

EED 363: APPLIED MACHINE LEARNING

SPRING 2022

Sentiment Analysis on twitter Data

Group Members

Vandavasi Charan (1910110468)

Taduvai Sai Saketh (1910110413)

Kamisetty Sudhamsh(1910110189)

Professor

Madan Gopal Sir

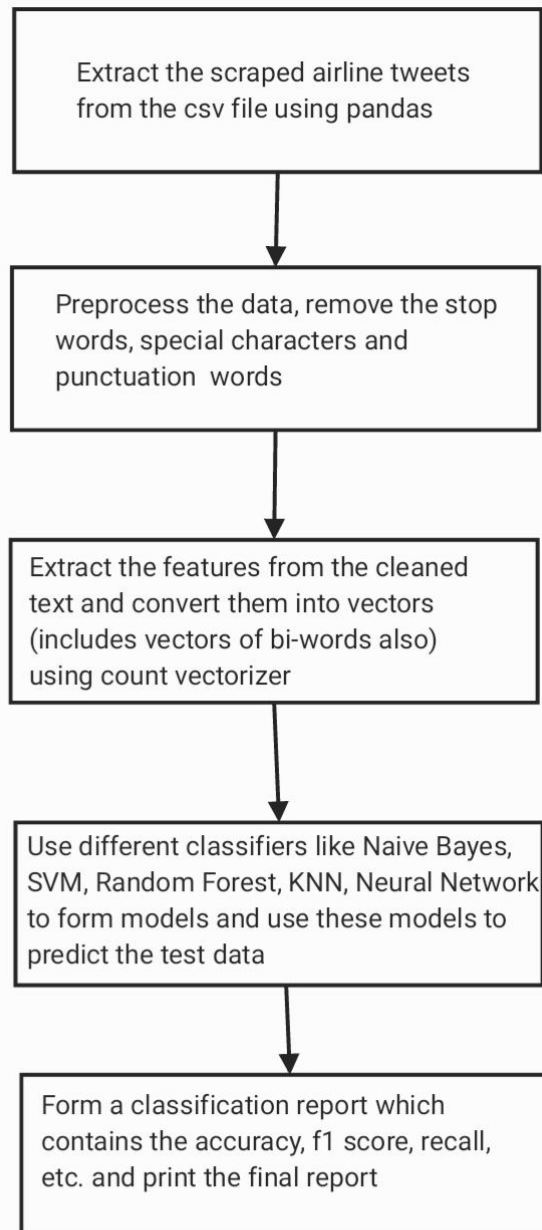
Abstract:

The main objective of this project is to identify the sentiment of tweets generated by the users on Twitter. In this project multiple machine learning techniques are examined. The proposed framework involves pre-processing, Exploratory Data Analysis(EDA), feature extraction, and classification. As will be seen, pre-processing and feature extraction play crucial roles in this experiment to reach the highest accuracy. The classification task is performed by a few classifiers namely, Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbours (K-NN), and Random Forest (RF) to determine which classifier has the highest accuracy. Moreover, few experimental results are analyzed and evaluated by a series of tools such as the confusion matrix, and error rates. Python libraries like sklearn, seaborn, NumPy, nltk, smote, and TfidfVectorizer+.

Introduction:

The sentiment is opinion or judgment prompted by feeling. Sentiment analysis involves extracting the opinion of the user on a scale based on the feedback given by the user. It is an approach to Natural language processing that identifies the emotional tone behind a body of text. It can be applied to any type of event or category; the results can be used to make satisfactory changes to them. The purpose of this sentiment analysis is to determine the subjectivity of opinion, the result of a review, or a tweet. It helps a business to understand the social sentiment of their brand, product, or service while monitoring online conversations.

Proposed Model:

