Analysis and Tweets sentiment prediction using US Airline Twitters Dataset

Minakshi Kesarwani

11/21/2019

1) Project Goal:

To build a machine learning model that tries to predict the negative and positive sentiment of tweets relating to US airlines using **Twitter US airline sentiment** dataset followed by categorizing negative reasons (such as "late flight" or "rude service").

It will provide the airline industry more comprehensive views about the sentiments of their customers. With recent development in machine learning algorithms, using the power of sentiment analysis now the airline industry can analyze tweets and enhance their customer services.

2) Executive Summary :

This project uses **sentiment analysis** which is one of the most popular applications of **natural language processing (NLP)** that determines the sentiment or emotion of a piece of text. The dataset is called **Twitter US Airline Sentiment** which was downloaded from Kaggle as a CSV file. Its source was **Crowdflowers Data for Everyone library**. It contains details of **14640 tweets** posted on Twitter in a week of **February 2015** about each major US airline. In this project, I started with **exploratory data analysis** to analyze the distribution of sentiment of the tweets, identify which airline has a greater number of negative tweets and what is the reason behind those negative tweets. Furthermore, using the **bag of word technique** transformed document into a vector to prepare train and test data and used various machine learning algorithms like **Decision Tree, Random Forest and SVM (Support Vector Machine)** to train corresponding models and validate their accuracy in predicting the sentiment of the test set. Since the data set is highly imbalanced (79% negative tweets vs 21% positive tweets), I have used **F1 score** and **AUC score** evaluation metrics to identify the best machine learning model built in this project.

3) Dataset Summary :

- This dataset contains tweets about 6 US airlines(American, Delta,Southwest, United, US Airways, Virgin America).
- Mean length of tweets:- (about 17.61 words in length),
- Max length of tweets: 35
- Min length of tweets: 1
- Shape of dataset: 14640(observation), 15(variables)
- 'Negative tweets' will be the class of Interest.
- **Datsource**: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Load required packages

Load Packages only if not present locally

4) Exploratory data analysis:

a) Read dataset, check its structure and dimension

```
## 'data.frame':    14640 obs. of  15 variables:
##  $ tweet_id                    : num  5.7e+17 5.7e+17 5.7e+17 5.7e+17 5.7e+17 ...
##  $ airline_sentiment           : Factor w/ 3 levels "negative","neutral",..: 2 3 2 1 1 1 3 2 3 3 ...
##  $ airline_sentiment_confidence: num  1 0.349 0.684 1 1 ...
##  $ negativereason              : Factor w/ 11 levels "","Bad Flight",..: 1 1 1 2 3 3 1 1 1 1 ...
##  $ negativereason_confidence   : num  NA 0 NA 0.703 1 ...
##  $ airline                     : Factor w/ 6 levels "American","Delta",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ airline_sentiment_gold      : Factor w/ 4 levels "","negative",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ name                        : Factor w/ 7701 levels "___the___","__betrayal",..: 1073 3477 7666 34
##  $ negativereason_gold         : Factor w/ 14 levels "","Bad Flight",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ retweet_count               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ text                        : Factor w/ 14427 levels "\"LOL you guys are so on it\" - me, had this
##  $ tweet_coord                 : Factor w/ 833 levels "","[-33.87144962, 151.20821275]",..: 1 1 1 1 1
##  $ tweet_created               : Factor w/ 14247 levels "2015-02-16 23:36:05 -0800",..: 14212 14170 1
##  $ tweet_location              : Factor w/ 3082 levels "","'Greatness has no limits'",..: 1 1 1465 1
##  $ user_timezone               : Factor w/ 86 levels "","Abu Dhabi",..: 32 64 29 64 64 64 64 64 64 31
```

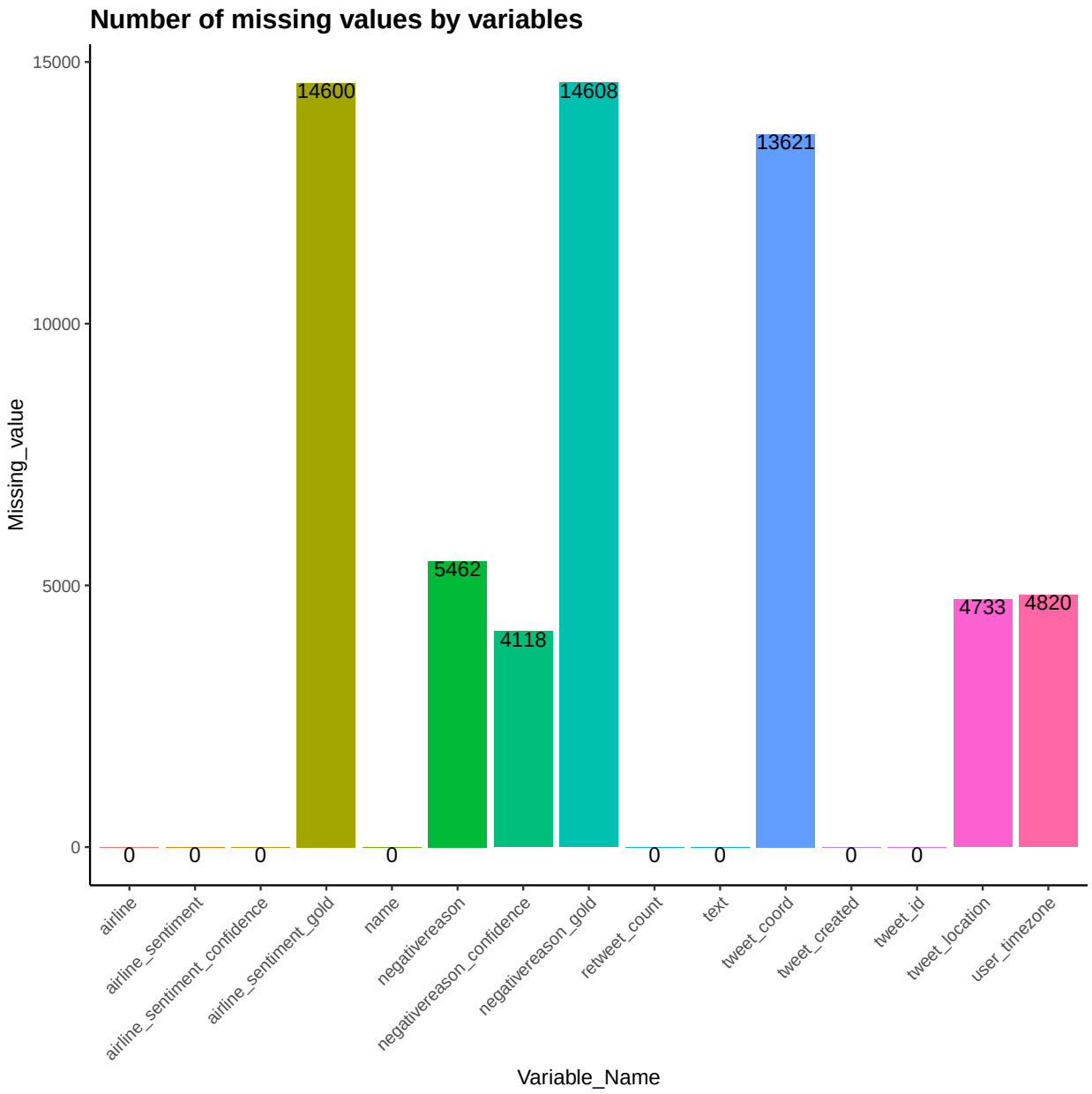
```
## [1] 14640    15
```

Interpretation: The dataset contains **14640 tweets (observation) and 15 variables (columns)**.

b) Check for missing values in the data set

Table 1: Number of missing value per column

Variable_Name	Missing_value
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



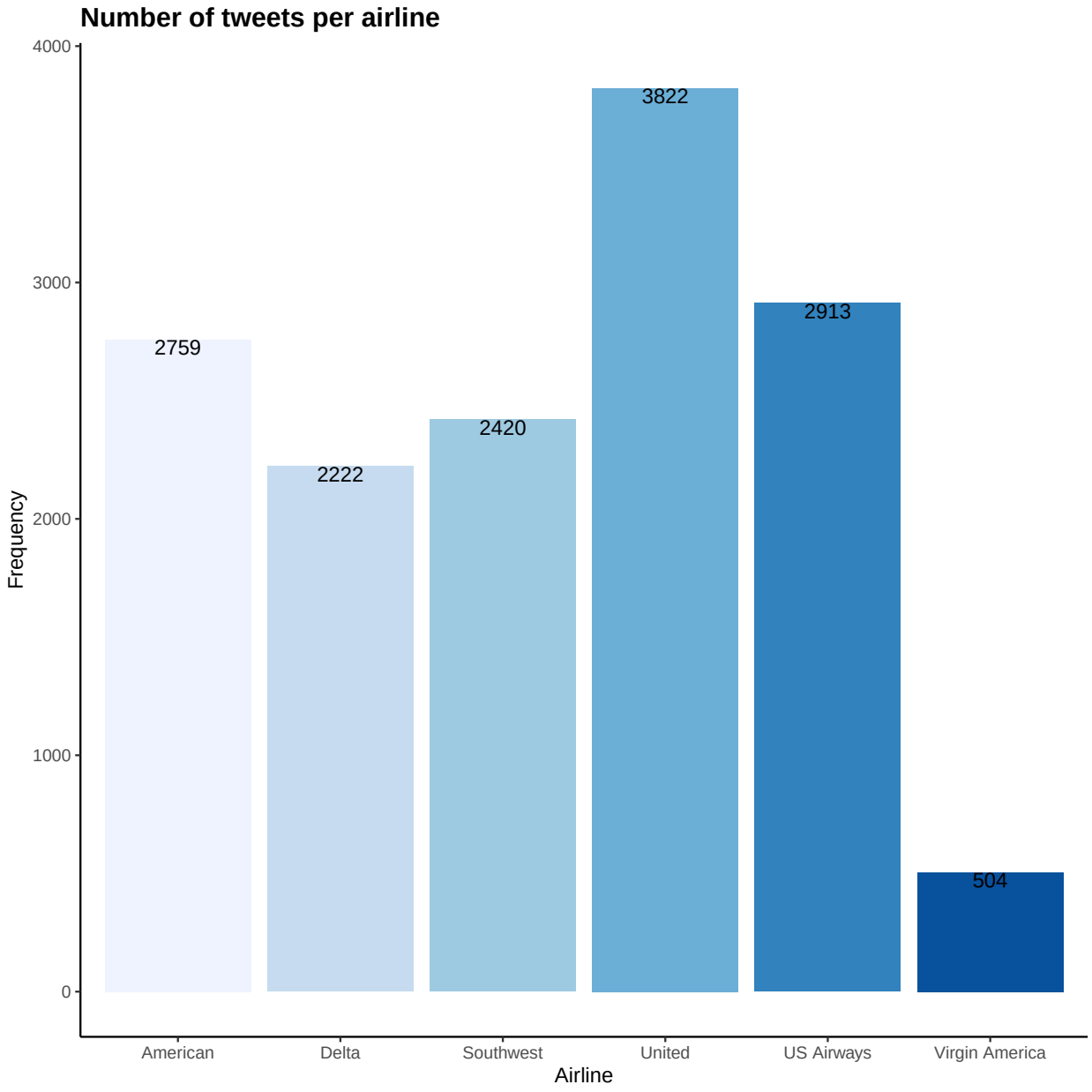
Interpretation: Out of 15 columns, 7 columns contains missing values. Since the column of interest are **airline_sentiment** and **text** and these do not contain any missing value, there is no need to handle missing values.

