Airline Sentiment Analysis

1. **Project Topic**

   Assess public sentiment towards major U.S. airlines based on Twitter data collected in February 2015.

2. **Motivation**

   Whenever a customer flies with an airline, they often tend to post about their experience with flying with the airlines. These can be positive, negative or neutral. Regardless of what they post, all this information can be quite useful for airline companies to understand to help improve their services. The process of analyzing such information is known as Sentiment Analysis. Through utilizing sentiment analysis we can filter through hundreds if not thousands of tweets and obtain useful information for what an airline company should do to focus on their services.

   Given that twitter is a common social media source people often use to post about their reviews of flying with specific airlines, we've decided to analyze sentiment data from it to derive insightful information for companies to utilize. In order to determine the sentiment of the text, we plan to utilize VADER and TextBlob techniques. In addition, we plan to train various machine learning models, such as Logistic Regression, Naive Bayes, Decision Trees, and ensemble methods (Random Forest, Gradient Boosting, XGBoost) to classify sentiment and show their accuracy

3. **Methodology/Approach**

   Data Preprocessing

   Twitter dataset is used as input, cleaned, preprocessed and feature engineered to get cleaned dataset as output.

   - **Cleaning and Preprocessing**: Tokenization, lemmatization, and spell-checking of text data.
   - **Feature Engineering**: Creating features using bag-of-words and TF-IDF vectorization techniques.

   Model Training

   Cleaned twitter dataset is used as input to the machine learning and sentiment scoring models.

- **Machine Learning Models**: Training various models including Logistic Regression, Naive Bayes, Decision Trees, and ensemble methods (Random Forest, Gradient Boosting, XGBoost) to classify sentiment.
- **Sentiment Scoring**: Utilizing tools like VADER and TextBlob for additional sentiment analysis.

Evaluation Metrics

Both the machine learning and sentiment scoring models were evaluated using the below metrics and the best model gets chosen based on these metrics.

- Accuracy
- AUC-ROC
- F1-score

User Interface

Best model chosen from the evaluation is used in the backend to predict sentiment about the airlines.

- **Development**: Building a user interface with the sentiment scoring engine running in the backend.
- **Functionality**: Predicting sentiment for new tweets about airlines.

4. Software Implementation Details

The application is built entirely using open-source Python frameworks and libraries. Some of the key libraries used, include:

1. **Interface**
    a. **React:** React is an open-source JavaScript library for building user interfaces and developing reusable UI components. It is maintained by Meta and is widely used for creating complex, interactive web applications. React supports many popular JavaScript libraries and frameworks, such as Redux, React Router, and Jest, making it a versatile choice for front-end development
2. **Text Preprocessing**
    a. NLTK: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging,

parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

b. **WordCloud:** WordCloud is a Python library for generating word clouds, a visual representation of text data where the size and prominence of words indicate their frequency or importance. It provides a simple and customizable way to create word clouds from text data, allowing users to visualize and explore the underlying themes and patterns in their data.