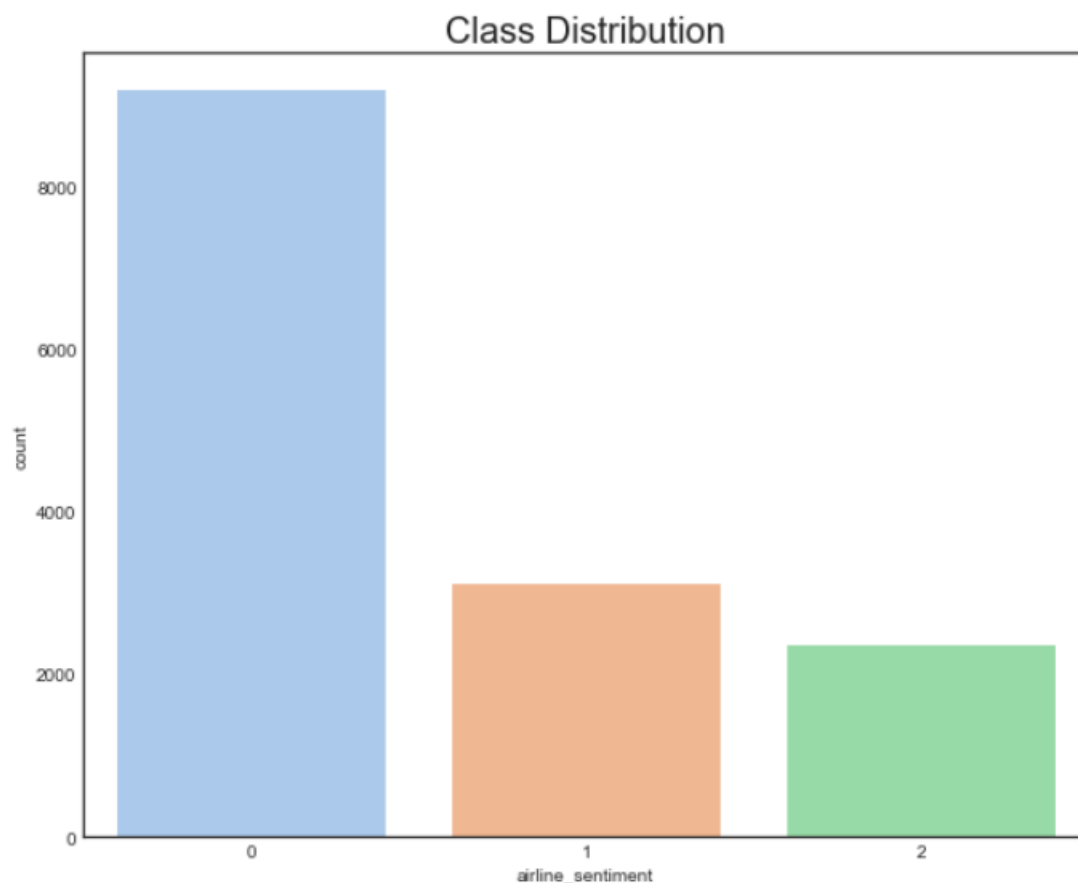


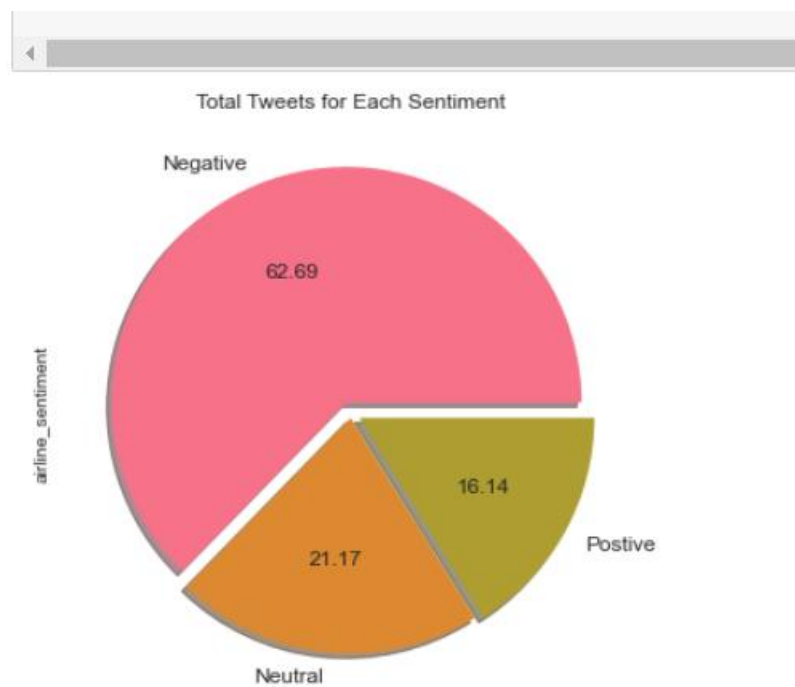
Data Description:

For our project, we are going to perform sentiment analysis on Twitter data. This data was generated through the users of major airlines on Twitter. This dataset contains tweets mentioning each Major US airline. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as “late flight” or “rude service”). The classified data also has reasons for negative, positive, and neutral feedback. The dataset contains 15 columns mentioning aspects such as the location, user id, tweet time, airline sentiment, negative reason, tweet text, etc., and 14640 rows containing the required data. We will be using this data to train and test our model.

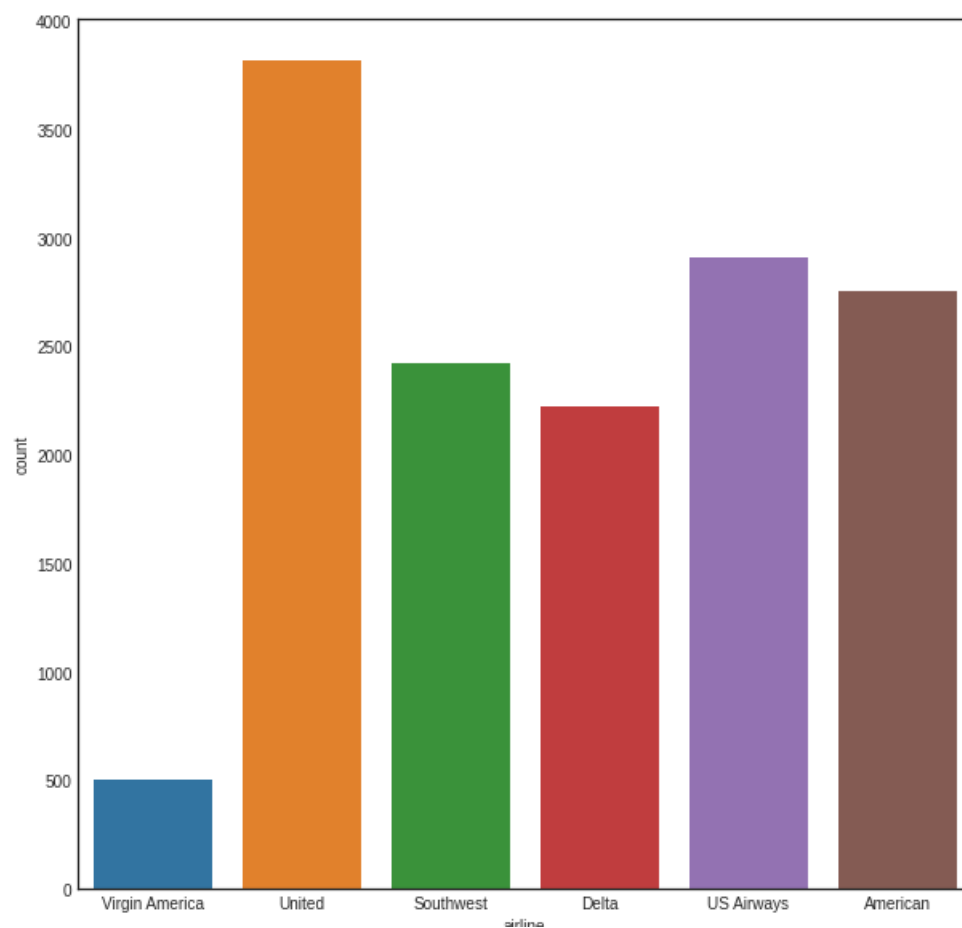
Dataset source: <https://www.kaggle.com/joparga3/us-twitter-airline-sentiment/data>

```
plt.figure(figsize = (10, 8))
ax = sns.countplot(x = 'airline_sentiment', data = tweets, palette = 'pastel')
ax.set_title(label = 'Class Distribution', fontsize = 20)
plt.show()
```





```
plt.figure(figsize=(10,10))  
ax = sns.countplot(x="airline", data=tweets)
```



```
tweets.shape
```

```
(14640, 16)
```

```
tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   tweet_id                             14640 non-null  int64
1   airline_sentiment                    14640 non-null  object
2   airline_sentiment_confidence         14640 non-null  float64
3   negativereason                      9178 non-null   object
4   negativereason_confidence            10522 non-null  float64
5   airline                             14640 non-null  object
6   airline_sentiment_gold               40 non-null     object
7   name                                14640 non-null  object
8   negativereason_gold                 32 non-null     object
9   retweet_count                       14640 non-null  int64
10  text                                14640 non-null  object
11  tweet_coord                         1019 non-null   object
12  tweet_created                       14640 non-null  object
13  tweet_location                      9907 non-null   object
14  user_timezone                       9820 non-null   object
dtypes: float64(2), int64(2), object(11)
memory usage: 1.7+ MB
```

