Twitter Sentiment Prediction of US Airlines

Milestone: Project Report

Group 12

Student-1 Name : Amitoj Singh Kohli

Student-2 Name : Aryaman Kakad

857-313-0650 (Tel of Student 1)

857-313-5234 (Tel of Student 2)

kohli.am@northeastern.edu

kakad.a@northeastern.edu

Percentage of Effort Contributed by Student1: 100%

Percentage of Effort Contributed by Student2: 100%

Signature of Student 1: Amitoj

Signature of Student 2: AKakad

Submission Date: 04 / 23 / 2023

Business Problem Definition:

Airline companies in the United States face a significant challenge in understanding and managing customer satisfaction. Negative sentiment towards an airline can lead to decreased bookings and revenue. Social media platforms, such as Twitter, provide extensive customer feedback on airline service quality that can be used to gain insights into customer sentiment toward the airlines. However, the sheer volume of data can make it challenging to manually analyze and understand the overall customer sentiment.

Problem Setting:

This project aims to predict the sentiment of tweets about US airlines using supervised machine learning techniques. It will understand customer perceptions and predict the sentiment of tweets as positive, negative, or neutral. This model will help airlines make data-driven decisions to identify areas of improvement and tailor their services to meet customer needs better.

Data Source and Description:

- Dataset is obtained from Kaggle, and the link is dataset link.
- The original source of this data is *Crowdflower's Data for Everyone library*.
- The dataset contains tweets expressing positive, negative, and neutral sentiments towards 6 US Airlines. The 6 US airlines are American airline, Delta, US airways, United airline, Virgin airline, and Southwest airline.
- The data contains tweets from users from the date 16th February 2015 to 24th February 2015 i.e., 9 days of tweets from users. The tweets on a specific day are as follows: 4, 1408, 1344, 1376, 1500, 1557, 3079, 3028 and 1344 for these 9 days.
- The shape of data frame is (14640, 15).
- The dataset contains variables like tweet_id, airline_sentiment, airline_sentiment_confidence, negativereason, negativereason_confidence, airline, airline_sentiment_gold, name, negativereason_gold, retweet_count, text, tweet_coord, tweet_created tweet_location, user_timezone.

Solution Design:

The proposed solution uses supervised machine learning to predict the sentiment of tweets about US airlines. The solution will consist of the following steps:

1. Cleaning and preprocessing the data for various data issues / error pattern.
2. Present the insights clearly and concisely using data visualization techniques.
3. Training a supervised machine learning model using labeled data on sentiment.
4. Using the trained model to predict the sentiment of tweets about US airlines.
5. Continuously evaluate the model's performance and make necessary improvements.

Data Exploration and Visualization:

It appears that many passengers are not having excellent flight experience through the visualization of count of tweets for each sentiment type. As a result, it is crucial to identify which airlines are excelling at satisfying their customers and which are falling apart. To accomplish this insight, this report shows the proportion of negative reviews for each airline.

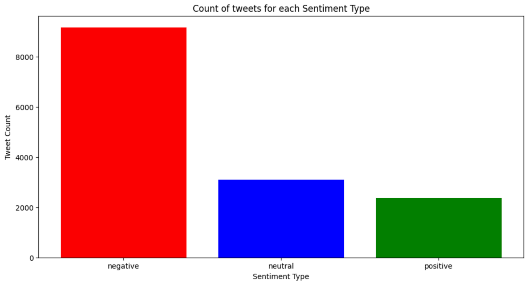


Figure 1: Count of Tweets for Each Sentiment Type

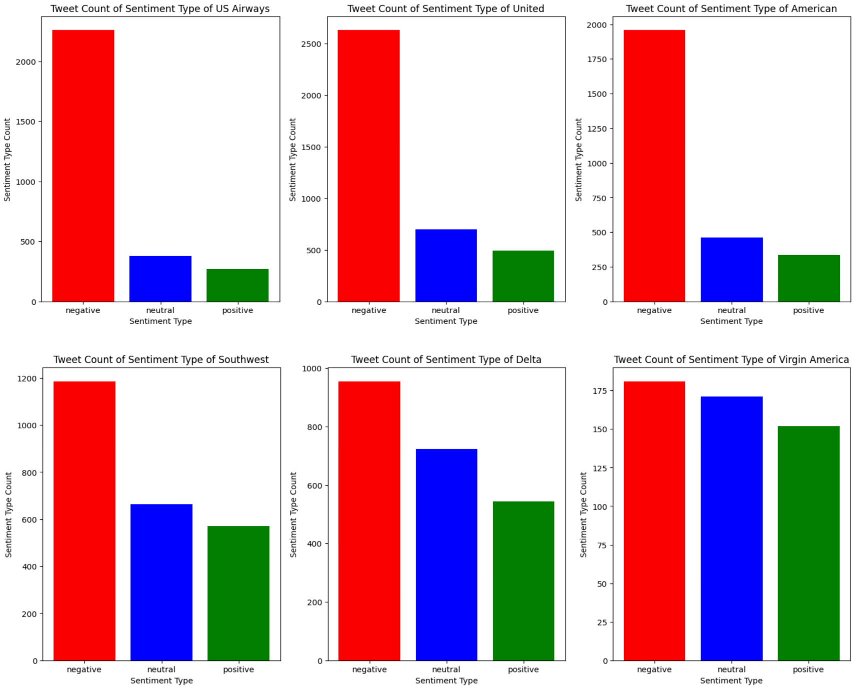


Figure 2: Tweet Count of Sentiment Type for Each Airline

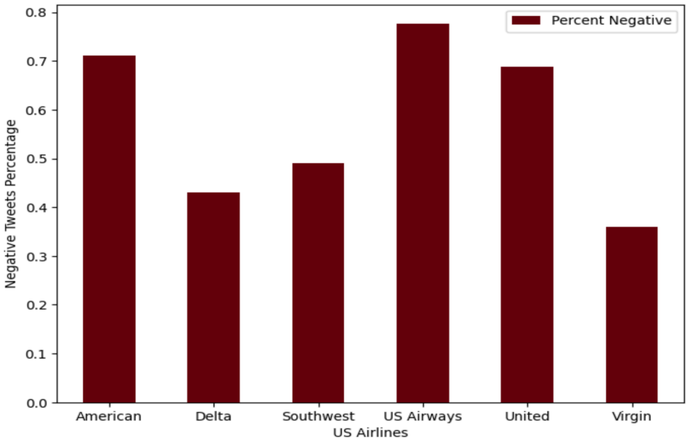


Figure 3: Negative Tweets Percentage for Each Airline

It is found that Southwest Airline has the most balanced reviews in terms of sentiment of the customer experience. Whereas, US Airways have around 80% of negative sentiment from the customers travelling through their airlines. Among the six airlines, Virgin Airlines seems to be the best in terms of sentiments from the customers.