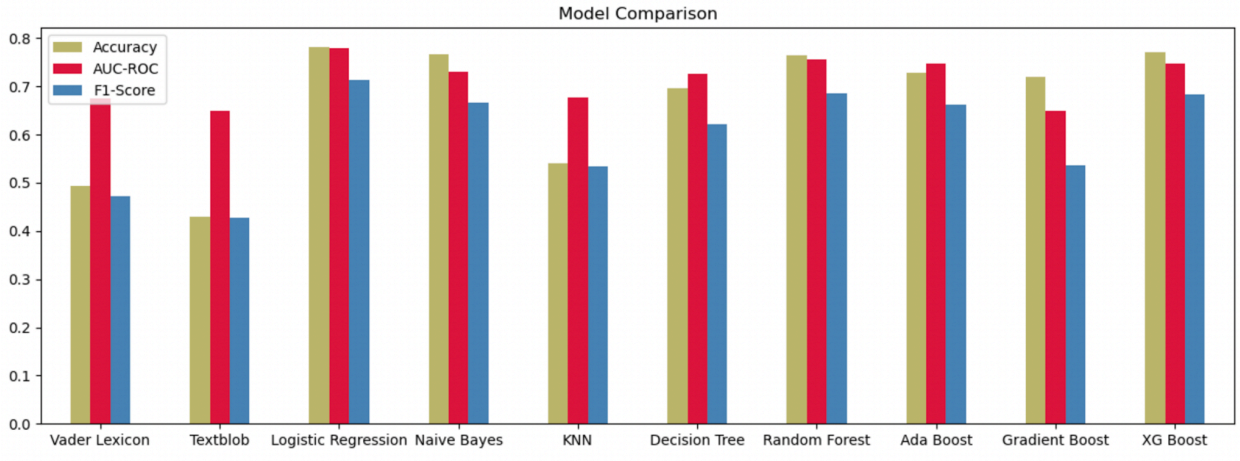
3. **NLP analysis**
    a. **TextBlob:** TextBlob is a python library for Natural Language Processing (NLP). TextBlob actively uses Natural Language ToolKit (NLTK) to achieve its tasks. TextBlob returns polarity and subjectivity of a sentence. Polarity lies between [-1,1],-1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information.
    b. **VADER:** (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based sentiment analysis tool specifically designed for social media text, providing a simple and effective way to analyze the sentiment of text data, including emotions, intensity, and polarity.
4. **ML Models**
    a. Machine learning models such as Logistic Regression, Naive Bayes, and Decision Trees can be used for sentiment analysis, each with their own strengths and weaknesses. Ensemble methods like Random Forest, Gradient Boosting, and XGBoost combine multiple models to improve accuracy and robustness, with XGBoost being a highly optimized and efficient option for large-scale tasks.
5. **Data visualization**
    a. **Matplotlib**: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python

6. **Software Usage Details**

Airline_Sentiment_Analysis_CS410 - Illinois Media Space

7. Results:

Here is the evaluation result based on the performance of different models using three metrics: Accuracy, AUC-ROC, and F1-Score.

