

Project Report on:

Sentiment Analysis on Airline Tweets

Student ID: 893200535

Student Name: Md Saiful Islam Sajol

Louisiana State University

Fall 2022

Course Name: EXST 7142 Statistical Data Mining

Instructor: Dr. Bin Li

1. Introduction

In this project, I will analyze the user tweets about the airlines performance in the US which was taken from Twitter in 2015. I will clean and preprocess the data and try to fix the imbalance of the data and finally create four models to analyze Sentiment of the tweets, and that will return whether the tweets convey positive, neutral or negative sentiment.

This analysis can be useful for the airline company to know what are the reasons for the disappointments of their company and where they should focus to work on.

For this work Python with: Pandas, jupyter notebook and scikit learning libraries will be used.

2. Data description and data cleaning

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 original dataset was collected from <https://data.world/crowdflower/airline-twitter-sentiment>. Then it was processed using this repository <https://github.com/benhamner/crowdflower-airline-twitter-sentiment/blob/master/Makefile>.

The dataset has 14640 rows and 15 columns. The dataset has the following columns:

- 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

First 9 columns of the dataset:

airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name	negativereason_gold	retweet_count
neutral	1			Virgin America		cairdin		0
positive	0.3486		0	Virgin America		jnardino		0
neutral	0.6837			Virgin America		yvonnalynn		0
negative	1	Bad Flight	0.7033	Virgin America		jnardino		0
negative	1	Can't Tell	1	Virgin America		jnardino		0
negative	1	Can't Tell	0.6842	Virgin America		jnardino		0

Figure 1

Last 5 columns of the dataset:

text	tweet_coord	tweet_created	tweet_location	user_timezone
@VirginAmerica What @dhepburn said.		2/24/2015 11:35		Eastern Time (US & Canada)
@VirginAmerica plus you've added commercials to the experience... tacky.		2/24/2015 11:15		Pacific Time (US & Canada)
@VirginAmerica I didn't today... Must mean I need to take another trip!		2/24/2015 11:15	Lets Play	Central Time (US & Canada)
@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces & they have little recourse		2/24/2015 11:15		Pacific Time (US & Canada)
@VirginAmerica and it's a really big bad thing about it		2/24/2015 11:14		Pacific Time (US & Canada)
@VirginAmerica seriously would pay \$30 a flight for seats that didn't have this playing. it's really the only bad thing about flying VA		2/24/2015 11:14		Pacific Time (US & Canada)
@VirginAmerica yes, nearly every time I fly VX this æœear wormâ€ wonâ€™t go away :)		2/24/2015 11:13	San Francisco CA	Pacific Time (US & Canada)
@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https://t.co/mWpG7grEzP		2/24/2015 11:12	Los Angeles	Pacific Time (US & Canada)

Figure 2

The information of main attributes for this project as follows;

- **airline_sentiment** : Sentiment classification.(positive, neutral, and negative)
- **negativereason** : Reason selected for the negative opinion
- **airline** : Name of 6 US Airlines('Delta', 'United', 'Southwest', 'US Airways', 'Virgin America', 'American')
- **text** : Customer's opinion
- **airline** : Does not contain null data. I will add this feature in stopwords.
- **name** : Will add this feature in stopwords.

- **text** : Does not contain null data. Every text begins with @ due to the characteristic of twitter. In addition, we can see emoji on the text.
- **negativereason** : Blank if sentiment is not 'negative'.
- **airline_sentiment** : Does not contain null data.

Some variables have null values, but the key variables like tweet_id, airline_sentiment, and text do not contain any missing values

Figure 3 represents the number of counts for each column and their corresponding datatype.

```
1 df.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
```

Figure 3

The following table describes the mean, standard deviation, minimum and maximum value for each of the column

```
1 df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
tweet_id	14640.0	5.692184e+17	7.791112e+14	5.675883e+17	5.685592e+17	5.694779e+17	5.698905e+17	5.703106e+17
airline_sentiment_confidence	14640.0	9.001689e-01	1.628300e-01	3.350000e-01	6.923000e-01	1.000000e+00	1.000000e+00	1.000000e+00
negativereason_confidence	10522.0	6.382983e-01	3.304398e-01	0.000000e+00	3.606000e-01	6.706000e-01	1.000000e+00	1.000000e+00
retweet_count	14640.0	8.265027e-02	7.457782e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.400000e+01

Figure 4

I also calculated the number of null values presented in each column.

```
1 #checking null values in our data
2 df.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
dtype: int64	

Figure 5

Figure 6 shows number of null values in percentage form.

Percentage null or na values in the dataframe

```

1 print("Percentage null or na values in d
2 ((df.isnull() | df.isna()).sum() * 100 /

Percentage null or na values in df
tweet_id                0.00
airline_sentiment        0.00
airline_sentiment_confidence  0.00
negativereason           37.31
negativereason_confidence  28.13
airline                  0.00
airline_sentiment_gold    99.73
name                     0.00
negativereason_gold       99.78
retweet_count            0.00
text                     0.00
tweet_coord              93.04
tweet_created            0.00
tweet_location           32.33
user_timezone            32.92
dtype: float64

```

Figure 6

- **airline_sentiment_gold, negativereason_gold** have more than 99% missing data And **tweet_coord** have nearly 93% missing data.
- Additionally column: negativereason tweet_location, and user_timezone have more than 30% missing values.
- For better analysis I delete these columns as they will not provide any constructive information

3. Exploratory data analysis EDA

In this section I will try to show an in depth analysis that the dataset is bearing.

Figure 7 shows the total number of negative, neutral and positive tweets that have been made of the time duration.

```
Total number of sentiments of tweets :  
negative    9178  
neutral     3099  
positive    2363  
Name: airline_sentiment, dtype: int64
```

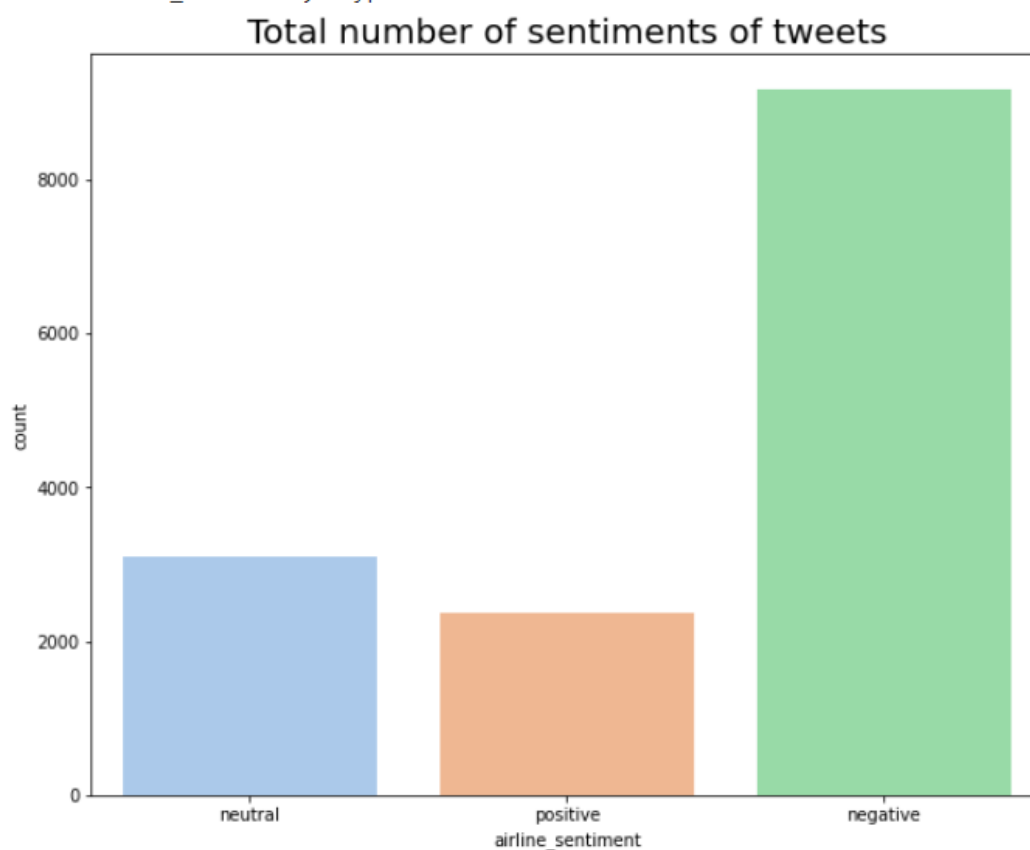


Figure 7

The above bar chart shows that majority (around 9000) of the tweets about the airlines carry negative sentiment. On the other hand, only around 2500 tweets carry positive reviews.

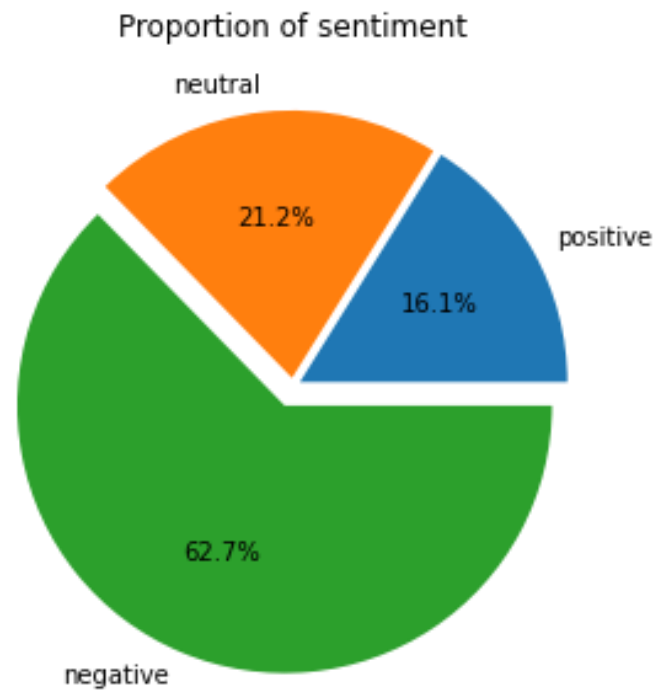


Figure 8

- The above pie chart (Figure 8) convey the same message in percentage form.
- Here it is seen that around 62.7% tweets convey negative sentiments, whereas only 21.2% percent tweets are neutral of point of view.

