Sentiment Analysis on Airline Tweets Dataset

Problem Statement / Opportunity

- The airline industry is a very competitive market
- Traditional customer feedback forms are tedious and time consuming
- Tweeter data serves as a good source to gather customer feedback
- Tweeter is a gold mine of data with nearly 100 million subscribers
- More than half a billion tweets are tweeted, and keeps growing

Through machine learning and text analytics, sentiment analysis is used to determine sentiments such as positive, neutral or negative.

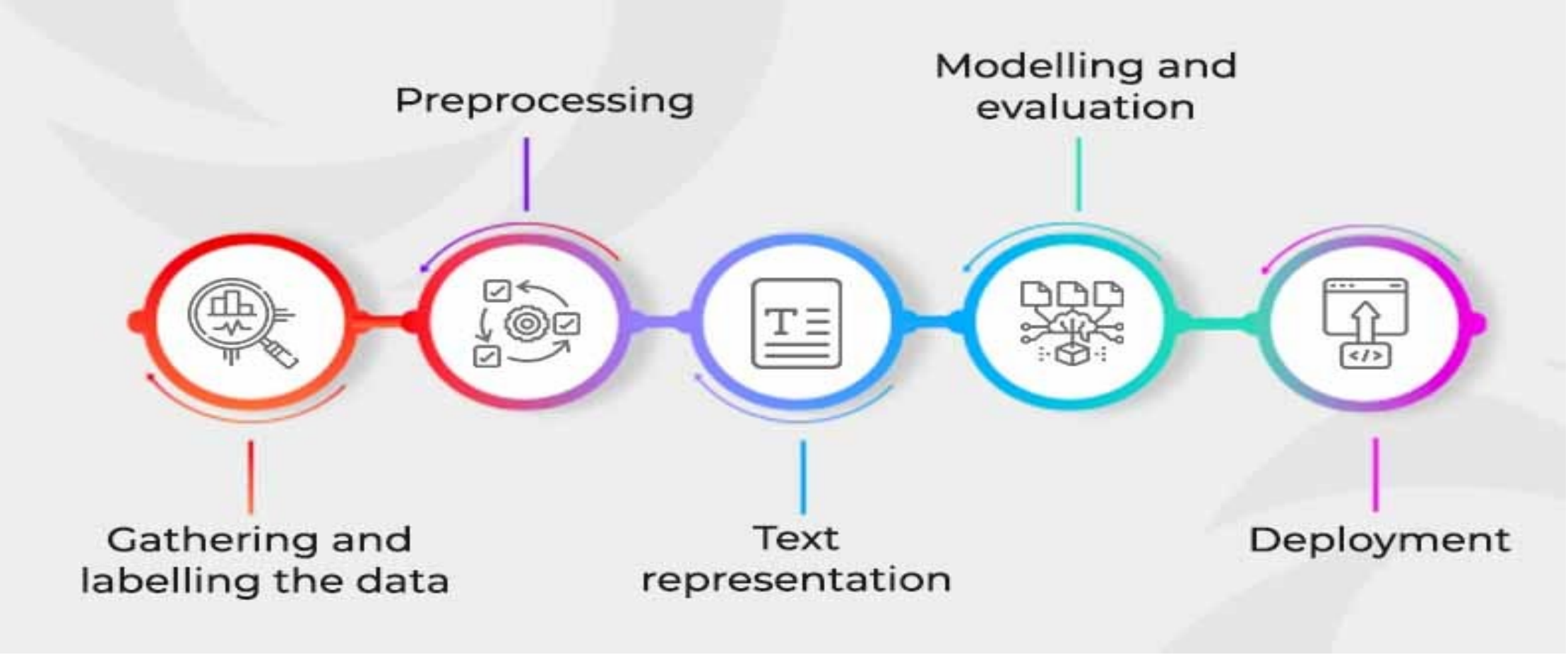
Benefit of Sentiment Analysis

- Sorting Data at Scale
- Real-Time Analysis

Business Impact

- Real-time sentiment analysis helps monitor social media mentions and proactively manage negative comments.
- It also offers insights into customer reactions to marketing campaigns and product launches.
- Periodic sentiment analysis helps understand customer preferences and concerns related to specific business aspects

Sentiment Analysis using Data Science



Data Pre-processing Steps

- Tokenization (e.g. convert to words or n-grams).
- Removing Stopwords (remove uninformative words).
- Lexicon Normalization: Either Stemming or Lemmatization (reduce vocabulary to essense)
- POS Tagging (parameterize information beyond words -- noun, verb, etc.).

