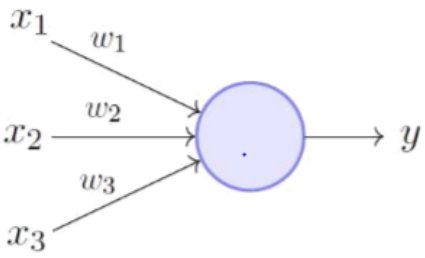
3) K- NEAREST NEIGHBOURS (KNN):

In this machine learning algorithm we will analyze our dataset and after this we will select number of nearest neighbors from which we are going to measure the distance and will decide to which category the new introduced point will belong.

We can change the number of nearest neighbors to increase the efficiency of our algorithm. So this is how will decide to which category the user will belong with the help of this method.



4) PERCEPTRON MODEL:



The perceptron model is a more general computational model than McCullochPitts neuron. It overcomes some of the limitations of the M-P neuron by introducing the concept of numerical weights (a measure of importance) for inputs, and a mechanism for learning those weights. Inputs are no longer limited to boolean values like in the case of an M-P neuron, it supports real inputs as well which makes it more useful and generalized.

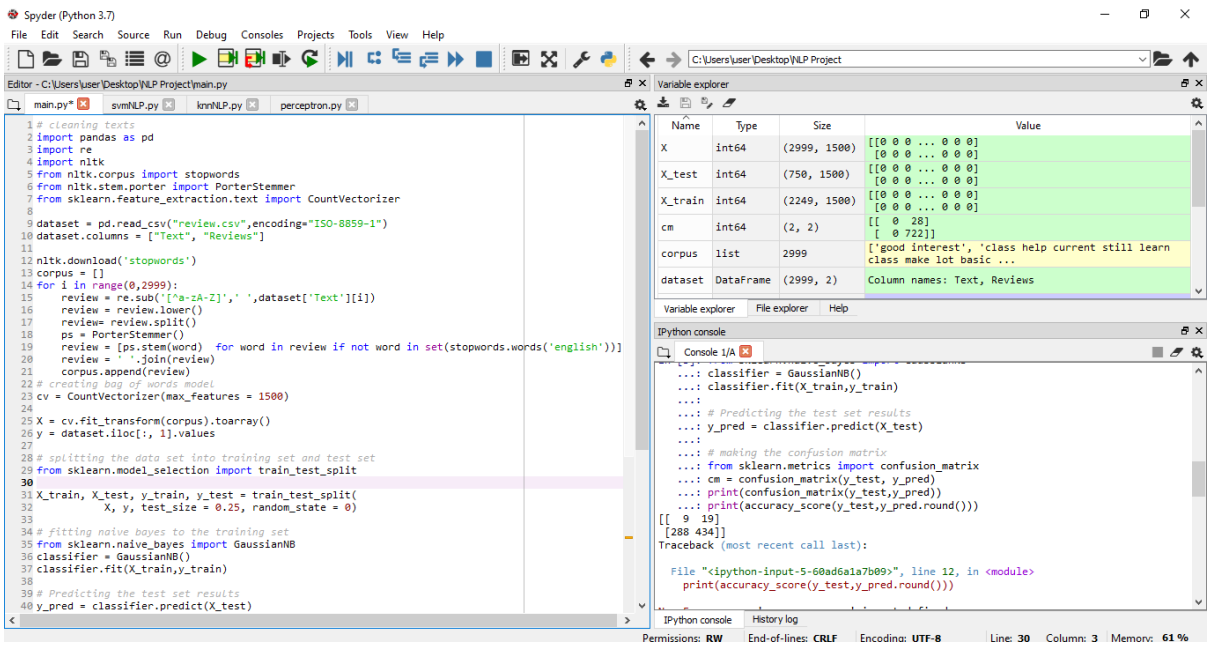
SOURCE CODE:

1) naiveBayesClassifier.py

```
# cleaning texts
import pandas as pd
import re
```

```
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

dataset = pd.read_csv("review.csv",encoding="ISO-8859–1")
dataset.columns = ["Text", "Reviews"]

nltk.download('stopwords')

corpus = []
for i in range(0,2999):
   review = re.sub('[^a-zA-Z]',' ',dataset['Text'][i])
   review = review.lower()
   review= review.split()
   ps = PorterStemmer()
   review = [ps.stem(word)  for word in review if not word in
set(stopwords.words('english'))]
   review = ' '.join(review)
   corpus.append(review)


# creating bag of words model
cv = CountVectorizer(max_features = 1500)

X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# splitting the data set into training set and test set
from sklearn.model_selection import train_test_split
 X_train, X_test, y_train, y_test = train_test_split(
         X, y, test_size = 0.25, random_state = 0)

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

# Predicting the test set results
y_pred = classifier.predict(X_test)

# making the confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred.round()))
```

SCREENSHOT:



2) svmClassifier.py

```
# cleaning texts
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

dataset = pd.read_csv("review.csv",encoding="ISO-8859–1")
dataset.columns = ["Text", "Reviews"]

nltk.download('stopwords')

corpus = []
for i in range(0,2999):
    review = re.sub('[^a-zA-Z]',' ',dataset['Text'][i])
    review = review.lower()
    review= review.split()
    ps = PorterStemmer()
    review = [ps.stem(word)  for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```

 # creating bag of words model
cv = CountVectorizer(max_features = 1500)

X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

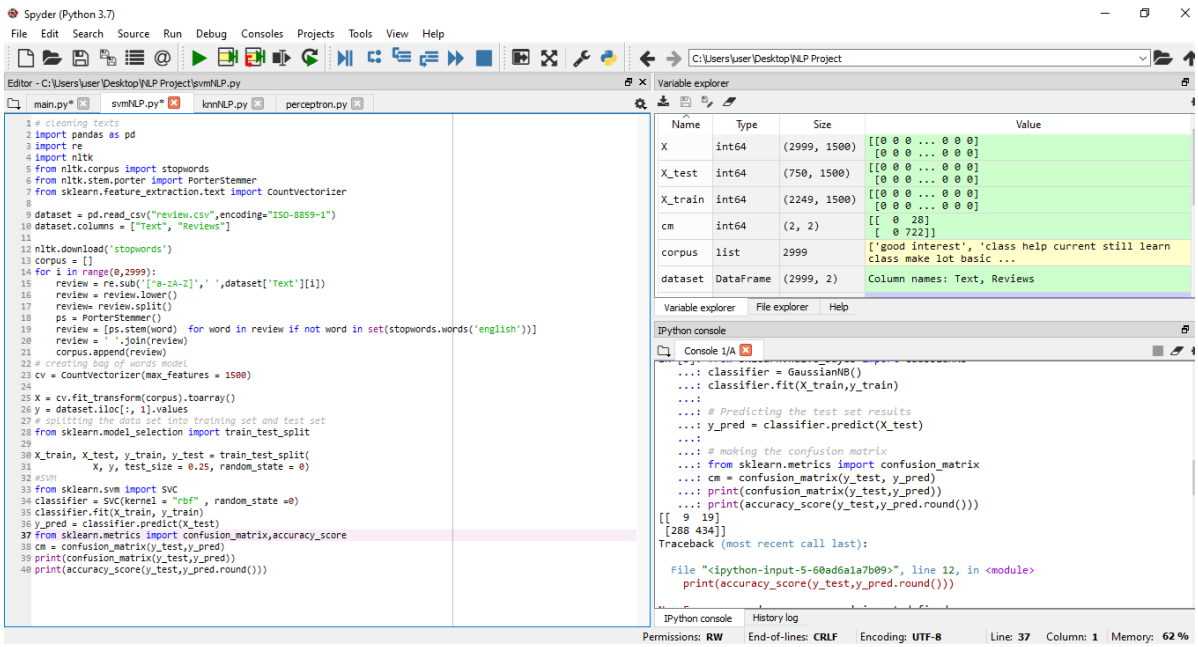
 # splitting the data set into training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size = 0.25, random_state = 0)

#SVM
from sklearn.svm import SVC
classifier = SVC(kernel = "rbf" , random_state =0)
classifier.fit(X_train, y_train)

#prediction
y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred.round()))

SCREENSHOT:

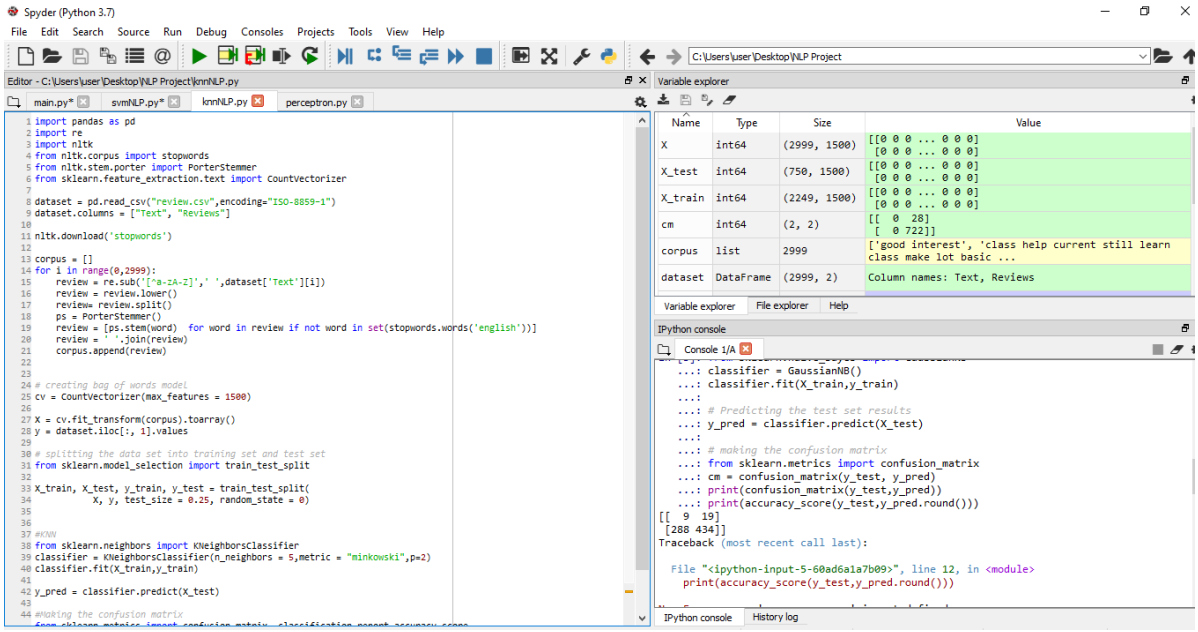


3) KNNClasifier.py