Total number of tweets for each airline

```
1 print(df.groupby('airline')['airline_sentiment'].count())
```

airline	airline_sentiment
American	2759
Delta	2222
Southwest	2420
US Airways	2913
United	3822
Virgin America	504

Name: airline_sentiment, dtype: int64

Figure 9

- The above code snippet (Figure 9) finds out which airlines people mostly tweet about.

- It can be seen that United airline has the most tweets, The US airways placing the second on the table.
- Figure 10 shows Virgin America has the least amount of tweets over that time duration.

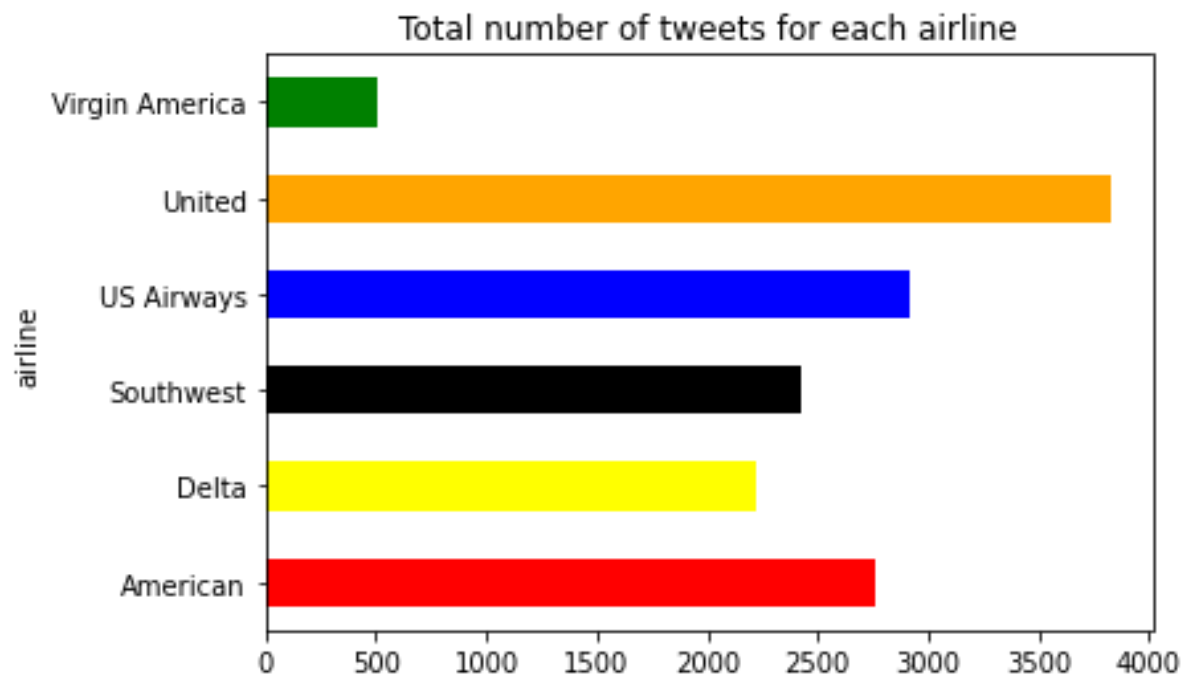


Figure 10

figure 11 shows the proportion of sentiment for each airlines.

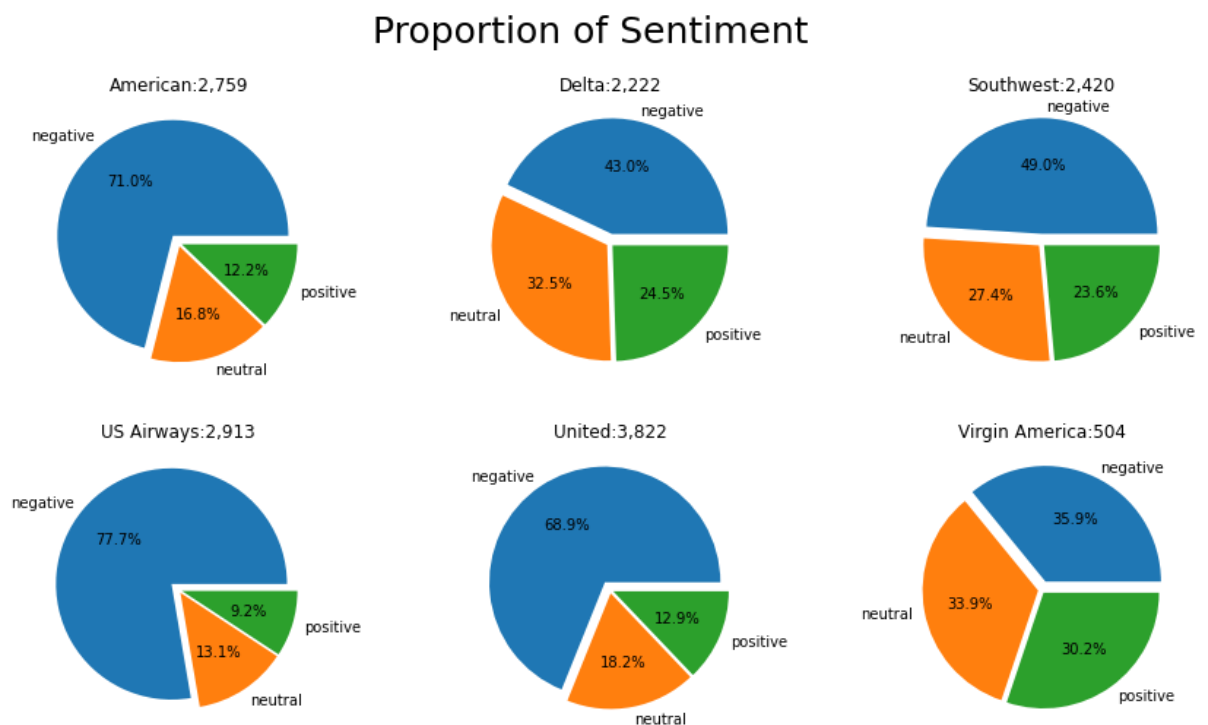


Figure 11

To conclude, the following information can be drawn by the above pie chart.

- In general, most of the airlines got negative reviews on Tweets
- The US Airways had the highest percentage of negative tweets and Virgin America had the lowest percentage of negative tweets.
- Virgin America had the highest percentage of positive tweets although it was not definitely the most popular airline in February 2015
- Proportion of sentiments for Virgin Airways are almost equal.

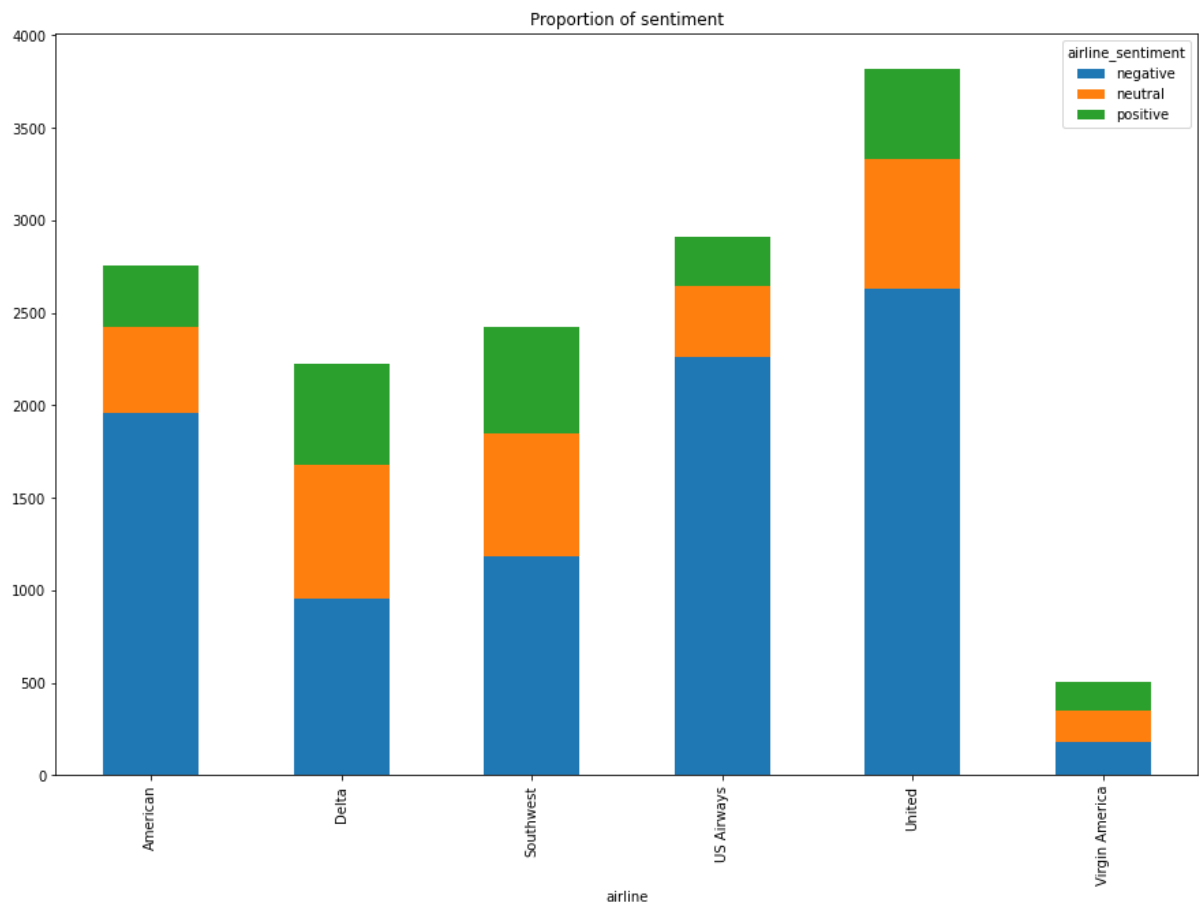


Figure 12

Figure 12 (The above bar chart) refers the same information in bar chart form



Virgin America : Out of total 504customers, 35.9% feel negative.