The top 5 reasons for negative sentiments are:

- Customer service issue
- Late flight
- Can’t tell.
- Cancelled flight.
- Lost luggage.

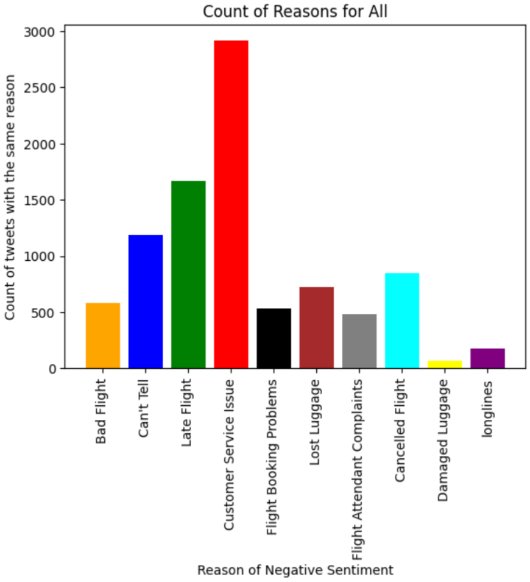


Figure 4: Count of Negative Reasons for All Airlines

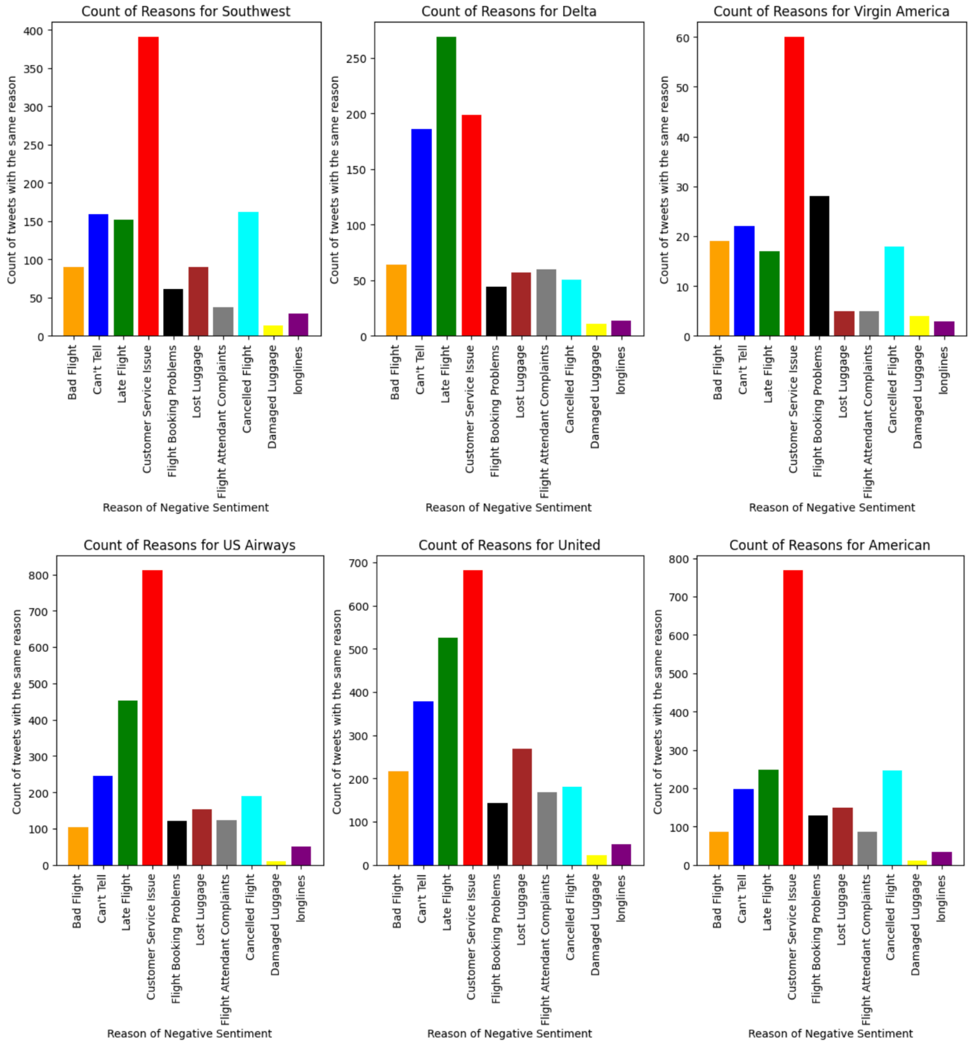


Figure 5: Count of Negative Reasons for Each Airlines

From the above visualization, it is found that Delta Airlines main reason for negative sentiment is late flight. The main reason for negative sentiment for US Airways, United, American, Southwest, Virgin America is Customer Service issue. Virgin America airline's highest count for each of the negative sentiment reason is not more than 60. Unlike Virgin America, airlines such as United Airline, US Airways, and American Airline are burdened with over than 500 count of negative sentiments, ranging from delayed flights to poor customer service.