Airline Tweets Dataset

- we worked on a dataset comprising of tweets for 6 major US Airlines and performed a multi-class sentiment analysis.

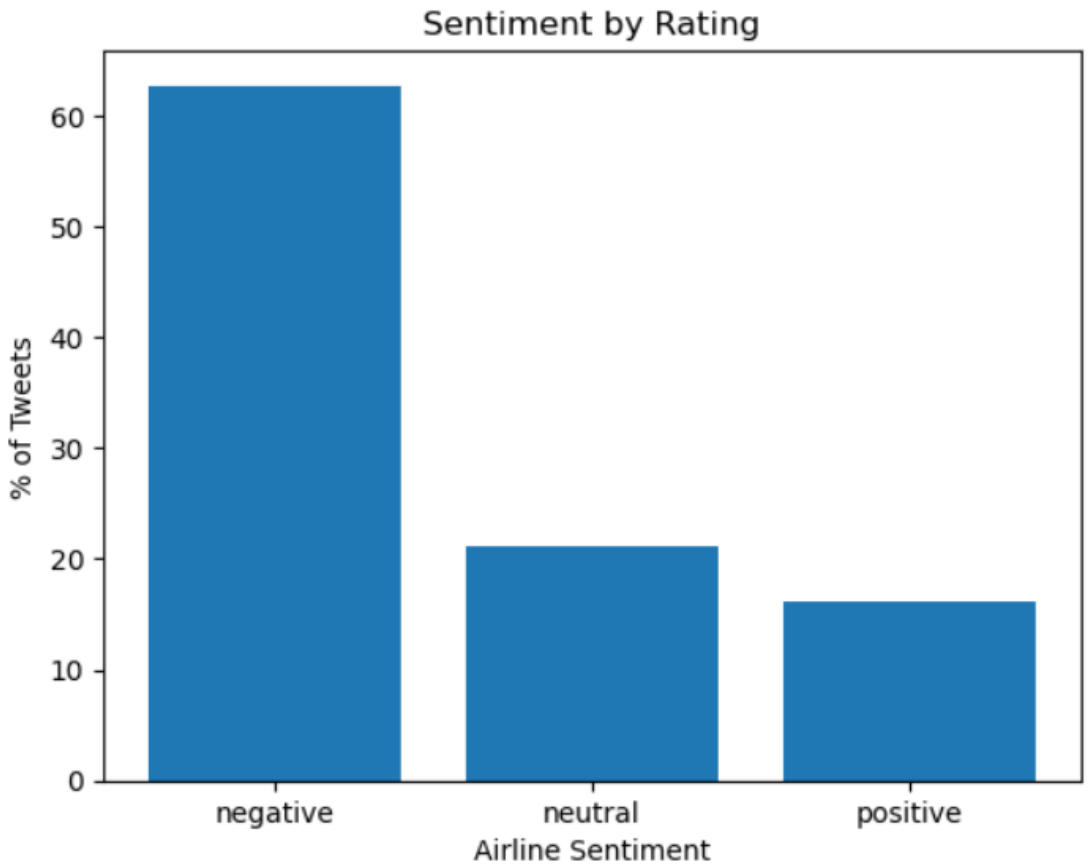
```
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   tweet_id                     14640 non-null  int64
 1   airline_sentiment            14640 non-null  object
 2   airline_sentiment_confidence 14640 non-null  float64
 3   negativereason               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
```