United : Out of total 3,822customers, 68.9% feel negative.

Southwest : Out of total 2,420customers, 49.0% feel negative.

Delta : Out of total 2,222customers, 43.0% feel negative.

US Airways : Out of total 2,913customers, 77.7% feel negative.

American : Out of total 2,759customers, 71.0% feel negative.

Figure 13

Top 5 Reasons Of Negative Tweets :
AxesSubplot(0.125,0.125;0.775x0.755)

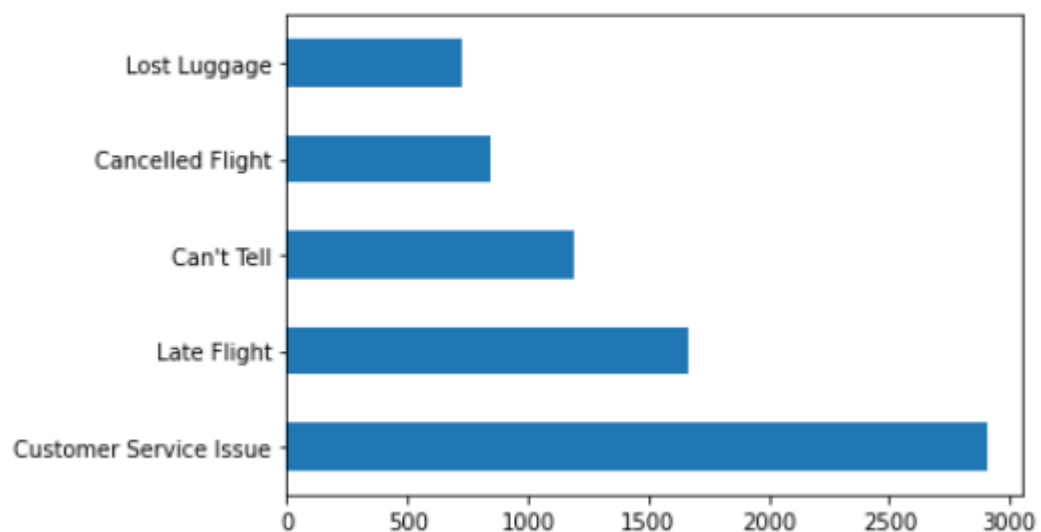


Figure 14

- This above bar chart (Figure 14) represents top 5 reasons of Negative tweets.
- Clearly most of people tweets negative for Customer Service Issue.
- Late flight and Can't Tell are the second and third reasons for customer dissatisfaction.
- Whereas, lost luggage is fifth most important reason for bad reviews on Tweeter.

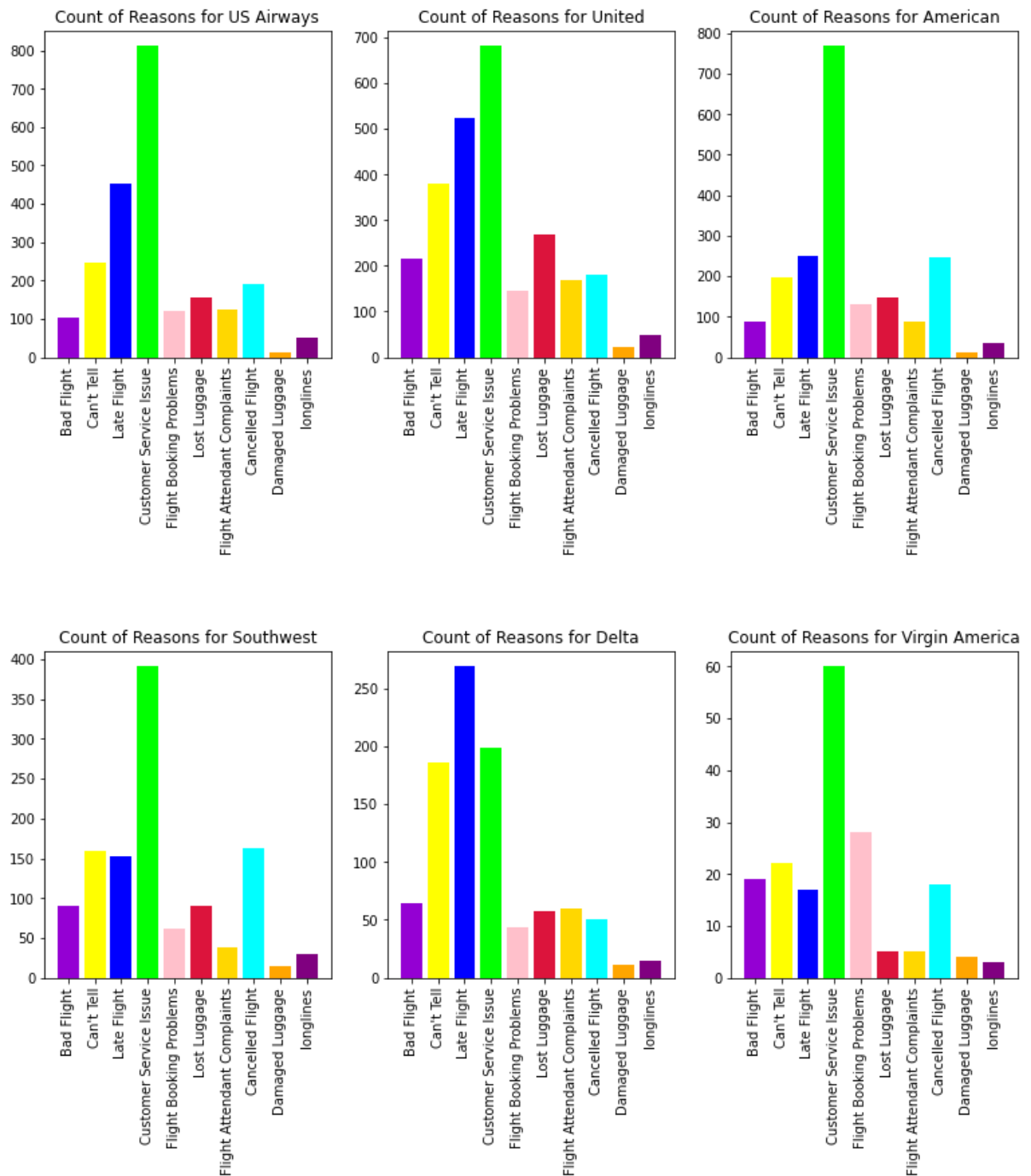


Figure 15

The above bar charts (Figure 15) provide information for top 10 main reasons of complain for each of the airline company.

The following information can be summarized:

- **American, US Airways, Southwest:** Complaints about customer service issue is relatively high. Followed by Late flight for US Airways and United airlines.
- **Delta:** Late flight was the main reason for customer dissatisfaction, Customer Service Issue being the second one.

- **United** : Customer service issue is the most complains, also customers for this airline experienced late flight making it the second most important reason for customer dissatisfaction.
- **Virgin America**: Mostly about customer service followed by flight booking problem.

Overall Customer Service Issue is the main negative reason for US Airways, United, American, Southwest, Virgin America frequently than other airlines. Lost luggage issue happened relatively high than other reasons.

4. Data cleaning and preprocessing

Since the project is about Tweeter Sentiment Analysis i.e prediction of sentiment of the tweets based on the Tweets posted in February 2015, only the “text ” column is needed as training data and the “airline_sentiment” column can be used as the label for the dataset.

Labels

The column “airline_sentiment” contains three categories of sentiment. Those are positive, neutral and negative. Since the machine doesn’t understand string values, I converted them to digits indicating 0 as negative, 1 as neutral and 2 as positive. So, the converted dataframe contains three labels 0, 1 and 2.

```
1 # convert Sentiments to 0,1,2
2 def convert_Sentiment(sentiment):
3     if sentiment == "positive":
4         return 2
5     elif sentiment == "neutral":
6         return 1
7     elif sentiment == "negative":
8         return 0
```

Training Data

The column “text” is used as the training set for the model.

The text contains many special characters, http links and other additional characters and word. So it needs pre-processing, like removal of special characters, conversion to lower case, removal of stopwords, treating of accented characters etc.