Exploratory Data Analysis:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. It's a method of visualizing, summarizing, and analyzing data that are hidden behind rows and columns. EDA is a critical step in data science because it helps us to gain specific information and statistical measurements that are critical for business continuity, stockholders, and data scientists. It performs to define and refine our important features variable selection, that will be used in our model. In EDA for our dataset, we will first identify various parameters for all the features such as their

probability distribution, mean, variance, and their correlation with the output. Also, we will be identifying and removing the outliers in each feature using a boxplot and identifying and removing null values from the data set if any.

```
In [47]: tweets.isnull().any().describe()
```

```
Out[47]: count      16  
         unique      2  
         top        False  
         freq        9  
         dtype: object
```

```
tweets.describe()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason_confidence	retweet_count
count	1.464000e+04	14640.000000	14640.000000	10522.000000	14640.000000
mean	5.692184e+17	0.534495	0.900169	0.638298	0.082650
std	7.791112e+14	0.756084	0.162830	0.330440	0.745778
min	5.675883e+17	0.000000	0.335000	0.000000	0.000000
25%	5.685592e+17	0.000000	0.692300	0.360600	0.000000
50%	5.694779e+17	0.000000	1.000000	0.670600	0.000000
75%	5.698905e+17	1.000000	1.000000	1.000000	0.000000
max	5.703106e+17	2.000000	1.000000	1.000000	44.000000

```
tweets.isnull().sum()
```

```
tweet_id          0
airline_sentiment  0
airline_sentiment_confidence  0
negativereason    5462
negativereason_confidence  4118
airline           0
airline_sentiment_gold  14600
name              0
negativereason_gold  14608
retweet_count      0
text              0
tweet_coord       13621
tweet_created      0
tweet_location     4733
user_timezone      4820
final_text         0
dtype: int64
```