```
# cleaning texts
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

dataset = pd.read_csv("review.csv",encoding="ISO-8859–1")
dataset.columns = ["Text", "Reviews"]

nltk.download('stopwords')

corpus = []
for i in range(0,2999):
    review = re.sub('[^a-zA-Z]',' ',dataset['Text'][i])
    review = review.lower()
    review= review.split()
    ps = PorterStemmer()
    review = [ps.stem(word)  for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# creating bag of words model
cv = CountVectorizer(max_features = 1500)

X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# splitting the data set into training set and test set
from sklearn.model_selection import train_test_split
 X_train, X_test, y_train, y_test = train_test_split(
         X, y, test_size = 0.25, random_state = 0)

#KNN
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5,metric = "minkowski",p=2)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

#Making the confusion matrix
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score
```

```
cm = confusion_matrix(y_test,y_pred)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred.round()))
```

SCREENSHOT:



4) perceptron.py

```
# cleaning texts
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

dataset = pd.read_csv("review.csv",encoding="ISO-8859–1")
dataset.columns = ["Text", "Reviews"]

nltk.download('stopwords')

corpus = []
for i in range(0,2999):
    review = re.sub('[^a-zA-Z]',' ',dataset['Text'][i])
    review = review.lower()
    review= review.split()
    ps = PorterStemmer()
    review = [ps.stem(word)  for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```