c) Number of tweets per airline

Table 2: Timezone vs Frequency of tweets

Airline	Frequency
American	2759
Delta	2222
Southwest	2420

Airline	Frequency
United	3822
US Airways	2913
Virgin America	504

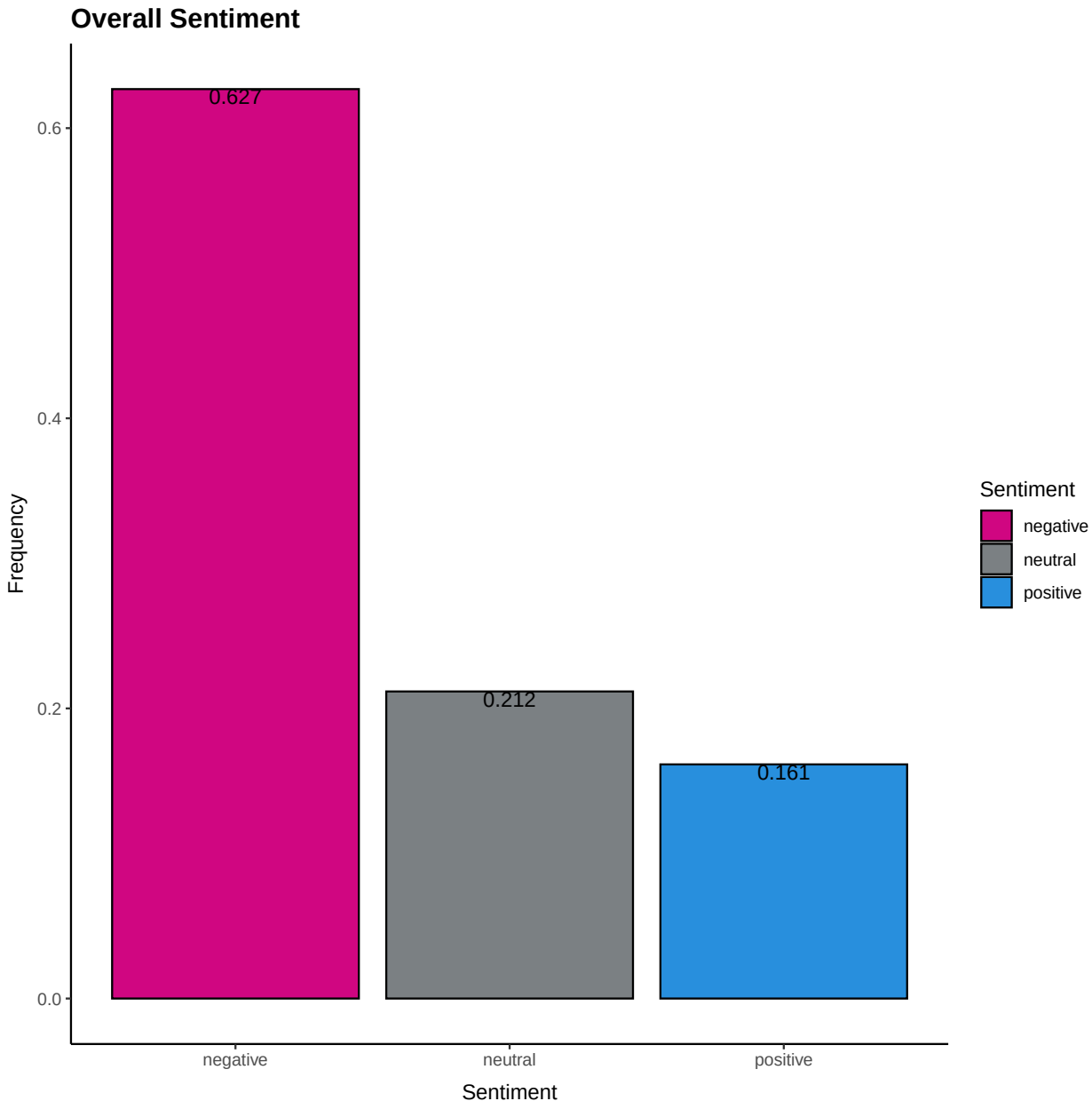


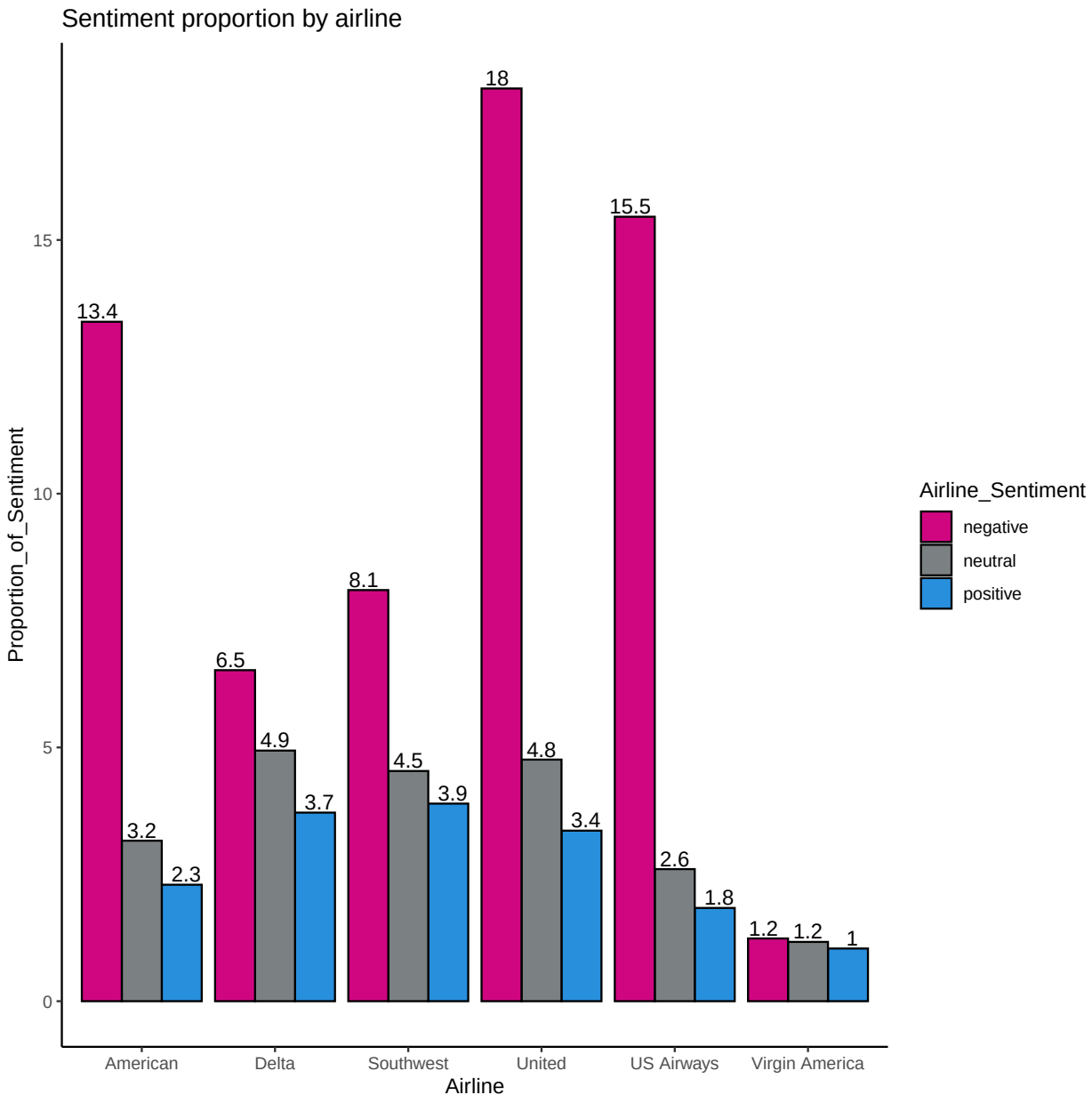
Interpretation: Most of the tweets are directed towards **United Airlines**, followed by **American Airlines** and **US Airways**. Very few tweets are targeted towards **Virgin America**.

d) Checking ratio of negative, positive and neutral sentiment

Table 3: Distribution of negative, positive and neutral sentiment

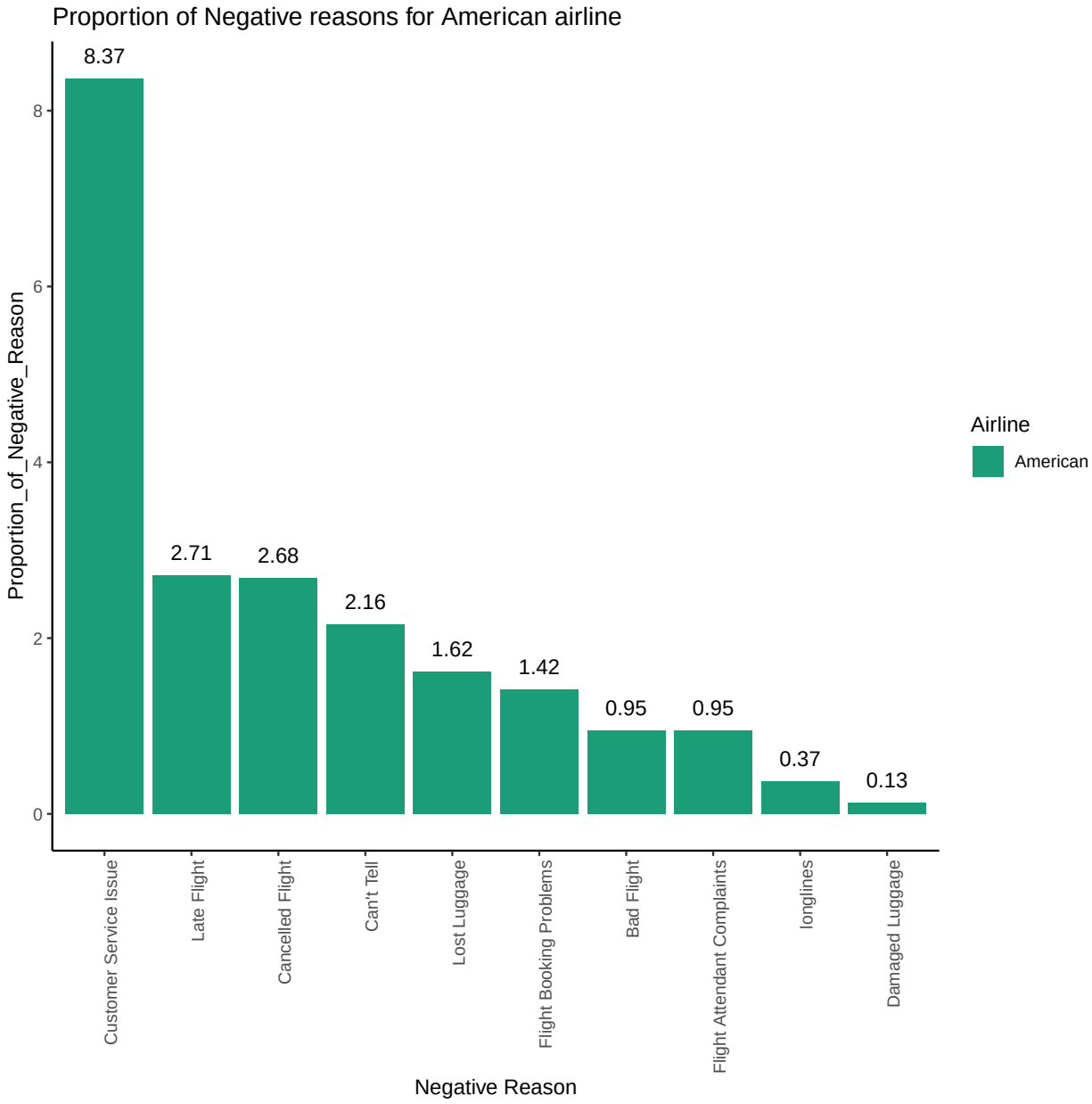
Sentiment	Frequency
negative	0.627
neutral	0.212
positive	0.161

