U.S. Airline Sentiment Analysis

Using Tweets

**Milestone: Project Report Due: 8/14/2023**

# Group 4:

Arthik Alexander — 857-544-9304 — [alexander.ar@northeastern.edu](mailto:alexander.ar@northeastern.edu) Reha Patel — 863-397-3985 — [patel.reh@northeastern.edu](mailto:patel.reh@northeastern.edu)

Siddhesh Koparde — 617-308-3420 —[koparde.s@northeastern.edu](mailto:koparde.s@northeastern.edu) Vinit Nagap — 857-381-2299 — [nagap.v@northeastern.edu](mailto:nagap.v@northeastern.edu)

***Percentage of Efforts:***

*Arthik — 25%*

*Reha — 25%*

*Siddhesh — 25%*

*Vinit — 25%*

# Problem Setting:

In this project proposal, we aim to analyze sentiment of tweets scraped from Twitter/X that mention select US airlines. Twitter has become a popular platform for users to express their opinions and experiences with various airlines. With the increasing dominance of social media platforms, airlines are now facing unprecedented challenges in managing their reputation, as customers actively share their experiences, opinions, and grievances on platforms like Twitter. The vast amount of unstructured data in the form of tweets can provide valuable insights into the sentiment and perception of airline services, enabling airlines to make data-driven decisions to enhance customer experience and address potential issues promptly.

*Domain:*

The domain of U.S. airline sentiment analysis involves analyzing and classifying tweets mentioning different U.S. airlines into sentiment categories such as positive, negative, or neutral. The sentiment analysis process allows airlines to understand public opinion, gauge customer satisfaction, and identify areas that need improvement in their services. By extracting meaningful insights from the tweets, airlines can proactively respond to negative feedback, resolve customer complaints, and reinforce positive experiences, thereby enhancing brand loyalty and customer retention.

*Application*:

The aim of this project is to identify robust and accurate sentiment analysis models capable of categorizing tweets about U.S. airlines into positive, negative, or neutral sentiments. The models will be trained on a labeled dataset containing historical tweets that express opinions and experiences related to various aspects of airline services, such as flight punctuality, customer service, in-flight amenities, baggage handling, and more. The sentiment analysis models will ideally be used to process real-time data, continuously monitoring and classifying incoming tweets, providing airlines with valuable insights to manage their brand reputation effectively. However, it must be noted that consumers are more likely to share negative feedback than positive, so we will also be particularly focusing on how well our models perform against negative tweets.

*Challenges*:

Several challenges were addressed while undertaking this project:

1. *Data Volume and Noise*: The tweets column contains noise in the form of slang, typos, abbreviations and emojis.
2. *Subjectivity and Context*: Determining sentiment from tweets can be challenging due to inherent subjectivity and ambiguity of language.
3. *Handling Negations and Emphasis*: Some tweets contain negations or emphasize certain words to convey a specific sentiment.
4. *Real-Time Processing:* The sentiment analysis model must be efficient enough to process incoming tweets in real-time, as timely responses are crucial in managing brand reputation.

# Problem Definition:

The specific problem addressed within the scope of this project is to identify a sentiment analysis model capable of accurately categorizing tweets mentioning U.S. airlines into one of the three sentiment classes: positive, negative, or neutral. The primary objective is to gain insights into public sentiment and opinions regarding different aspects of airline services, enabling airlines to enhance customer satisfaction, manage their brand reputation, and make data-driven improvements to their services.

*List of questions addressed:*

1. *What is the overall sentiment distribution?*

* What percentage of tweets are positive, negative, and neutral?
* Is there any imbalance in the sentiment classes, and does it affect analysis?

1. *What is the overall accuracy of the sentiment models?*

* What is the model’s accuracy and F-1 score?
* How well does the model generalize the unseen data?

1. *How well does the model handle noise and contextual language?*

* How does the model deal with slang, typos, abbreviations, and emojis?
* Does the model consider the context in which the tweets are written?

1. *Which sentiment model to be used for real-time monitoring*?

* Is the model efficient enough for processing incoming tweets in real time situations?
* Can the model provide timely insights for airlines to respond to customer feedback?

1. *How can airlines adopt sentiment analysis?*

* How can airlines proactively respond to negative sentiment to address customer concerns?
* Can positive sentiment be leveraged for reinforcing positive customer experiences and strengthening brand loyalty?

# Data Sources:

The dataset that we have chosen for our project is a twitter dataset from [Kaggle](https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment?resource=download).

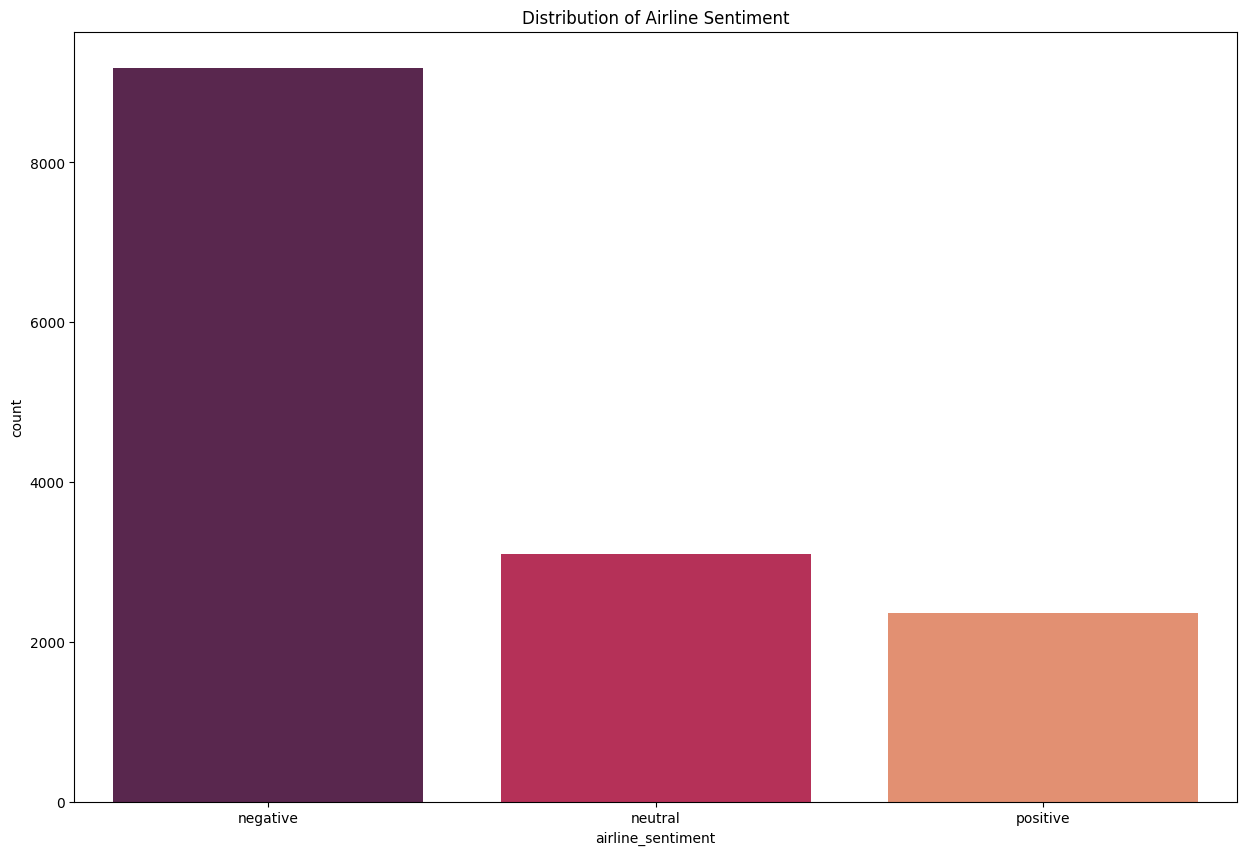
# Data Description:

The collected data includes tweets, user information (names), timestamps, location, airline, tweet ID, user timezone, tweet coordinates, retweet counts and other metadata. In total, the dataset consists of 14640 rows and 16 columns. The twitter data was scraped from February 16, 2015 and February 24, 2015. Here are the columns in the dataset alongside a short description:

1. tweet\_id - This column contains a unique identifier (ID) for each tweet. The tweet ID helps differentiate one tweet from another and is crucial for referencing specific tweets.
2. airline\_sentiment - This column indicates the sentiment expressed in the tweet. Sentiment refers to the emotional tone of the text. The sentiment can be categorized as "positive," "negative," or "neutral".
3. airline\_sentiment\_confidence - This column provides a confidence score or level for the sentiment classification. It indicates how certain the sentiment analysis model is about its assigned sentiment label. Higher values suggest higher confidence.
4. negativereason - If the sentiment is negative, this column might contain the reason behind the negative sentiment. For instance, if a customer expressed dissatisfaction with a "Bad Flight," this column would capture that reason. It helps identify specific issues customers are unhappy about.
5. negativereason\_confidence - Similar to airline\_sentiment\_confidence, this column provides a confidence level for the negative reason classification. It indicates the model's confidence in correctly identifying the specific negative reason.
6. airline - This column holds the name of the airline in question. It helps categorize the dataset based on the airline being discussed.
7. airline\_sentiment\_gold and negativereason\_gold - These columns might contain additional sentiment and negative reason information in a gold standard format. The "gold" standard might refer to a more accurate or validated annotation for reference.
8. name - This column contains the Twitter username of the person who posted the tweet. It provides information about the author of the tweet.
9. retweet\_count - This column represents the number of times the tweet has been retweeted by other Twitter users. A higher retweet count suggests that the tweet gained more attention and possibly went viral.
10. text - This is the main content of the tweet. It consists of the actual text written by the Twitter user, expressing their opinion, feedback, or experience related to a particular airline.

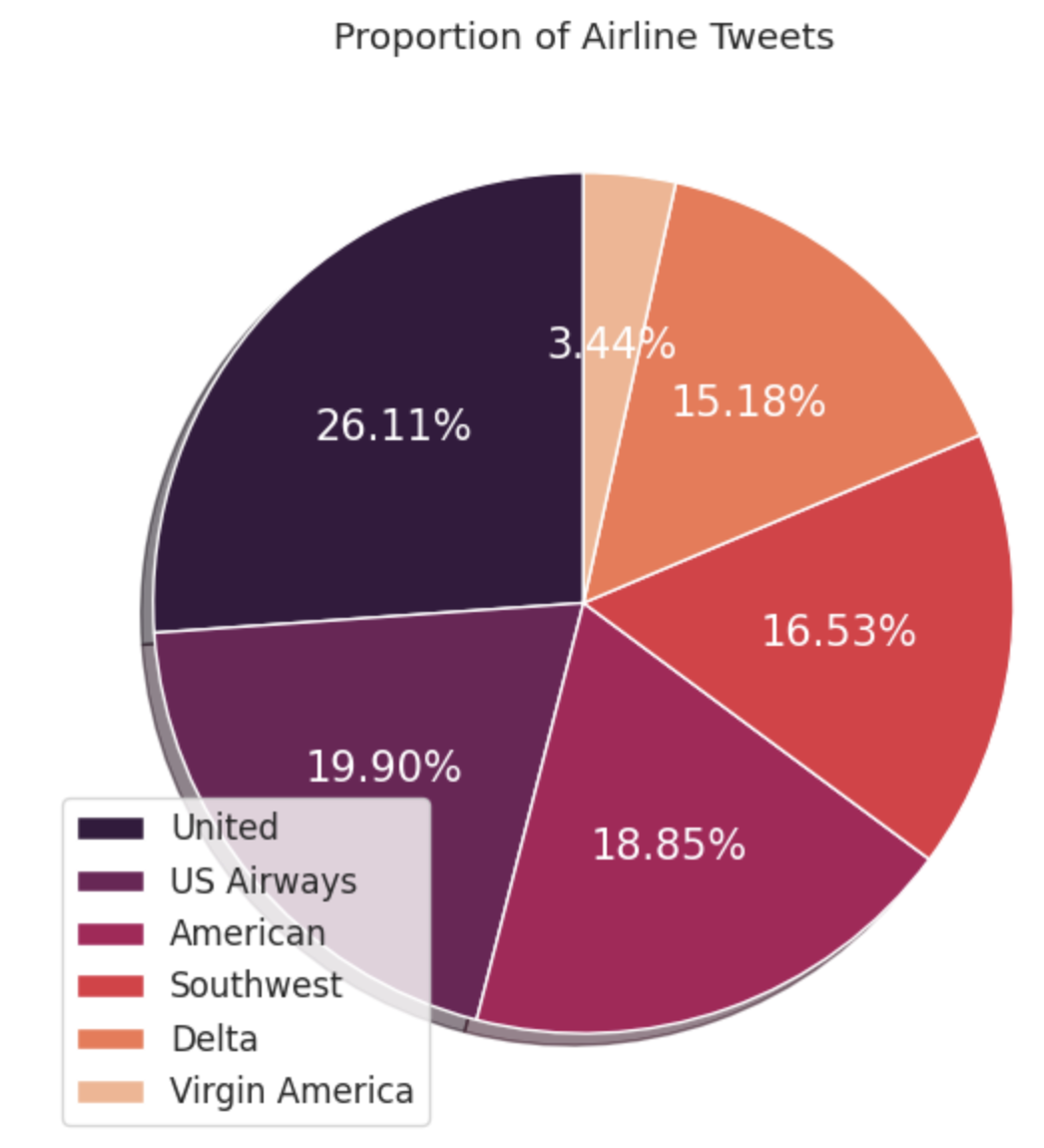
# **Data Exploration:**

Data exploration is a crucial preliminary step in the data analysis and modeling building process. In our project, we have examined and understood the characteristics of our dataset to uncover patterns, relationships, and potential insights. As our dataset consists mainly of textual data (tweets) and figuring out the sentiment for those tweets, we made a bar plot of all the tweets with the respective sentiments.