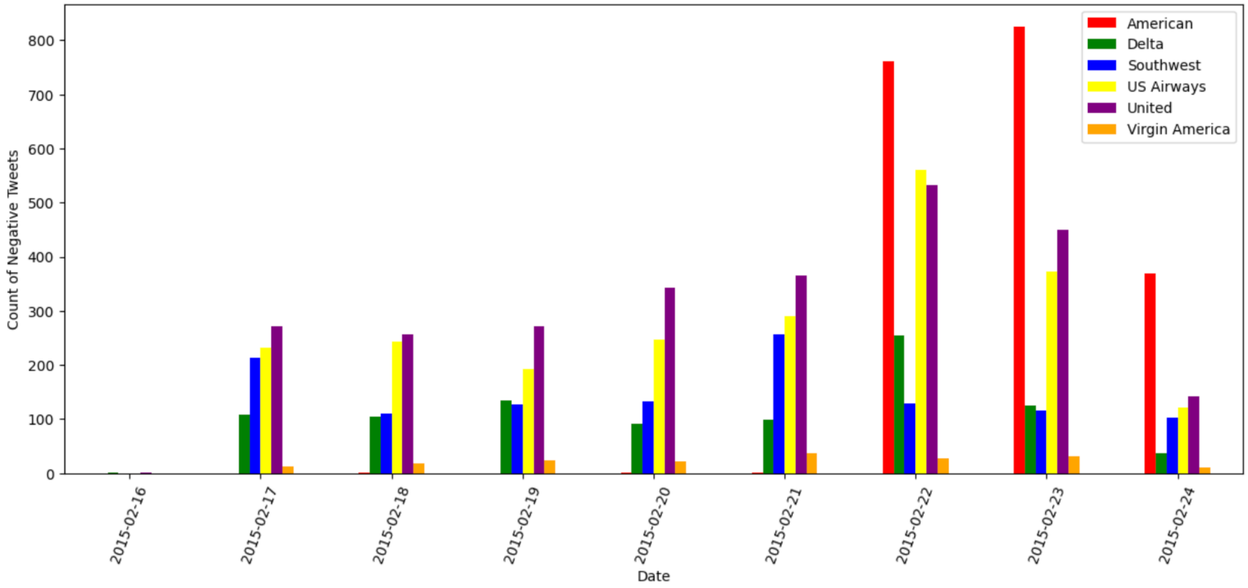


Figure 6: Relationship Between Date and Negative Sentiment

By visualizing the date's impact on tweet sentiments (especially negative ones!), we can derive diverse conclusions. It'll be captivating to observe whether the date influences the tone of tweets. Most of the airlines had a slightly higher proportion of negative tweets towards the end of the week. In contrast, Virgin America had the fewest negative tweets compared to other airlines in the weekly data, which could be due to their significantly lower total number of tweets. On February 23rd, 2015, there was a sudden surge in negative sentiment tweets directed at American, but it decreased by half on the following day.

Data Mining and Cleaning:

Three columns, that is, ‘airline_sentiment_gold’, ‘negativereason_gold’, and ‘tweet_coord’ are having more than 90% of missing values. These columns do not contribute to determine the dependent variable ‘airline_sentiment’ and therefore, it is deleted. This dataset have 4 numerical predictors, out of which one is the tweet id which is a unique value for each of the tweet in the dataset. Columns such as 'airline_sentiment_confidence' and 'negativereason_confidence' have values between 0 and 1 to determine how confident each of the user was in tweeting their tweet about the airlines and about the tweet itself. The last numerical predictor is retweet count which contains the number of users who have retweeted a particular tweet and it is a discrete value, so there is no possibility of an outlier.

The data is cleaned, and the text is pre-processed using the TF-IDF (Term Frequency – Inverse Document Frequency) vectorizer, and the output is in the form of matrix. The resultant matrix is converted to a dataframe and then it is merged with columns like 'airline_sentiment_confidence', 'negativereason_confidence', 'retweet_count', 'airline_sentiment' from the original dataset. Additionally, the rows with the missing values were dropped.

The dataset became unbalance upon dropping the null values. A function called 'compute_class_weight' is used to deal with this issue. This function computes the weight of each class. This variable was then used in ML models like Logistic Regression, Random Forest Classifier, and Support Vector classifier.

Data Mining Models:

As the business problem is related to Classification Supervised Machine Learning. We have applied seven techniques related to classification supervised learning.

1. Naïve Bayes Classification
2. Random Forest Classification
3. Logistic Regression
4. K-Nearest Neighbors Classification
5. Support Vector Classifier
6. Linear Discriminant Analysis
7. Neural Network

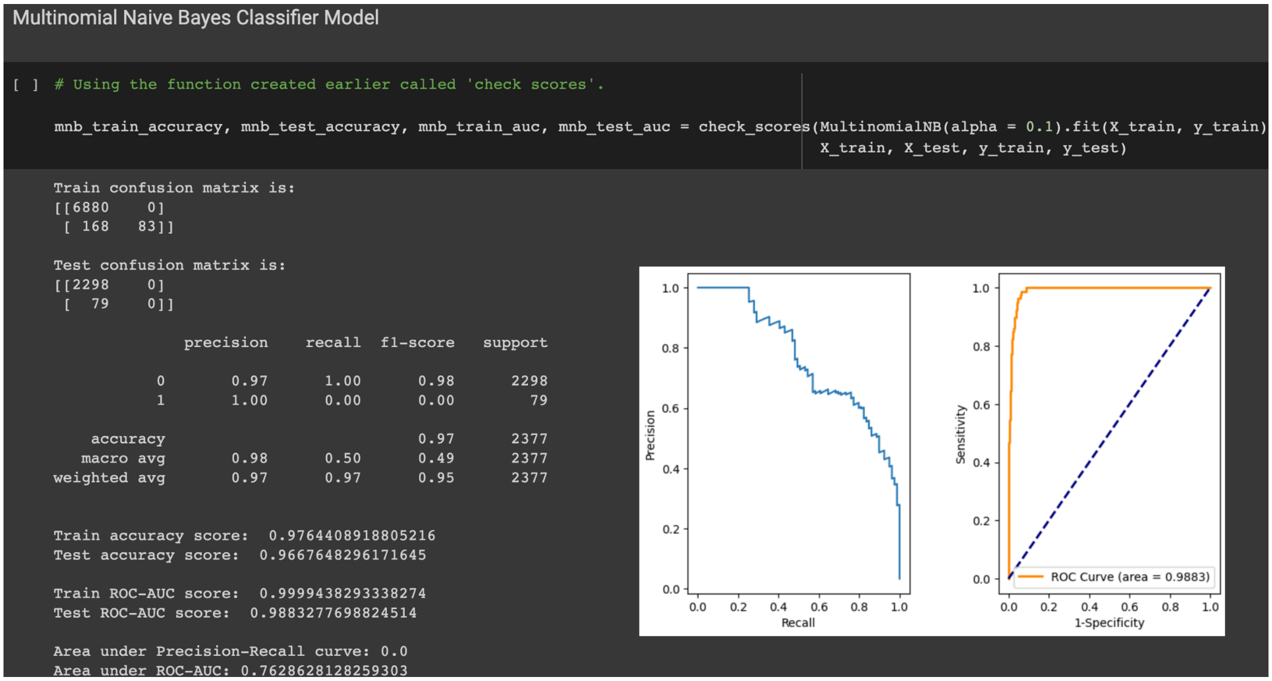
The training and validation accuracy of each of these models as well as the classification summary which includes accuracy, precision, recall and f-1 score. The graphs of the Precision vs Recall and ROC-AUC Curve for each of the models is plotted to decide which out these seven models is the best for our classification problem.

1. Naïve Bayes Classification:

Naive Bayes Classification is a common Sentiment Analysis technique. The chance of a specific tweet or document belonging to a specific class (positive, negative, or neutral) is estimated using Bayes' Theorem in this method. The primary benefit of Naive Bayes Classification is its ease of use and efficiency in processing vast amounts of text data. In addition, it is less prone to overfitting than other more sophisticated algorithms. Furthermore, Naive Bayes can handle many features, making it perfect for text classification issues.