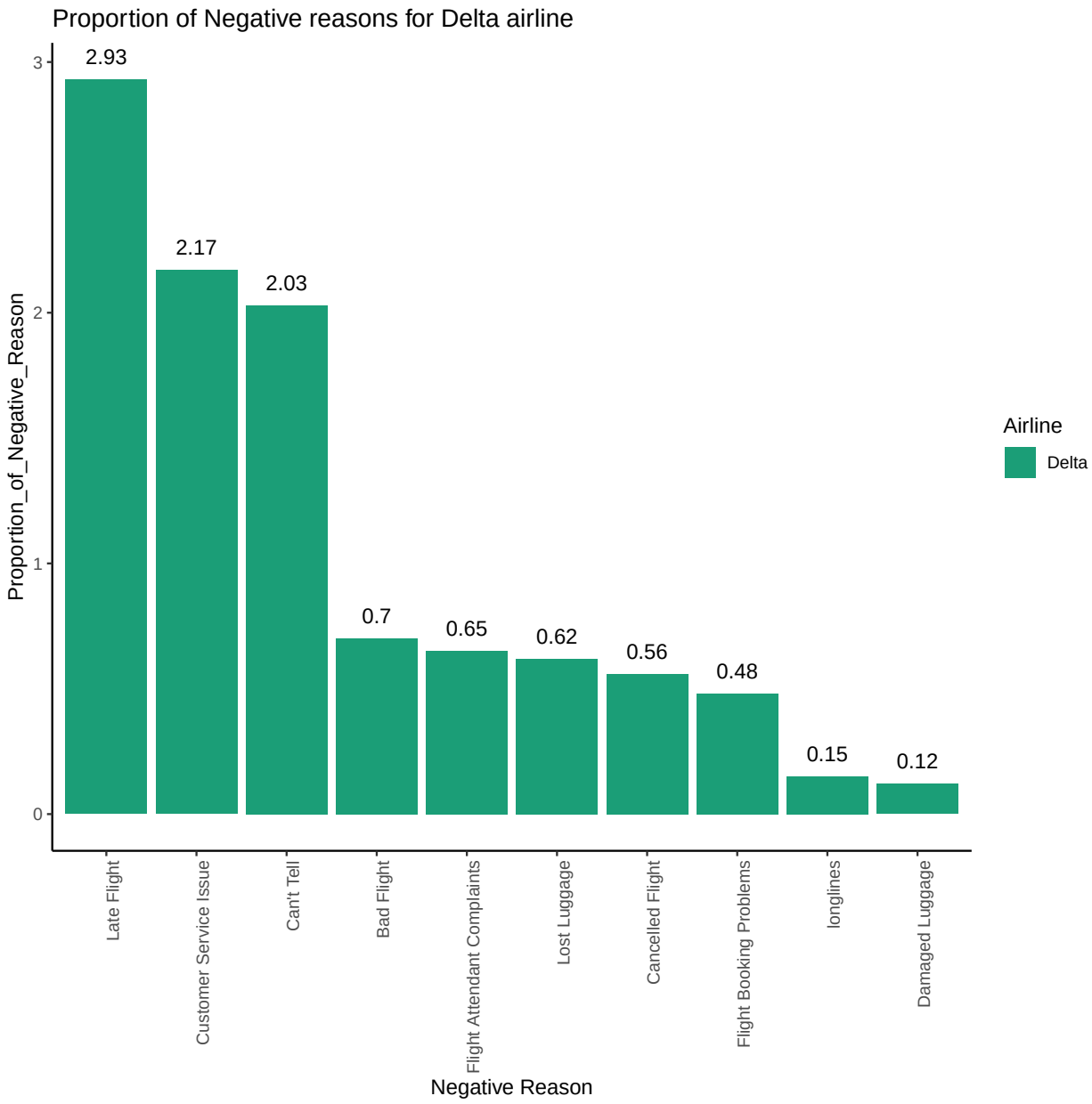


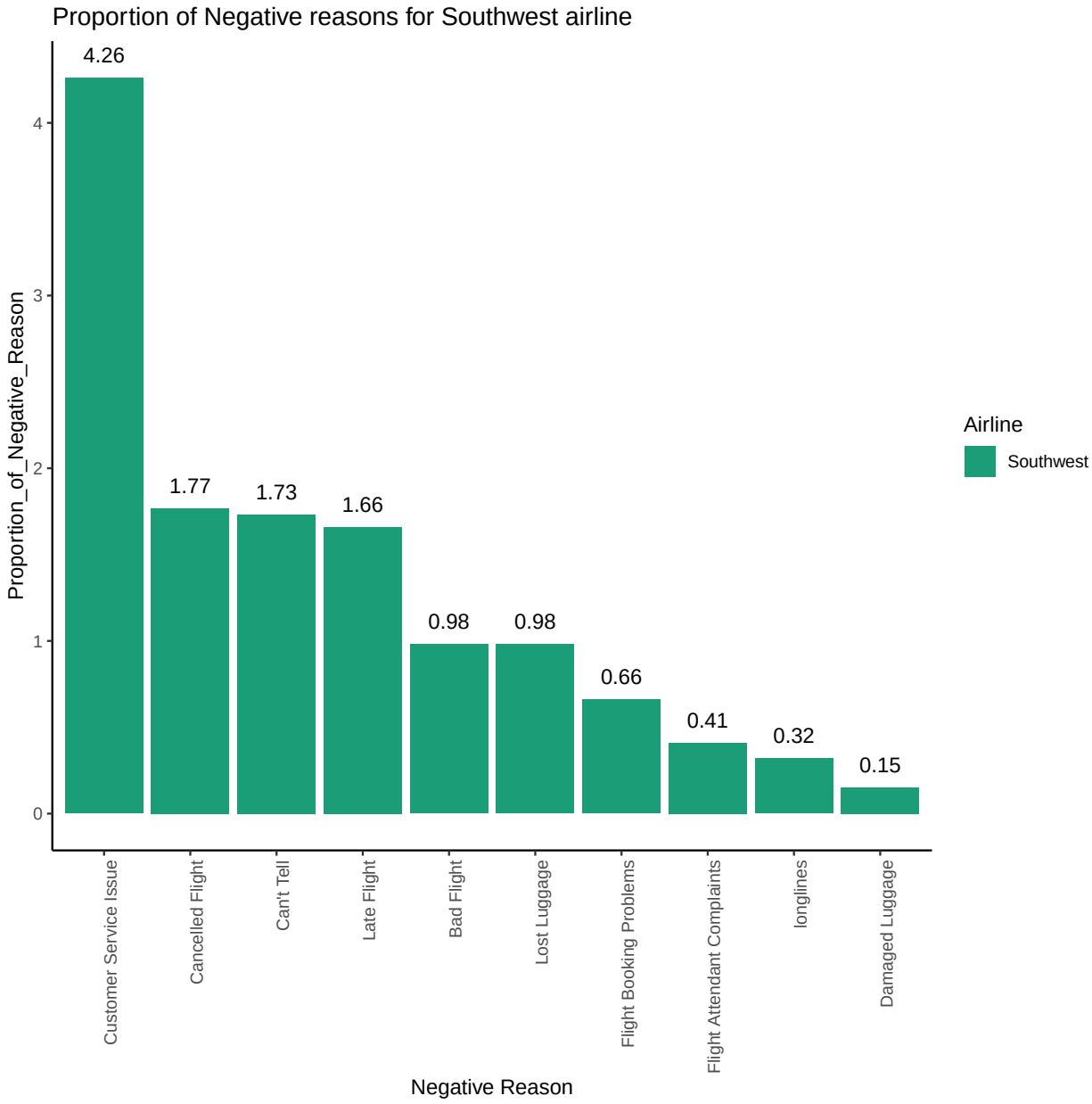


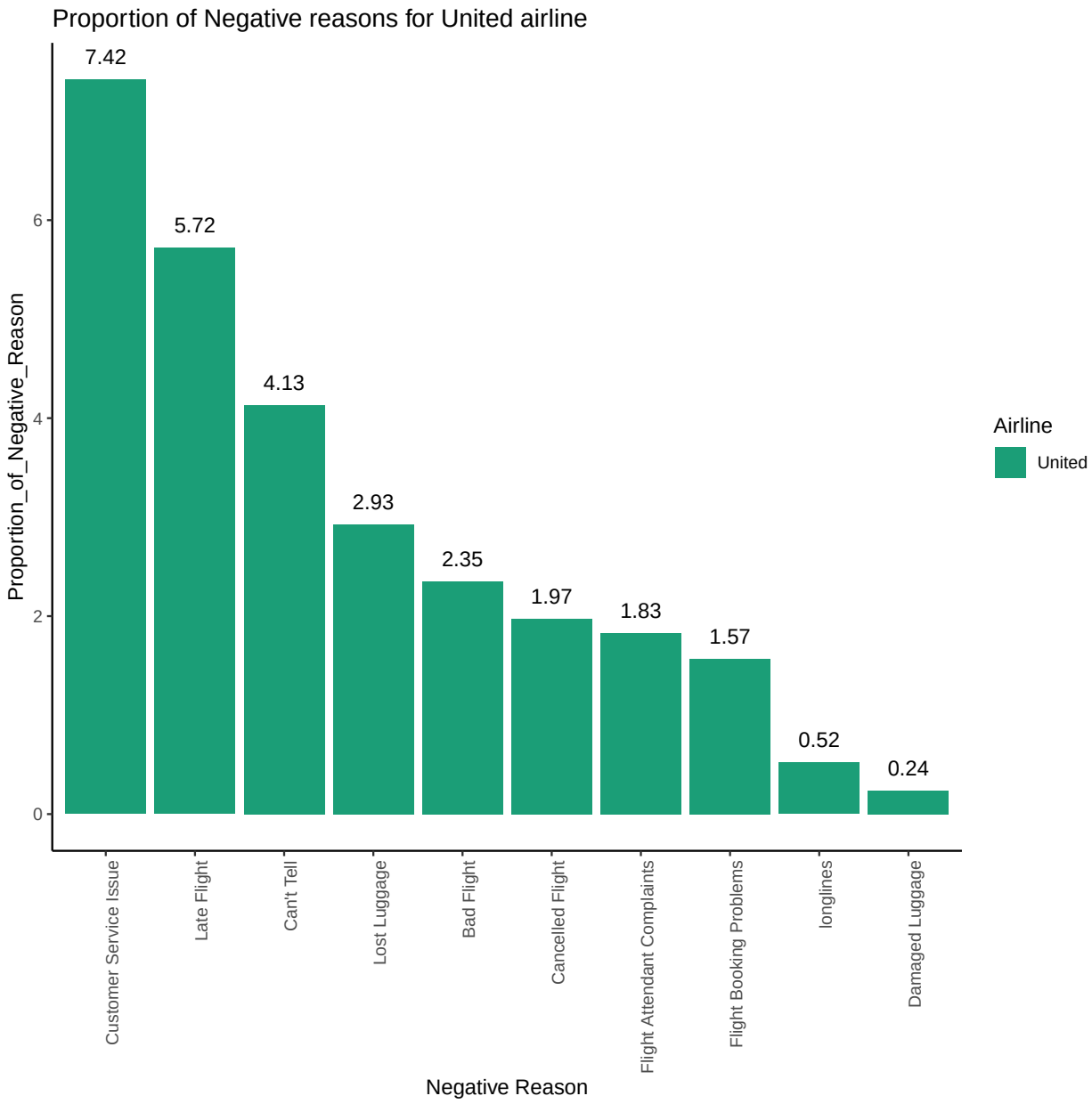
Interpretation: 1) It can be seen from the **Overall Sentiment** bar plot as well as from the pie chart that about *62.7%* tweets contain negative sentiment. 2) **Sentiment proportion per airline** bar plot displays that **United airlines** has maximum negative tweet and **Virgin America** has least negative tweet.

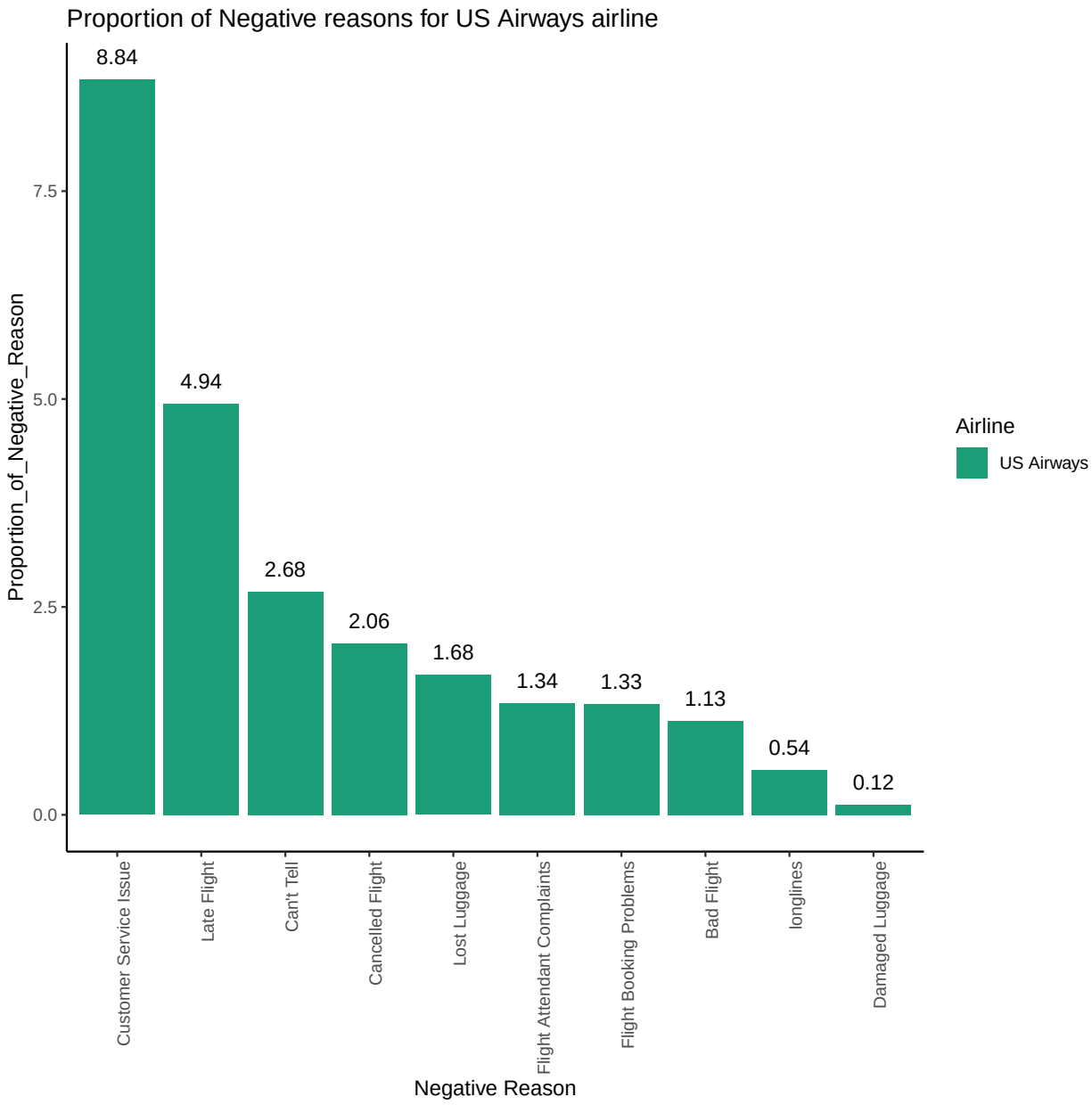
e) Reason for negative sentiment tweet per flight

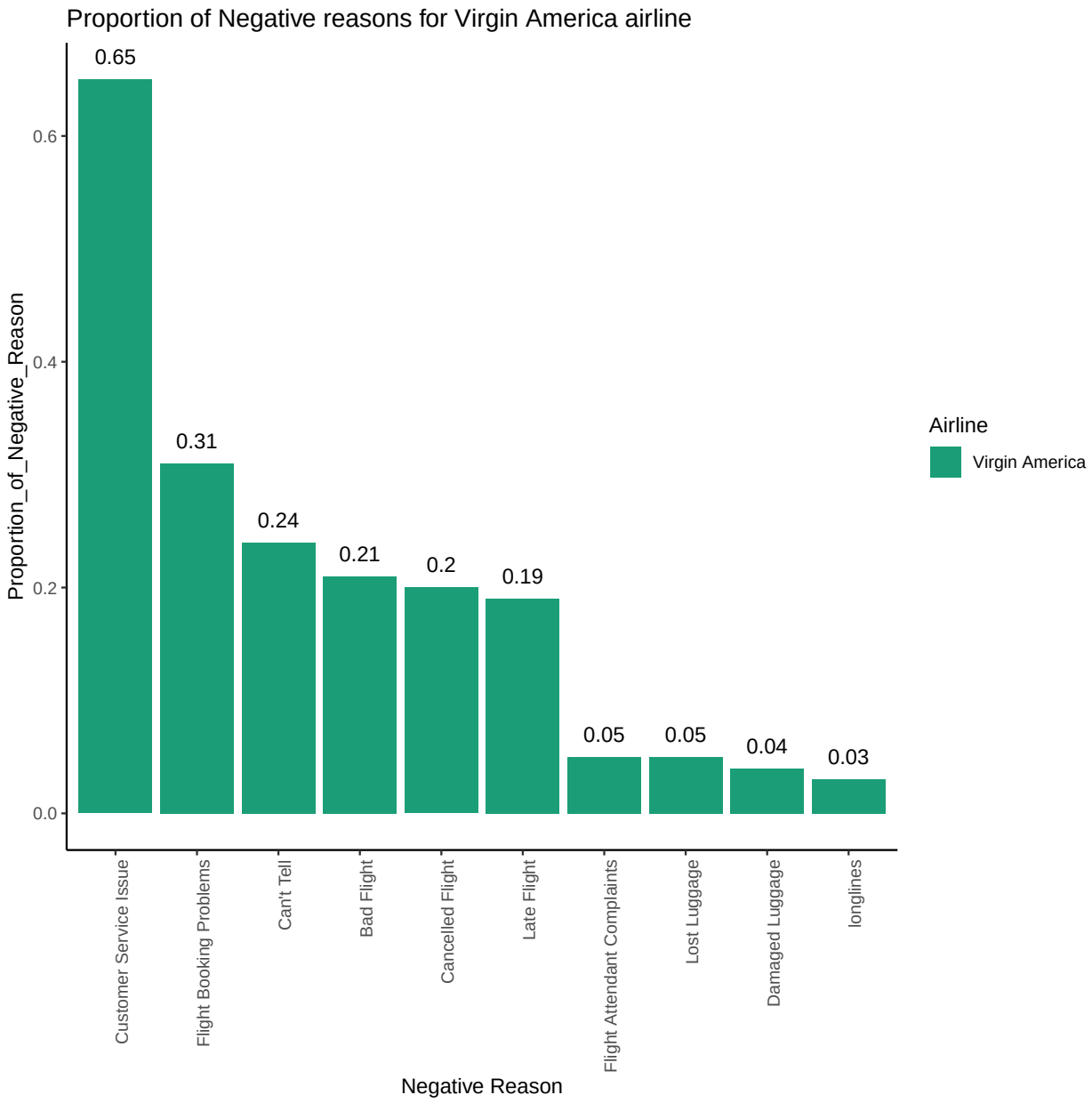



Proportion of Negative reasons for Delta airline
Proportion_of_Negative_Reason
3
2
1
0
2.93
2.17
2.03
0.7
0.65
0.62
0.56
0.48
0.15
0.12
Late Flight
Customer Service Issue
Can't Tell
Bad Flight
Flight Attendant Complaints
Lost Luggage
Cancelled Flight
Flight Booking Problems
longlines
Damaged Luggage
Negative Reason
Airline
Delta


Proportion of Negative reasons for Southwest airline
Proportion_of_Negative_Reason
Negative Reason
Airline
Southwest
4.26
1.77
1.73
1.66
0.98
0.98
0.66
0.41
0.32
0.15
Customer Service Issue
Cancelled Flight
Can't Tell
Late Flight
Bad Flight
Lost Luggage
Flight Booking Problems
Flight Attendant Complaints
longlines
Damaged Luggage
0
1
2
3
4


Proportion of Negative reasons for United airline
Proportion_of_Negative_Reason
0
2
4
6
7.42
5.72
4.13
2.93
2.35
1.97
1.83
1.57
0.52
0.24
Customer Service Issue
Late Flight
Can't Tell
Lost Luggage
Bad Flight
Cancelled Flight
Flight Attendant Complaints
Flight Booking Problems
longlines
Damaged Luggage
Negative Reason
Airline
United


Proportion of Negative reasons for US Airways airline
Proportion_of_Negative_Reason
0.0
2.5
5.0
7.5
8.84
4.94
2.68
2.06
1.68
1.34
1.33
1.13
0.54
0.12
Customer Service Issue
Late Flight
Can't Tell
Cancelled Flight
Lost Luggage
Flight Attendant Complaints
Flight Booking Problems
Bad Flight
longlines
Damaged Luggage
Negative Reason
Airline
US Airways



Interpretation:
1) Overall, we see that most negative sentiments are due to ***Customer Service Issue (presumably bad customer service)***. 2) ***United*** and ***US airways*** have a number of complaints for ***Customer Service Issues*** followed closely by ***Late Flights***. 3) For ***American airlines***, negative sentiment is elicited mostly by ***Customer Service Issues***, and not so much for ***Late Flights***. ***Southwest and virgin Airline*** have similar reasons for negative tweets. 4) ***Virgin America*** seems to have a sub-optimal booking system, as ***booking problems*** is the second reason eliciting bad sentiment in the tweets. 5) For ***Delta***, on the contrary, most of the complaints are due to ***late flights***. They show a perhaps ***better customer service*** compared to other airlines.

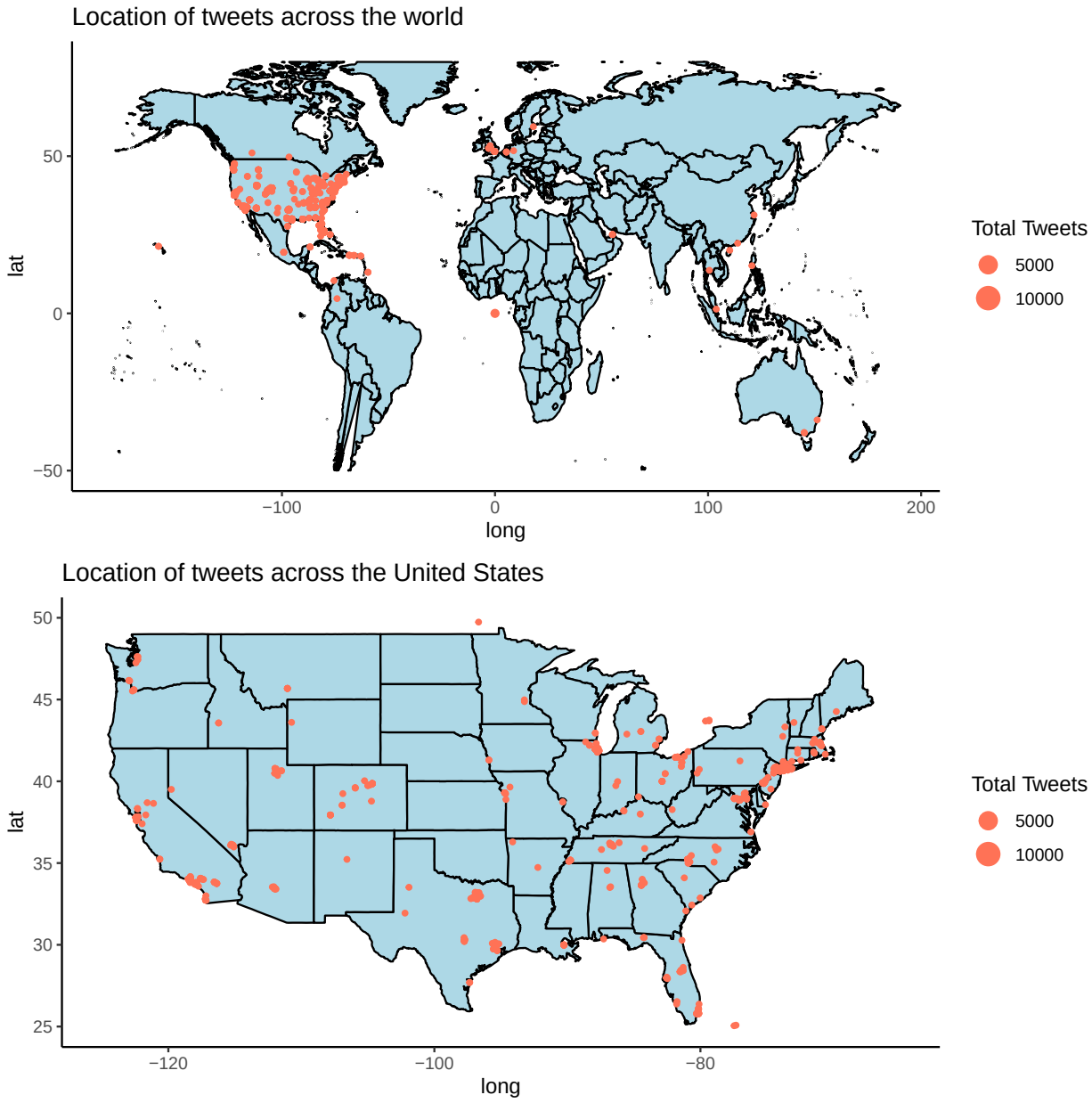
f) Timezone of tweets

Table 4: Timezone vs Frequency Table

Timezone	Frequency
	0.329
Eastern Time (US & Canada)	0.256
Central Time (US & Canada)	0.132
Pacific Time (US & Canada)	0.083
Quito	0.050
Atlantic Time (Canada)	0.034
Mountain Time (US & Canada)	0.025
Arizona	0.016
London	0.013
Alaska	0.007