The “text” column shows some sample texts. Some sample data from “text” column can be seen from Figure 16.

```

1 df["text"]

0          @VirginAmerica What @dhepburn said.
1    @VirginAmerica plus you've added commercials t...
2    @VirginAmerica I didn't today... Must mean I n...
3    @VirginAmerica it's really aggressive to blast...
4    @VirginAmerica and it's a really big bad thing...
...
14635 @AmericanAir thank you we got on a different f...
14636 @AmericanAir leaving over 20 minutes Late Flig...
14637 @AmericanAir Please bring American Airlines to...
14638 @AmericanAir you have my money, you change my ...
14639 @AmericanAir we have 8 ppl so we need 2 know h...
Name: text, Length: 14640, dtype: object

```

Figure 16

Figure 17 represents the string length distribution of the dataset.

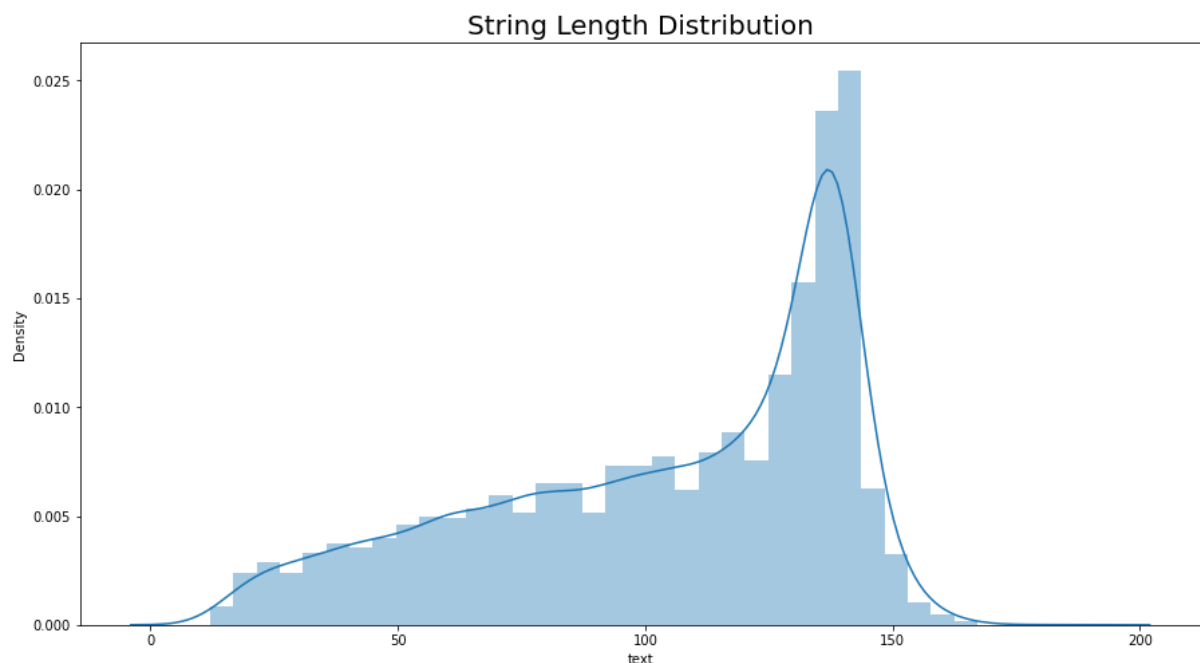


Figure 17

From the figure it can be said that

- Tweet text is consisted of 103 length in average.
- Minimum length is 12, and maximum one is 186.

The following preprocessing things are done.

Other Preprocessing steps:

- Removal of url
- Convert all the reviews to lower case.
- Removal of punctuation
- Removal of html
- Removal of username
- Removal of emojis
- Decontraction of texts
- Separation of alphanumeric

```
88 # Apply functions on tweets
89 df['final_text'] = df['final_text'].apply(lambda x : remove_username(x))
90 df['final_text'] = df['final_text'].apply(lambda x : remove_url(x))
91 df['final_text'] = df['final_text'].apply(lambda x : remove_emoji(x))
92 df['final_text'] = df['final_text'].apply(lambda x : decontraction(x))
93 df['final_text'] = df['final_text'].apply(lambda x : separate_alphanumeric(x))
94 df['final_text'] = df['final_text'].apply(lambda x : unique_char(cont_rep_char,x))
95 df['final_text'] = df['final_text'].apply(lambda x : char(x))
96 df['final_text'] = df['final_text'].apply(lambda x : x.lower())
97 #df['final_text'] = df['final_text'].apply(lambda x : remove_stopwords(x))
```

The final dataframe looks like this:

```
1 # result
2 df['final_text']
```

0	what said
1	plus you have added commercials to the experie...
2	i did not today must mean i need to take anoth...
3	bad flight it is really aggressive to blast ob...
4	ca not tell and it is a really big bad thing a...
...	
14635	thank you we got on a different flight to chicago
14636	customer service issue leaving over minutes...
14637	please bring american airlines to blackberry
14638	customer service issue you have my money you c...
14639	we have ppl so we need know how many seats...

Name: final_text, Length: 14640, dtype: object

Text to Vectorization

Convert a collection of raw documents to a matrix of TF-IDF features. TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. TF-IDF is an algorithm to transform text into a meaningful representation of numbers, then this representation is used to fit machine algorithm for prediction.

Count Vectorizer gives number of frequency with respect to index of vocabulary. Here the tf-idf consider overall documents of weight of words.

Handling Imbalance data

On the dataset the number of negative class sample are much higher than number of positive and neutral sentiment tweets.

As a result, the network may have biased towards negative sentiment texts, which ultimately led to poor performance of the model. To handle the issue data augmentation for the minority class was done using Synthetic Minority Oversampling Technique or in short SMOTE

Splitting Training and Testing Set

Finally, I split the data with 80% for training and 20% for testing. The final dataset length for training and testing look like below.

```
1 print("X_train.shape",X_train.shape)
2 print("X_test.shape",X_test.shape)

X_train.shape (22027, 11124)
X_test.shape (5507, 11124)
```

5. Predictive modelling

1. Support Vector Machine

I obtained around 97% testing accuracy using Support Vector Machine.

```
| 1 accuracy_score(svm_prediction,y_test)

0.9734882876339205
```


To improve the accuracy even further we used GridSearch techniques with different combinations of hyperparameters.

```
1 # Tuning the hyperparameters
2 parameters = {
3     "C": [0.1, 1, 10],
4     "kernel": ['linear', 'rbf', 'sigmoid'],
5     "gamma": ['scale', 'auto']
```

The grid search technique using the above parameters suggest that $C = 0.1$, $\gamma = \text{scale}$, and $\text{kernel} = \text{linear}$ should give the best performance.

```
[CV] END .....C=10, gamma=auto, kernel=rbf; total time= 2.0min
[CV] END .....C=10, gamma=auto, kernel=rbf; total time= 2.0min
[CV] END .....C=10, gamma=auto, kernel=sigmoid; total time= 1.9min
[CV] END .....C=10, gamma=auto, kernel=sigmoid; total time= 1.9min
```

Best parameters are:

```
{'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}
```

However, in practice, the highest accuracy wasn't found with this set of hyperparameters. So, it seems GridSearch method isn't working for this type of dataset.

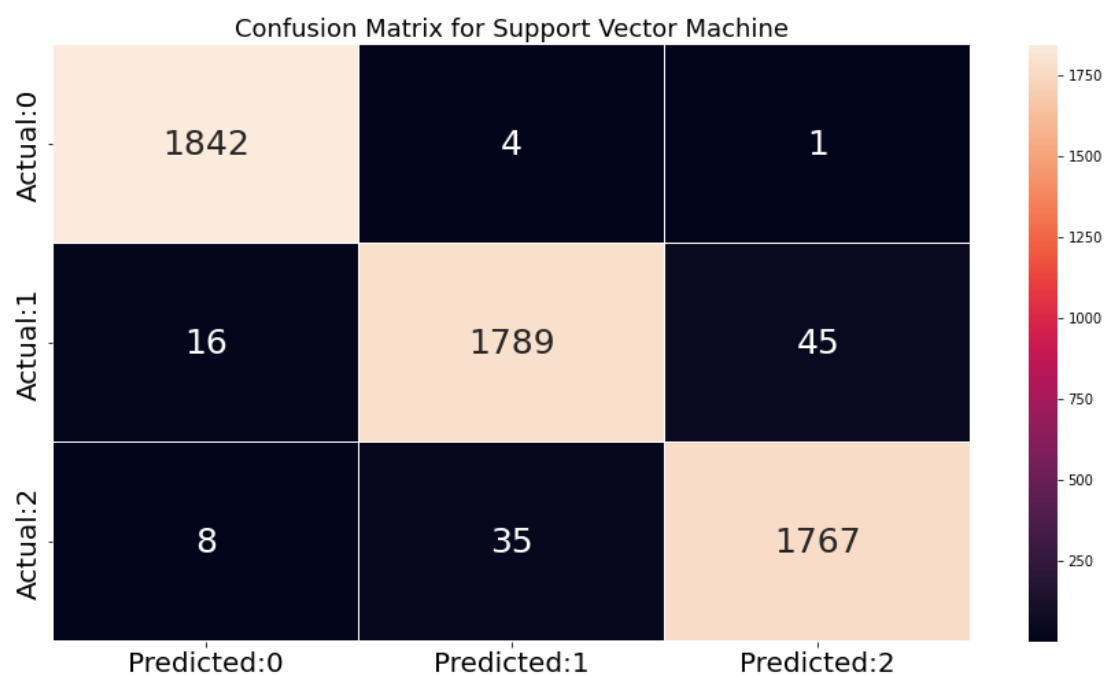
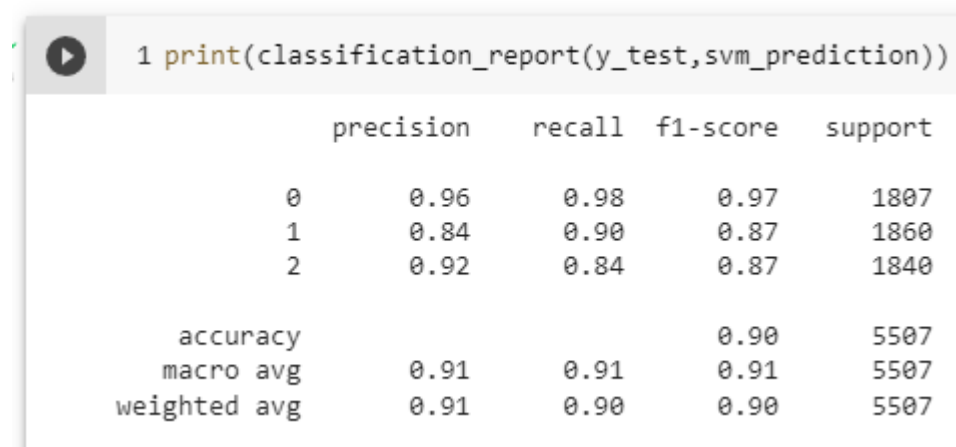