However, there are several disadvantages to employing Naive Bayes for Sentiment Analysis. The algorithm presupposes that the features are independent, which in natural language literature may not always be the case. Furthermore, the efficacy of Naive Bayes is largely dependent on the data quality and feature selection method. It may not perform well with noisy or incomplete data, and feature selection necessitates thorough evaluation of the domain and type of text data to be studied.

Overall, because of its simplicity, efficiency, and capacity to handle enormous volumes of text data, Naive Bayes Classification is a strong candidate for Sentiment Analysis. To obtain the optimum results, however, the data quality and feature selection procedure must be carefully considered.



Multinomial Naive Bayes Classifier Model

```
[ ] # Using the function created earlier called 'check scores'.

    mnb_train_accuracy, mnb_test_accuracy, mnb_train_auc, mnb_test_auc = check_scores(MultinomialNB(alpha = 0.1).fit(X_train, y_train)
                                                                                    X_train, X_test, y_train, y_test)
```

```
Train confusion matrix is:
[[6880    0]
 [ 168   83]]

Test confusion matrix is:
[[2298    0]
 [  79    0]]

              precision    recall  f1-score   support

           0       0.97      1.00      0.98      2298
           1       1.00      0.00      0.00        79

    accuracy                           0.97      2377
   macro avg       0.98      0.50      0.49      2377
weighted avg       0.97      0.97      0.95      2377


Train accuracy score:  0.9764408918805216
Test accuracy score:  0.9667648296171645

Train ROC-AUC score:  0.9999438293338274
Test ROC-AUC score:  0.9883277698824514

Area under Precision-Recall curve: 0.0
Area under ROC-AUC: 0.7628628128259303
```

Figure 7: Naïve Bayes Classifier Model

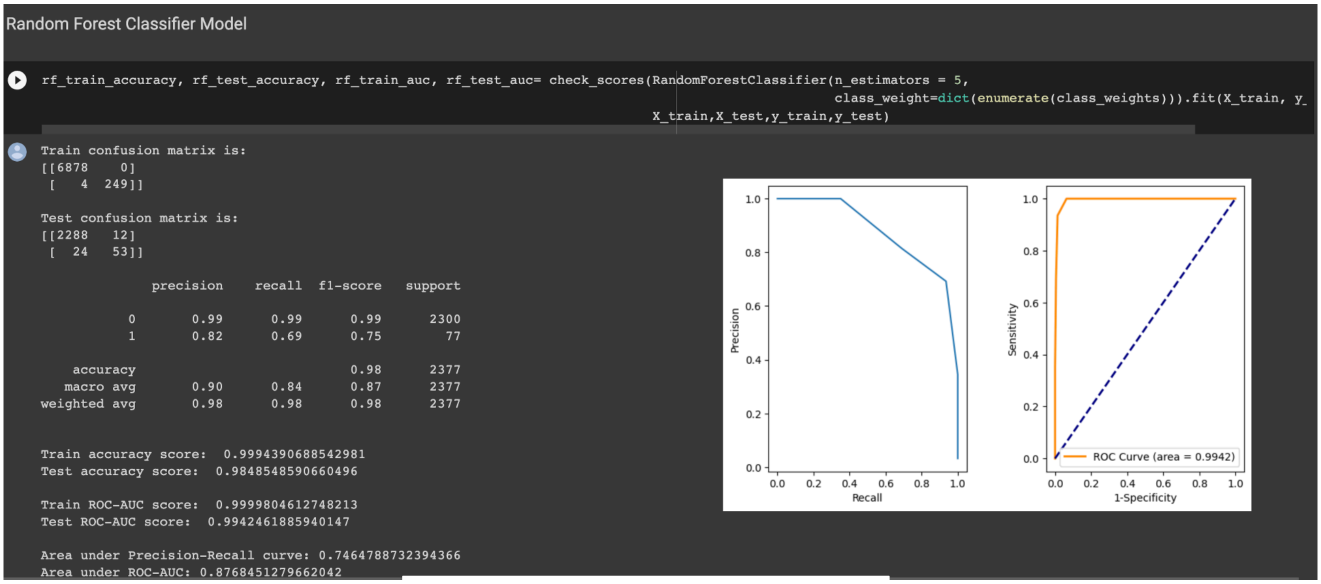
2. Random Forest Classification:

The Random Forest Classification is a commonly utilized machine learning technique in sentiment analysis, which refers to the process of examining textual data to identify the underlying sentiment or viewpoint. In this approach, the algorithm creates multiple decision trees, each using a random subset of the input data and features. The output of the algorithm is a vote from each decision tree, which is used to determine the final classification into one of the three sentiment classes - positive, negative, or neutral.

Random Forest Classification has several advantages for sentiment analysis. Firstly, it can handle many features and can identify important features for classification. This is particularly useful in sentiment analysis as it can help identify key words or phrases that indicate positive, negative, or neutral sentiment. Secondly, it has a lower likelihood of overfitting, which can happen when a model is overly complicated and achieves good performance on the training data but performs poorly on novel, unseen data. Thirdly, it can handle imbalanced classes, where there are more instances of one class than the other. This is useful in sentiment analysis as data is often imbalanced, with more instances of neutral sentiment than positive or negative sentiment.

However, there are also some limitations to using Random Forest Classification for sentiment analysis. Firstly, it can be computationally expensive and may require significant computing resources to train and evaluate the model. Secondly, it can be difficult to interpret the results of the model, as the final classification is based on the votes of multiple decision trees rather than a single decision. Lastly, it may not perform as well as other algorithms when dealing with very high-dimensional data, where the number of features is very large.

To sum up, the Random Forest Classification is a potent machine learning method that can manage various classification assignments, which includes sentiment analysis involving three categories: positive, negative, and neutral. While it has several advantages for sentiment analysis, including the ability to handle large feature spaces and imbalanced classes, it also has some limitations, including computational complexity and interpretability.