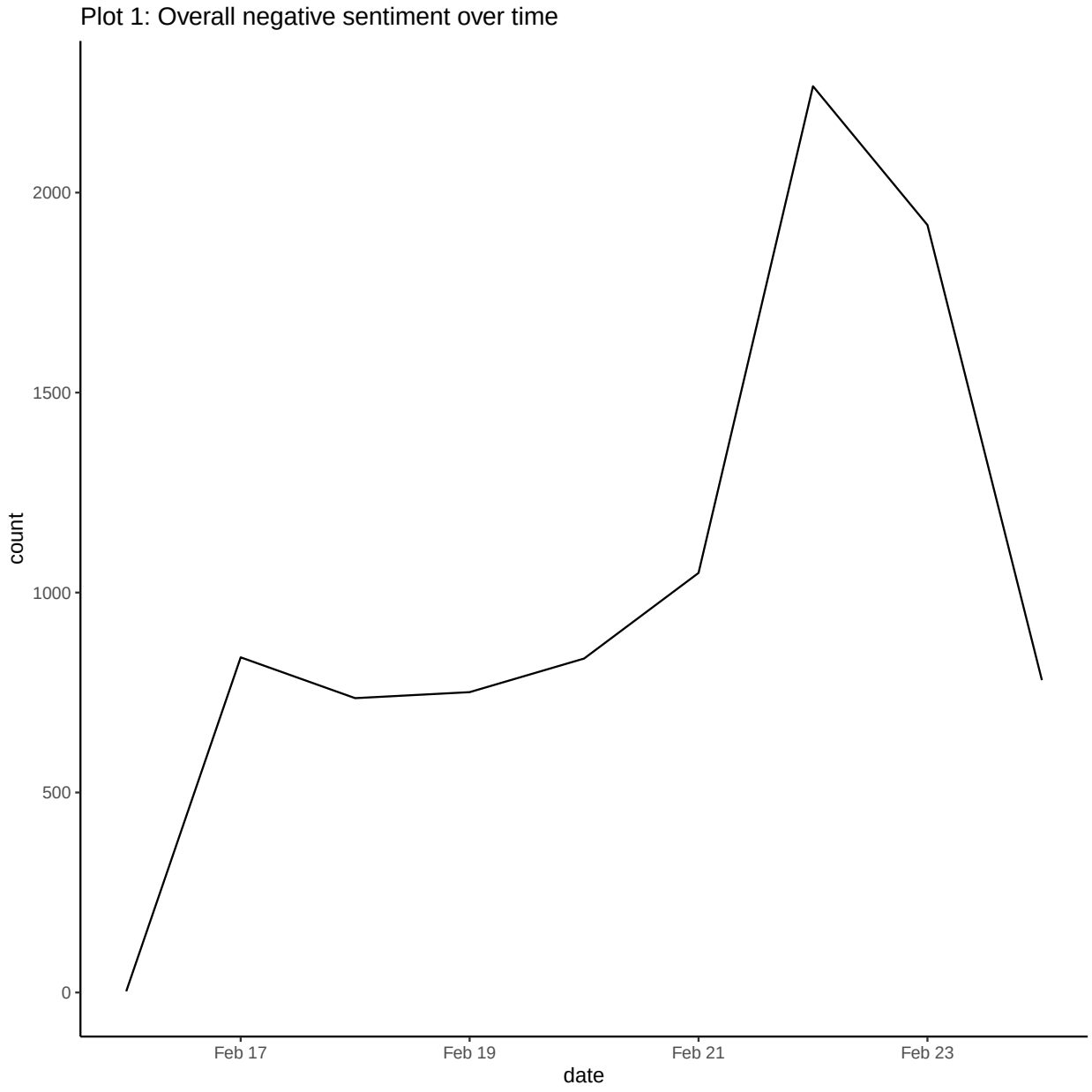
Interpretation: Majority of tweets coming from *Eastern time zone* and almost all the tweets come from *US & Canada time zone*.

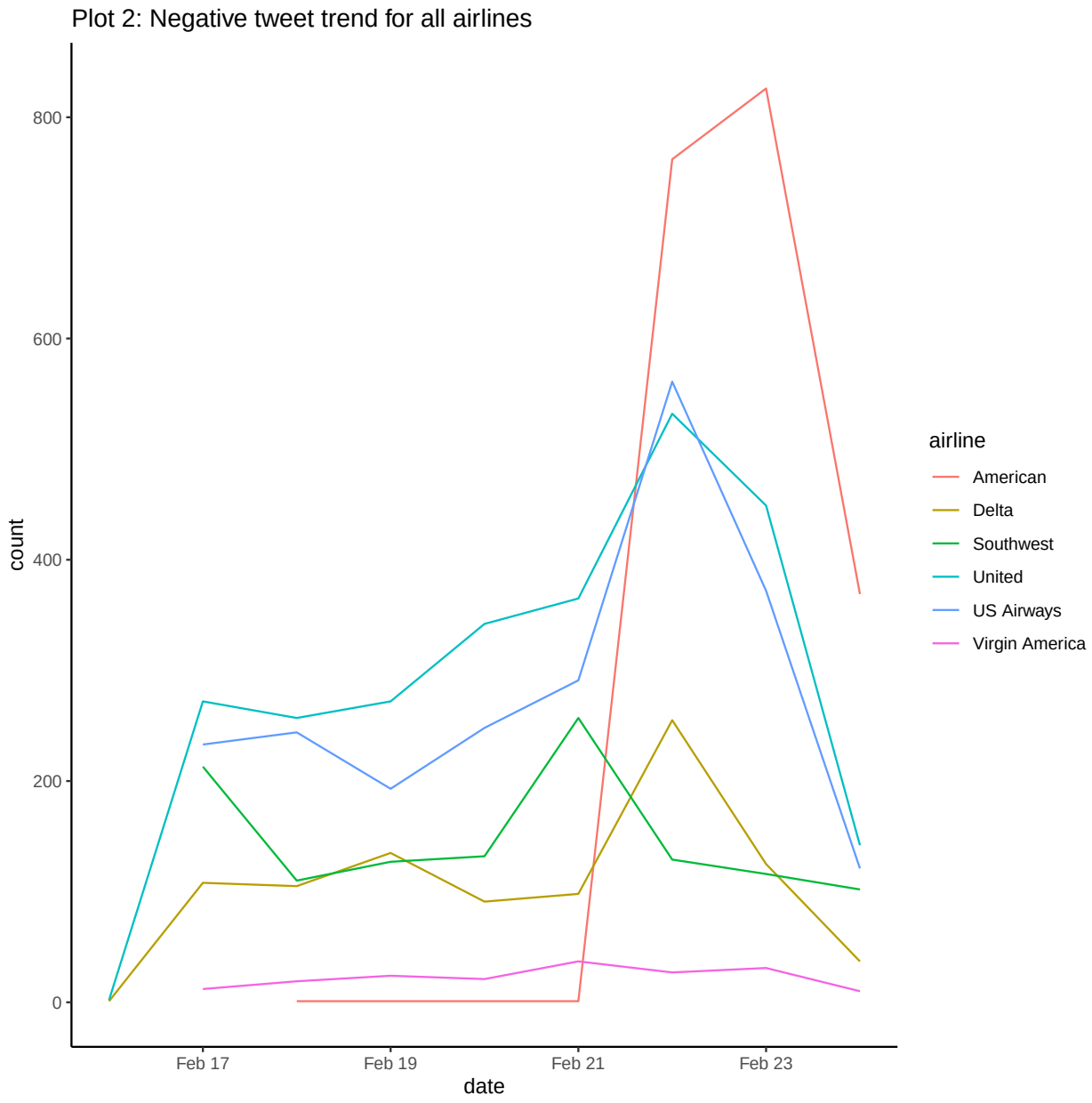
g) Location of tweets



Interpretation: Majority of tweets are coming from United states.

h) Timeline for negative tweets





Tweet Timeline Interpretation: 1) Plot 1- It shows that there was a *spike in negative tweet on 22ND February*. 2) Plot 2- When analysed for each flight it appeared that the **negative tweets were more for American airline on 22ND February**. Something look suspicious as till 21st February count was 0 and suddenly it spiked and reached maximum.

i) Word cloud for most frequent negative sentiment

Negative Word cloud Interpretation: From above displayed word cloud, as expected words like *Cancelled, delayed, customet service* are appearing in negative tweets.

j) Word cloud for most frequent positive sentiment

Positive Word cloud Interpretation: From above displayed word cloud, as expected words like **thanks, great, love, amazing, and Virgin America** (which had least negative tweet appearing in word cloud)

5) Tweet text visualization to strategies classification algorithm :

a) Remove neutral tweets from dataset and check proportion of negative and positive tweets

- Created a new dataset containing only negative and positive sentiment tweets and airline sentiment and text column only

- Removed @ sign from each text

Sentiment	Frequency
negative	79.5
positive	20.5

Interpretation: Dataset contains **79.5% Negative tweet and 20.5% positive tweets.**

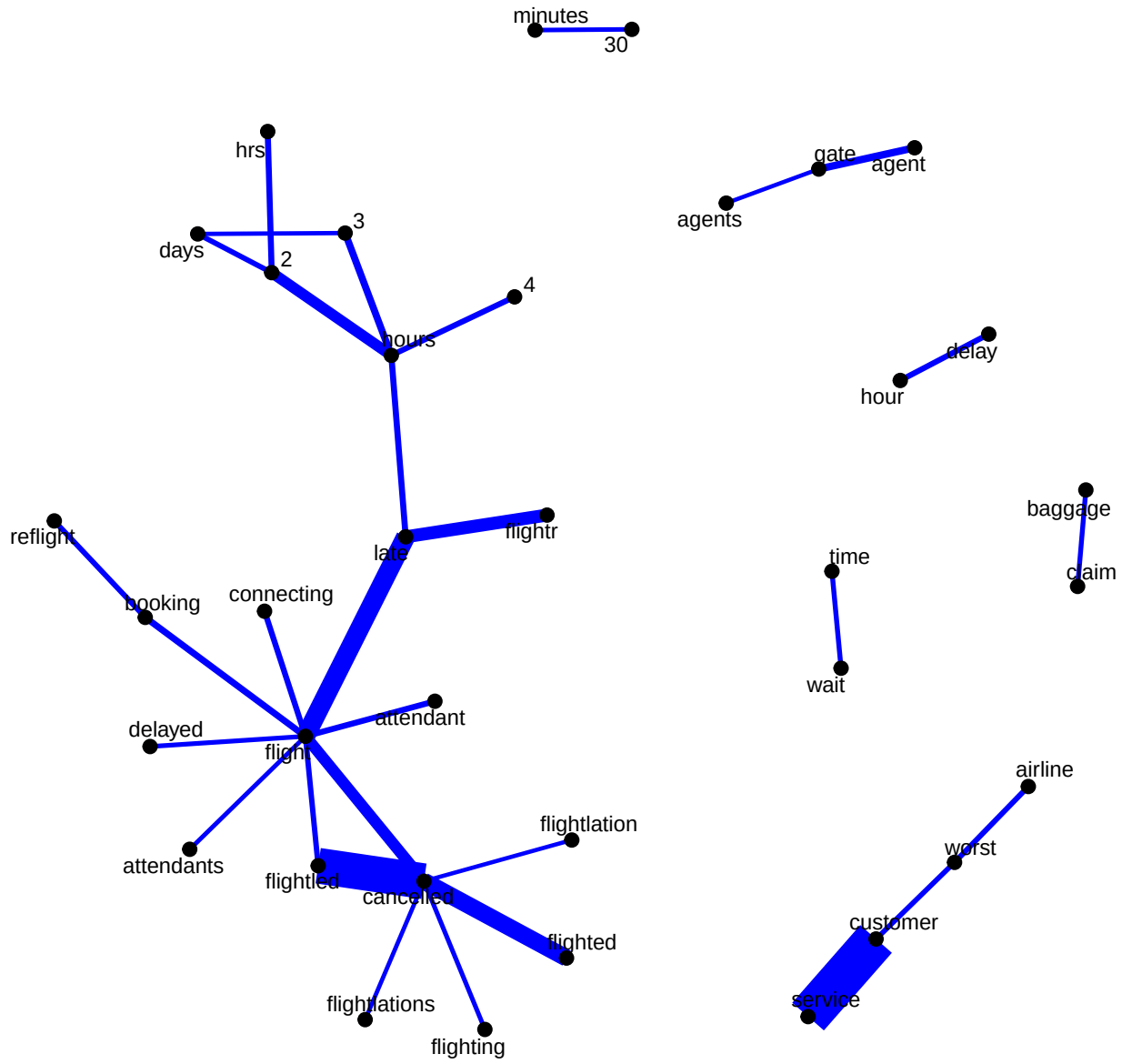
b) Tweet text Unigram and Bigram word frequency and Bigram network visualization

Table 6: Twitter data Uni-Grams

word	n
flight	3270
cancelled	957
service	896
customer	716
time	677
hours	650
hold	614
im	610
2	589
plane	565

Table 7: Twitter data Bi-Grams

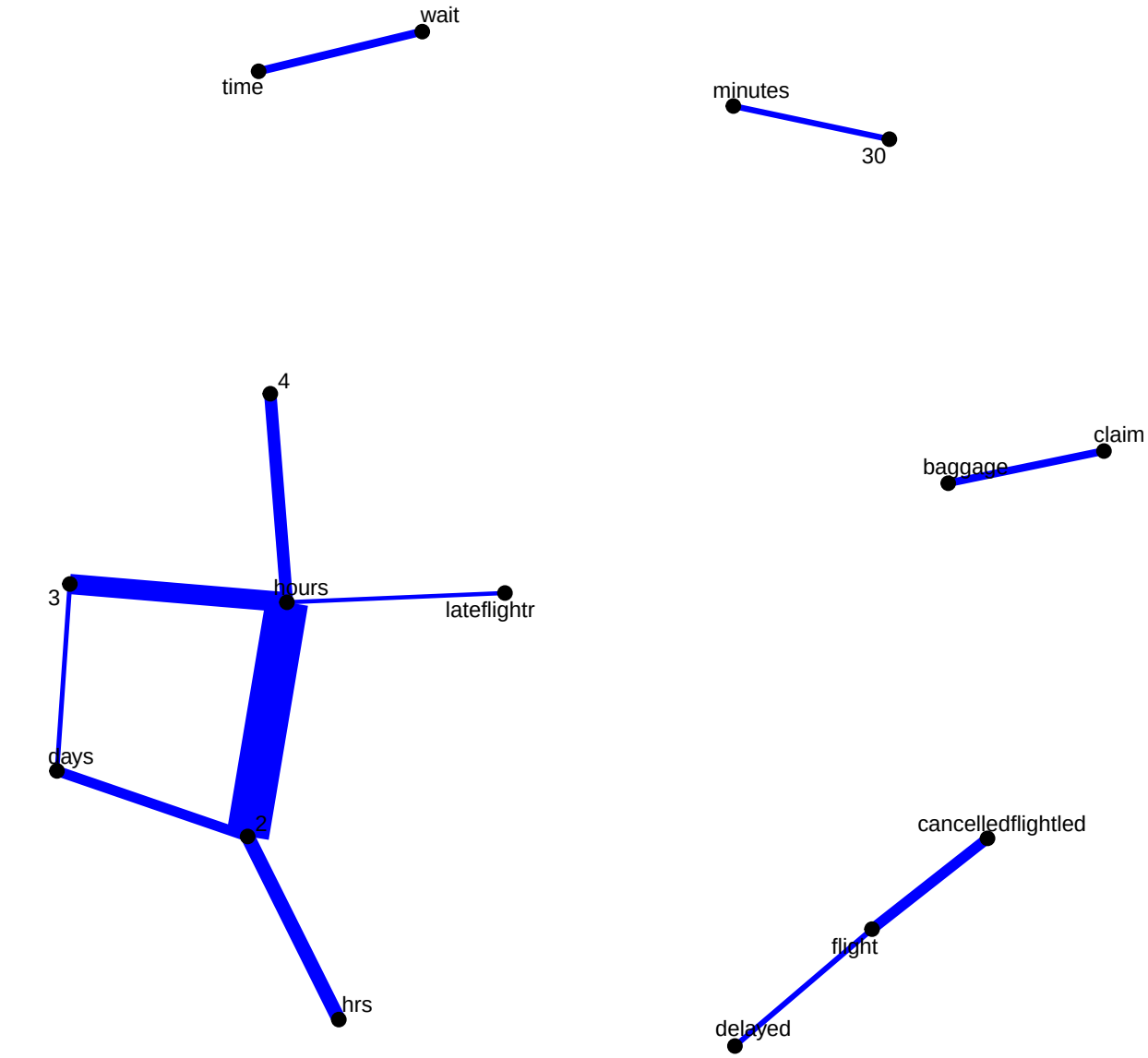
word1	word2	n
customer	service	534
cancelled	flightled	450
late	flight	229
cancelled	flighted	203
late	flightr	147
cancelled	flight	127
2	hours	121
flight	cancelled	77
gate	agent	73
3	hours	69



Interpretation: From above bigram network visualization, it can be seen that **Customer and service** word appeared almost 534 times adjacent to each other, similarly **Cancelled flightled** appeared 450 adjacent to each other similarly late flight.

c) Merged words which commonly occur together based on Bigram high frequency value

To make our bag of word technique more meaningful, we can merge frequently occurring adjacent words. Let say, if we have customer and service as separate word It will not make much sense but if we can merge whenever customer and service are present together then we will get inference tweet is talking about customer service.