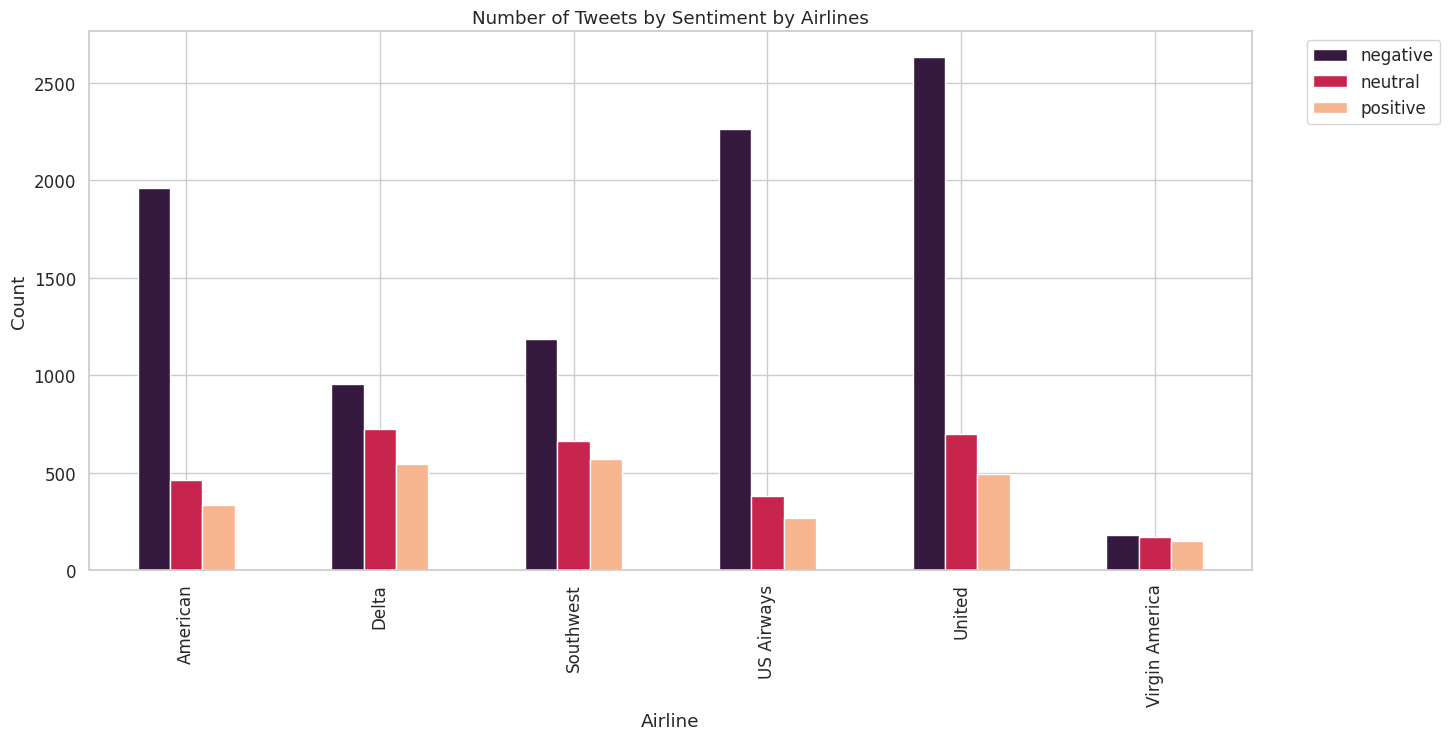
As it can be seen from the plot, the proportion of negative tweets is way more than positive and neutral tweets. This means that the sentiment data present in the dataset is either incorrect or the airlines are providing a bad service.

To understand our data more, we plotted a pie chart to gain an insight to the number of tweets associated with each airline.



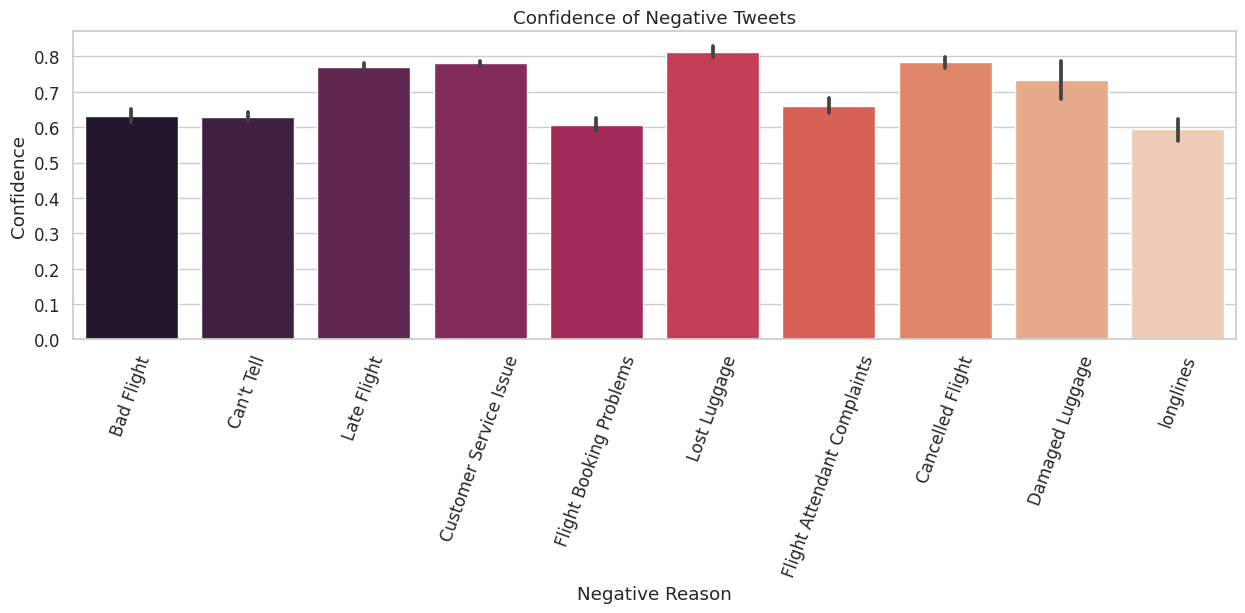
As it can be seen from the pie chart, the majority of the tweets are for United Airlines. This could mean that a majority chunk of negative tweets are for United Airlines.

To gauge how the pre-defined sentiment for all airlines is distributed, we plotted a bar plot for the same.



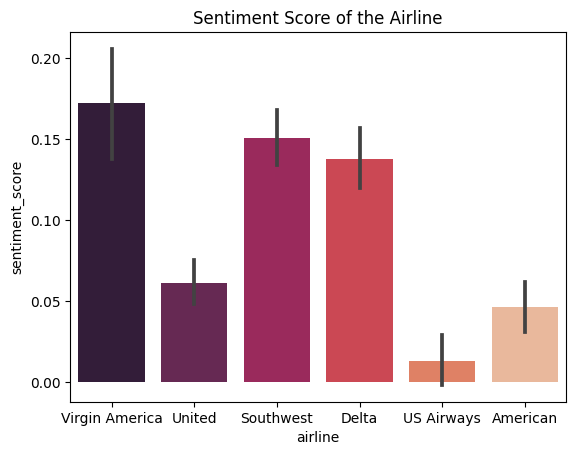
Our initial guess was correct, as most of the negative tweets are for United Airlines, with US Airways and American Airlines following.