```
print(tweets.nativereason.value_counts())
```

```
Customer Service Issue    2910
Late Flight                1665
Can't Tell                1190
Cancelled Flight           847
Lost Luggage               724
Bad Flight                 580
Flight Booking Problems    529
Flight Attendant Complaints 481
longlines                  178
Damaged Luggage            74
Name: negativereason, dtype: int64
```

Literature Survey:

“Sentiment Analysis using SVM and Naïve Bayes Classifiers on Restaurant Review Dataset”

This is an IEEE 2021 conference paper

As the customer review on food, service of restaurant is very important for growth of business. Sentiment analysis is a technique that was used in order to identify people’s opinions.

Sentiment analysis technology is very beneficial for organizations and businesses as it allows them to understand customer needs and monitor the reputation of their products.

Reason for using SVM and Naïve Bayes is because of the popular usage of those algorithms in research of sentiment analysis. However, it is still quite uncertain which of them perform better. The used data set contains 1000 reviews.

The steps involved to construct their model is as follows:

Data collection, data pre-processing, feature selection, sentiment classification, and evaluation.

They researched and found that SVM classifier shows highest accuracy for one data set (and another paper shows that Naïve Bayes highest accuracy for movie reviews but lower accuracy in hotel reviews).

Because of the compact size of the available dataset, the resulting models are evaluated by cross validation. For each model, 700 reviews are selected non-specifically for training and the leftover 300 reviews is used for testing. For each training and testing dataset, there is a balance between constructive and destructive evaluations. Each model will be evaluated based on accuracy, F1 score, and confusion matrix.

Finally, they compared the two classifiers SVM and Naïve Bayes classifiers. Using grid search to assess different hyperparameter combinations for each classifier, a total of 36 models were fitted. Evaluation of these models shows that Naïve Bayes resulted in the single best model with the highest precision of 77.33% and F1 score of 0.7792, but for the overall performance, SVM slightly outperformed Naïve Bayes, also reaching accuracies of up to 77%.

“Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning”

This is an IISA 2013 conference paper by Christos Troussas, Maria Virvou

In this paper the primary and underlying idea is that the fact of knowing how people feel about certain topics can be considered as a classification task. People’s feelings can be positive, negative, or neutral. A sentiment is often represented in subtle or complex ways in a text. An online user can use a diverse range of other techniques to express his or her emotions. Apart from that, s/he may mix objective and subjective information about a certain topic. On top of that, data gathered from the World Wide Web often contain a lot of noise. Indeed, the task of automatic sentiment recognition in online text becomes more difficult for all the aforementioned reasons. Hence, we present how sentiment analysis can assist language learning, by stimulating the educational process and experimental results on the Naive Bayes Classifier.

Their study was mainly on the focus of users' Facebook status updates. They taken limited quantity 5000 for positive sentiments and the other 5000 for the negative sentiments.

In one survey, the authors found that symbolic techniques achieve accuracy lower than 80% and are generally poorer than machine learning methods on movie review sentiment analysis. Among the machine learning methods, they considered three supervised approaches: Support Vector Machine (SVM), Naive Bayes Multinomial (NBM), and maximum Entropy (Maxent). They found that all of them deliver comparable results on various feature extraction (unigrams, bigrams, etc) with high accuracy at 80%~87%.