Random Forest Classifier Model

```
rf_train_accuracy, rf_test_accuracy, rf_train_auc, rf_test_auc= check_scores(RandomForestClassifier(n_estimators = 5,
                                                                                                      class_weight=dict(enumerate(class_weights))).fit(X_train, y_
                                                                                        X_train,X_test,y_train,y_test)
```

```
Train confusion matrix is:
[[6878    0]
 [   4  249]]

Test confusion matrix is:
[[2288   12]
 [  24   53]]

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2300
           1       0.82      0.69      0.75        77

    accuracy                           0.98      2377
   macro avg       0.90      0.84      0.87      2377
weighted avg       0.98      0.98      0.98      2377


Train accuracy score:  0.9994390688542981
Test accuracy score:  0.9848548590660496

Train ROC-AUC score:  0.9999804612748213
Test ROC-AUC score:  0.9942461885940147

Area under Precision-Recall curve: 0.7464788732394366
Area under ROC-AUC: 0.8768451279662042
```

Figure 8: Random Forest Classification Model

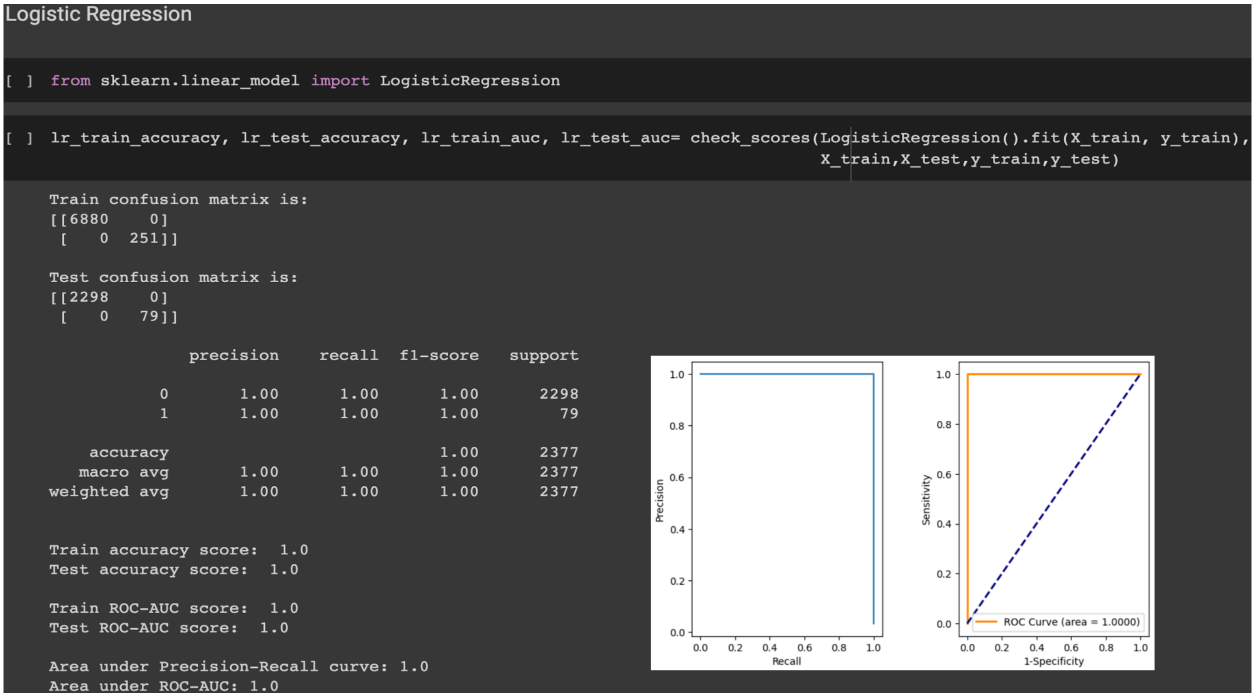
3. Logistic Regression Model:

Logistic Regression Model is a commonly used machine learning algorithm for sentiment analysis, which involves analyzing text data to determine the underlying sentiment or opinion. In this approach, the algorithm creates a model that predicts the probability of a text belonging to each sentiment class - positive, negative, or neutral - based on the input features.

Logistic Regression Model has several advantages for sentiment analysis. The first advantage is that it is relatively simple to execute and comprehend, which makes it a commonly preferred option for novice machine learning practitioners. Moreover, it can accommodate both linear and non-linear connections between the input features and the output categories. Thirdly, it can produce probabilistic outputs, which can be useful in some applications where confidence scores are required.

However, there are also some limitations to using Logistic Regression Model for sentiment analysis. Firstly, it can struggle with high-dimensional data, where the number of features is very large, and can suffer from overfitting. Another aspect is that it presupposes that the input features are unrelated to each other, which may not reflect real-world scenarios. Lastly, it may not perform as well as other algorithms, such as Random Forest or Support Vector Machines, when dealing with imbalanced classes or non-linear relationships between the features and the output.

In summary, Logistic Regression Model is a popular and useful machine learning algorithm for sentiment analysis with several advantages, including ease of implementation and interpretation and the capability to manage non-linear associations between input features and output classes. However, it also has some limitations, including difficulty handling high-dimensional data and non-linear relationships between features and output, as well as the assumption of feature independence.



Logistic Regression

```
[ ]  from sklearn.linear_model import LogisticRegression
```

```
[ ]  lr_train_accuracy, lr_test_accuracy, lr_train_auc, lr_test_auc= check_scores(LogisticRegression().fit(X_train, y_train),
                                                                                  X_train,X_test,y_train,y_test)
```

```
Train confusion matrix is:
[[6880    0]
 [   0  251]]

Test confusion matrix is:
[[2298    0]
 [   0   79]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2298
           1       1.00      1.00      1.00        79

    accuracy                           1.00      2377
   macro avg       1.00      1.00      1.00      2377
weighted avg       1.00      1.00      1.00      2377


Train accuracy score:  1.0
Test accuracy score:  1.0

Train ROC-AUC score:  1.0
Test ROC-AUC score:  1.0

Area under Precision-Recall curve: 1.0
Area under ROC-AUC: 1.0
```