Figure 18

Figure 18 represents the Confusion matrix for Support vector Machine. The above figure shows the different values of the matrix in different colors, which makes the distribution of observed values more obvious. The values are listed in each square and match the colors. It

can be seen that there is a significant difference between the colors of the diagonal and those of other positions, which directly indicates that the SVM model has good performance. The values of the light color position in the figure do not exceed 45. So, It can be concluded that text classification can be completed with a low error rate.

Figure 19 provides the classification report for the SVM model. It should be noted that higher False Positive may miss lead the airline companies to understand higher satisfaction rate by the customers, which against the objective of this project. The objective of this project is to know the reasons for customer dissatisfaction and take proper steps to reduce customer dissatisfaction. So, for this task lower False Positives are expected, that means higher precision value is needed. The SVM model gives 96% precision value for the class zero (i.e for negative sentiment class) and 92% for the class 2 (i.e or positive class).

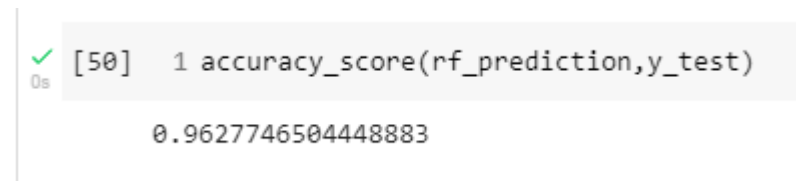


	precision	recall	f1-score	support
0	0.96	0.98	0.97	1807
1	0.84	0.90	0.87	1860
2	0.92	0.84	0.87	1840
accuracy			0.90	5507
macro avg	0.91	0.91	0.91	5507
weighted avg	0.91	0.90	0.90	5507

Figure 19

2. Random Forest

The same training and testing set were fed to test accuracy on Random Forest algorithm.



```
[50] 1 accuracy_score(rf_prediction,y_test)
```

0.9627746504448883

I found around 96% accuracy with Random Forest model.

The following figure shows confusion matrix for the Random Forest model. It can be seen from the figure that the False Positive value is little bit higher than the SVMs'.

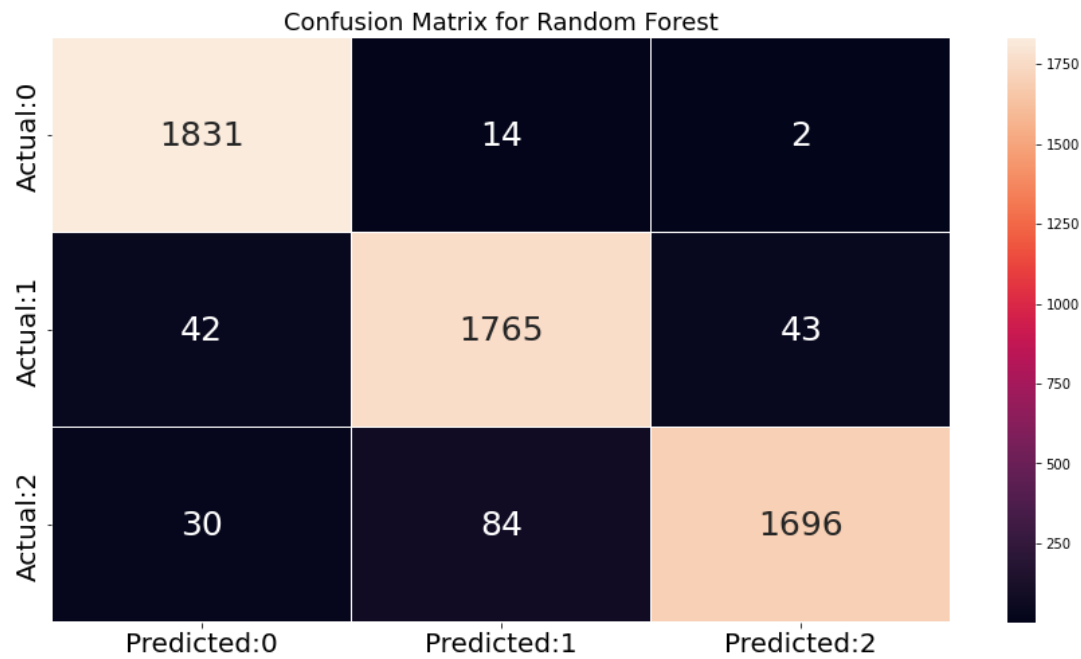


Figure 20

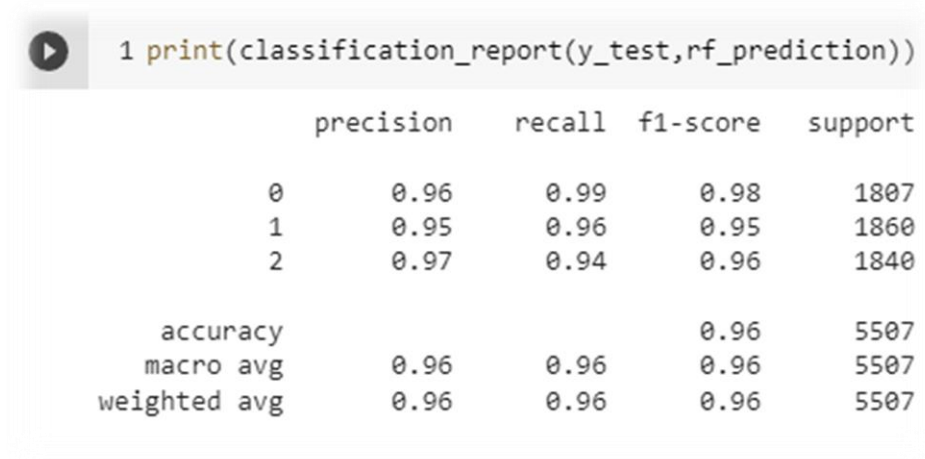


Figure 21

On the other hand, the classification report for Random Forest provides more than 95% precision value for every class. Since our task desires more precision value so Random Forest is definitely a good model for this task.

3. Logistic Regression

I also checked the dataset with Logistic regression algorithm.

Here are the following outputs.