We further plotted a graph to gain insights into the negative reasons that have higher confidence levels in the sentiment analysis and how frequently each negative reason appears in the dataset.



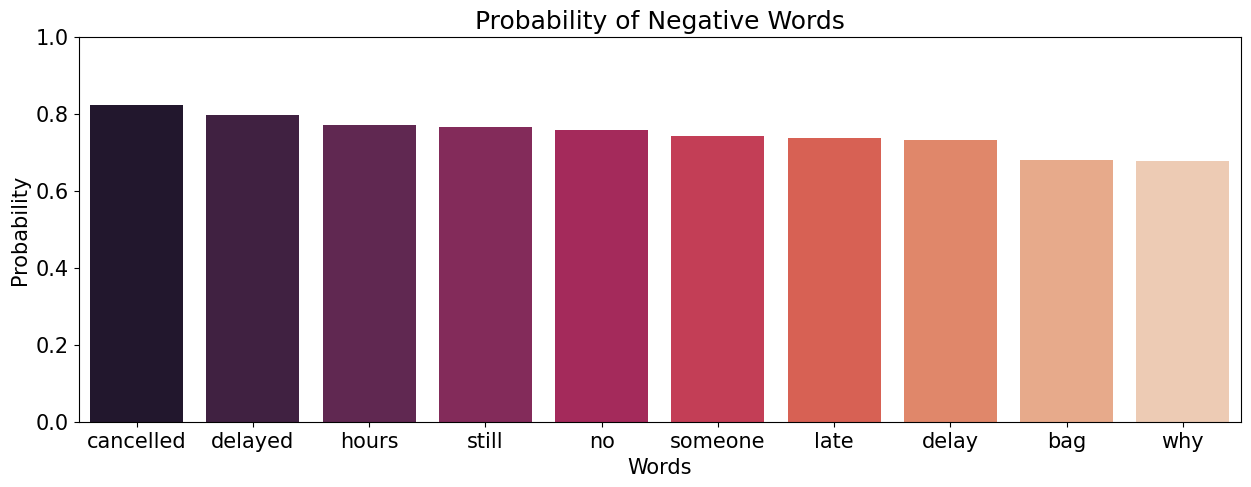
Late flights, Customer Service Issues, Lost Luggage, Canceled Flights, Damaged Luggage were the main reasons that the customers were facing with the airline.

To dig deeper into how the airlines are performing, we found the sentiment scores and visualized them.



Virgin America seems to have the highest sentiment score, which means compared to all the other airlines, Virgin America is performing well, however, previously we had noted that Virgin America also had the least number of tweets mentioning them, at 3% of the entire dataset.

To understand which words cause a negative sentiment, we found and plotted the probability of the negative words that occur frequently in tweets for all airlines. The following graph is an example of Delta Airlines.



Similarly, the plots for all other airlines can be found in the *Google Colab File.*

# Data Mining Tasks:

The dataset consists of a total of 16 columns out of which only 11 columns are used. In the context of our dataset, various data mining tasks were undertaken to extract meaningful insights and patterns from the raw data. These tasks encompassed data reduction, data transformation, missing data imputation, and manually inputting the y variable. Each of these tasks played a distinct role in enhancing the understanding of the dataset and enabling informed decision-making.

In the US Airlines Twitter dataset project, data reduction methods were applied to handle the large volume of tweets and their associated features. In our initial data reduction phases, we chose to remove any columns where over 90% of the rows were null. We chose to do this because at the percentage of 90%, it would be ineffective to do any sort of imputing on the rows. These rows were “airline\_sentiment\_gold”, “negativereason\_gold”, and “retweet\_count.”

A key task which all members of the team performed was that a new y value was manually created. The initial dataset provided us with a sentiment of the tweets alongside the confidence of the sentiment classification, but upon further examination even when the confidence was 1.0, the sentiment classification was wrong. This could have been due to factors such as the classifier not understanding sarcasm, lingo, etc. As a result, we randomly picked 4000+ tweets across all the airlines to manually classify by examination. We have named the new y variable as *‘Sid, Arthik, Reha, Vinit,’* although it was sometimes renamed to ‘y’ for our analysis.

Given that we were doing sentiment analysis, cleaning the text was another key task which we have performed. Some of the tweets contained additional text such as URLs, Twitter handles, etc. so we removed those as well as standardized capitalization to be all lower case, removed punctuation, and also removed any stop words.

# Data Mining Models/Methods:

Prior to beginning any modeling, we did a substantial amount of research on how other data scientists had performed sentiment analysis, mistakes they made, and a variety of steps we could follow. Specifically, the paper *Sentiment Analysis of Facebook Comments Using Various Machine Learning Techniques* by S. Veni et al. performed sentiment analysis using Naive Bayes, Support Vector Machine (SVM), Random Forests, k - Nearest Neighbours (KNN), and Decision Trees, so we decided to pick a few of those to use as baseline models that we could compare our trained models against. Specifically, we chose Naive Bayes, Decision Trees, Random Forests, KNN, and an addition that we were curious about, Logistic Regression. Alongside this, we chose to also incorporate pre-trained models for sentiment analysis, known as VADER and RoBERTa. All of these models were then used to compare against two models which we trained, tuned and tested from scratch, SVM and Neural Networks. We chose these to be the models which we created from scratch because SVM on one hand was also found in the paper we referenced and is well known for its effectiveness in text classification, whereas neural networks were presented in class as a powerful machine learning tool with which we could modify the number of classes we wanted to predict. As a result, we decided to incorporate Neural Networks because of its high potential effectiveness in text classification. Here are some further explanations on each of our models:

*Vader Model:*

The VADER (Valence Aware Dictionary and sentiment Reasoner) model is a popular sentiment analysis tool that is specifically designed for social media text. It's often used for analyzing sentiments in short, informal texts such as tweets. VADER doesn't just rely on a predefined dictionary of words; it also takes into account the context, intensity, and valence of words to determine the sentiment of a text.

In our project, the VADER model can be used as follows:

1. Data Preprocessing: Collect the tweets related to airlines, clean the data, remove any irrelevant information, and preprocess the text (e.g., lowercasing, removing special characters, etc.).
2. VADER Model: Use the VADER sentiment analysis tool to analyze the sentiment of each tweet. VADER assigns a polarity score to each tweet, indicating its positivity, neutrality, or negativity. The scores are calculated based on the valence of words and the intensity of their usage.
3. Thresholds and Labels: Based on the polarity scores provided by VADER, we have set threshold values to classify the tweets into positive, negative, or neutral categories. For example, if the compound score is greater than 0.05, we have classified the tweet as positive, if it's less than -0.05, we have classified it as negative, and anything in between is considered neutral.
4. Visualization and Analysis: Once we have classified the tweets into sentiment categories, we have visualizations by using pie-chart to show the distribution of sentiments in the airline tweets. This helps us understand the overall sentiment of the users towards the airlines.
5. Accuracy and Evaluation: We have calculated the accuracy of the VADER model's predictions by comparing its classifications with manually annotated sentiment labels. We have evaluated the model's performance using metrics like precision, recall, F1-score, etc.