6) Preprocess the Data :

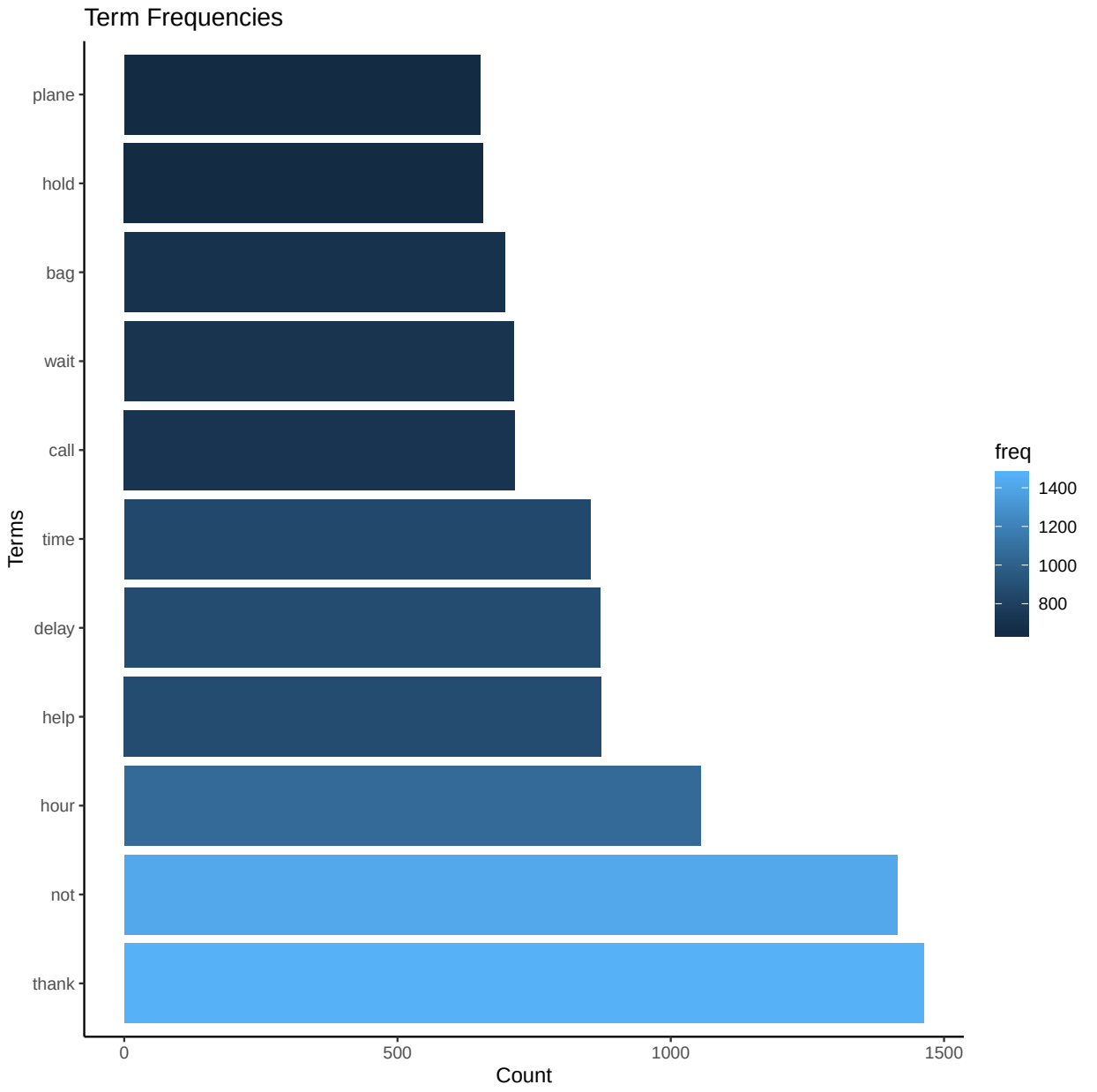
a) Tweet cleaning and creating corpus

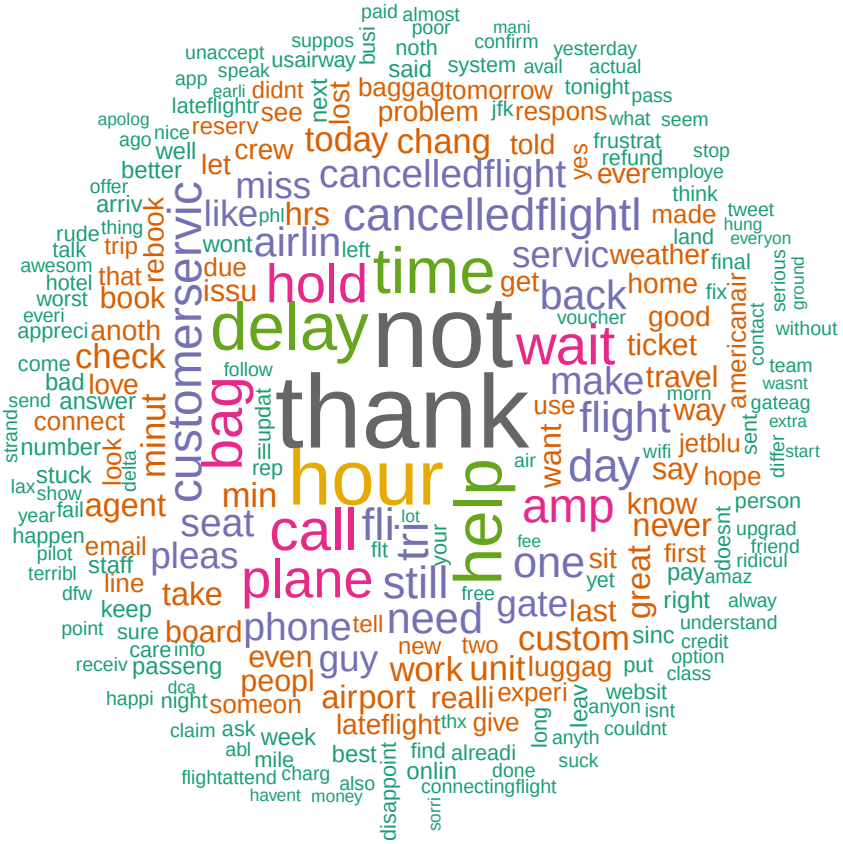
- Created a corpus contain each possible word from all sentences written in the text column
- Put every word into lower case
- Then removed punctuation, English stop words, strip white spaces, and stem each word.
- **Note 1:** Here I have made changes in English.dat file. As on removing stop words all the negative words were also getting removed. Therefore, i have removed those negative words from enlish.dat file such as **isn't, aren't, wasn't, weren't, hasn't, haven't, hadn't, doesn't, don't, didn't,**

won't, wouldn't, shan't, shouldn't, can't, cannot, couldn't, mustn't, no , not. As some tweets conatians sarcastic word incorporating these negative word in tweet really helped in improving the accuracy of model.

- **Note 2:** To execute these code, user has to delete above mentioned word from enlish.dat file.

b) Corpus frequency plot and word cloud





Interpretation: We can see that both word cloud and Bar plot shows a high frequency of these words **thank, not, hour, cancelledflight, 'customerservice'** in our tweets. Similar words were observed when we made word cloud for positive and Negative tweets individually. As most of the tweets are negative some with the sarcastic word. So having a high frequency of words like cancelled flight, customerService , thank, wait , hour, delay support our investigation.

c) Create a Document Term Matrix and split data into training(0.70) and test set(0.30)

Note: As we are dealing with tweet texts here and every tweet will have different text, to train other models it is better to incorporate more word but as computer computation power slows down when we train with more words, I kept only terms that appear in 0.7% or more of tweets).

```
## <<DocumentTermMatrix (documents: 11541, terms: 9813)>>
## Non-/sparse entries: 96204/113155629
## Sparsity           : 100%
## Maximal term length: 160
## Weighting          : term frequency (tf)

## <<DocumentTermMatrix (documents: 11541, terms: 254)>>
## Non-/sparse entries: 53566/2877848
## Sparsity           : 98%
## Maximal term length: 16
## Weighting          : term frequency (tf)
```

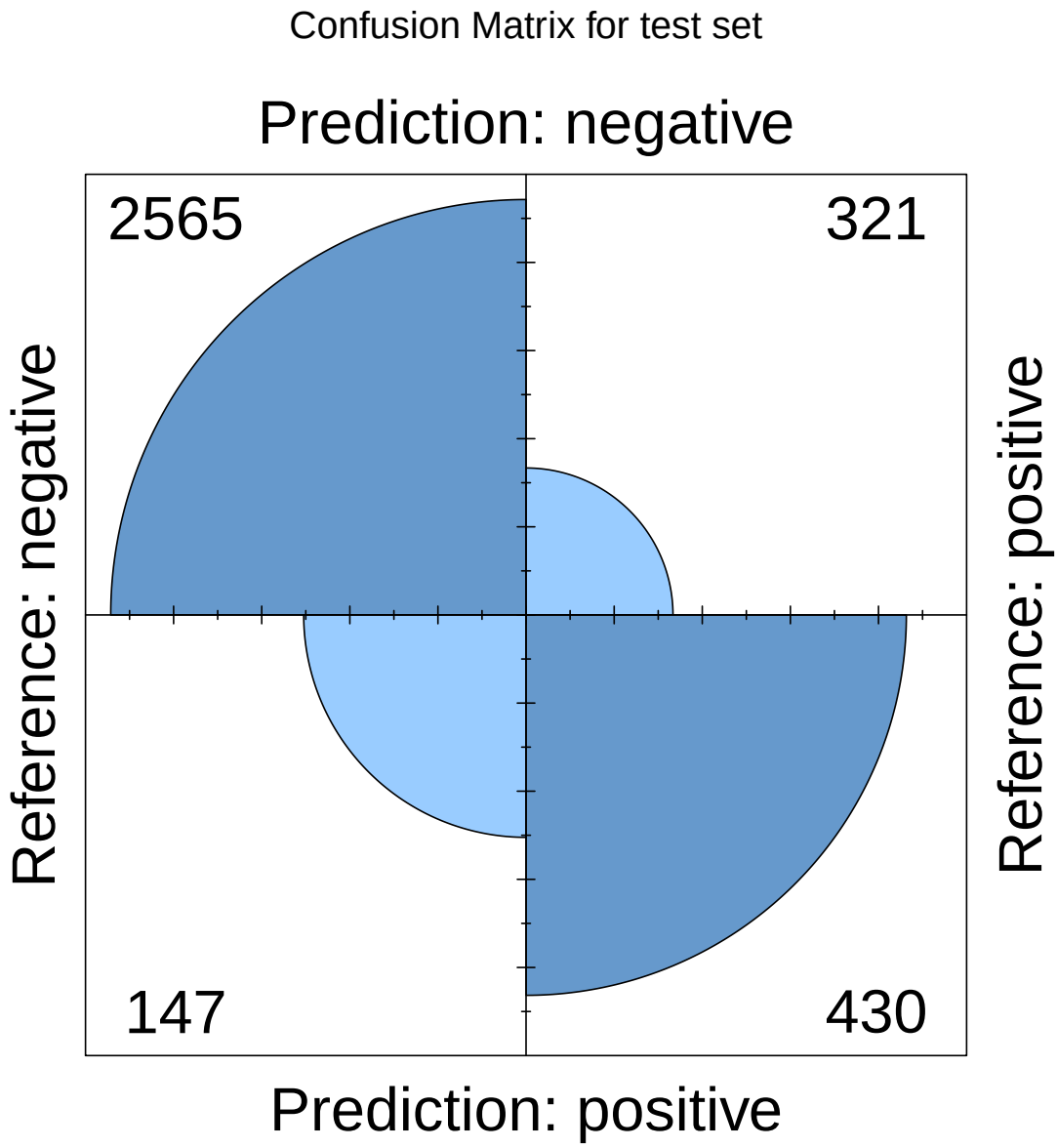
Interpretation: * We can see that DTM has 11541 documents and 9813 terms. This document has a lot of sparse terms. * Removed terms that dont appear often (keep only terms that appears in 0.7% or more of tweets).