They used three classifiers to their dataset. The data is pre-processed and developed the code for three classifiers Naive Bayes Classifier, Rocchio Classifier, Perceptron Classifier.

The results are as follows:

Naïve Bayes classifier	Actual Positive	Actual Negative
Predicted Positive	0.76	0.23
Predicted Negative	0.35	0.65

Rocchio Classifier	Actual Positive	Actual Negative
Predicted Positive	0.73	0.24
Predicted Negative	0.27	0.76

Perceptron Classifier	Actual Positive	Actual Negative
Predicted Positive	0.65	0.35

Predicted Negative	0.52	0.48
--------------------	------	------

	Naïve Bayes classifier	Rocchio Classifier	Perceptron Classifier
Precision	0.77	0.65	0.75
Recall	0.68	0.56	0.73
F -score	0.72	0.60	0.74

Based on the F-score, Rocchio classifier has the best performance and Perceptron the least. Naive Bayes performed almost as well as Rocchio but has a significantly lower recall than the latter.

Finally, the authors present their experimental results, which show that the accuracy in analyzing the sentimental state of Facebook users, using the Naive Bayes Classifier, is really high.

Pre-processing /Cleaning the data:

By observing the data, we find that there is some unnecessary information that doesn't affect the sentiment. So, the data should be pre-processed to remove this unnecessary information. Listed below are some of the examples of what needs to be removed from the data to get the data ready for further processing

Removal of URLs: There's a high possibility of having some URLs in the tweet. So, they need to be deleted for further study.

Removal of HTML Tags: It is a common pre-processing technique that will be useful as the data is scraped from various websites which can contain HTML strings as part of the text which needs to be removed.

Removal of special characters: This is a text pre-processing technique that can be used to remove special characters such as “@, ! & , #” which can better manage 'hurray' and 'hurray! 5

Removal of emojis: The emojis are to be removed because this would not affect the sentiment Lowering case: We had used the lowercasing technique to eliminate the sparsity issue and reduce the vocabulary size.

Removals of stop words: Stop words are words that are commonly used in a language. 'a', 'an', 'the', 'is', 'what' etc. are examples.

Removal of Username, Airline name: There might be a change that when we download the tweet it contains the username and airline name which can be identified with @ symbol. This will not affect the sentiment. So, we can remove them.

Tokenization: Tokenization is a technique by which vast quantities of text are redivided into smaller parts.

Vectorization Approach:

In other approaches all the words in the text are treated as equally important there's no notion of some words in the document being more important than others. TF-IDF, or term frequency-inverse document frequency, addresses this issue. It aims to quantify the importance of a given word relative to other words in the document and the corpus.

The intuition behind TF-IDF is as follows: if a word w appears many times in a sentence S_1 but does not occur much in the rest of the Sentences S_n in the corpus, then the word w must be of great importance to the Sentence S_1 . The importance of w should increase in proportion to its frequency in S_1 (how many times that word occurs in sentence S_1), but at the same time, its importance should decrease in proportion to the word's frequency in other Sentence S_n in the corpus. Mathematically, this is captured using two