*Roberta Model:*

The RoBERTa (A Robustly Optimized BERT Pretraining Approach) model is a variant of BERT (Bidirectional Encoder Representations from Transformers), which is a highly advanced deep learning model for natural language processing tasks, including sentiment analysis. BERT belongs to a class of models known as transformers, which are neural networks that process data in parallel using attention mechanisms. Unlike traditional models that read text input sequentially (left to right or right to left), transformers can consider the entire context of a word by attending to all other words in a sentence simultaneously. This bidirectional approach is a key innovation in BERT. RoBERTa further optimizes the training approach of BERT, resulting in improved performance on various NLP tasks.

1. Data Preprocessing: Collect airline-related tweets, preprocess the data (cleaning, tokenization, etc.), and prepare it for input into the RoBERTa model.
2. RoBERTa Model: Utilize a pre-trained RoBERTa model for sentiment analysis. There are pre-trained RoBERTa models available that have been fine-tuned on sentiment analysis tasks. We use the Hugging Face (cardiffnlp/twitter-roberta-base-sentiment) Transformers library in Python to load and use these pre-trained models.
3. Text Encoding and Prediction: We encoded the preprocessed tweet text using RoBERTa's tokenizer and then passed it through the RoBERTa model for sentiment prediction. The model will output probabilities or scores for each sentiment class.
4. Thresholds and Labels: Similar to the VADER model approach, we have set thresholds on the predicted scores to classify the tweets into different sentiment categories.
5. Visualization and Analysis: We have used visualizations to represent the distribution of sentiments in the airline tweets using pie charts, just like with the VADER model.
6. Accuracy and Evaluation: We have evaluated model's performance using metrics like accuracy, precision, recall, F1-score, etc. We have compared the model's predictions against manually labeled sentiment categories.

*Comparison of Logistic Regression, Multinomial Naive Bayes, Decision Trees, k- Nearest Neighbours and Random Forest using TF-IDF:*

In this approach, we have utilized the term frequency - inverse document frequency (TF-IDF) that uses frequency of words to determine relevancy. The dataset is split into training and tests dataframe using the split function allocating 70% to training, 30% to testing and random state for reproducibility. Two empty lists, train\_tweets and test\_tweets are established to hold the cleaned tweets from training and testing datasets correspondingly. Through a loop, each text data ‘tweet’ in the train and test dataframe undergoes processing and is appended to the appropriate list. The text data (tweets) is then transformed into a TF-IDF matrix through 'TfidfVectorizer' from 'sklearn.feature', with 'fit\_transform' applied to train split for training and 'transform' applied to test split using the same vectorizer. Additionally, the TF-IDF matrix from training is converted into 'train\_tfidf' with columns representing feature names obtained from the vectorizer. Finally, a list of classifier instances including 'LogisticRegression', 'MultinomialNB', 'DecisionTreeClassifier', 'RandomForestClassifier', and 'KNeighborsClassifier' is generated, accompanied by an initialized list 'cls\_name' to store their names.

*Support Vector Machine:*

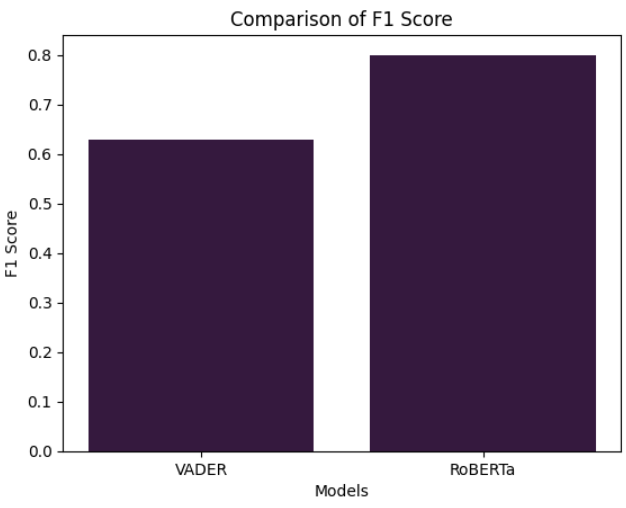
We chose to implement Support Vector Machines, SVM, because of its abilities to classify vectors of the features presented to us in the tweets in our dataset. We chose to implement Model 1 using CountVectorizer() with no additional parameters, Model 2 using TfidfVectorizer() with parameters we found in our research, and then a final model, Model 3, using CountVectorizer() and parameters that we hypertuned. Notably, in Model 3 we also implemented GridSearchCV() so that we could pass a parameter grid to determine the best combination of parameters for the model. The result of this after running for about 15 minutes total was that the metrics were optimized when the regularization parameter C was 10, the kernel coefficient Gamma was 0.1 and the kernel was rbf (Radial Basis Function) with the class weights being balanced. It was interesting to us that the grid search had suggested balanced class weights even though the dataset presented an imbalance between positive, negative, and neutral tweets.

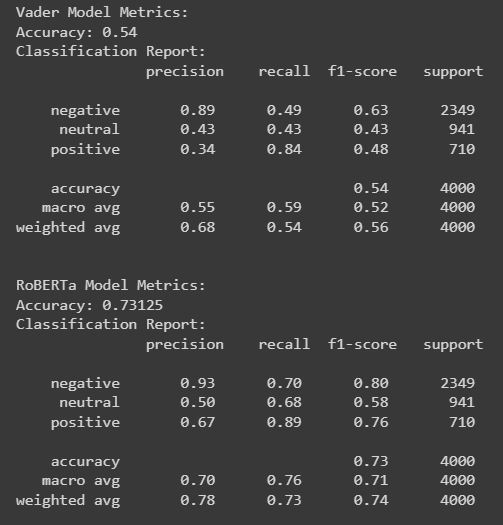
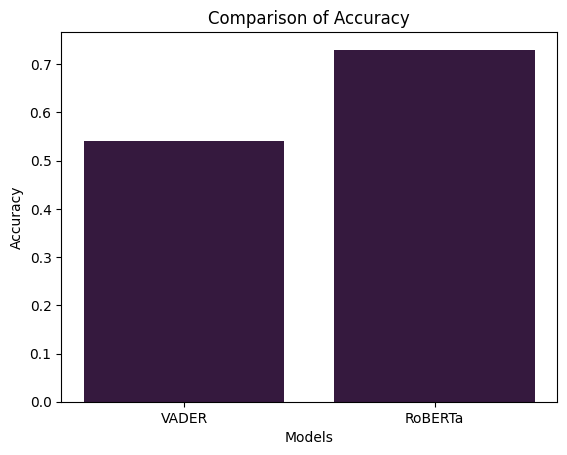
*Neural Network :*

For constructing the neural network architecture, we utilized the sequential model provided by Keras. The architecture comprised several layers, each serving a specific purpose. An embedding layer was employed to convert the tokenized integers into dense vectors. This was followed by an LSTM layer to capture the sequential dependencies in the data. To prevent overfitting, we included dropout layers after the LSTM and an intermediate Dense layer. The final Dense layer, with a softmax activation function, provided the sentiment classification probabilities. The model was compiled with the categorical cross-entropy loss function and the Adam optimizer. Accuracy was chosen as the evaluation metric. We split the preprocessed data into training and testing sets to validate the model's performance. The neural network was trained on the training data using a batch size of 32 for 5 epochs.