Figure 9: Logistic Regression Model

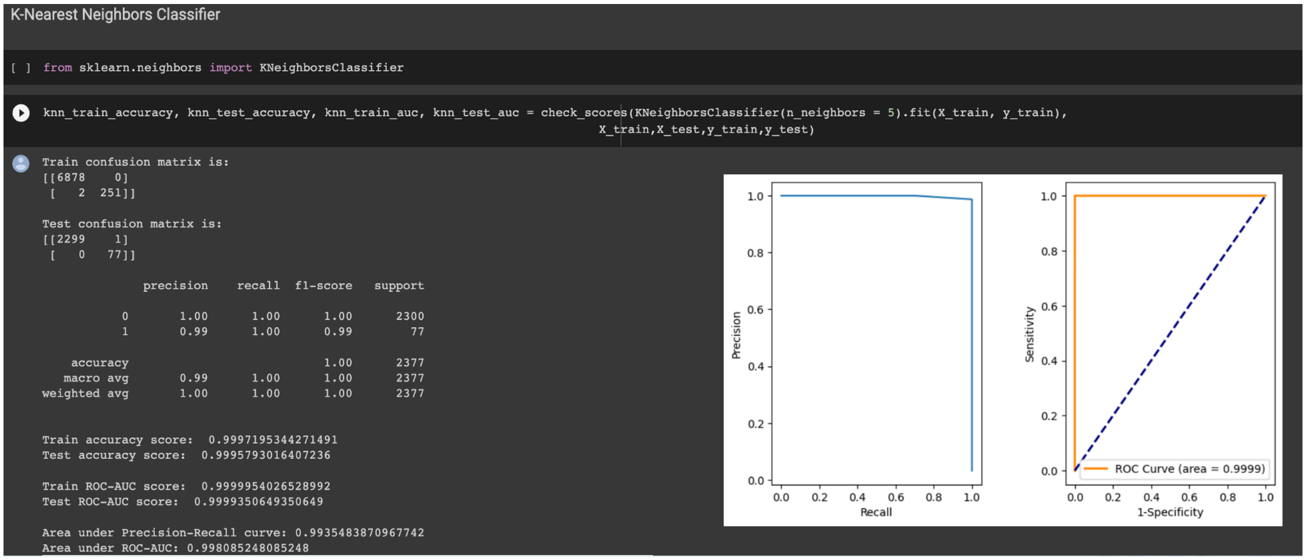
4. K-Nearest Neighbors Classification

K-Nearest Neighbors (KNN) Classification is one of the machine learning technique that is often used in sentiment analysis, which involves analyzing text data to determine the underlying sentiment or opinion. In this approach, the algorithm classifies a new input text by finding the K closest training instances based on their feature similarity and assigning the most common class label among those K instances as the predicted sentiment class - positive, negative, or neutral.

K-Nearest Neighbors Classification has several advantages for sentiment analysis. Firstly, it is a simple algorithm that is easy to implement and interpret, making it a popular choice for beginners in machine learning. Secondly, it can handle non-linear relationships between features and the output classes, which can be useful in sentiment analysis, where the relationship between words and sentiment can be complex. Thirdly, it is a lazy learning algorithm, meaning that the training data is used directly during the classification phase, making it quick to train and allowing for real-time classification of new text data.

However, there are also some limitations to using K-Nearest Neighbors Classification for sentiment analysis. Firstly, it can be computationally expensive, especially when the training dataset is large, as it demands the computation of distances between each novel input and every training example. Additionally, it can be influenced by the selection of distance measurement and the K value, which can affect the overall effectiveness of the algorithm. Lastly, it may not perform well when dealing with imbalanced classes or high-dimensional data, where the number of features is very large.

In summary, K-Nearest Neighbors Classification is a simple and flexible machine learning algorithm that can be useful for sentiment analysis with several advantages, including the ability to handle non-linear relationships between features and output classes and real-time classification. However, it also has some limitations, including computational complexity, sensitivity to distance metric and K value, and difficulty handling imbalanced classes and high-dimensional data.



K-Nearest Neighbors Classifier

```
[ ] from sklearn.neighbors import KNeighborsClassifier
```

```
knn_train_accuracy, knn_test_accuracy, knn_train_auc, knn_test_auc = check_scores(KNeighborsClassifier(n_neighbors = 5).fit(X_train, y_train),
                                                                                   X_train,X_test,y_train,y_test)
```

```
Train confusion matrix is:
[[6878    0]
 [   2  251]]

Test confusion matrix is:
[[2299    1]
 [   0   77]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2300
           1       0.99      1.00      0.99        77

    accuracy                           1.00      2377
   macro avg       0.99      1.00      1.00      2377
weighted avg       1.00      1.00      1.00      2377


Train accuracy score:  0.9997195344271491
Test accuracy score:  0.9995793016407236

Train ROC-AUC score:  0.9999954026528992
Test ROC-AUC score:  0.9999350649350649

Area under Precision-Recall curve: 0.9935483870967742
Area under ROC-AUC: 0.998085248085248
```

Figure 10: K-Nearest Neighbors Classification

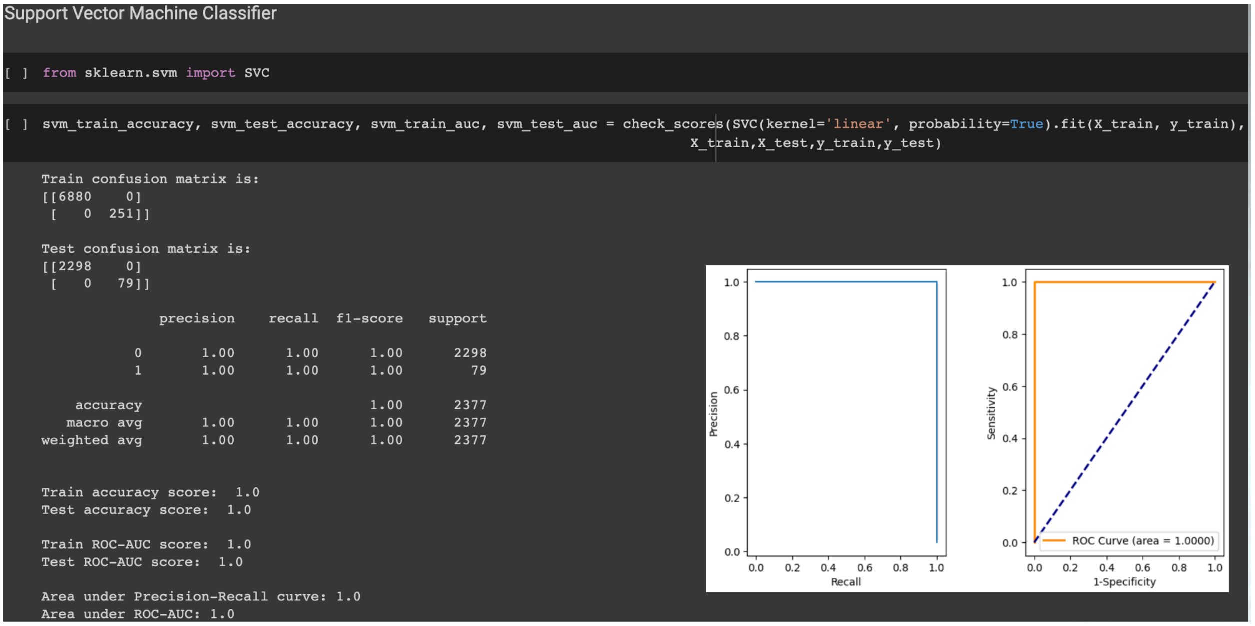
5. Support Vector Machines

Support Vector Machines (SVMs) are a popular machine learning algorithm used in sentiment analysis, which involves analyzing text data to determine the underlying sentiment or opinion. In this approach, the algorithm creates a hyperplane that separates the different classes - positive, negative, or neutral - by maximizing the margin between them.

SVMs have several advantages for sentiment analysis. Firstly, they are effective at handling non-linear relationships between the input features and the output classes, which is useful in sentiment analysis, where the relationship between words and sentiment can be complex. Secondly, they can handle high-dimensional data well, where the number of features is very large, by using a kernel function to project the data into a higher-dimensional space. Thirdly, they are less prone to overfitting than other algorithms, such as Logistic Regression, which can improve their generalization performance.