quantities: TF and IDF. The two are then multiplied to arrive at the TF-IDF score.

TF (term frequency) measures how often a term or word occurs in a given document.

IDF (inverse document frequency) measures the importance of the term across a corpus. In computing TF, all terms are given equal importance (weightage). However, it's a well-known fact that stop words like is, are, am, etc., are not important, even though they occur frequently. To account for such cases, IDF weighs down the terms that are very common across a corpus and weighs up the rare terms

SMOTE Methodology:

“Synthetic Minority Oversampling Technique” (SMOTE) methodology is a widely used technique to balance the training set before the learning phase. It is used to deal with class imbalance problems.

In this technique, we simply duplicate examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model. An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective. Prior to fitting a model, we simply duplicate examples from the minority class in the training dataset. This can help to balance the class distribution, but it doesn't give the model any extra information.

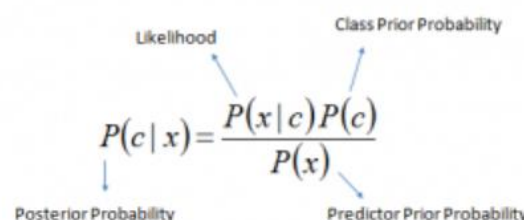
Classification Models

Training and Test Data:

Data needs to be separated into 2 sections- training data and testing data. This is done in order to fit our model according to training data so that we can test its results on testing data. Test data provides us with ideal standards and is used once the training of model is complete. In our project, we have decided to split dataset in 80%-20% ratio for training and testing respectively.

Naïve Bayes Classifier:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on the Bayes theorem (Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred) and it is used for solving classification problems. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions. The fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome.



The diagram shows the formula for Bayes' Theorem: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from labels to parts of the formula: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

$P(c|x)$ is the posterior probability of class (c, target)
given predictor (x, attributes).

$P(c)$ is the prior probability of class.

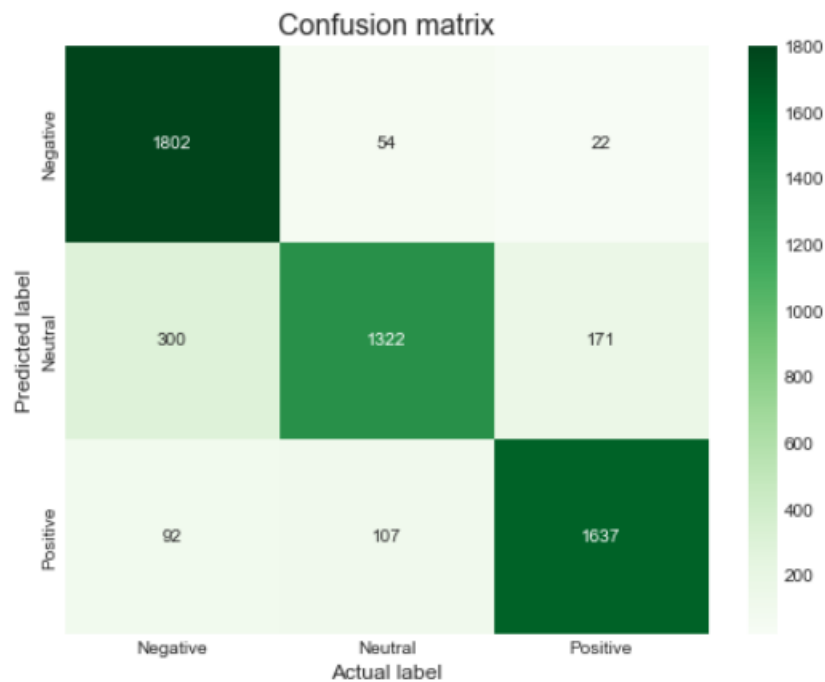
$P(x|c)$ is the likelihood which is the probability of predictor given class.

$P(x)$ is the prior probability of predictor.

We have used multinomial naïve bayes classifier for prediction.

Results:

0.8645360450335936					
	precision	recall	f1-score	support	
0	0.82	0.96	0.89	1878	
1	0.89	0.74	0.81	1793	
2	0.89	0.89	0.89	1836	
accuracy			0.86	5507	
macro avg	0.87	0.86	0.86	5507	
weighted avg	0.87	0.86	0.86	5507	

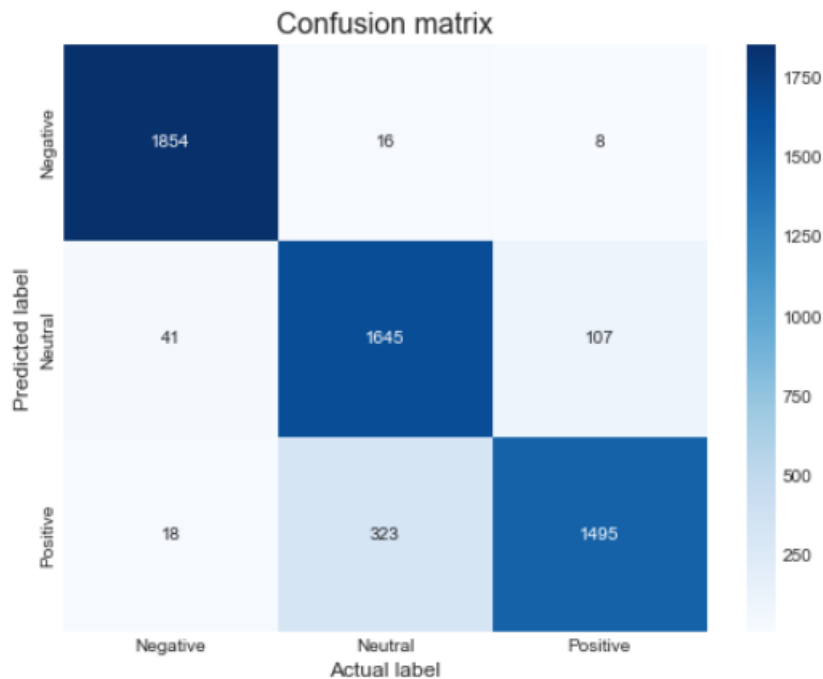


Logistic Regression Model :

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Unlike linear regression, which finds the best fitting line, logistic regression finds the best fitting curve and works well for classification problems. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value which in the case of our project is 'positive', 'negative', or 'neutral'.

Logistic Regression uses a simple equation that shows the linear relation between the independent variables. These independent variables along with their coefficients are united linearly to form a linear equation that is used to predict the output. This algorithm is entitled logistic regression as the key method behind it is the logistic function. The output can be predicted from the independent variables, which form a linear equation.

0.9068458325767206				
	precision	recall	f1-score	support
0	0.97	0.99	0.98	1878
1	0.83	0.92	0.87	1793
2	0.93	0.81	0.87	1836
accuracy			0.91	5507
macro avg	0.91	0.91	0.91	5507
weighted avg	0.91	0.91	0.91	5507