 # creating bag of words model
cv = CountVectorizer(max_features = 1500)

X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

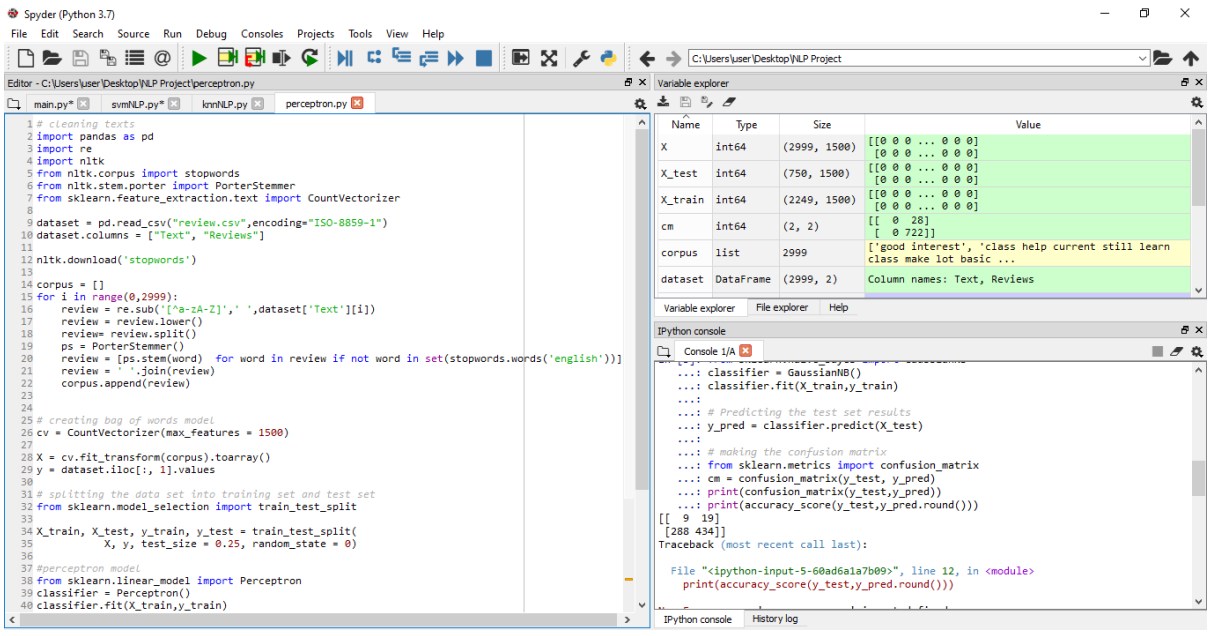
 # splitting the data set into training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size = 0.25, random_state = 0)

#perceptron model
from sklearn.linear_model import Perceptron
classifier = Perceptron()
classifier.fit(X_train,y_train)

#Predicting testset
y_pred = classifier.predict(X_test)

#confusion matrix and accuracy
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred.round()))

SCREENSHOT:



RESULTS AND DISCUSSIONS:

DATASET:

dataset - DataFrame

Index	Text	Reviews
0	good and interesting	1
1	This class is very helpful…	1
2	like!Prof and TAs are help…	1
3	Easy to follow and i…	1
4	Really nice teacher!I co…	1
5	Great course - I recommen…	1
6	One of the most useful …	1
7	I was disappointed…	1
8	Super content. I'l…	1
9	One of the excellent co…	1
10	Is there any reason why y…	1
11	Excellent course and t…	1
12	This is a good course …	1
13	Good content, but the cour…	1

Format Resize Background color Column mi

CORPUS CREATED:

corpus - List (2999 elements)