# Performance Evaluation:

For each of the models that we did analysis on, we chose to primarily focus on the accuracy as well as F1 score. This was because we wanted to ensure that the negative sentiment classification was accurate so the models could be used to identify negative tweets on a larger scale.

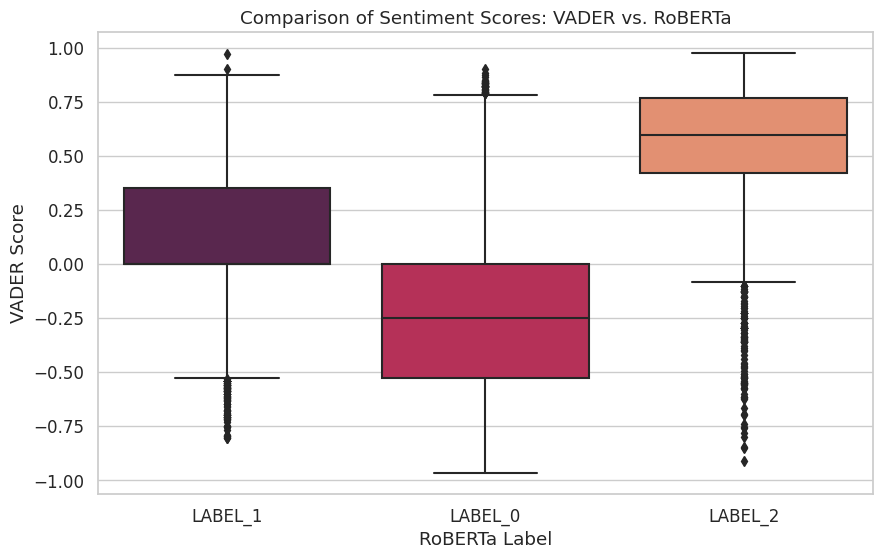
*Vader Model & RoBERTa Model*



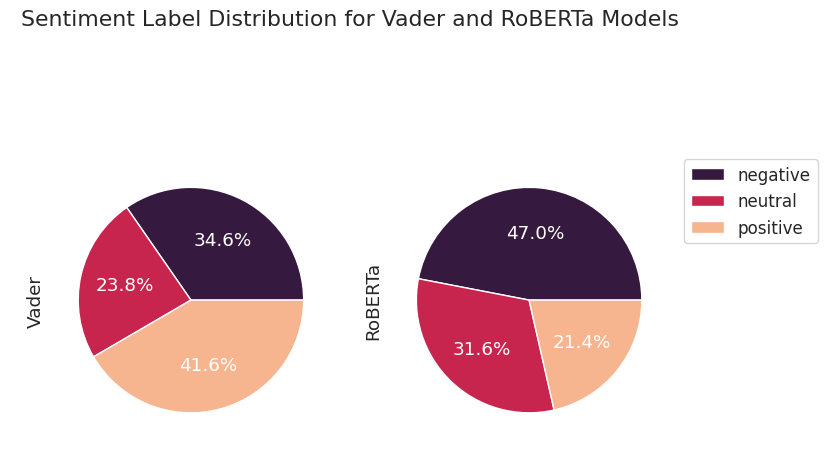
Comparing the Vader and RoBERTa model we got an accuracy of around 53% in Vader and about 73% in RoBERTa, whereas the F1 score of Vader is 0.63 and of RoBERTa is 0.8. As we can see upon first glance, RoBERTa performed better overall, compared to VADER. Upon doing research into this difference in accuracy and F1 score, we came across the fact that VADER attempts to balance the sentiment distribution whereas RoBERTa performed better on unbalanced datasets such as the one we have.

*Interpretation Between Vader & Roberta Models:*

In the below box plot Label\_1 is a neutral tweet, Label\_0 is a negative tweet, Label\_2 is a positive tweet. In this box plot we compared the RoBERTa label with Vader Score and found out that RoBERTa negative has a Vader Score from -0.5 to 0, RoBERTa neutral has a Vader Score from 0 to 0.35, and RoBERTa positive has a Vader Score from 0.5 to 0.75. There are also few outliers present in this comparison of two models.



The RoBERTa model outperforms the Vader model in terms of accuracy, macro average F1-score, and weighted average F1-score. RoBERTa's higher precision, recall, and F1-scores for all sentiment categories indicate that it provides better overall performance in classifying sentiments compared to the Vader model.



In Vader, we have higher positive tweets compared to negative and neutral, whereas in RoBERTa negative tweets are on the higher side compared to neutral and positive tweets.

*Logistic Regression, Multinomial Naive Bayes, Decision Trees, k- Nearest Neighbors and Random Forest using TF-IDF:*

A key analysis we wanted to perform in our comparison was creating baseline models of different classification algorithms we had come upon in our research of sentiment analysis. We have compared Logistic Regression, Multinomial Naive Bayes, Decision Trees, k- Nearest Neighbours and Random Forests:

1. *Logistic Regression:*

Accuracy: 77.98%

This model correctly predicted around 78% of the outcomes.

It performed well in identifying "negative" and "positive" cases, but not as well in "neutral" cases.

On average, its predictions were about 78% accurate.

1. *Multinomial Naive Bayes:*

Accuracy: 68.19%

This model correctly predicted around 68% of the outcomes.

It was good at identifying "positive" cases, but not very good at "neutral" cases.

On average, its predictions were about 68% accurate.

1. *Decision Tree Classifier:*

Accuracy: 70.2%

This model correctly predicted around 70% of the outcomes.

It performed similarly across all three classes: "negative," "neutral," and "positive."

On average, its predictions were about 70% accurate.

1. *Random Forest Classifier:*

Accuracy: 76.39%

This model correctly predicted around 76% of the outcomes.

It performed well in identifying "negative" and "positive" cases, but not as well in "neutral" cases.

On average, its predictions were about 76% accurate.

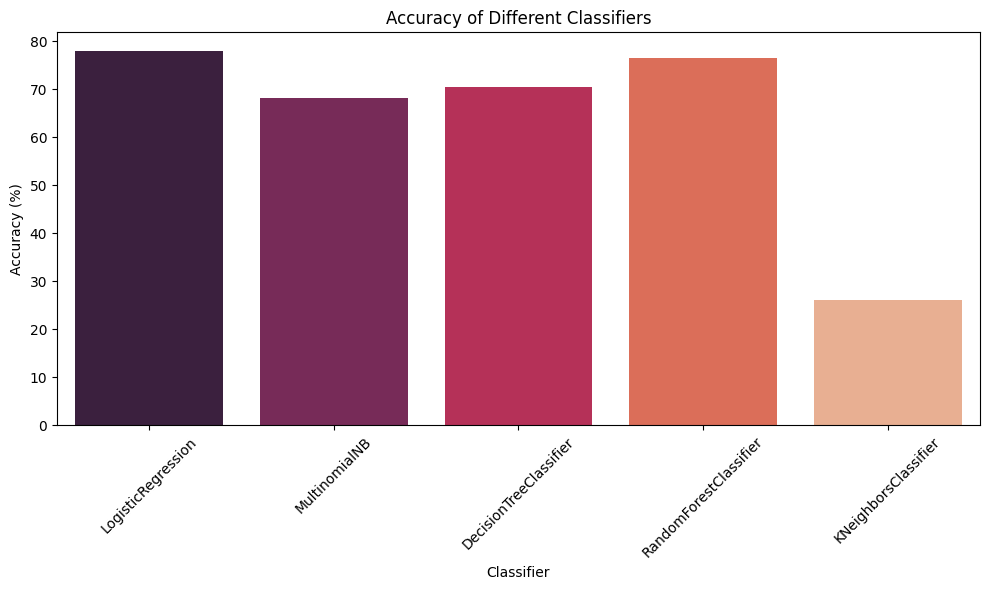
1. *k-Nearest Neighbors Classifier:*

Accuracy: 26.02%

This model had the lowest accuracy among the models, correctly predicting only around 26% of the outcomes.