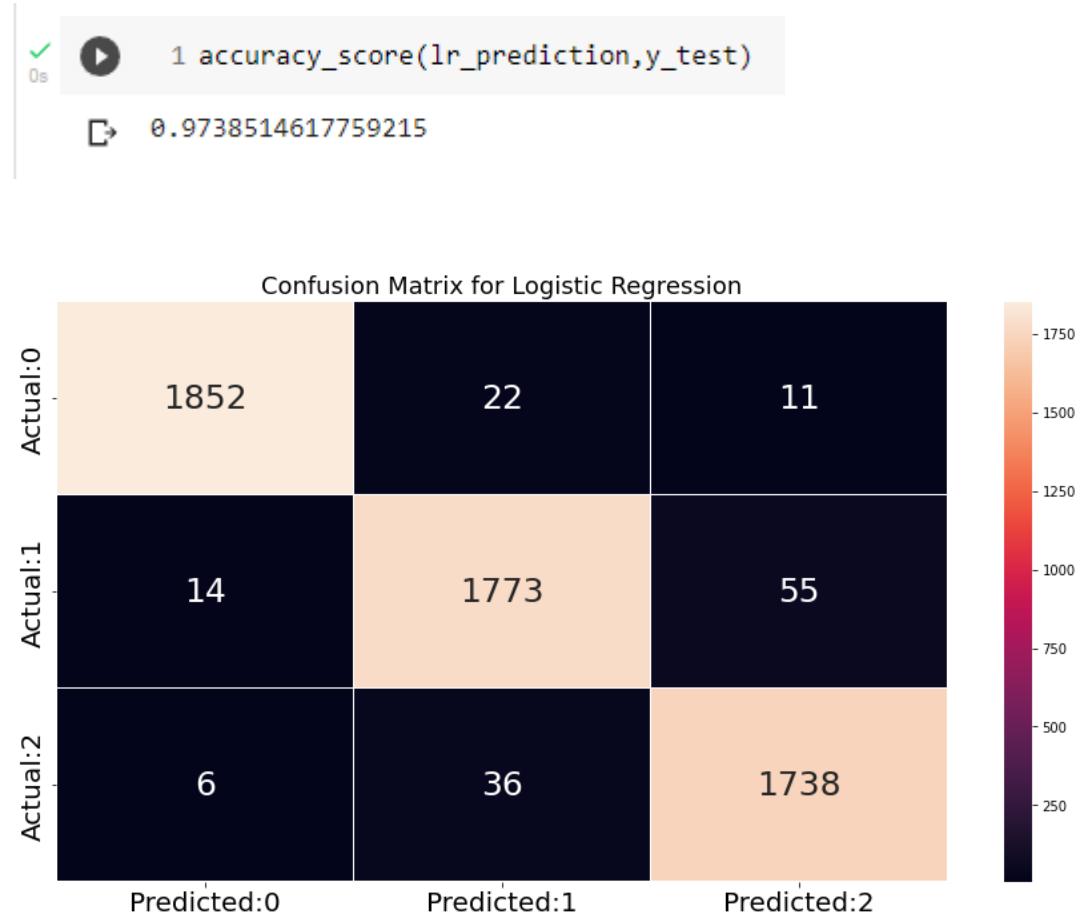


Figure 22

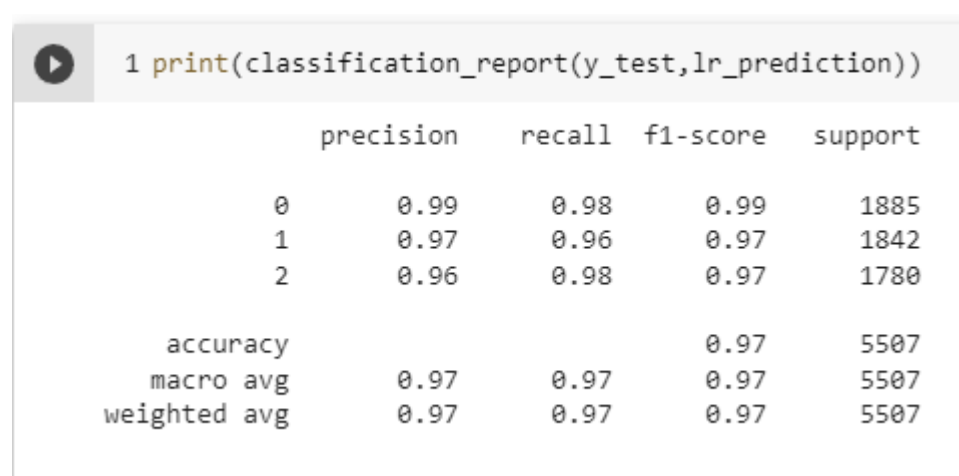


Figure 23

IV. Naïve Bayes

Finally, I checked the dataset with Naïve Bayes algorithm. Here are the outputs.

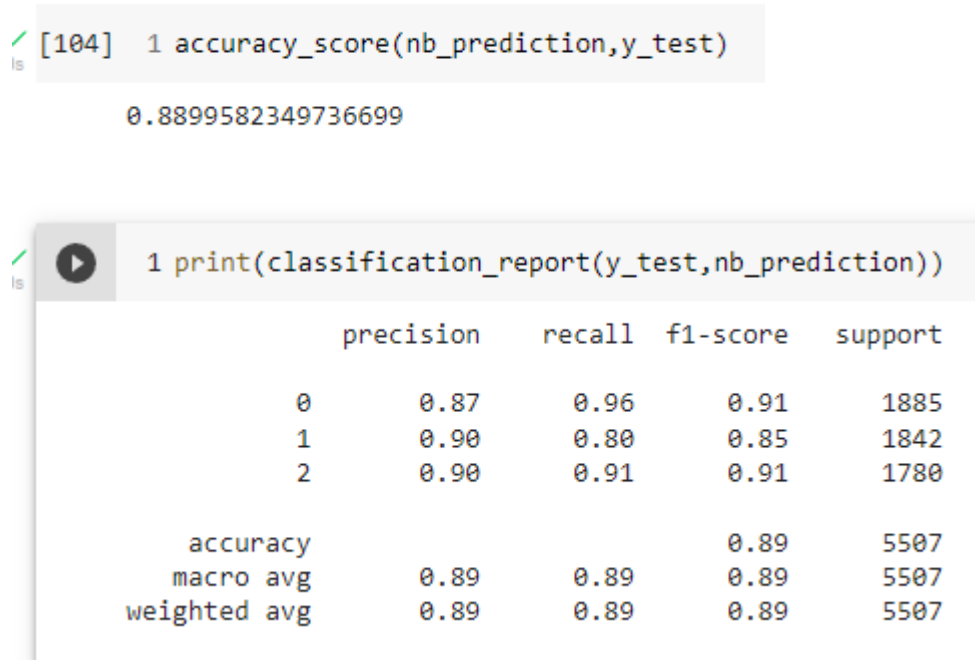


Figure 24

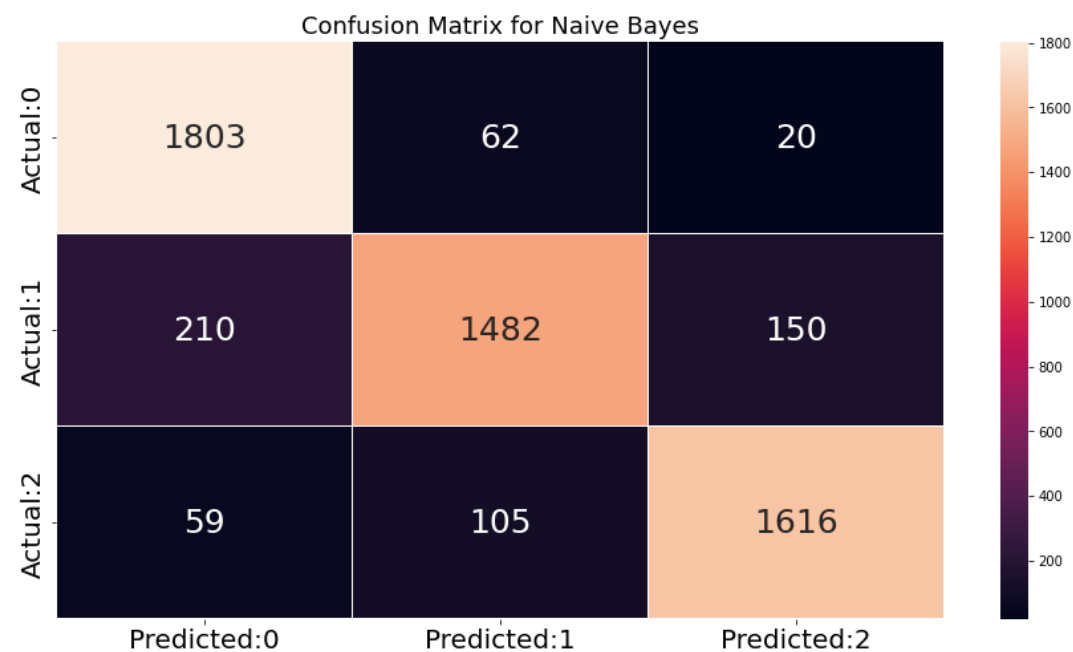


Figure 25

To sum up all of the models worked relatively well. However, considering the accuracy, precision, recall and f1-score Logistic Regression seems best among the models.

6. What did you learn about the data and the underlying data generating mechanism

First let's have a look on texts of different sentiment that were obtained by data cleaning and preprocessing.

Texts with negative sentiment:

4 negative		
	final_text	airline_sentiment
3	bad flight it is really aggressive to blast obnoxious entertainment in your guests faces amp the...	0
4	ca not tell and it is a really big bad thing about it	0
5	ca not tell seriously would pay a flight for seats that did not have this playing it is reall...	0
15	late flight sfo pdx schedule is still mia	0
17	bad flight i flew from nyc to sfo last week and could not fully sit in my seat due to two large ...	0
...
14631	bad flight thx for nothing on getting us out of the country and back to us broken plane come on ...	0
14633	cancelled flight my flight was cancelled flightled leaving tomorrow morning auto rebooked for a ...	0
14634	late flight right on cue with the delays	0
14636	customer service issue leaving over minutes late flight no warnings or communication until we...	0
14638	customer service issue you have my money you change my flight and do not answer your phones any ...	0

9178 rows × 2 columns

Figure 26

From the texts of negative, neutral and positive sentiments I will try to find out most frequents words used to express different sentiments.

Texts with Neutral Sentiment:

4 neutral		
	final_text	airline_sentiment
0	what said	1
2	i did not today must mean i need to take another trip	1
7	really missed a prime opportunity for men without hats parody there	1
10	did you know that suicide is the second leading cause of death among teens	1
23	will you be making bos gt las non stop permanently anytime soon	1
...
14607	i need someone to help me out	1
14611	guarantee no retribution if so i would be glad to share	1
14632	george that does not look good please follow this link to start the refund process	1
14637	please bring american airlines to blackberry	1
14639	we have ppl so we need know how many seats are on the next flight plz put us on standby for ...	1

3099 rows × 2 columns

Figure 27

Texts with positive sentiment:

4 positive		
	final_text	airline_sentiment
1	plus you have added commercials to the experience tacky	2
6	yes nearly every time i fly vx this ear worm won t go away	2
8	well i did not but now i do d	2
9	it was amazing and arrived an hour early you are too good to me	2
11	i lt pretty graphics so much better than minimal iconography d	2
...
14623	love the new planes for the jfk lax run maybe one day i will be on one where the amenities all f...	2
14625	flight was great fantastic cabin crew a landing thankyou jfk	2
14628	thank you customer relations will review your concerns and contact you back directly john	2
14630	thanks he is	2
14635	thank you we got on a different flight to chicago	2

2363 rows × 2 columns

Figure 28

Visualizing Important Words with Wordcloud

Wordcloud data visualization technique is used on text data in which the size of each word indicates its frequency or importance. Word clouds are commonly used for analyzing data from social network websites.

To generate word cloud in Python, module- wordcloud is installed.