K Nearest Neighbours:

KNN (K nearest neighbours) is the non-parametric method or classifier used for classification as well as regression problems. This is the lazy or late learning classification algorithm where all of the computations are derived until the last stage of classification, as well as this, is the instance-based learning algorithm where the approximation takes place locally. Being simplest and easiest to implement there is no explicit training phase earlier and the algorithm does not perform any generalization of training data. KNN explains categorical value using majority votes of K nearest neighbors where the value for K can differ, so on changing the value of K, the value of votes can also vary.

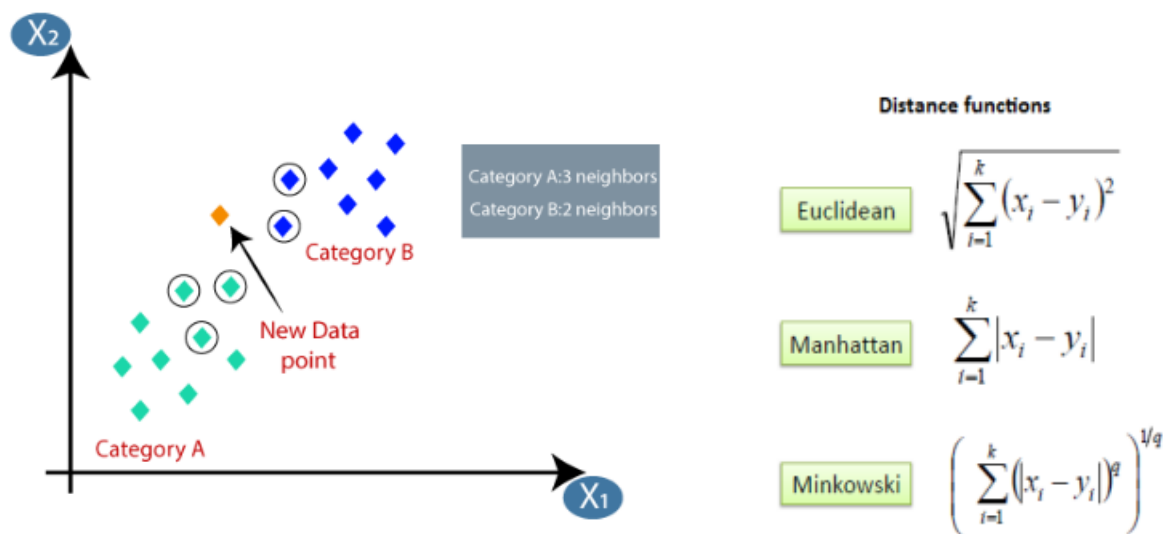
Algorithm:

Compute the distance metric between the test data point and all labeled data points.

Order the labeled data points in increasing order of distance metric.

Select the top K-labeled data points and look at class labels.

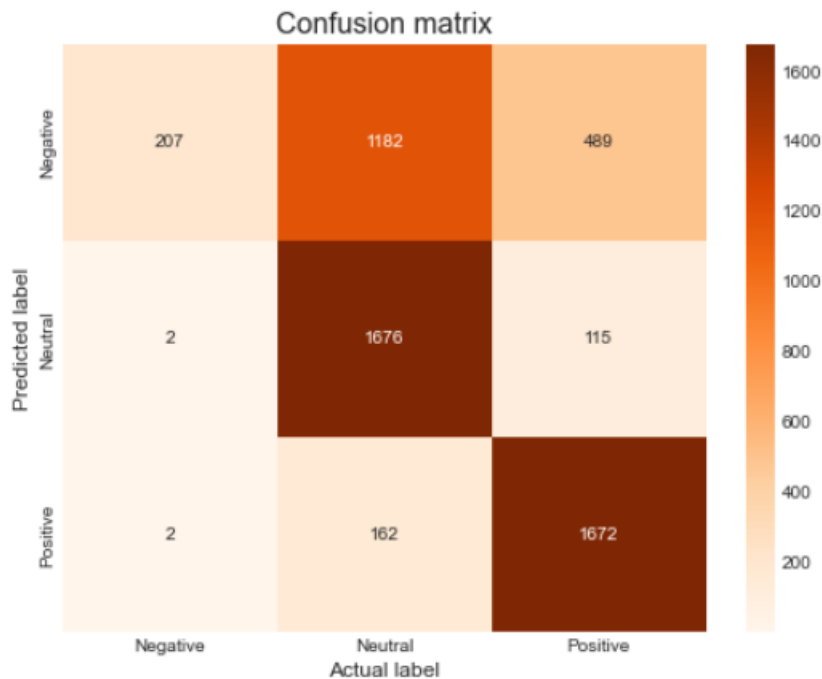
Look for the class labels that the majority of these K-labeled data points have and assign them to test data points.



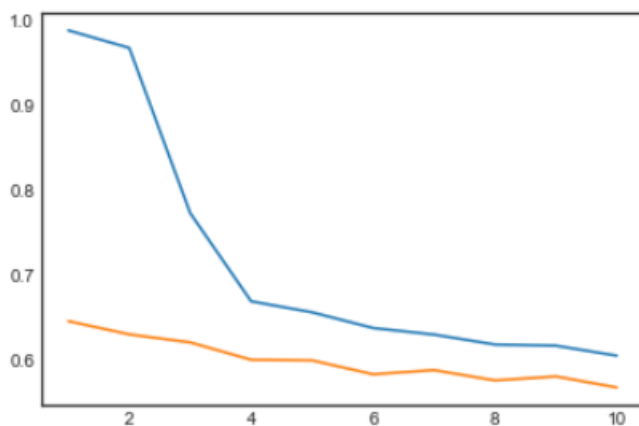
For changing the value of K, the output for the test data point can also vary. So, it's necessary to choose the value of K wisely. The large value for K can reduce the overall noise but there is no guarantee. The k-Nearest Neighbors algorithm is a simple and effective way to classify data. But the algorithm has to carry around the full dataset; for large datasets, this implies a large amount of storage. In addition, you need to calculate the distance measurement for every piece of data in the database, and this can be cumbersome.

0.6455420374069366				
	precision	recall	f1-score	support
0	0.98	0.11	0.20	1878
1	0.55	0.93	0.70	1793
2	0.73	0.91	0.81	1836
accuracy			0.65	5507
macro avg	0.76	0.65	0.57	5507
weighted avg	0.76	0.65	0.57	5507

plot.show()



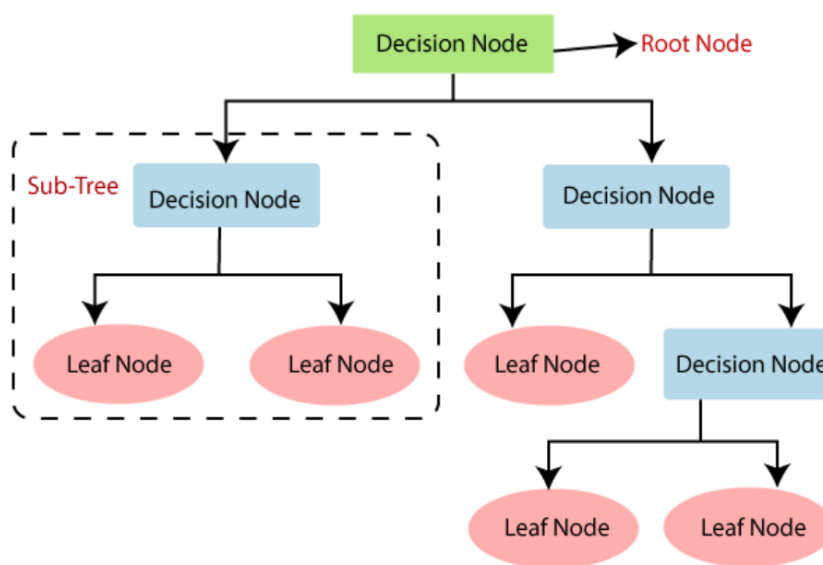
Accuracy: 0.6455420374069366



Decision tree Classifier:

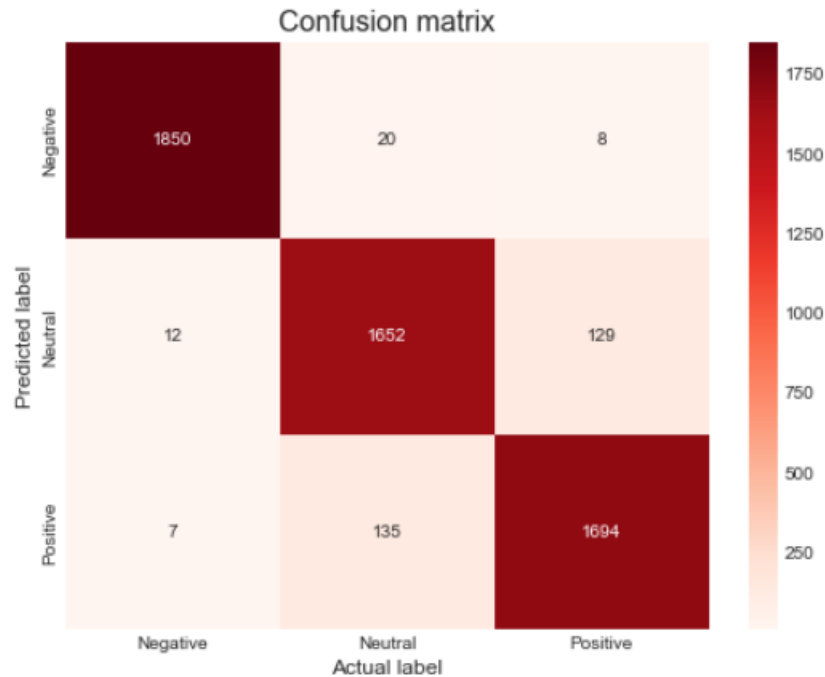
Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes

are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated for the subtree rooted at the new node. By using Decision Trees, both numerical and categorical data can be processed.



Results :

0.9420737243508263					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	1878	
1	0.92	0.92	0.92	1793	
2	0.92	0.92	0.92	1836	
accuracy			0.94	5507	
macro avg	0.94	0.94	0.94	5507	
weighted avg	0.94	0.94	0.94	5507	



Random Forest:

Actually, Random Forest is a general term for classifier combination that uses L tree-structured classifiers $\{h(x, \theta_k), k = 1, \dots, L\}$ where θ_k are independent identically distributed random vectors and x is an input. With respect to this definition, one can say that Random Forest is a family of methods in which we can find several algorithms, such as the Forest-RI algorithm proposed by Breiman in and cited as the reference method in all RF-related papers. In the Forest-RI algorithm, Bagging is used in tandem with a random feature selection principle. The training stage of this method consists in building multiple trees, each one trained on a bootstrap sample of the original training set i.e., the Bagging principle - and with a CART-like induction algorithm. This tree induction method, sometimes called RandomTree, is a CART-based algorithm that modifies the feature selection procedure at each node of the tree, by introducing a random pre-selection i.e., the Random Subspace principle.

Each tree is grown as follows:

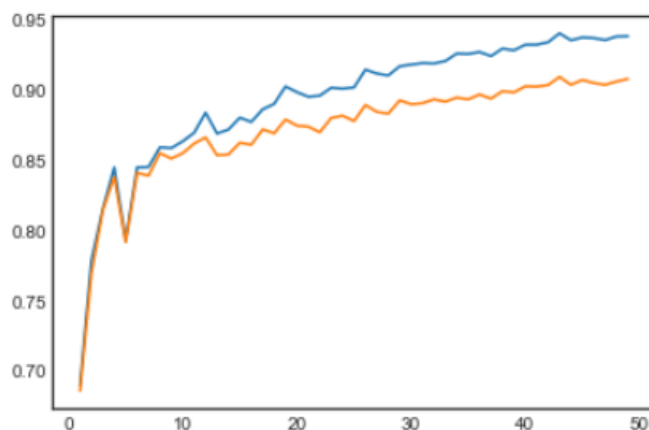
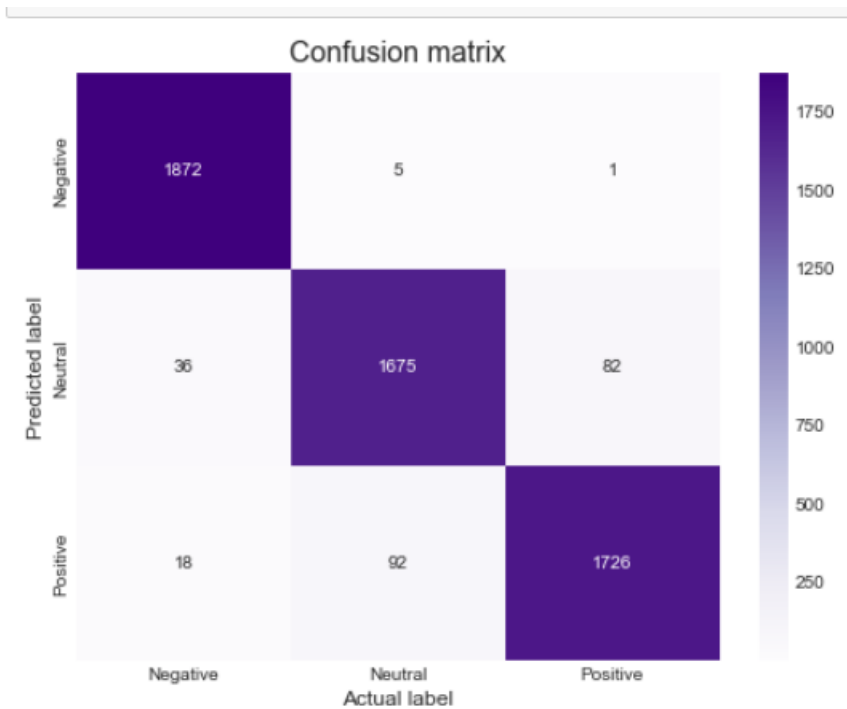
For N instances in the training set, sample N cases at random with replacement. The resulting set will be the training set of the tree.