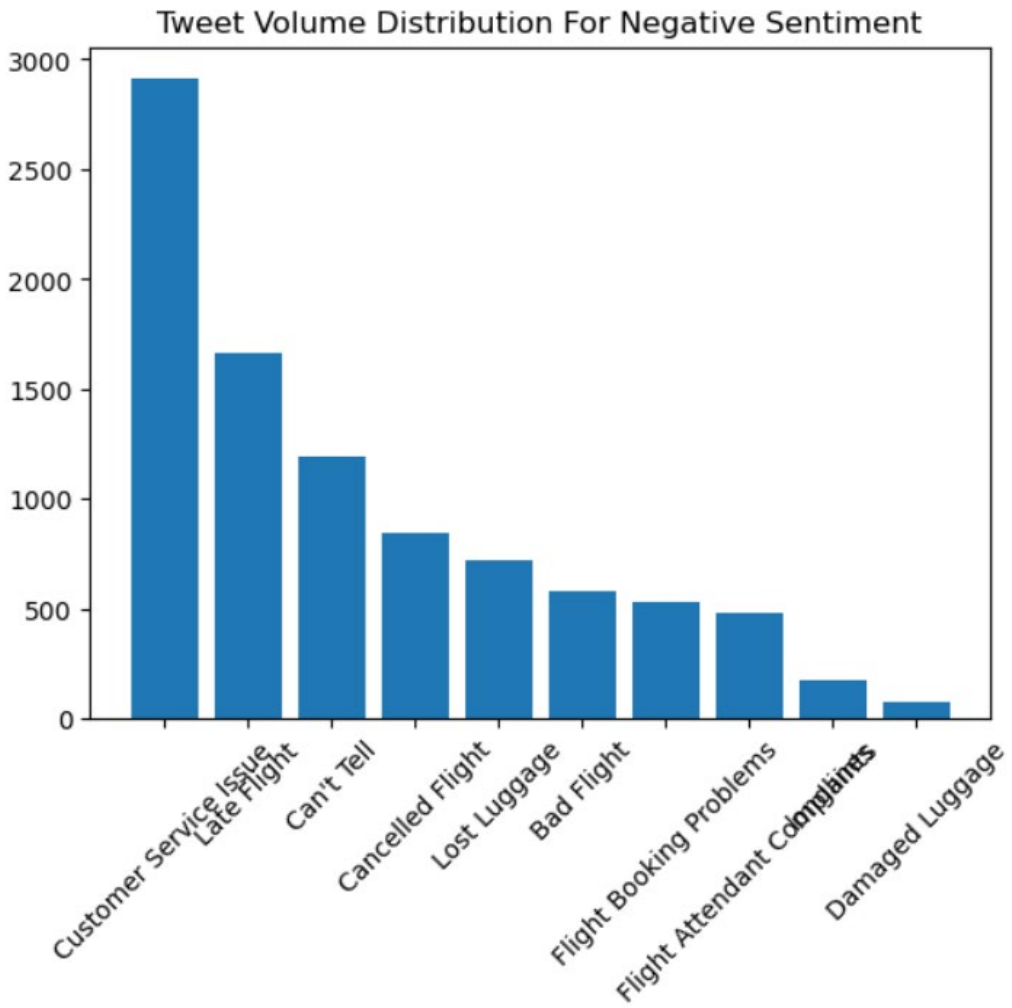
- ✓ This dataset contains 14,640 observations and 15 features.
- ✓ We will focus on “Text” and “airline_sentiment”
- ✓ Some feathers “Negative Reason” and “Airline” are also useful for model building
- ✓ Other feature “tweet_id” ,”tweet_creation” are less important.