It was particularly good at identifying "neutral" cases, but not very good at "negative" and "positive" cases.

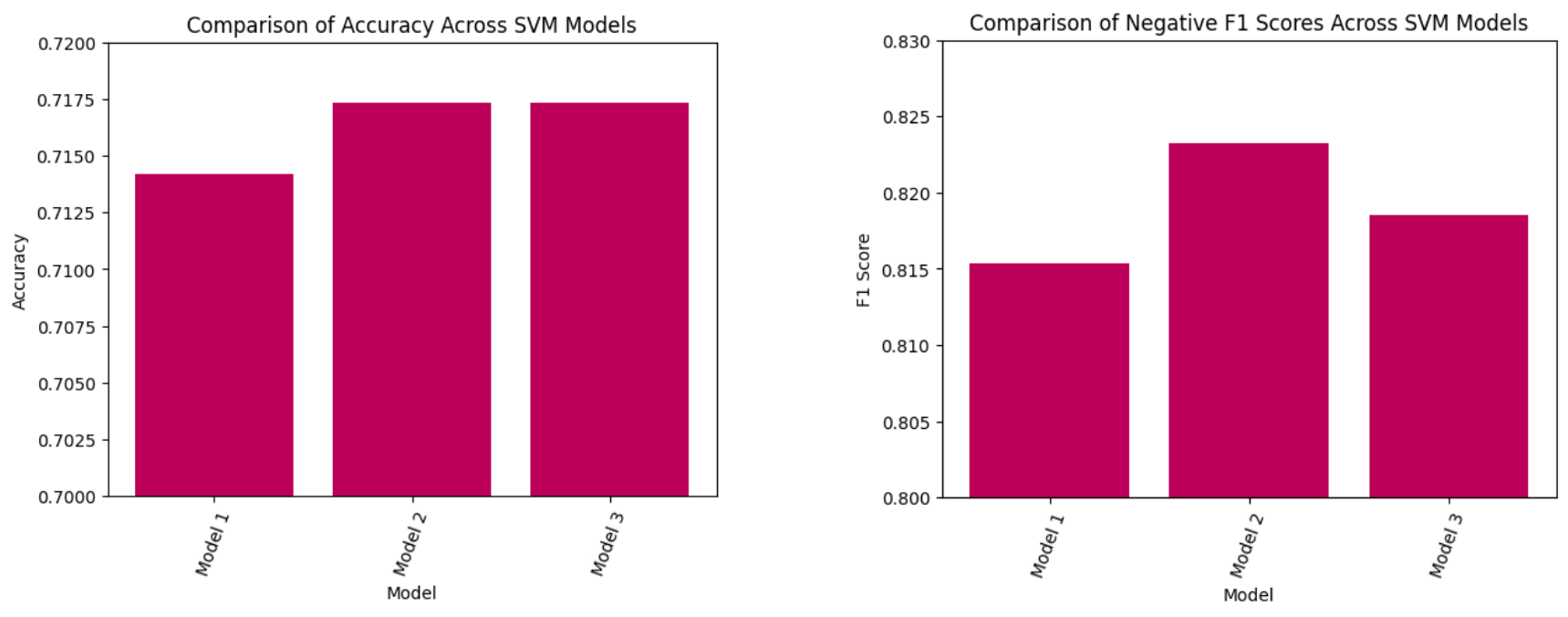
On average, its predictions were about 26% accurate.



While the accuracies of the first four classifiers were in the high 60s-70s, KNN was in the 20s. This is most probably due to the fact there are no appropriate distance measures in our dataset, so any adjustment of the neighbors would not have been relevant. However, if we were to further train the other model, we would have tried different parameter values to optimize the accuracy and F1 scores.

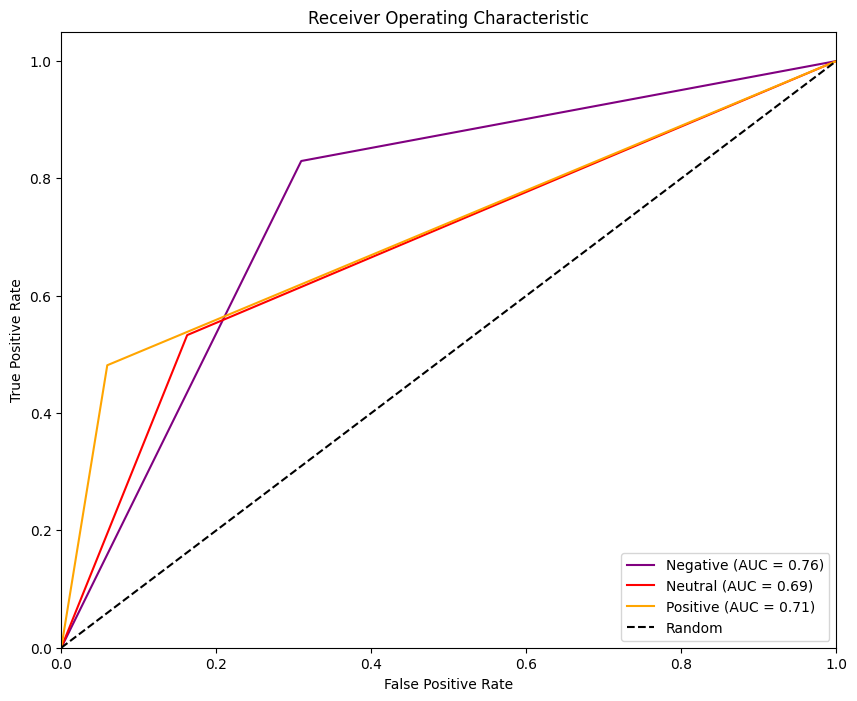
*SVM*

Across the three SVM models that we trained, our accuracy across sentiments ranged from 0.71-0.72. This was surprising to us because even after hypertuning the parameters, the accuracy only increased by roughly 0.01. Because the accuracy between the models was so similar, we also chose to analyze the F1 score as well due to the unbalanced nature of the data. For the F1 score comparison for the negative sentiment, the second model we trained using TfidfVectorizer() performed the best with an F1 score of about 0.82 whereas the first model we trained with no parameter tuning performed the worst with an F1 score of about 0.81, again a relatively small difference between the models.