Index	Type	Size	Value
7	str	1	disappoint name mislead cours provid good introduct overview respons c ...
8	str	1	super content definit cours
9	str	1	one excel cours coursera inform technolog boss manag
10	str	1	reason appli cours bcg content pretti uniqu includ high level analysi ...
11	str	1	excel cours teacher congratul
12	str	1	good cours cio non technic compani
13	str	1	good content cours set least allow learn content long term due miss re ...
14	str	1	structur approach thank share
15	str	1	program demystifi evolv world cio typic global corpor coverag introduc ...
16	str	1	relev use cours design cio
17	str	1	cours say anyth digit core subject digit wave
18	str	1	video present french could translat english
19	str	1	cours content quit good though could deeper area peer review system wo ...
20	str	1	great piec work especi like lifehack cio
21	str	1	excel cours reward term use tool given excel today put use job thank i ...
22	str	1	excel represent day day thank share vision experi advic regardsdaniel ...
23	str	1	interest well design mooc
24	str	1	excel cours total recommend
25	str	1	great cours inform content interest unfortun peer grade assign classma ...
26	str	1	interest cours learn lot histori aborigin educ

- **NAÏVE BAYES CLASSIFIER:**

  Accuracy and Confusion Matrix:

```
In [6]:

In [6]: from sklearn.metrics import confusion_matrix,accuracy_score
   ...: cm = confusion_matrix(y_test,y_pred)
   ...: print(confusion_matrix(y_test,y_pred))
   ...: print(accuracy_score(y_test,y_pred.round()))
[[  9  19]
 [288 434]]
0.5906666666666667
```

- **SVM CLASSIFIER :**

  Accuracy and Confusion Matrix:

```
   ...:
   ...: from sklearn.metrics import confusion_matrix,accuracy_score
   ...: cm = confusion_matrix(y_test,y_pred)
   ...: print(confusion_matrix(y_test,y_pred))
   ...: print(accuracy_score(y_test,y_pred.round()))
C:\Users\user\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: Futu
default value of gamma will change from 'auto' to 'scale' in version 0.
better for unscaled features. Set gamma explicitly to 'auto' or 'scale'
warning.
  "avoid this warning.", FutureWarning)
[[  0  28]
 [  0 722]]
0.9626666666666667
```

- **KNN- Classifier:**

  Accuracy and Confusion Matrix:

```
   ...:
   ...: #Making the confusion matrix
   ...: from sklearn.metrics import confusion_matrix,
classification_report,accuracy_score
   ...: cm = confusion_matrix(y_test,y_pred)
   ...: print(confusion_matrix(y_test,y_pred))
   ...: print(accuracy_score(y_test,y_pred.round()))
[[  0  28]
 [  1 721]]
0.9613333333333334
```

- **PERCEPTRON MODEL:**

  Accuracy and Confusion Matrix:

```
   ...: from sklearn.metrics import confusion_matrix,
classification_report,accuracy_score
   ...: cm = confusion_matrix(y_test,y_pred)
   ...: print(confusion_matrix(y_test,y_pred))
   ...: print(accuracy_score(y_test,y_pred.round()))
C:\Users\user\Anaconda3\lib\site-packages\sklearn\linear_model
\stochastic_gradient.py:166: FutureWarning: max_iter and tol parameters have been
added in Perceptron in 0.19. If both are left unset, they default to max_iter=5 a
tol=None. If tol is not None, max_iter defaults to max_iter=1000. From 0.21, defa
max_iter will be 1000, and default tol will be 1e-3.
  FutureWarning)
[[ 11  17]
 [ 33 689]]
0.9333333333333333
```

So in this project we have used four Machine Learning algorithms namely Naïve Bayes, SVM, KNN, and Perceptron algorithm. We have taken a dataset from Kaggle to predict whether a MOOCS reviews are positive or negative. So we have taken some dependent and one independent variable. For dependent variable (Reviews) if value is 1 then we can conclude that the review is positive and if the value is 0 then we can say that the review is negative. So for our algorithms we have got the accuracy of 59% (approx.) for Naïve Bayes algorithm, 96.5% (approx.) for SVM algorithm, 96% (approx.) for KNN algorithm and 93% (approx.) for Perceptron algorithm. So from here we can conclude that SVM algorithm is more efficient in calculating the result that whether the review is positive or not. And it will provide us more accurate results with less number of faulty results.