For M input features, a number $K \ll M$ is specified such that at each node, a subset of K features is drawn at random, and among which the best split is selected.

Each tree is grown to its maximum size and unpruned.

The idea of our experiments is to tune the RF main parameters in order to analyse the correlation between the RF performances and the parameter values. In this section, we first detail the parameters studied in our experiments and we explain the way they have been tuned. We then present our experiment protocol, by describing the database, the test procedure, the results recorded, and the features extraction technique used.

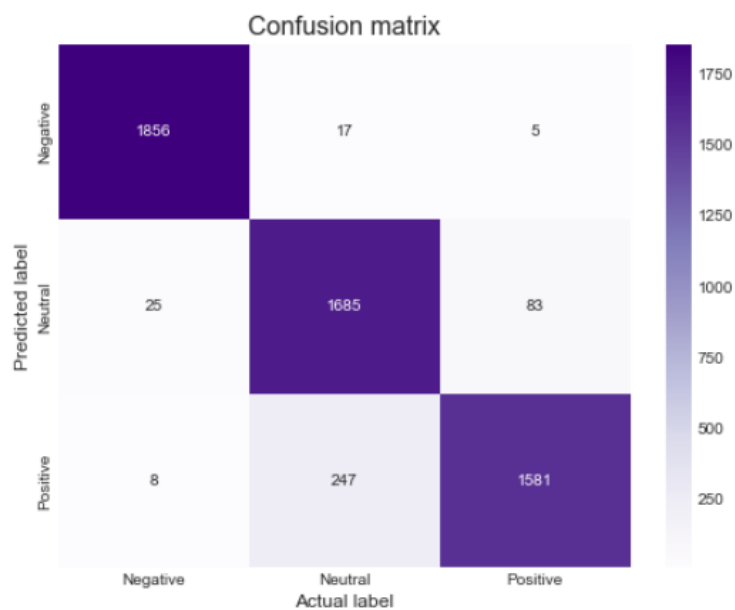
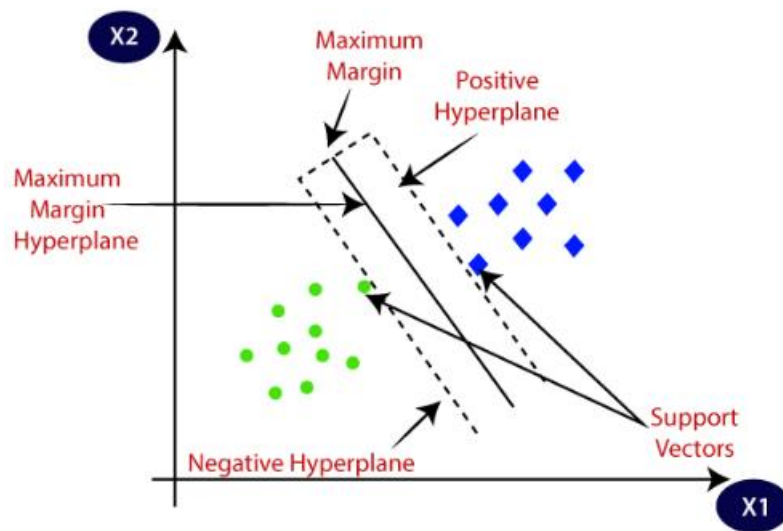
0.9607771926638823					
	precision	recall	f1-score	support	
0	0.98	1.00	0.99	1827	
1	0.94	0.95	0.94	1795	
2	0.96	0.94	0.95	1885	
accuracy			0.96	5507	
macro avg	0.96	0.96	0.96	5507	
weighted avg	0.96	0.96	0.96	5507	



Support Vector Machine :

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

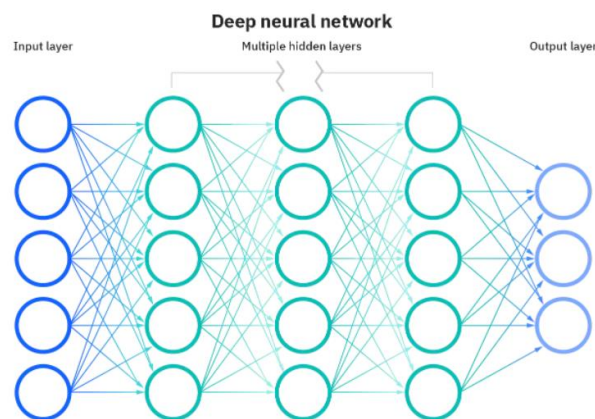
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which two different categories are classified using a decision boundary or hyperplane:



0.9300889776647903					
	precision	recall	f1-score	support	
0	0.98	0.99	0.99	1878	
1	0.86	0.94	0.90	1793	
2	0.95	0.86	0.90	1836	
accuracy			0.93	5507	
macro avg	0.93	0.93	0.93	5507	
weighted avg	0.93	0.93	0.93	5507	

Neural Networks:

Neural networks are a class of machine learning algorithms used to model complex patterns in datasets using multiple hidden layers and non-linear activation functions. Artificial neural networks (ANNs) are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next

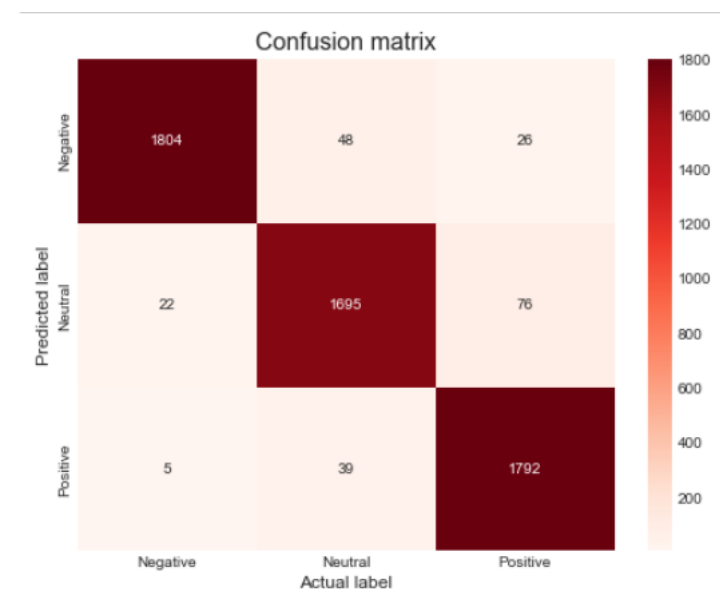


layer of the network.

Neural Networks take in the weights of connections between neurons. The weights are balanced, learning data point in the wake of learning data point. When all weights are trained, the neural network can be utilized to predict the class or a quantity, if there should arise an occurrence of regression of a new input data point. With Neural networks, extremely complex models can be trained and they can be utilized as a kind of black box, without playing

out an unpredictable complex feature engineering before training the model. Joined with the “deep approach” even more unpredictable models can be picked up to realize new possibilities.

For our project, we will be using an MLP classifier from the scikit library. MLPClassifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of classification.



```
0.9607771926638823
```

	precision	recall	f1-score	support
0	0.99	0.96	0.97	1878
1	0.95	0.95	0.95	1793
2	0.95	0.98	0.96	1836
accuracy			0.96	5507
macro avg	0.96	0.96	0.96	5507
weighted avg	0.96	0.96	0.96	5507

Results:

Below is the comparison of all the models.

Model	Accuracy	Precision	Recall	F1 score
Naïve bayes	0.86	0.89	0.89	0.89
Logistic Regression	0.91	0.93	0.81	0.87
KNN	0.65	0.73	0.91	0.81
Decision Tree	0.94	0.92	0.92	0.92
Random Forest	0.961	0.96	0.94	0.95
SVM	0.93	0.95	0.86	0.80
Neural Network	0.96	0.95	0.98	0.96

We can observe that Random Forest and neural network have the best accuracy among the models. Even the other models perform pretty well for this dataset.

References:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609776>

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6623713>

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8567243>

<https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>

<https://www.geeksforgeeks.org/machine-learning>

<https://www.javatpoint.com/machine-learning>

<https://scikit-learn.org/stable/>

<https://www.ibm.com/cloud/learn/neural-networks>

<https://www.tutorialspoint.com/>