7) Build the Models :

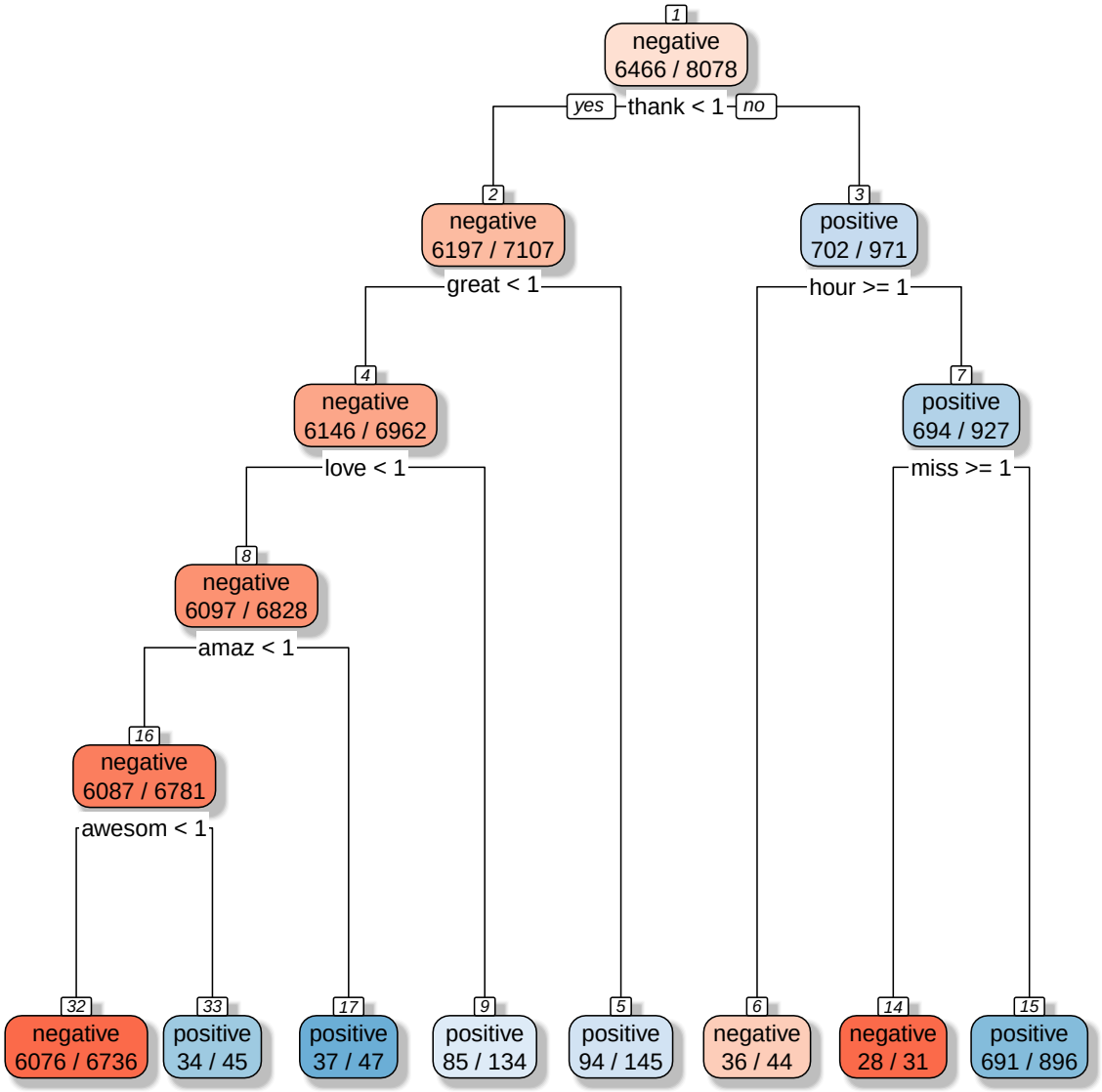
- **Strategy to build model:**
- Started with finding out how to identify word out of tweets which will help in the proper prediction of the class of interest.
- Initially, used unigram method, was getting good accuracy with the decision tree model. Then tried Bigram and again used the decision tree algorithm, accuracy was little less than the unigram model.
- Later, merged high-frequency words which were adjacent in the bigram network. After merging again executed the decision tree on Unigram with merged word and bigram with the merged word. As expected accuracy was improved from 0.84 to 0.86 for unigram with the merged word not much although.
- Executed Models: Logistic Regression(performed really bad in terms of prediction), Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (Kernel: Linear) **Note:** As Logistic regression and Naive Bayes was not giving good accuracy kept their execution result in appendix

(A) Model based on Unigram

a) Decision Tree Unigram(merged word) Model

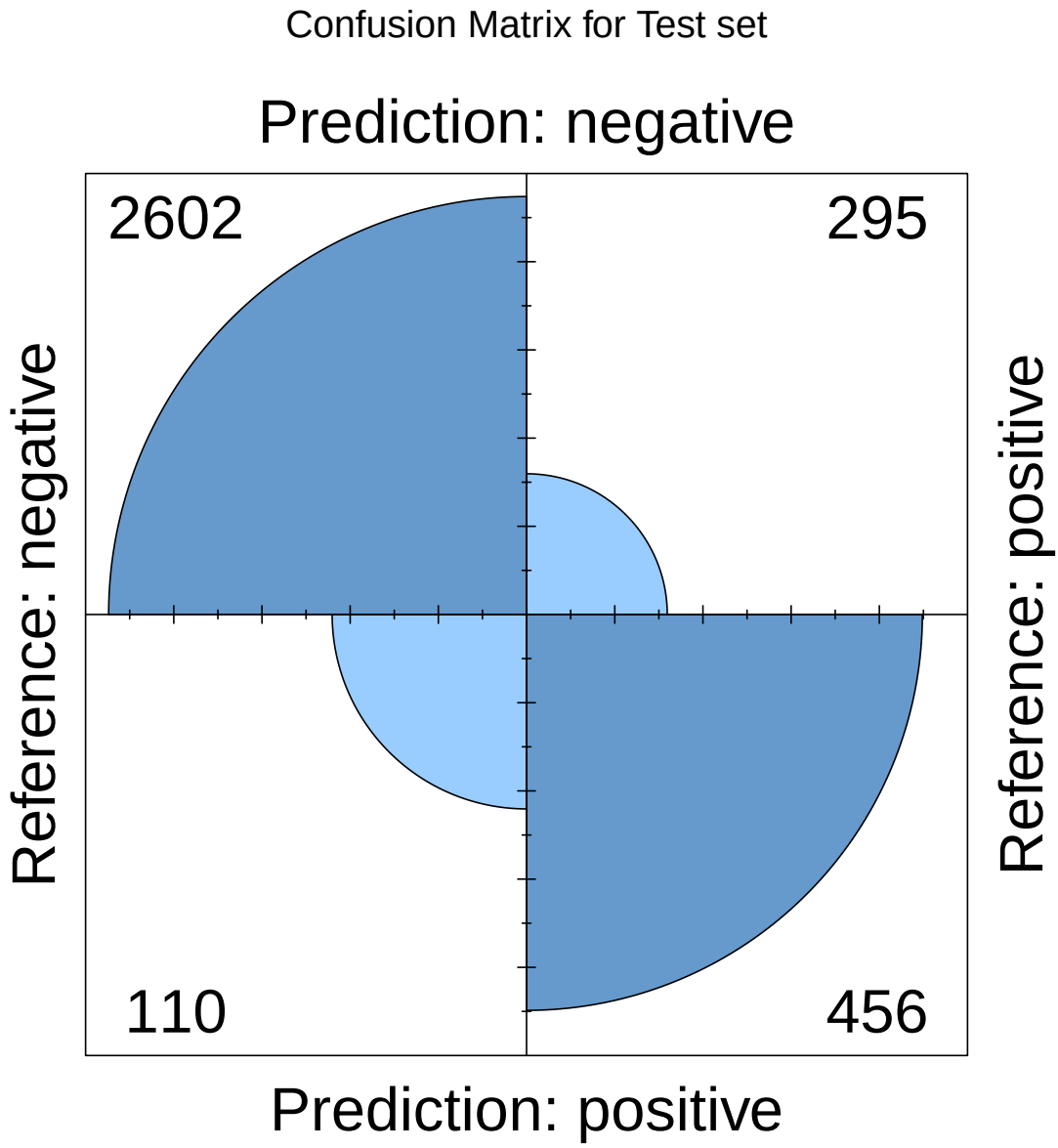



```
## Train score error:
## 0.123
## Test score error:
## 0.135
```

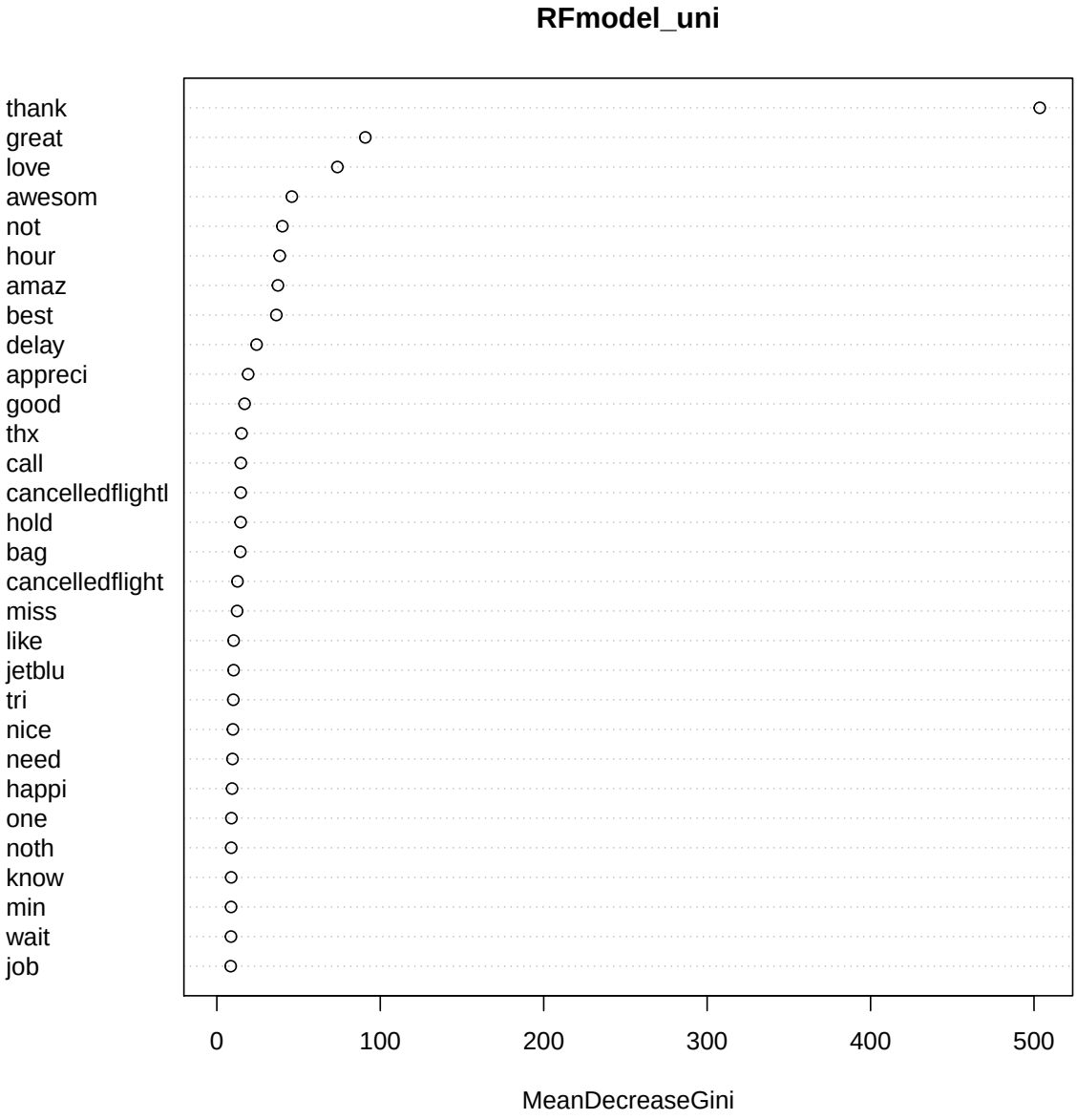


Decision Tree Unigram Interpretation: 1) Accuracy of Train Set:- 0.877 2) Accuracy of Test Set:- 0.865 3) It shows that the model is not overfitting 4) Looking at the Decision tree Plot following points has observed: a) If we look at the left branch of the tree, the text which does not contains words like **Thank, great, love, amaz, awesome** will classify as Negative. b) If we look at the right branch of the tree, the text which contains thank but hour is >= 1 is classified as negative. As expected in the case of the sarcastic tweet. c) Similar behavior can be observed in the case of text which contains thank, has **hour<1 and miss > 1** is classified as negative.

b) Random Forest Unigram(merged word) Model



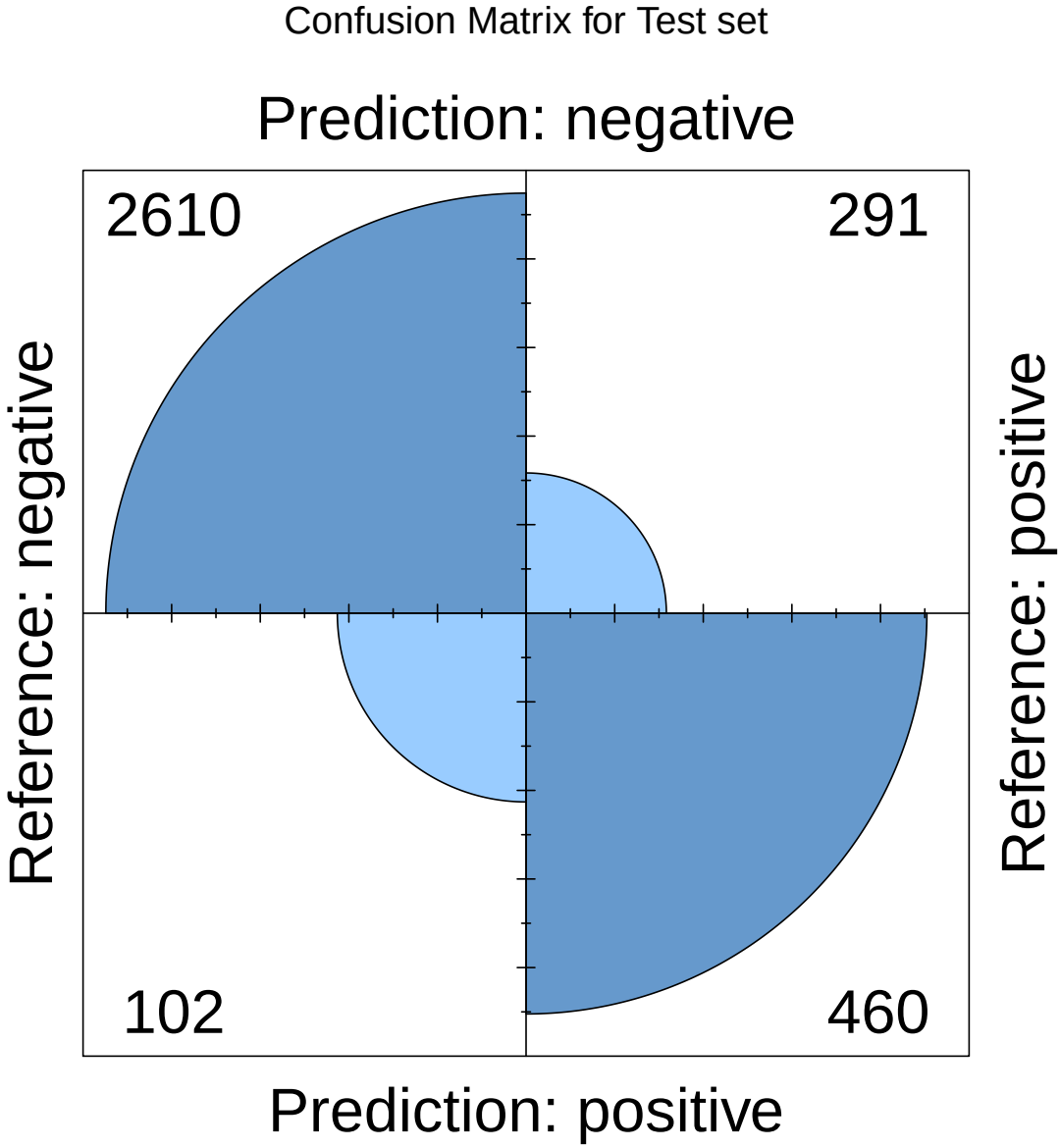

```
## Train score error:
## 0.0532
## Test score error:
## 0.117
```



Random Forest Unigram Interpretation: Train set accuracy: 0.947, Test set accuracy: 0.883. Its shows sign of little overfitting. The tree splitted on the word Thank.

d) SVM Linear Unigram (merged word) Model

Executed SVM Linear unigram model for hyperparameter tuning and identified optimize cost parameters.As its Execution nearly took more than 2 hours so after finding optimized C value used that in code and commented code for hyperparameter tuning.




```
## Train score error:
## 0.0923
## Test score error:
## 0.114
```

SVM Linear Model Interpretation: Train set accuracy: 0.908, Test set accuracy: 0.887. **SVM linear model shows best accuracy among all model.**

(B) Model based on Bigram

Create a Document Term Matrix for Bigram and Split data into training(0.70) and test set(0.30)