The following images show the frequency of neutral, negative and positive sentiment words respectively.

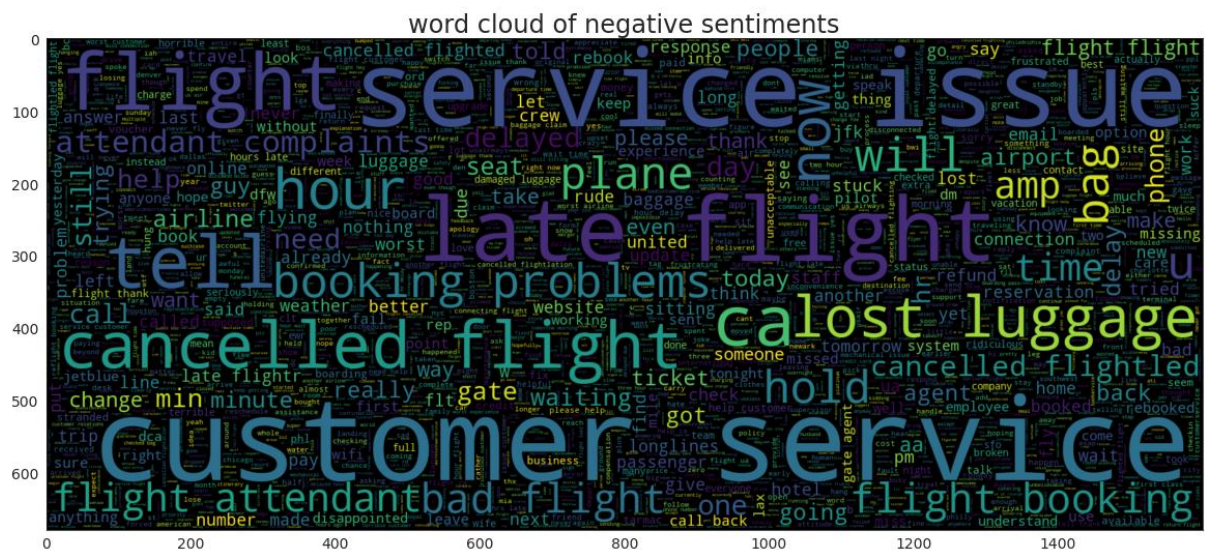


Figure 29

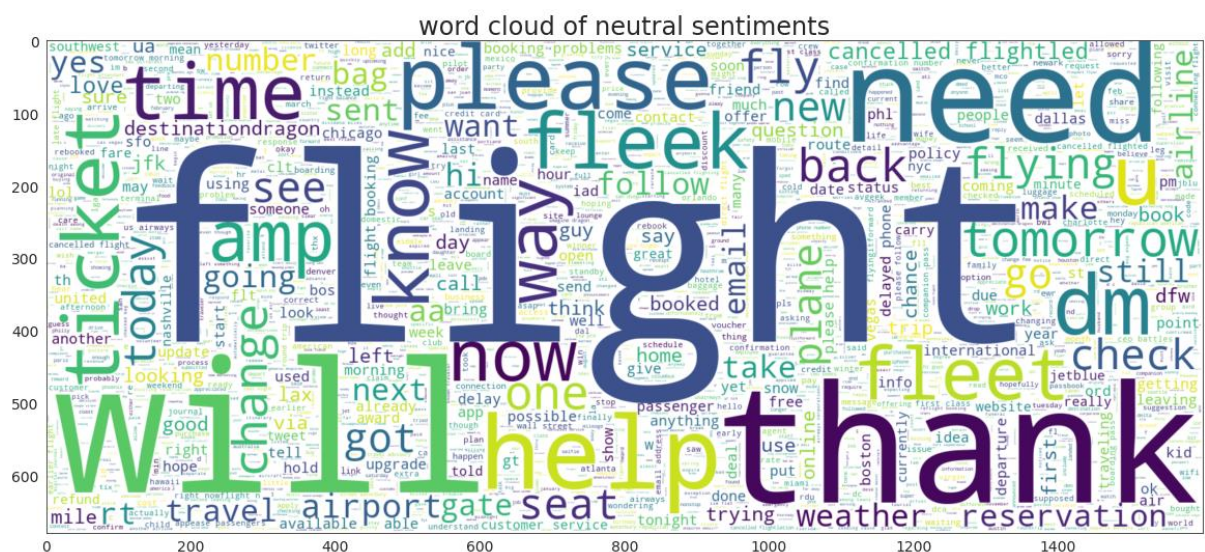


Figure 30



Figure 31

7. Which variable(s) are more related to the response?

Here I am using only the “text” column as feature for training the model. Generally, the words with higher frequency should have more impact on the model than that of lower frequency words. In this project only the column “final_text” is used as the main feature to train the model.



```
1 # result
2 df['final_text']
```



```
0                                     what said
1      plus you have added commercials to the experie...
2      i did not today must mean i need to take anoth...
3      bad flight it is really aggressive to blast ob...
4      ca not tell and it is a really big bad thing a...
                                     ...
14635    thank you we got on a different flight to chicago
14636    customer service issue leaving over      minutes...
14637      please bring american airlines to blackberry
14638    customer service issue you have my money you c...
14639    we have    ppl so we need    know how many seats...
Name: final text, Length: 14640, dtype: object
```

Figure 32

8. Model Inference and Comparison of the predictive modelling methods

To compare the model performance along with model accuracies, Precision, F1 score and ROC AUC scores will be considered

ROC-AUC

It is a chart that visualizes the tradeoff between true positive rate (TPR) and false positive rate (FPR). Basically, for every threshold, we calculate TPR and FPR and plot it on one chart.

Of course, the higher TPR and the lower FPR is for each threshold the better and so classifiers that have curves that are more top-left-side are better.

Model name	Precision			Recall			F1		
	Class 0	Class 1	Class 2	Class 0	Class 1	Class 2	Class 0	Class 1	Class 2
Random Forest	0.96	0.95	0.97	0.99	0.96	0.94	0.98	0.95	0.96
Logistic Regression	0.99	0.97	0.96	0.98	0.96	0.98	0.99	0.97	0.97
Support Vector	0.96	0.84	0.92	0.98	0.90	0.84	0.97	0.87	0.87
Naïve Bayes	0.87	0.90	0.90	0.96	0.80	0.91	0.91	0.85	0.91

Model name	Test Accuracy	ROC-AUC Score Class 0 (negative class)	ROC-AUC Score Class 1 (neutral class)	ROC-AUC Score Class 2 (positive class)
Random Forest	96.27%	1.00	0.99	0.99
Logistic regression	97.38%	1.00	0.99	1.00
Support Vector	97.34%	1.00	1.00	0.99
Naïve Bayes	88.98%	0.99	0.96	0.98