Given these minimal improvements in the accuracy and F1 scores of the models, we believe additional hypertuning or pre-processing measures could be taken. For example, we could try different libraries of stop words or different libraries for performing SVM. In our post-modeling research, we came across additional methods such as OneVsRestClassifier(), a function that would allow us to create different classifiers for our different classes.

*Neural Network- Based Sentiment Analysis Model*   
Upon evaluation, the neural network model achieved an accuracy of 71%. While the accuracy results may seem comparable to the SVM model earlier, it's important to note that neural networks often require more computational resources and data for achieving significant accuracy improvements. But at the same time it has much higher potential for creating highly accurate models. The balance of these elements is a crucial aspect of neural net model design.  
Recognizing the imbalanced sentiment dataset, further analysis was conducted using the F1 score. This metric, which delves into precision and recall capabilities, exhibited scores ranging from 0.53 to 0.81 for the diverse sentiment categories. This was particularly promising as it demonstrated the model's ability to handle negative sentiment classification effectively. While the model's accuracy and F1 score improvements were relatively modest despite hyperparameter tuning, it is crucial to recognize that neural networks have an inherent capacity to capture intricate relationships within the data.   
Now coming to the ROC curve, we get AUC for each class, i.e: 0.76 for “negative”, 0.69 for “neutral” and 0.72 for “positive” class. A higher AUC value signifies better performance in distinguishing between the positive and negative classes. For reference, An AUC of 0.5 corresponds to random classification, while an AUC of 1.0 indicates perfect classification. The AUC value of 0.76 for the "negative" class indicates that the model has a good ability to distinguish between true negative instances and false positive instances.

  
Accuracy: 0.71 (71%)

F1-score for negative sentiment: 0.81

F1-score for neutral sentiment: 0.53

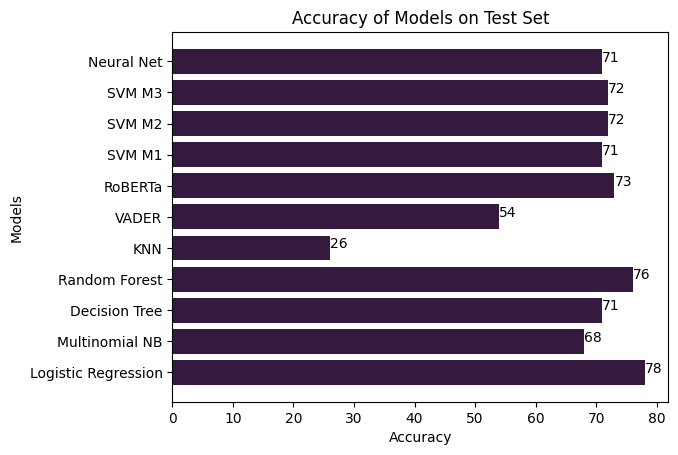
F1-score for positive sentiment: 0.55

Macro-averaged F1-score: 0.63

Weighted-averaged F1-score: 0.70

# Project Results:

Our project focused on an extensive comparative analysis of various sentiment analysis models, aiming to identify their accuracy levels and implications.The obtained accuracy scores can be seen in the graph. Interestingly, Logistic Regression emerged as a top performer showcasing its exceptional capability in accurately discerning sentiment all the while not having any parameter tuning done to it. Neural networks, Random Forest, SVM and RoBERTa also performed well.



The choice of model should align with the specific project requirements and data characteristics. Our analysis demonstrated that while pre-trained models exhibit competence, they may not always offer the optimal performance required for industry-specific sentiment analysis. In contrast, our custom models, fine-tuned to identify the intricacies of sentiments expressed towards companies, showcased the potential to provide more reliable and contextually accurate sentiment analysis, even though the accuracy metrics improved minimally after tuning.

# Project Impact:

The implications of our project extend to multiple customer facing sectors & industries, such as airlines, hospitality, electronics etc. Marketing teams can utilize our sentiment analysis models to gauge the effectiveness of campaigns and identify areas for improvement. Customer support departments can proactively address customer concerns and feedback, leading to enhanced customer satisfaction.  
People and companies within these industries would be able to run these models and classify their tweets so that they could determine what the overall response from consumers is towards them and their products. Based on these responses, they can take the appropriate steps to mend the customer responses or continue their positive actions. Thus, our project further pushes for redefining how businesses comprehend and respond to customer opinions within the dynamic and increasingly influential realm of social media.

# Project Next Steps:

In future iterations and expansions of this project, we would want to continue our work with the neural network model specifically, despite its average accuracy among all of the models.

In the project, our progress was restricted by the capabilities of our Google Colab Notebook. We began training our neural networks for 10 epochs initially, but due to timeouts (particularly during the 16-hour GridSearch for hyperparameter tuning), we had to shorten the epochs to 5. For upcoming phases, we aim to utilize higher performance hardware by running models locally on more powerful machines. This way, we can extend the epochs to 10-20, ensuring successful training without encountering timeouts. In addition to this, we would manually impute more y variables as well as consider expanding our search beyond just tweets related to airlines. We could incorporate tweets mentioning hotels, restaurants, and other entities where consumer response is relevant.   
Delving further into hyperparameter optimization could potentially unlock impactful improvements in our model's accuracy and precision. Also, implementing techniques to handle class imbalance, a common challenge in sentiment analysis, could enhance the model's performance. Finally, Incorporating pre-trained word embeddings like Word2Vec, GloVe, or BERT could significantly enhance our model. These embeddings capture intricate meanings within words, helping the model better understand text and improving sentiment analysis results.

However, we also came across different methodologies for training our models using SVM. As previously mentioned, it would be interesting to see the results of OneVsRestClassifier(), so we could also improve the classification of the positive and neutral sentiments, despite there not being as many positive and neutral tweets being found in the dataset. But with that, it would also benefit the model to have more positive and neutral tweets to ensure that the right sentiment specific words are appearing enough times in the text.

Finally, it would be interesting for us to be able to classify the negative tweets based on the cause and also further by the individual airlines, the way the initial dataset had done so. Doing this would allow individual airlines in the industry to further narrow down where exactly they are lacking to be receiving negative remarks. For example, if ‘late baggage’ was a category for tweets, an airline would be able to further investigate why there is an issue with baggage arriving on time.

# 

# **Conclusion:**

In conclusion, our project focused on understanding sentiments in U.S. airline-related tweets using advanced machine learning models. Through a comprehensive exploration of various techniques, we aimed to uncover the emotions conveyed by users towards U.S. airlines.

We found that Logistic Regression, a simpler model, performed remarkably well in sentiment analysis, accurately gauging sentiment without complex tuning. Our neural network models, powered by LSTM, showcased potential in understanding subtle text nuances. Additionally, SVM and RoBERTa demonstrated their effectiveness in sentiment classification.

The implications of our project extend beyond aviation, benefiting industries reliant on customer perception. Our findings can guide marketing strategies and empower customer support to address concerns effectively, boosting satisfaction and loyalty.

Looking ahead, we see room for refining models through hyperparameter tuning, addressing class imbalances, and using pre-trained embeddings. By expanding our dataset to include broader domains, our sentiment analysis solution can make a wider impact. Through continued development, our sentiment analysis models promise to reshape business-customer engagement, enriching experiences and strategic decisions.