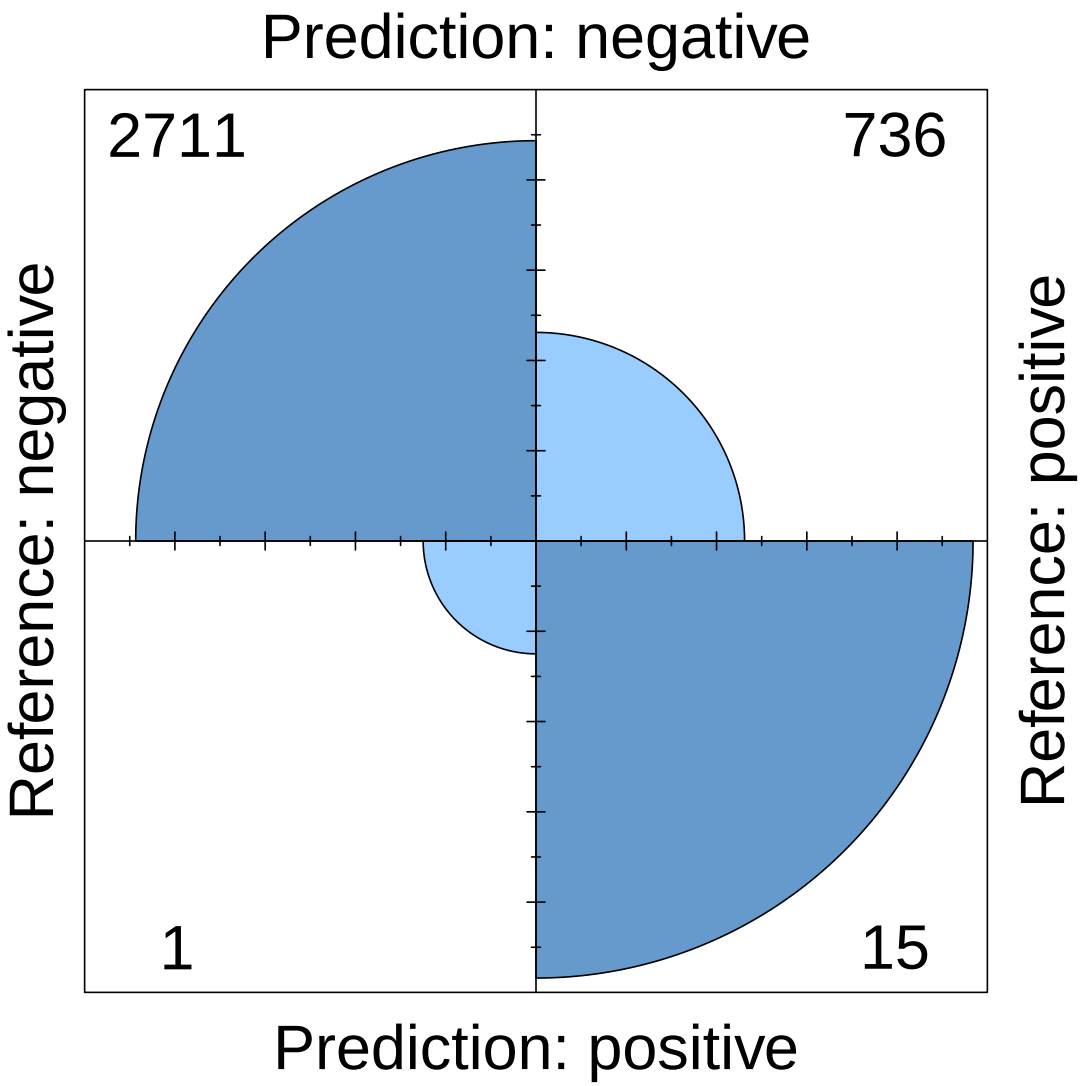
```
## <<DocumentTermMatrix (documents: 11541, terms: 67142)>>
## Non-/sparse entries: 93522/774792300
## Sparsity           : 100%
## Maximal term length: 166
## Weighting          : term frequency (tf)

## <<DocumentTermMatrix (documents: 11541, terms: 169)>>
## Non-/sparse entries: 5185/1945244
## Sparsity           : 100%
## Maximal term length: 23
## Weighting          : term frequency (tf)
```

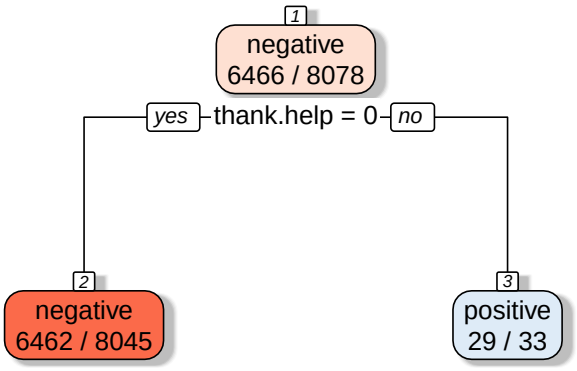
Interpretation:
1) We can see that DTM has 11541 documents and 67142 terms. This document has a lot of sparse terms.
2) Removed terms that dont appear often (keep only terms that appears in 0.15% or more of tweets).

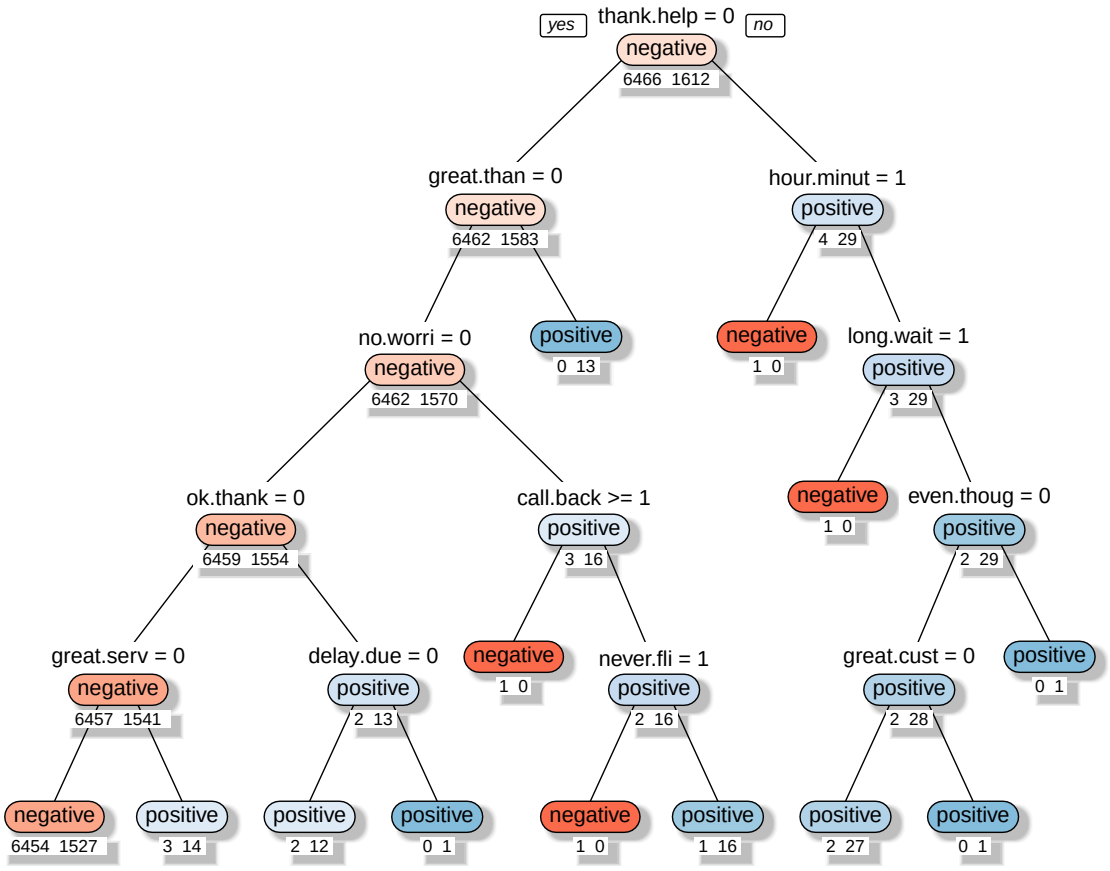
a) Decision Tree Bigram Model

The complexity parameter (cp) is used to control the size of the decision tree and to select the optimal tree size.The default value of cp is 0.01. A value of cp = 1 will result in a tree with no splits. Setting cp to a negative values ensures a fully grown tree.Therefore, As at CP =0, tree was not splitting into more branches so I have used cp=-1, which gives fully grown tree.



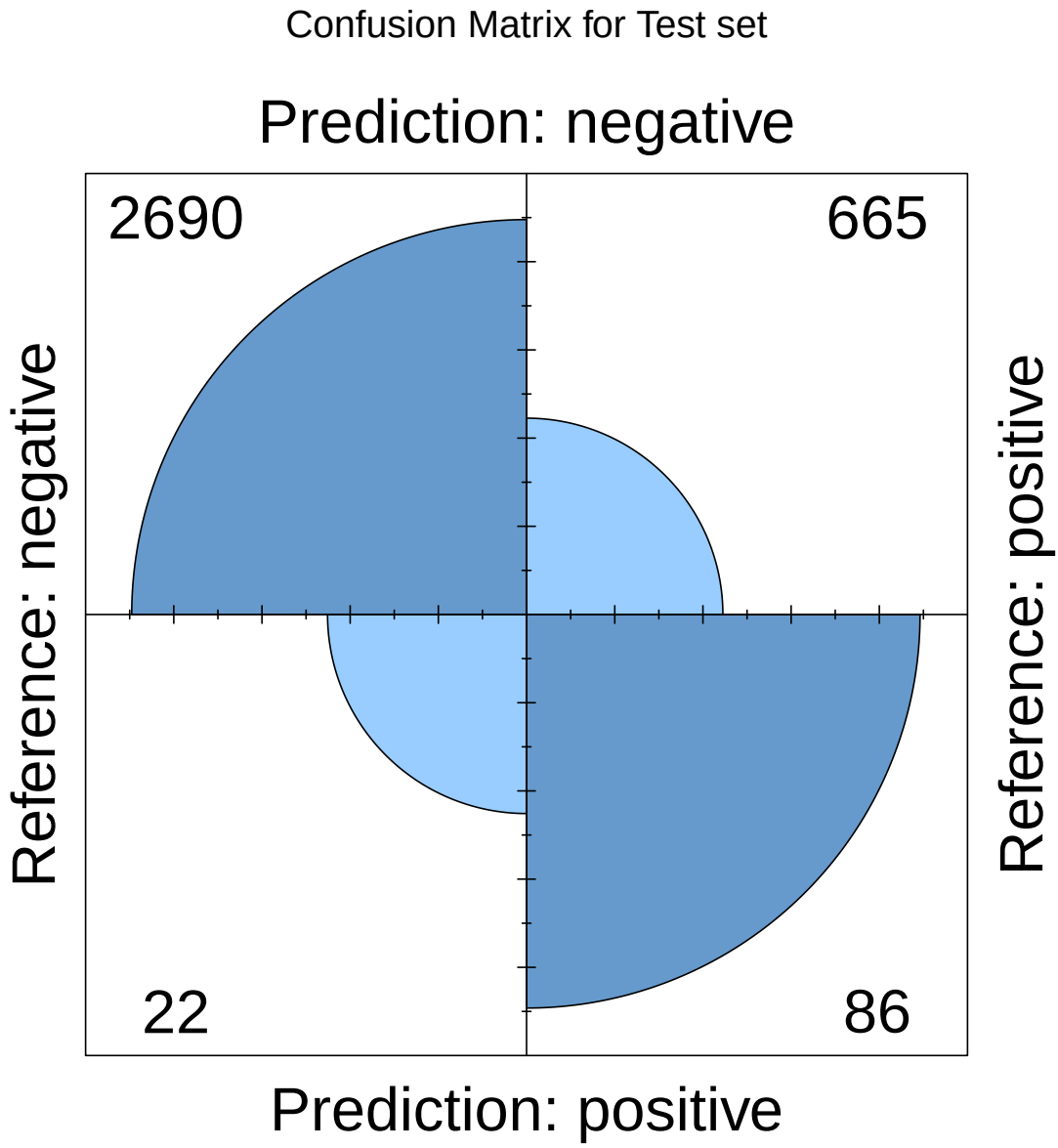

```
## Train score error:
## 0.197
## Test score error:
## 0.213
```


1
negative
6466 / 8078
yes
thank.help = 0
no
2
negative
6462 / 8045
3
positive
29 / 33

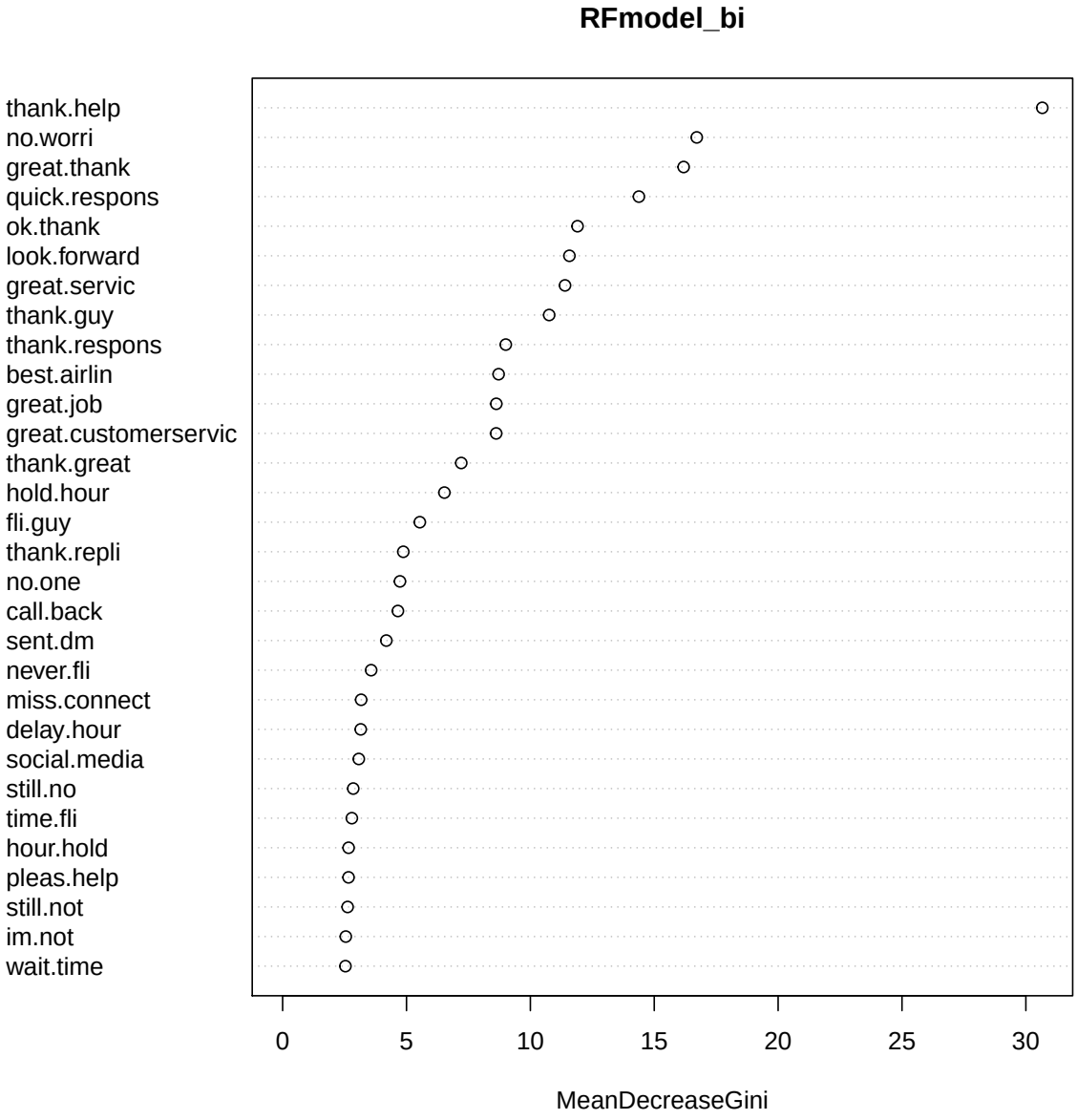


Decision Tree Bigram Interpretation: Train set accuracy: 0.804, Test set accuracy: 0.787. The word on which tree is splitting is **thank.help**. If **Thank.help** is present tweet is positive else negative. If tweets does not have word like great.thank, no worries, ok.thank and great.service. It has been classified as negative.