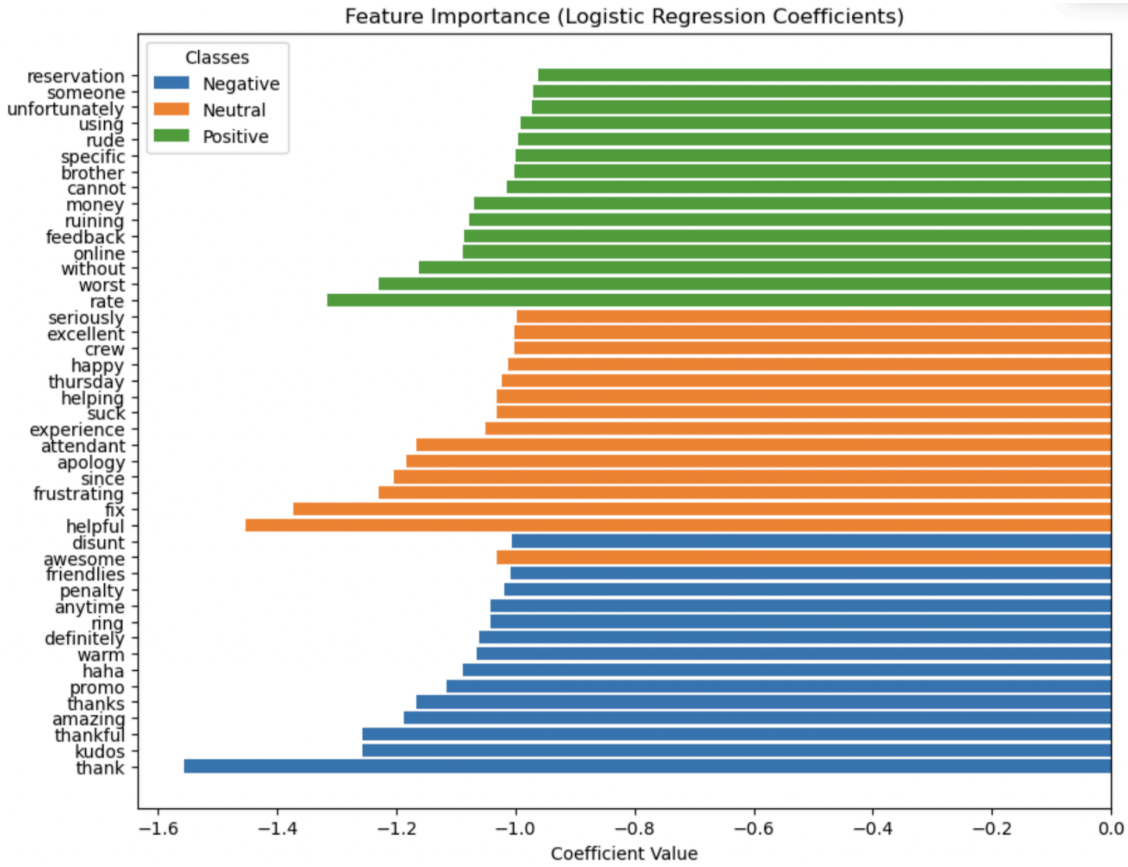


- The models compared include Vader Lexicon, Textblob, Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest, Ada Boost, Gradient Boost, and XG Boost. Each model has three bars representing the three metrics, with Accuracy in yellow, AUC-ROC in red, and F1-Score in blue.