Missing values detection

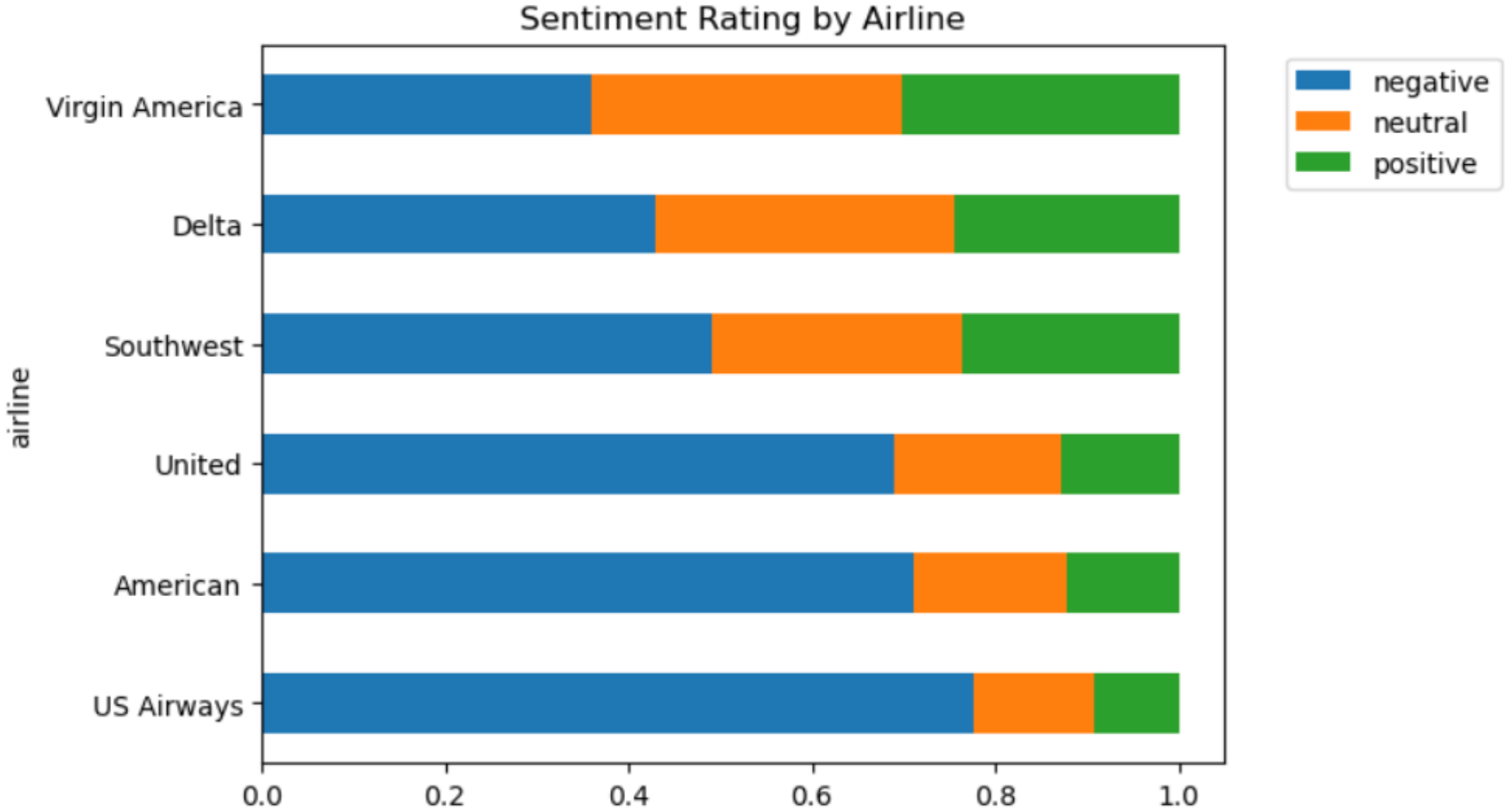
```
tweet_id                      0.000000
airline_sentiment             0.000000
airline_sentiment_confidence  0.000000
negativereason                0.373087
negativereason_confidence     0.281284
airline                       0.000000
airline_sentiment_gold        0.997268
name                          0.000000
negativereason_gold           0.997814
retweet_count                 0.000000
text                          0.000000
tweet_coord                   0.930396
tweet_created                 0.000000
tweet_location                0.323292
user_timezone                 0.329235
dtype: float64
```