However, there are also some limitations to using SVMs for sentiment analysis. Firstly, they can be computationally expensive, especially when the dataset is large, or the kernel function is complex. Secondly, they can be sensitive to the choice of hyperparameters, such as the kernel function and regularization parameter, which can impact their performance. Lastly, they may not perform well when dealing with imbalanced classes or noisy data.

In summary, Support Vector Machines are a powerful and effective machine learning algorithm for sentiment analysis with several advantages, including the ability to handle non-linear relationships and high-dimensional data and less prone to overfitting. However, they also have some limitations, including computational complexity, sensitivity to hyperparameters, and difficulty handling imbalanced classes and noisy data.



Support Vector Machine Classifier

```
[ ]  from sklearn.svm import SVC
```

```
[ ]  svm_train_accuracy, svm_test_accuracy, svm_train_auc, svm_test_auc = check_scores(SVC(kernel='linear', probability=True).fit(X_train, y_train),
                                                                                    X_train,X_test,y_train,y_test)
```

```
    Train confusion matrix is:
    [[6880    0]
     [   0  251]]

    Test confusion matrix is:
    [[2298    0]
     [   0   79]]

                  precision    recall  f1-score   support

               0       1.00      1.00      1.00      2298
               1       1.00      1.00      1.00        79

        accuracy                           1.00      2377
       macro avg       1.00      1.00      1.00      2377
    weighted avg       1.00      1.00      1.00      2377


    Train accuracy score:  1.0
    Test accuracy score:  1.0

    Train ROC-AUC score:  1.0
    Test ROC-AUC score:  1.0

    Area under Precision-Recall curve: 1.0
    Area under ROC-AUC: 1.0
```

Figure 11: Support Vector Machines

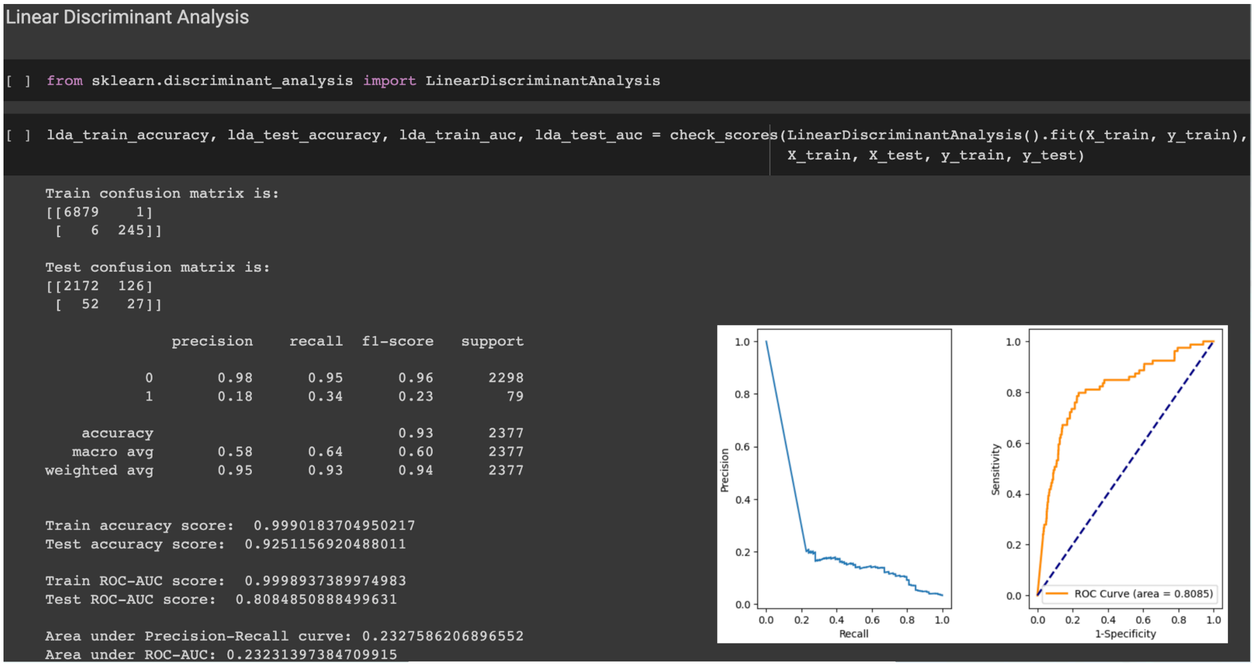
6. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a machine learning algorithm that is sometimes used in sentiment analysis, which involves analyzing text data to determine the underlying sentiment or opinion. In this approach, LDA creates a linear combination of input features that maximizes the separation between different classes - positive, negative, or neutral - while minimizing the variance within each class.

LDA has several advantages for sentiment analysis. Firstly, it is a simple algorithm that is easy to interpret and implement, making it a popular choice for beginners in machine learning. Secondly, it can handle high-dimensional data well, where the number of features is very large, by projecting the data into a lower-dimensional space while retaining the most relevant information. Thirdly, it assumes a normal distribution of input features, which can be useful in sentiment analysis, where some features (e.g. word frequencies) may follow a normal distribution.

However, there are also some limitations to using LDA for sentiment analysis. Firstly, it assumes that the input features are normally distributed, which may not always be the case in practice. Secondly, it is a linear algorithm, which means it may not capture complex relationships between features and the output classes as well as non-linear algorithms. Thirdly, it assumes that the covariance of input features is equal across all classes, which may not always be true in practice.

In summary, Linear Discriminant Analysis is a simple and effective machine learning algorithm for sentiment analysis with several advantages, including the ability to handle high-dimensional data and a normal distribution of input features. However, it also has some limitations, including the assumption of normally distributed input features, linear relationships between features and output classes, and equal covariance across all classes.