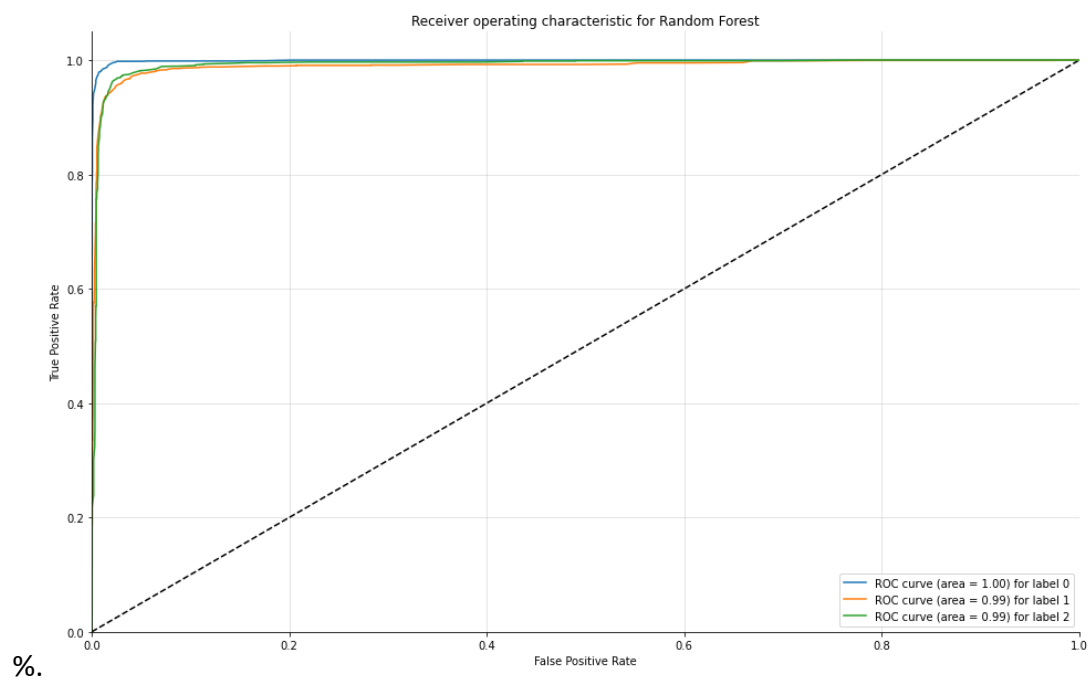


Figure 33

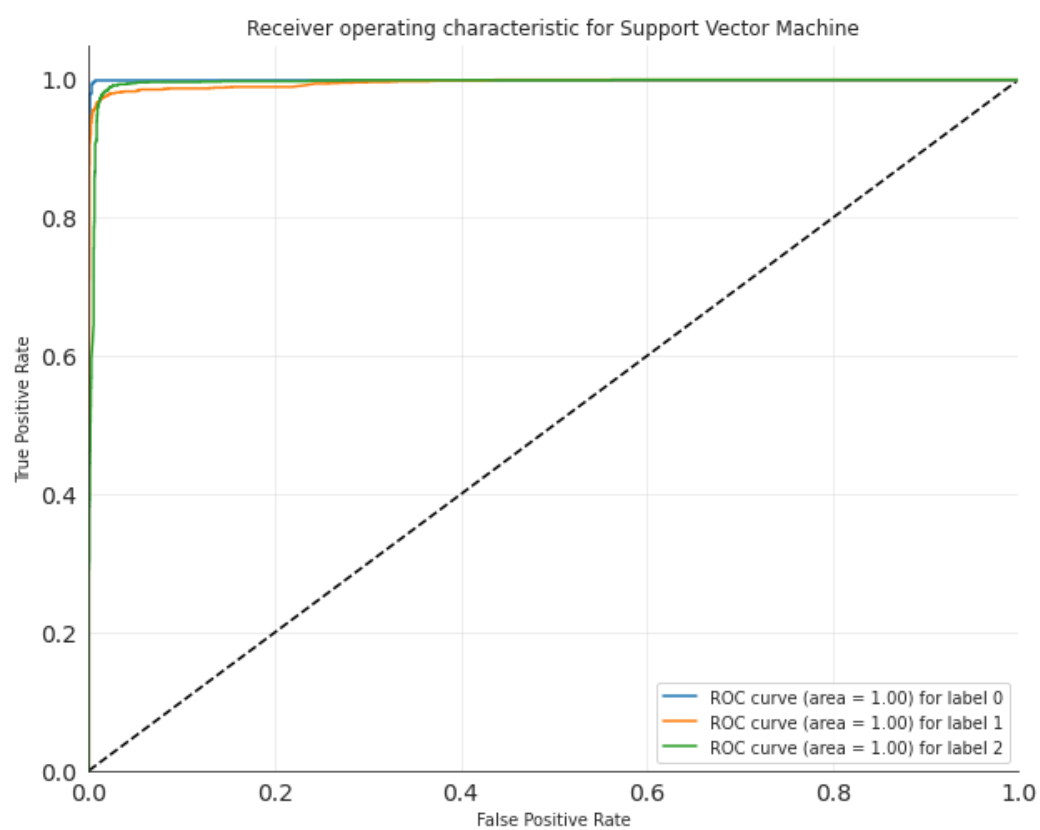


Figure 34

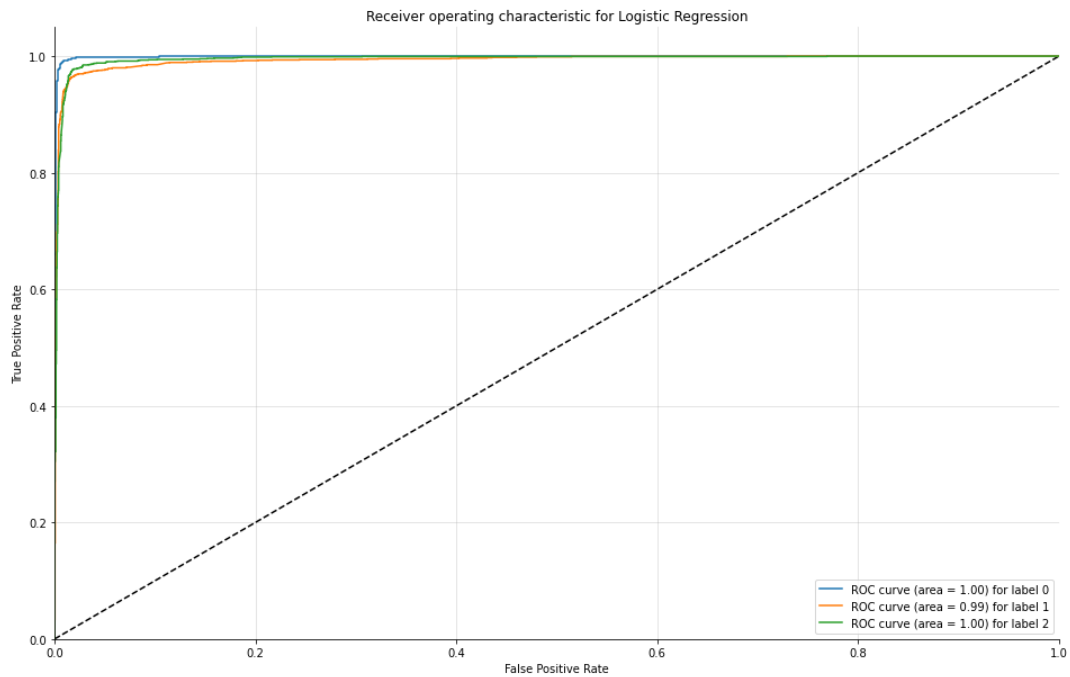


Figure 35

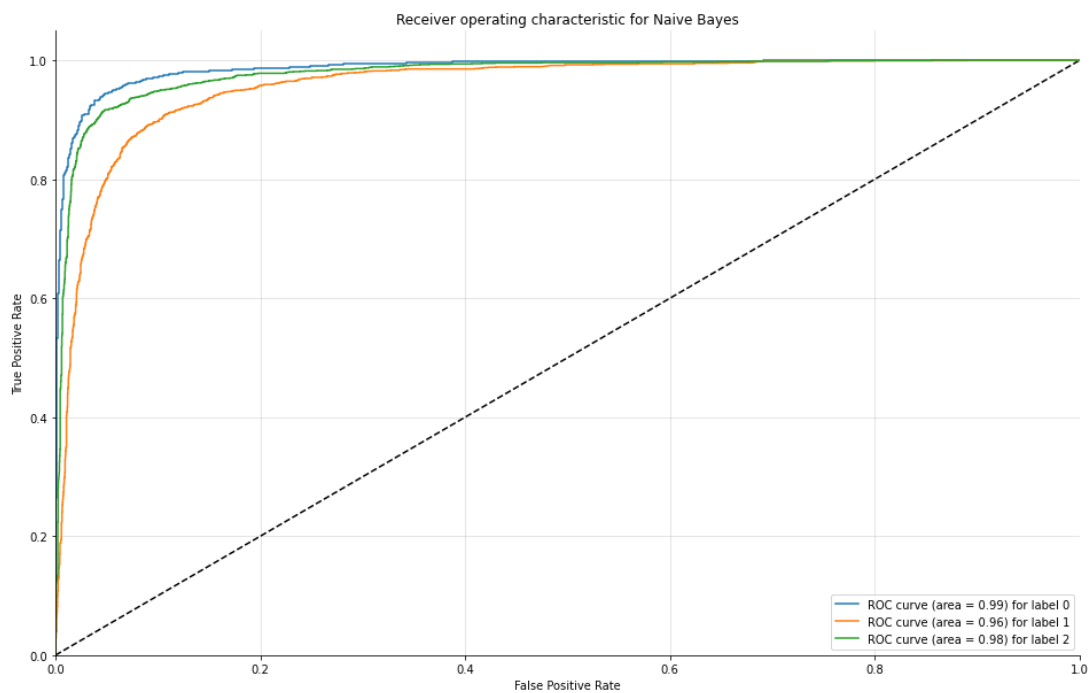


Figure 36

Among those 4 models, logistic regression using TFIDF had the highest test accuracy at 97.38%.

- Comparing the accuracies of different models are not always perfect to identify the right model or fine tune the model.

- In a classifier model, though accuracy is easily readable finding area under roc-curve / roc auc score is consistent and more discriminative.
- For four of our classifiers, we have plotted the ROC curve and roc auc scores.
- The roc auc scores of each model also tells that along with logistic regression, Support Vector Machine and Random Forest classifier also fit with the model better than Naïve bayes.

9. Model inference, knowledge discovery and discussion

Pros and Cons of Each Model

- Logistics regression is easy to train, implement and interpret
- However, it is difficult to establish complex relationship with this predicting method and it is only suitable for linear surface data.
- Random Forests work well with large set of data.
- It can be used for both classification and regression tasks.
- However, Random Forests are not easily interpretable and can be computationally intensive for large datasets. It is expensive for large datasets.
- Support vector machine is effective in high dimensional spaces, but it requires higher training time when the dataset is large enough.
- Naive Bayes is commonly associated with text-based classification. For example, it is applied in spam filtering and text categorization.
- A big advantage of Naive Bayes classifiers is that it is not prone to overfitting, because the algorithm “ignore” irrelevant features

10. Conclusion

In this project, I analyzed Sentiment on Tweeted text data, cleaned them, and extracted important words to analyze sentiment from the text context. Text to vector tokenizing method is used to convert words to vector. Then, dataset was used in four different machine learning methods and observed different performance metrics from them. Finally, we compared their results and concluded that, though three out of the four models (Logistic Regression, Random Forest and Support Vector Machine) work well, Logistic Regression beat other models in terms of Precision and F1 scores. So, Logistic regression seems the best model performing among them

One of challenges of performing sentiment analysis on Twitter is that, since each of Twitter user speaks differently about their experience, there are many slangs, new words, acronym,

abbreviation, curse, or simply misspelled words that can be hard to capture with current text cleaning packages especially when data is large. For the next step, we would like to search for better text cleaning packages that can reduce issues mentioned above.

11. References

1. <https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc>
2. <https://towardsdatascience.com/generate-meaningful-word-clouds-in-python-5b85f5668eeb>
3. <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>
4. <https://neptune.ai/blog/vectorization-techniques-in-nlp-guide>
5. <https://www.baeldung.com/cs/decision-tree-vs-naive-bayes>
6. <https://medium.datadriveninvestor.com/twitter-airline-sentiment-analysis-3b147b565027>