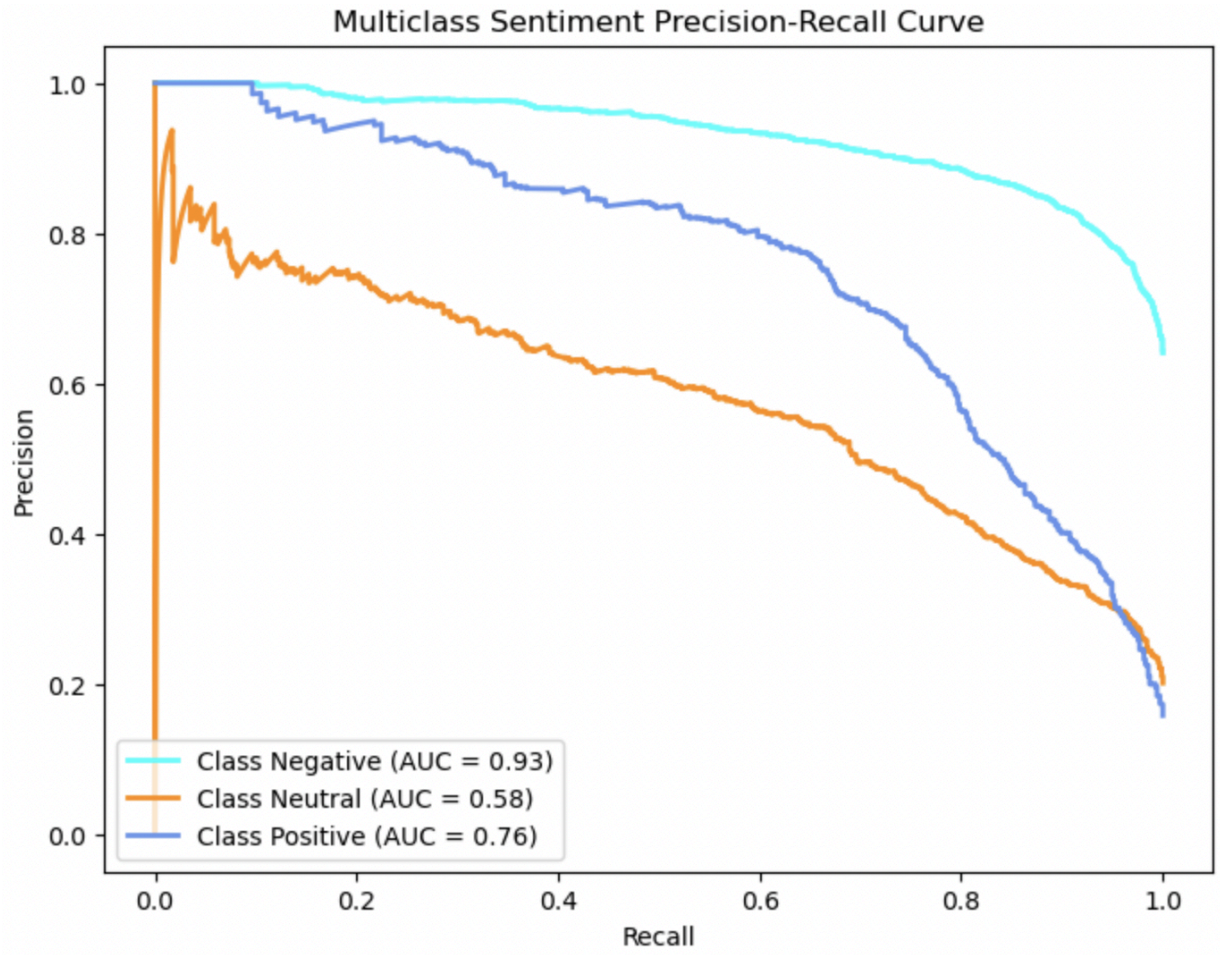
- The chart highlights that Logistic Regression, Random Forest, and XG Boost are the top performers across all metrics, with values close to or above 0.7. In contrast, Vader Lexicon and Textblob exhibit the lowest performance, particularly in Accuracy and F1-Score, where their values are below 0.5.