Exploratory Data Analysis

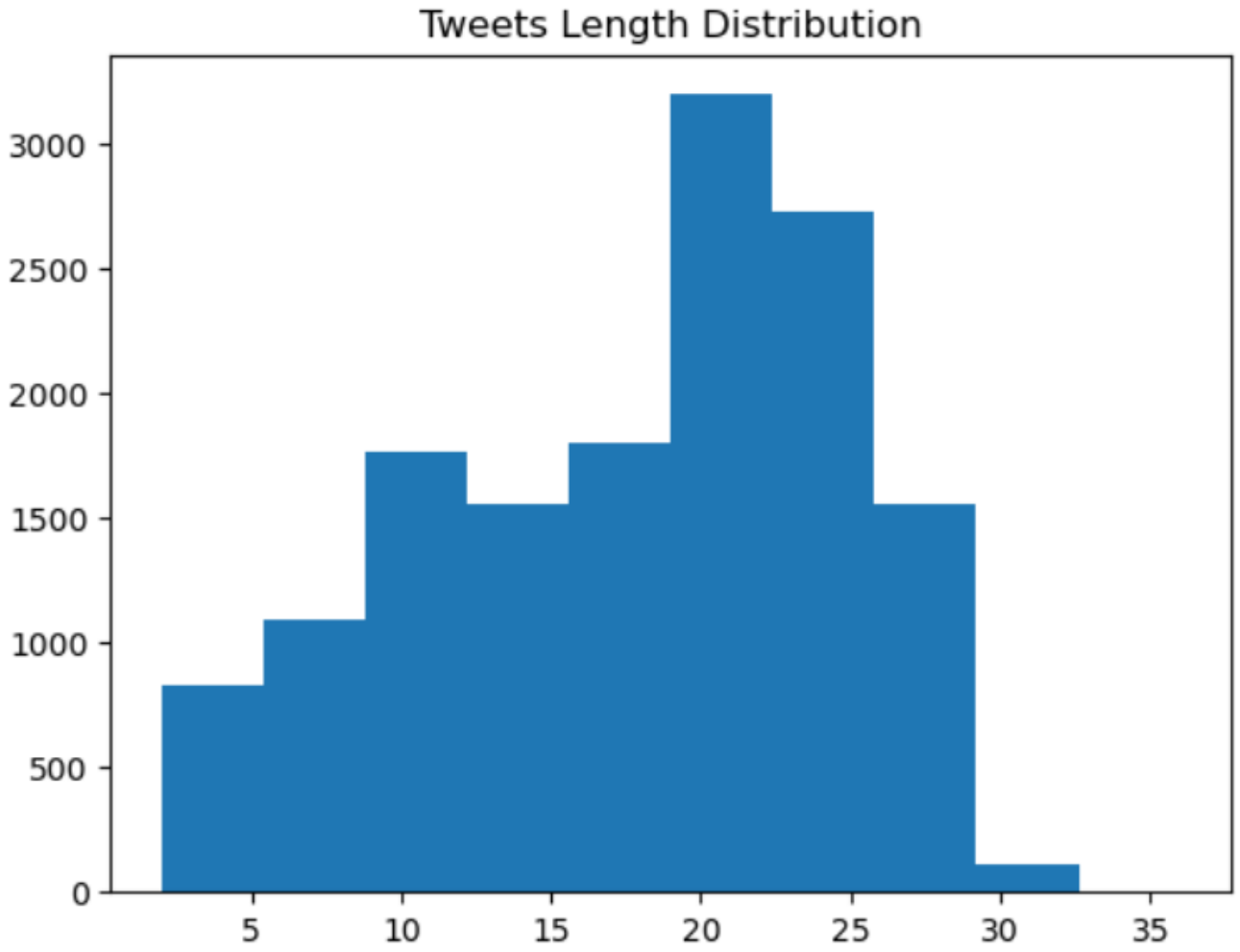




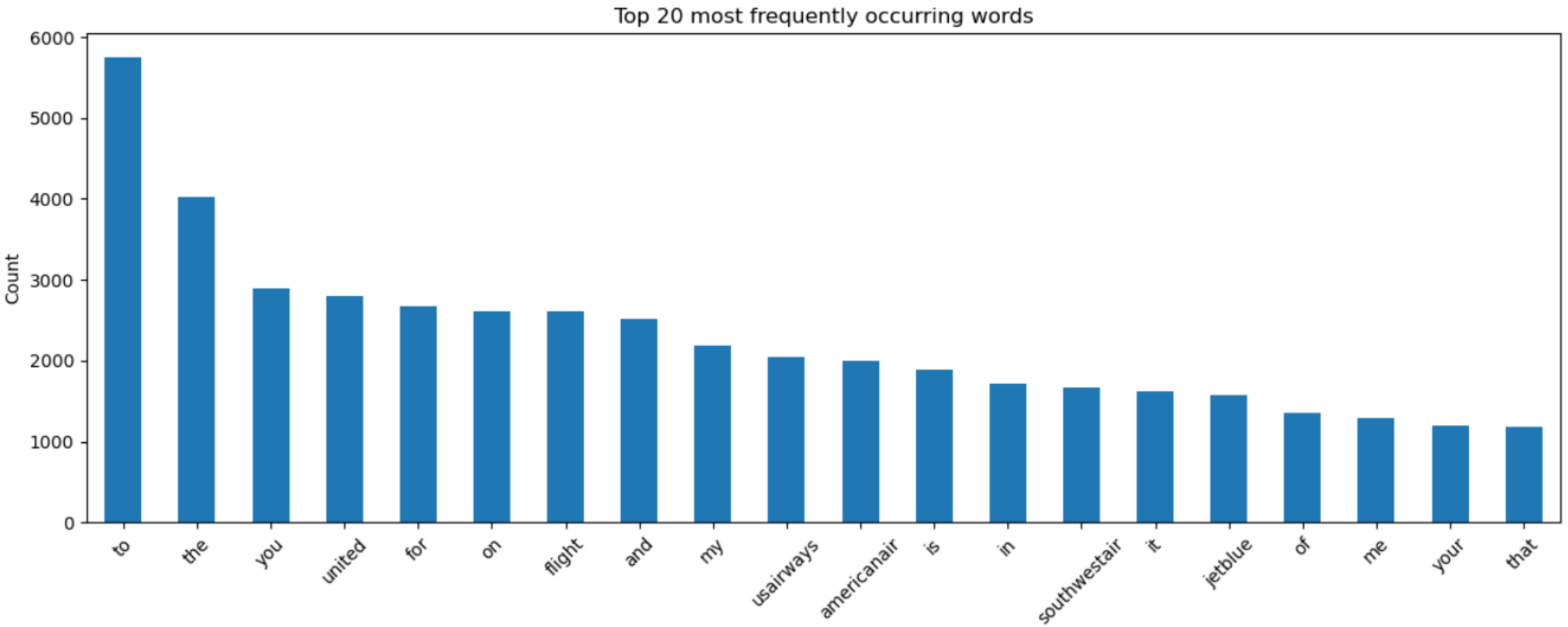
Exploratory Data Analysis



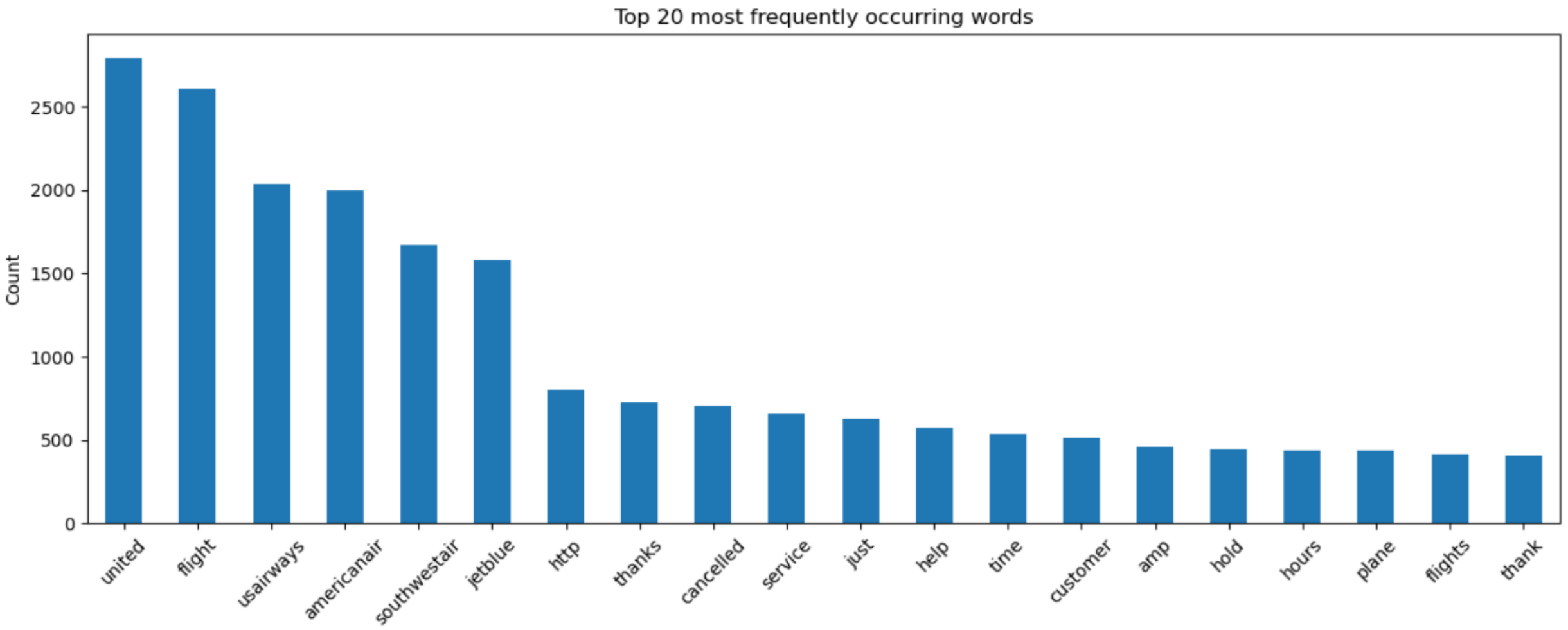
Exploratory Data Analysis



Bag of Words – Before stopwords removal



Bag of Words – After Stopwords removal