b) Random Forest Bigram Model



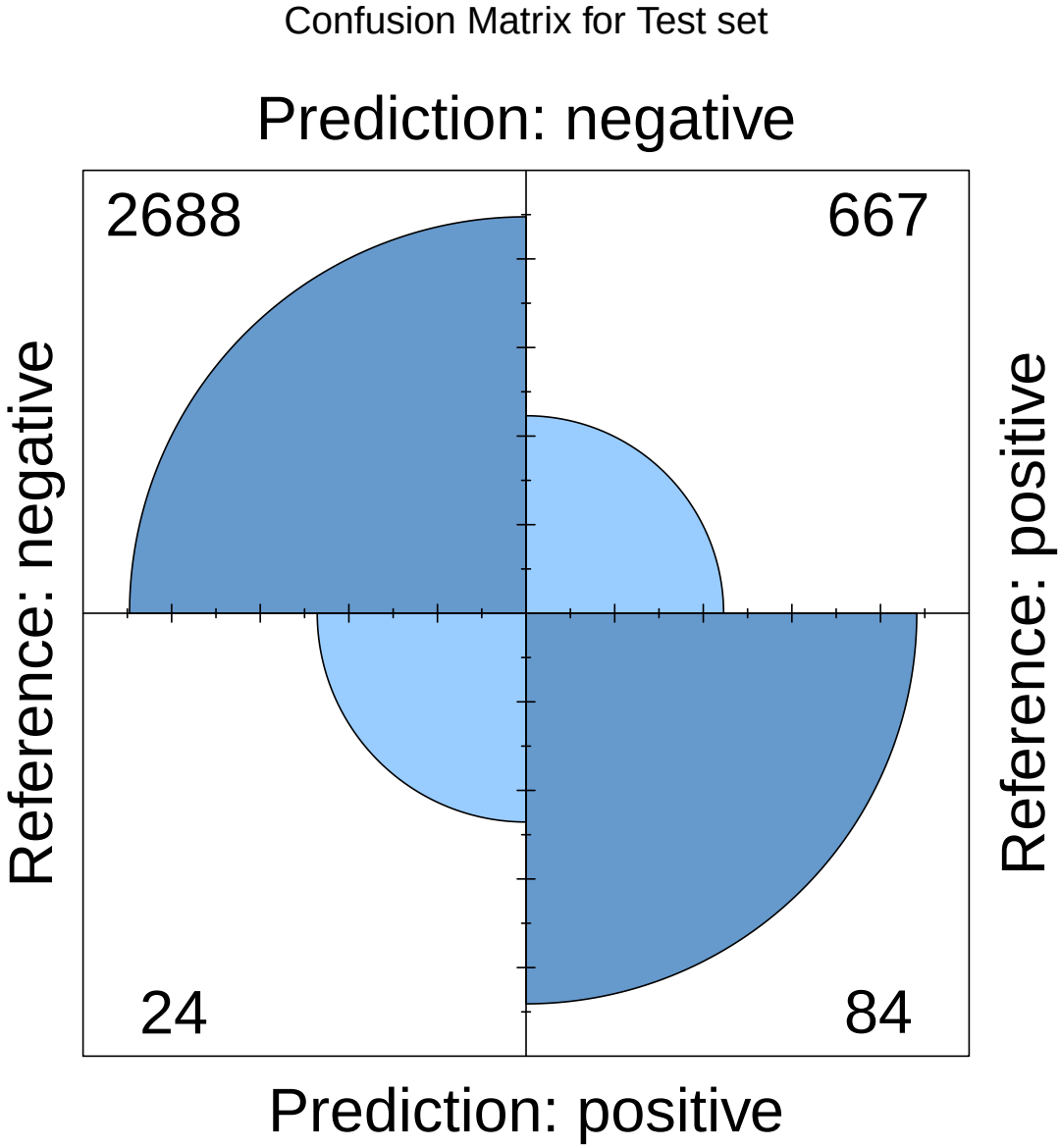

```
## Train score error:
## 0.181
## Test score error:
## 0.198
```



RF Model Bigram Interpretation: Train set accuracy: 0.819, Test set accuracy: 0.802.The word on which tree is splitting is *thank.help* which is similar to one which we had in Decision Tree.

c) SVM Linear Bigram Model

Executed the SVM Linear model to find out optimize cost parameters.As its Execution nearly took more than 2 hours so after finding an optimized C value which is 0.1 used that in the code directly.



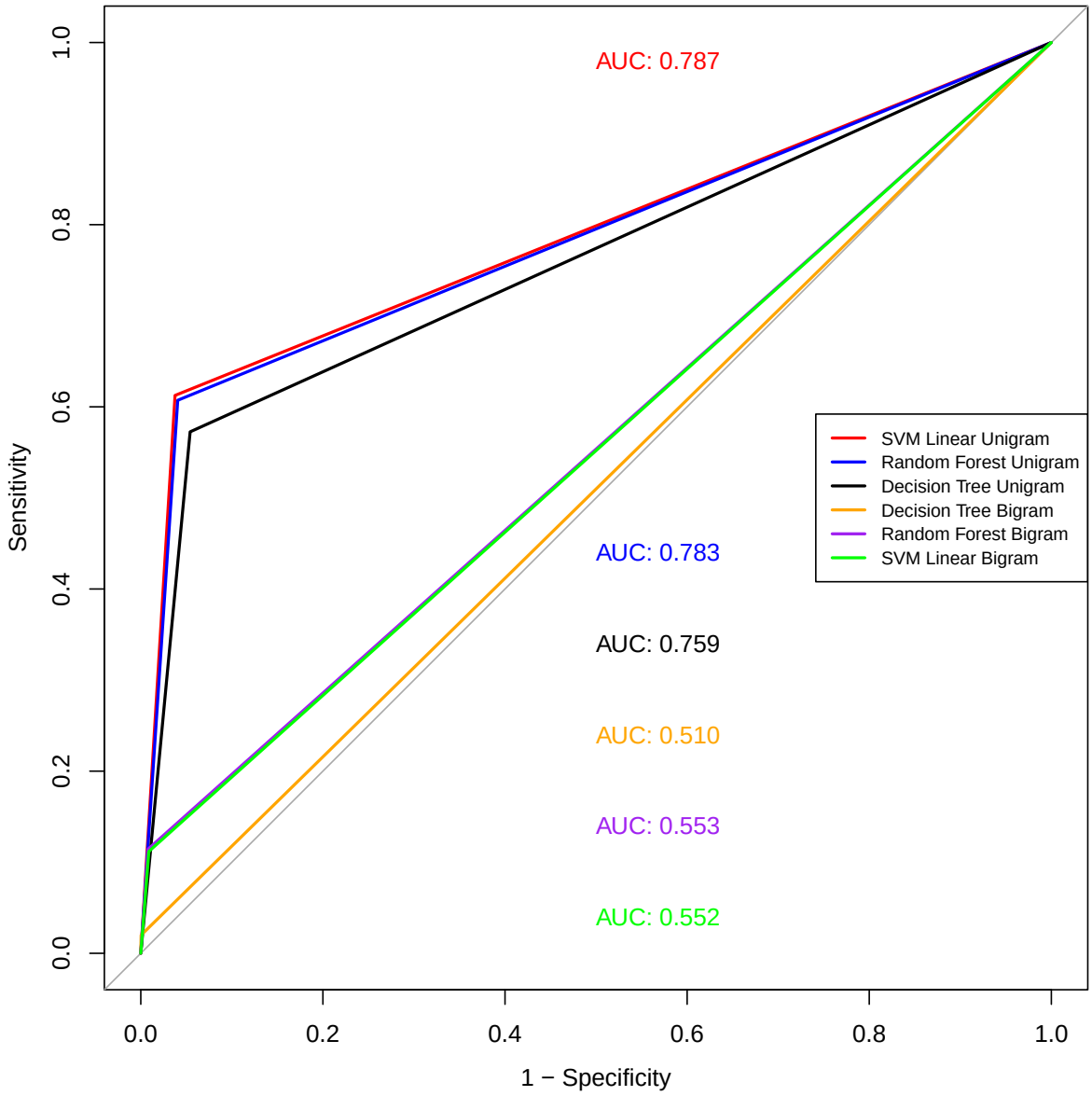

```
## Train score error:
## 0.179
## Test score error:
## 0.2
```

SVM(kernel:Linear) Bigram Interpretation: Train set accuracy: 0.821, Test set accuracy: 0.80

8) Conclusion :

a) Model Comparision and Reporting best model based on AUC value

AUC measures the performance of a binary classifier averaged across all possible decision thresholds.



ROC Curve Interpretation: It is evident from the above plot, that **SVM Linear(unigram)** is the best model for classifying US airline Twitter sentiments with the **highest area under curve value(0.787)**. Random Forest and Decision Tree also create a good model that has auc value pretty close to SVM Linear Model. Comparatively, Bigram Decision Tree and Random Forest model are not performing that well in classifying the tweet.

b) Model Comparision based On F1 score, Trainset and Testset accuracy and AUC score

As the dataset is highly imbalanced, So can not just rely on accuracy as our only scoring function hence will consider F1 and AUC for selecting the best model.

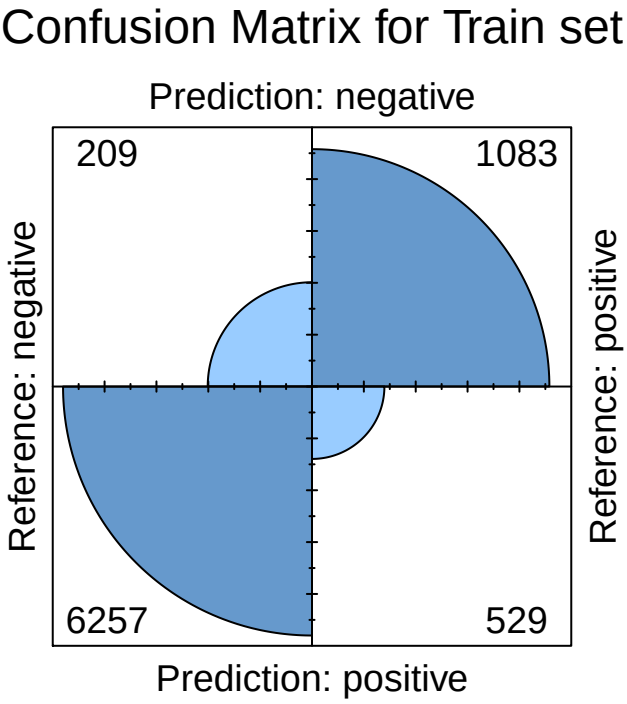
	Test Accuracy(%)	Train Accuracy(%)	F1 score	AUC score
Decision Tree Unigram	86.5	87.7	91.6	75.9
Random Forest Unigram	88.3	94.7	92.8	78.3
SVM Linear Unigram	88.7	90.8	93.0	78.8
Decision Tree Bigram	78.7	80.3	88.0	51.0
Random Forest Bigram	80.2	81.9	88.7	55.3
SVM Linear Bigram	80.0	82.1	88.6	55.1

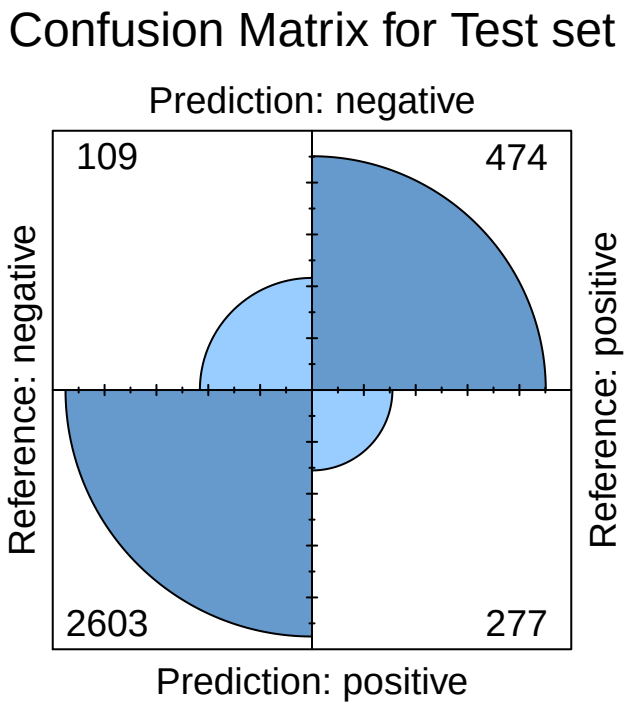
Interpretation: From the above model comparison table, it is evident that **SVM Linear Unigram** has the highest F1 score and AUC value. Its train and test set accuracy is also good. Therefore, It is the best algorithm to train our model for prediction. Although, Random Forest unigram is also giving good accuracy but shows the chance of overfitting.

9) Appendix :

a) Logistic Regression Unigram(merged word) Model

Executed Logistic Regression for Unigram with the merged word. Its accuracy was very less nearly 0.12. Therefore, did not include in the model comparison.

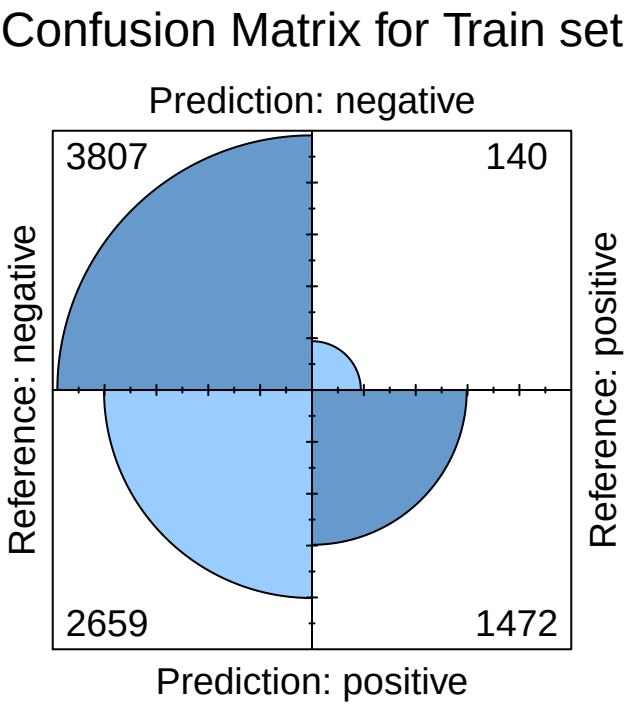


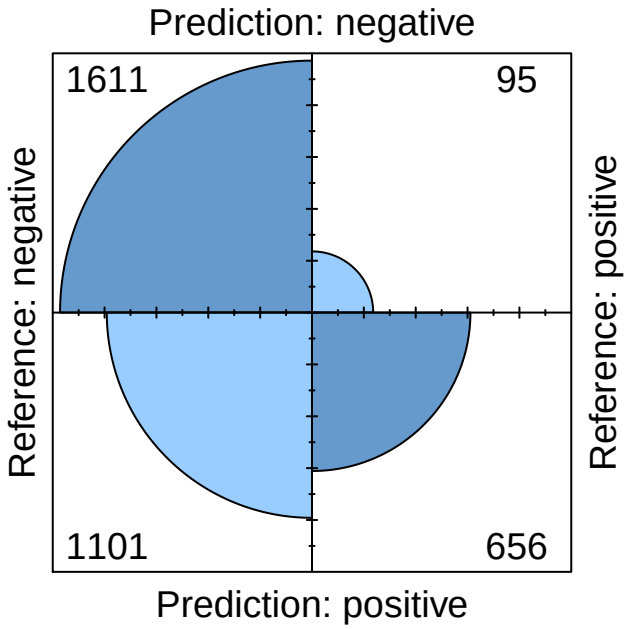



```
## Train score error:
## 0.909
## Test score error:
## 0.888
```

b) Naive Bayes Unigram(merged word) Model

Executed Naive Baise for Unigram with the merged word. Its accuracy was very good nearly 0.84. However, when I was sampling data e.g rather than taking full train dataset, if I will take only 5000 or 4000 samples in training set its accuracy was changing drastically from the range of 0.54 to 0.78. Therefore, did not include in the model comparison.






```
## Train score error:
## 0.179
## Test score error:
## 0.2
```