Linear Discriminant Analysis

```
[ ]  from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
```

```
[ ]  lda_train_accuracy, lda_test_accuracy, lda_train_auc, lda_test_auc = check_scores(LinearDiscriminantAnalysis().fit(X_train, y_train),
                                                                                        X_train, X_test, y_train, y_test)
```

```
Train confusion matrix is:
[[6879    1]
 [   6  245]]

Test confusion matrix is:
[[2172  126]
 [  52   27]]

              precision    recall  f1-score   support

           0       0.98      0.95      0.96      2298
           1       0.18      0.34      0.23        79

    accuracy                           0.93      2377
   macro avg       0.58      0.64      0.60      2377
weighted avg       0.95      0.93      0.94      2377


Train accuracy score:  0.9990183704950217
Test accuracy score:  0.9251156920488011

Train ROC-AUC score:  0.9998937389974983
Test ROC-AUC score:  0.8084850888499631

Area under Precision-Recall curve: 0.2327586206896552
Area under ROC-AUC: 0.23231397384709915
```

Figure 12: Linear Discriminant Analysis

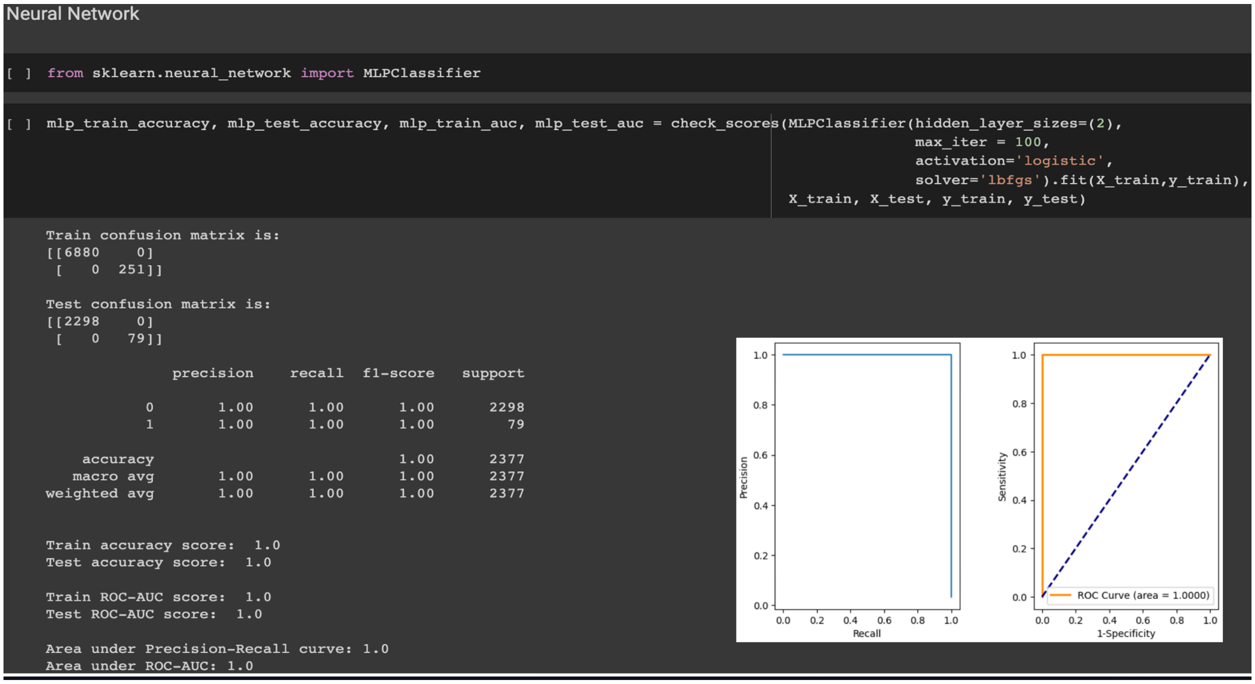
7. Neural Network

Neural networks are a popular machine learning algorithm used in sentiment analysis, which involves analyzing text data to determine the underlying sentiment or opinion. In this approach, a neural network is trained on a large dataset of labeled text data, where each text is associated with a sentiment - positive, negative, or neutral.

Neural networks have several advantages for sentiment analysis. Firstly, they can capture complex relationships between input features and output classes, which can improve their performance compared to simpler algorithms. Secondly, they can handle high-dimensional data well, where the number of features is very large, by using multiple layers of hidden nodes to extract relevant information from the input. Thirdly, they can be adapted to different types of text data, such as tweets or product reviews, by modifying the architecture of the network and the preprocessing steps.

However, there are also some limitations to using neural networks for sentiment analysis. Firstly, they can be computationally expensive, especially when the network is large, or the dataset is large. Secondly, they can be prone to overfitting if the network is too complex or if the dataset is small. Thirdly, they can pose difficulties in interpretation, which can make it difficult to comprehend the reason on how the network is making its predictions.

In summary, Neural networks are a powerful and flexible machine learning algorithm for sentiment analysis with several advantages, including the ability to capture complex relationships, handle high-dimensional data and adapt to different types of text data. However, they also have some limitations, including computational complexity, overfitting, and interpretability.



Neural Network

```
[ ]  from sklearn.neural_network import MLPClassifier
```

```
[ ]  mlp_train_accuracy, mlp_test_accuracy, mlp_train_auc, mlp_test_auc = check_scores(MLPClassifier(hidden_layer_sizes=(2),
                                                                                                   max_iter = 100,
                                                                                                   activation='logistic',
                                                                                                   solver='lbfgs').fit(X_train,y_train),
                                                                                    X_train, X_test, y_train, y_test)
```

```
Train confusion matrix is:
[[6880    0]
 [   0  251]]

Test confusion matrix is:
[[2298    0]
 [   0   79]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2298
           1       1.00      1.00      1.00        79

    accuracy                           1.00      2377
   macro avg       1.00      1.00      1.00      2377
weighted avg       1.00      1.00      1.00      2377


Train accuracy score:  1.0
Test accuracy score:  1.0

Train ROC-AUC score:  1.0
Test ROC-AUC score:  1.0

Area under Precision-Recall curve: 1.0
Area under ROC-AUC: 1.0
```

Figure 14: Neural Network

Project Results:

After looking at performance evaluation metrics like F-1 score, Accuracy, Precision, Recall, and the ROC-AUC curve, it is found that the top three models for our project is **Logistic Regression, Support Vector Machine (Classifier) and Neural Network.**

One of the major reasons for an accuracy of 1 for these 3 models, is because of the dataset being imbalanced. When using a TF-IDF vectorizer, it creates a sparse matrix containing 1's and 0's, therefore it is very easy for some of the classification models to classify the sentiment of the analysis as positive or negative based on the words used in the tweet.

The TF-IDF gives normalized term frequency values, which helps to reduce the impact of high-frequency words that may not be beneficial for class distinction. SVC and LR's linear character make them ideally suited for text classification tasks, since the relationships between features are frequently described using linear functions.

Impact of the Project Outcomes:

Potential improvements for this project can be to use a dataset that contains tweets for US airline passengers for more than 9 days which gives us an increased probability to make a robust classification model which does not over-fit on the data.

One key factor to investigate while selecting a dataset should be the nature of the dataset in terms of class labels. It would be highly beneficial to select a dataset which is balanced in nature in terms of its class labels.

Another improvement could be using additional classes like Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree so that the US airlines can address issues based on the priority of each of these classes.

Future Scope of the Project:

Deploy the trained model in the production environment and make predictions on new tweets in real-time.