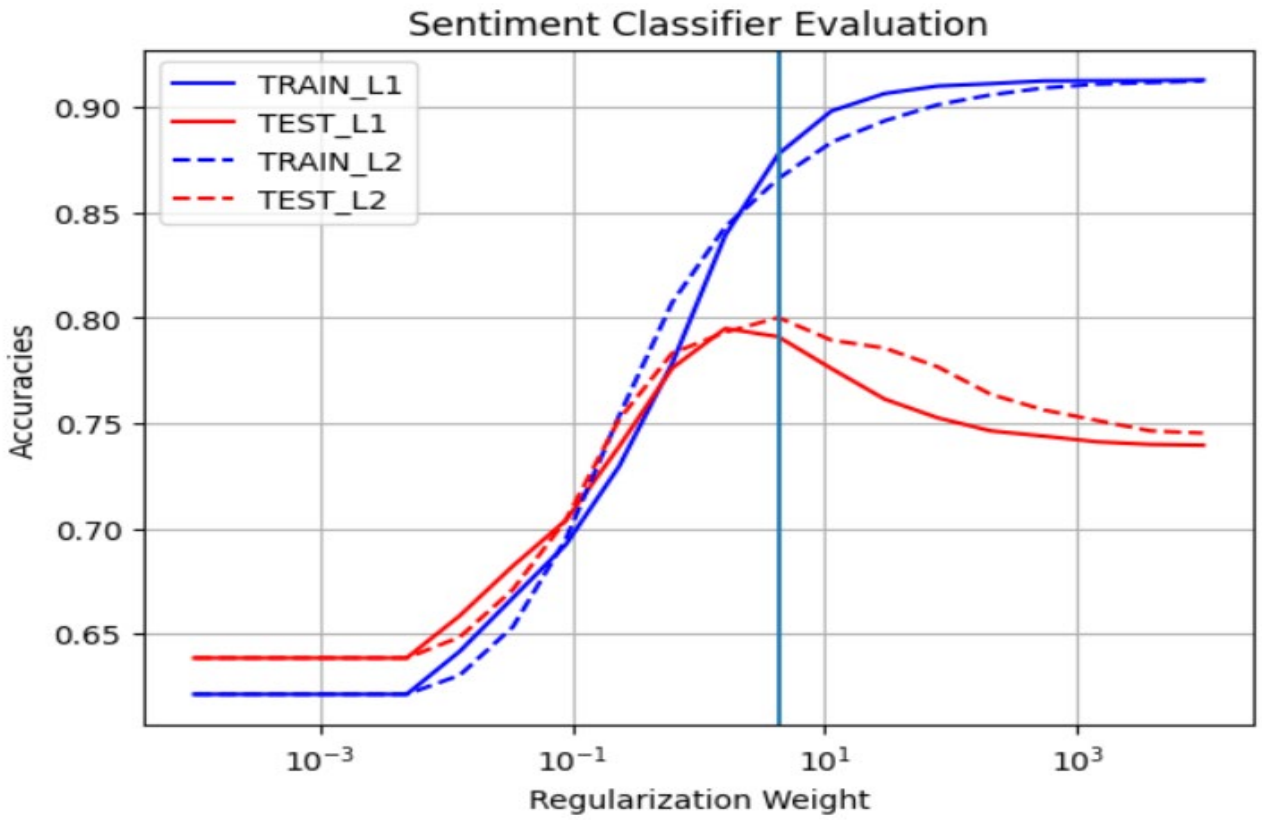
Model building -- Bag of Words

```
vect = CountVectorizer(stop_words='english',
                       max_features=1000,
                       min_df = 5,
                       max_df = 0.5)
```

Algorithms	Best Parameters	Test Score	
Logistic Regression	C=0.46	77.44% (Ngram=1)	76.84% (Ngram=(1,2))
Decision Tree	criterion='log_loss', max_depth=17, min_samples_leaf=7	70.71%	
Random Forest	Max_depth=60; n_estimators=100	74.89%	
Adaboost Classifier	N-estimators from 1-100	73%	
Gradient Boosting	Default setting	73%	