- Based on this comparison we determined **Logistic regression** as the most effective model as it showcases superiority in achieving higher predictive performance and reliability.

- The trained model is then used in the front end for predicting the sentiment of the airline tweets.

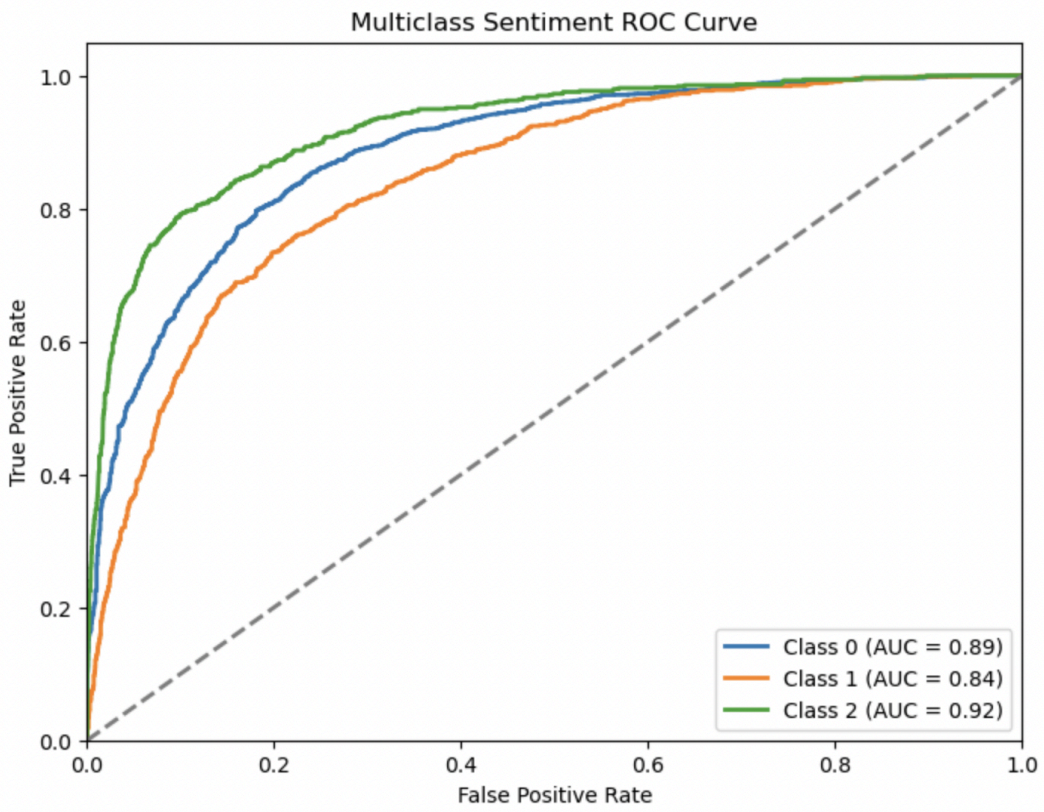


  - This graph highlights the feature importance of individual words for a Logistic Regression model trained on sentiment analysis. Specifically it showcases how specific words are associated with predicting **negative**, **neutral**, or **positive** sentiments. Words like "rude," "unfortunately," and "money" strongly indicate negative sentiment, while words such as "amazing," "thankful," and "awesome" are strong predictors of positive sentiment. Neutral words like "feedback," "helpful," and "online" are more factual and balanced. This information can be quite useful in prioritizing responses to negative tweets for customer service or amplifying positive tweets for marketing, while also identifying key features to refine the model or understand dataset biases.



-

- This graph represents the precision recall curves for a multiclass sentiment model. These 3 classes include Positive, Negative, and Neutral. The model that performs the best is the negative class since it has the largest AUC. This demonstrates that the model has high precision and recall for detecting negative sentiment. For positive class, it shows moderate performance as precision drops when recall increases. However, for neutral class the model performs the worst.




```
Average AUC: 0.88
```

- This graph represents the model's ability to distinguish between sentiment classes. In the graph the positive class and negative class models perform the best, indicating that the model reliably predicts positive and negative sentiments. However, the neutral class doesn't perform as well as the other 2, which suggests the model struggles to neutral tweets from positive or negative ones. However, given that the average AUC score is 0.88, it indicates strong overall model performance. This makes it effective to classify sentiment.

We built the following website pages to provide additional insights on the dataset, sentiment analysis and to predict sentiment based on new tweets.

1. Home Page - Provides an overview of our project with links to all the different pages.
2. Dataset Overview Page - Dive into the dataset we used as well as key metrics related to our model's performance.
3. Sentiment Analysis Page - Explore how our model performed on our dataset with key insights into why a particular sentiment was assigned to each tweet.

4. Sentiment Search Page - See our model in action! Our best performing model (Logistic Regression) is made available for you. Type in a sample tweet about an airline and see what sentiment our model assigns.

8. Limitations

The process of collecting tweets using the Twitter API is rate-limited, allowing a maximum of 50 requests per 15-minute window for the search API and similarly severe limitations on the number of tweets that could be collected during a month (only 200). This restriction slows down the data collection process, especially when dealing with large datasets. Furthermore, the API may return errors or timeout due to high traffic or server overload, requiring us to implement retry mechanisms and handle exceptions. As a result, we prioritize collecting tweets offline and feeding into the model rather than attempting to collect it realtime.

9. Team Members and Contributions

Name	Netid
Ameya Malekar	malekar2
Gokul Naidu	gnaidu2
Manoj Munaganuru	manojsm2
Praneeth Rangamudri	jrang3
Siva Paramasivam	sp108

All the tasks were split equally among all team members. This includes experimentation with various techniques, building the various components and creating all the necessary documentation.