CONCLUSION

In this project, we have performed sentiment analysis on each and every review given by the student for a particular course. The Outcome of the Sentiment analysis is positive or negative or neutral. For these outcomes, we have given the recommendations to be done in future. And these recommendations will be done in future based on the reviews given by the students. If the majority of students gives the positive polarity review then no action is taken, else if majority gives negative polarity review then that course is removed, else course is said to be improved. This leads to the beneficial of students and management.

REFERENCES:

- Adarsh, M. J., & Ravikumar, P. (2018, February). An Effective Method of Predicting the Polarity of Airline Tweets using sentimental Analysis. In *2018 4th International Conference on Electrical Energy Systems (ICEES)* (pp. 676-679). IEEE.
- Kaur, V. D. Sentimental Analysis of Book Reviews using Unsupervised Semantic Orientation and Supervised Machine Learning Approaches.
- Kim, J., Seo, J., Lee, M., & Seok, J. (2019, July). Stock Price Prediction Through the Sentimental Analysis of News Articles. In *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 700-702). IEEE.
- Angeline Prasanna, G., & Paulstin, K. A. S. Restaurant's Feedback Analysis System using Sentimental Analysis and Data Mining Techniques.
- Dey, R., & Chakraborty, S. (2015, October). Convex-hull & DBSCAN clustering to predict future weather. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)* (pp. 1-8). IEEE.
- Woldemariam, Y. (2016, March). Sentiment analysis in a cross-media analysis framework. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)* (pp. 1-5). IEEE.
- Parveen, H., & Pandey, S. (2016, July). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. In *2016 2nd international conference on applied and theoretical computing and communication technology (iCATccT)* (pp. 416-419). IEEE.
- Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)* (pp. 30-38).
- Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), 2009.
- Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, *4*(10), 1-14.
- Abdulla, N. A., Ahmed, N. A., Shehab, M. A., & Al-Ayyoub, M. (2013, December). Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)* (pp. 1-6). IEEE.
- Hussein, D. M. E. D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, *30*(4), 330-338.
- Mumtaz, D., & Ahuja, B. (2016, July). Sentiment analysis of movie review data using Senti-lexicon algorithm. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* (pp. 592-597). IEEE.
- Chawla, S., Dubey, G., & Rana, A. (2017, September). Product opinion mining using sentiment analysis on smartphone reviews. In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 377-383). IEEE.
- Moreno-Marcos, P. M., Alario-Hoyos, C., Muñoz-Merino, P. J., Estévez-Ayres, I., & Kloos, C. D. (2018, April). Sentiment analysis in MOOCs: A case study. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1489-1496). IEEE.
- Pandey, S. V., & Deorankar, A. V. (2019, February). A Study of Sentiment Analysis Task and It's Challenges. In *2019 IEEE International Conference on Electrical,*

Computer and Communication Technologies (ICECCT) (pp. 1-5). IEEE.

- Harisinghaney, A., Dixit, A., Gupta, S., & Arora, A. (2014, February). Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm. In *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)* (pp. 153-155). IEEE.
- Pratama, B. Y., & Sarno, R. (2015, November). Personality classification based on Twitter text using Naive Bayes, KNN and SVM. In *2015 International Conference on Data and Software Engineering (ICoDSE)* (pp. 170-174). IEEE.
- Kechaou, Z., Ammar, M. B., & Alimi, A. M. (2011, April). Improving e-learning with sentiment analysis of users' opinions. In *2011 IEEE global engineering education conference (EDUCON)* (pp. 1032-1038). IEEE.
- Kalaivani, P., & Shunmuganathan, K. L. (2014, March). An improved K-nearest-neighbor algorithm using genetic algorithm for sentiment classification. In *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]* (pp. 1647-1651). IEEE.
- Abdulla, N. A., Ahmed, N. A., Shehab, M. A., & Al-Ayyoub, M. (2013, December). Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)* (pp. 1-6). IEEE.