Model building -- TF-IDF

- Customized tokenizer to include stemming and set min_df=5;
- Total tokens: 1970
- Logistic Regression
- Penalty: l2 C: 4.281
- Accuracy: 80%



Model building -- TF-IDF

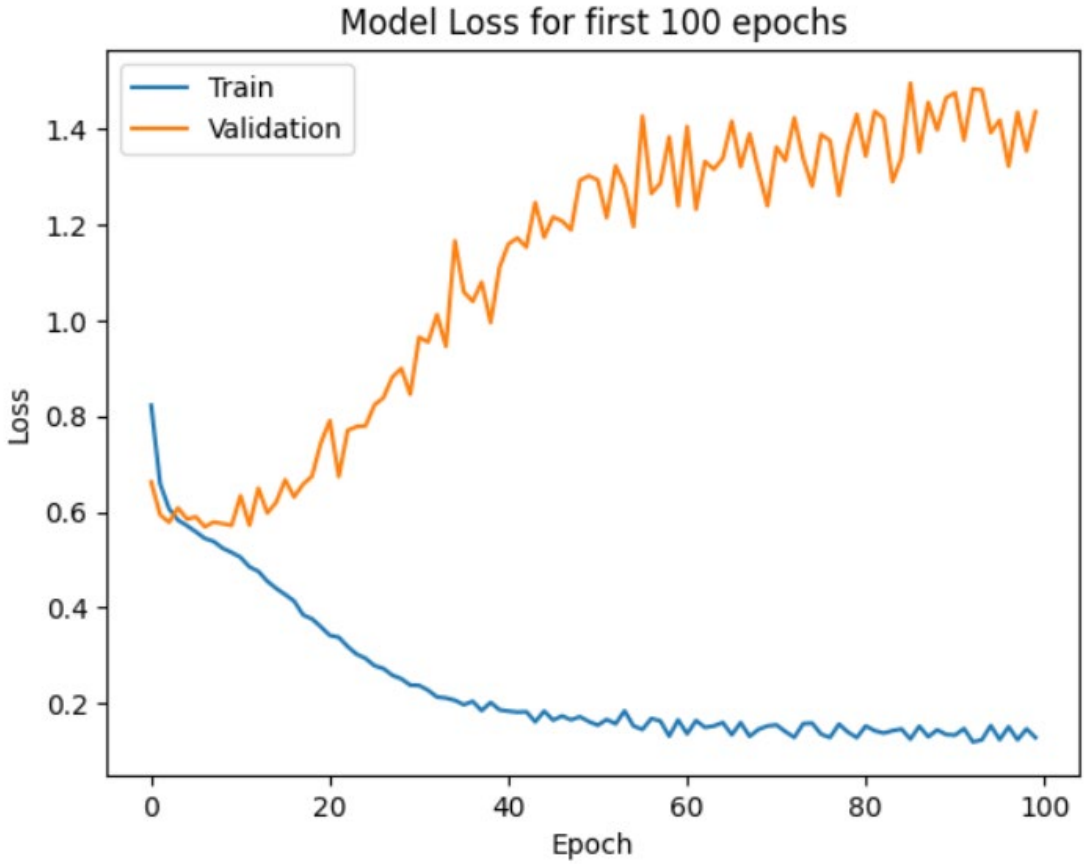
Algorithms	Best Parameters	Test Score	
Decision Tree		71%	
Adaboost Classifier	N_estimators =20	71%	
Gradient Boosting	Default setting	75%	

Model building -- Word Embedding

- Pretrained Word2Vec: LexVec
- The vector dimension reduced to 300.

Algorithms	Best Parameters	Test Score
Logistic Regression	Penalty: l2; C: 11.28	77.8%
Random Forest	Max_depth: 37	72%
Neural Network	5 Hidden layers Total params: 65667	88%

Model building -- Neural Network



	precision	recall	f1-score	support
0	0.94	0.91	0.92	6093
1	0.84	0.76	0.80	2115
2	0.74	0.91	0.82	1600
accuracy			0.88	9808
macro avg	0.84	0.86	0.85	9808
weighted avg	0.89	0.88	0.88	9808

Next Steps

- RNN (SimpleRNN, LSTM, RGU)
- Build models with attention,BERT
- Multiclass ROC curve and AUC
- Optimize the tokenizers
- Topic modeling investigation

Questions?