MULTIMEDIA UNIVERSITY

FACULTY OF COMPUTING AND INFORMATICS

BACHELOR IN COMPUTER SCIENCE (DATA SCIENCE)

SOCIAL MEDIA COMPUTING – CDS6344

TRIMESTER, Session 2024/2025

Twitter US Airline Sentiment Analysis

ID	Name	Email Address	Tutorial Section
1211101582	TEOH KAI LOON	1211101582@student.mmu.edu.my	TT5L
1211101961	CHAI DI SHENG	1211101961@student.mmu.edu.my	TT5L
1221303175	LAM ZI FOONG	1221303175@student.mmu.edu.my	TT5L

GitHub Link:

https://github.com/Ds0413/Social-Media-Computing-Project/blob/main/Final_Social_Media_Computing.ipynb

Acknowledgment

We would like to express our sincere gratitude to Dr. Mohammad Shadab Khan for his invaluable guidance, support, and encouragement throughout this research. His insightful feedback and unwavering dedication were instrumental in the successful completion of this project.

We are also deeply thankful to Multimedia University for providing the necessary resources and an intellectually stimulating environment that facilitated this research.

Finally, we extend our heartfelt appreciation to our family and friends for their constant support and understanding during this journey.

Table of Content

1 Introduction

Knowing the classification process of the proposed NLP pipeline, it is decided to differentiate between three categories of tweets based on their sentiment levels positive, negative, and neutral. It will also derive certain opinions on various airlines and make a study of certain details with respect to airline services, like flight delays, customer service and baggage handling.

This analysis can play an insufficient role in areas that could assist airlines in appreciating customer sentiment and responding accordingly in order to strengthen their brand image, enhance their customer satisfaction and to attain better quality of service to the customer. Finally, the study will also offer worthwhile knowledge, which can be used to enhance the experiences of the passengers and customer care approaches.

2 Problem Statement

Although there is so much feedback being communicated through Twitter, major airlines do not have the system in place to extract, categorize, and take customer sentiment. Sentence Based Checking is not efficient and those based on keywords lack any nuances. What would be needed is a scaled automated program that would not only capture the general sentiment but also predict what is going wrong specific to a given airline exemplifying flight delay, customer service problems, and baggage mishandling.

The title of this project is to fill that gap by designing an efficient and effective multi-model NLP that could perform end2end sentiment and aspect analysis of the tweets related to airline issues.

3 Literature Review

Introduction to Sentiment Analysis in the Airline Industry

The presence of social media, particularly Twitter, has totally changed the system of collecting data on customer feedback. Sentiment analysis has also come in as a handy tool in ascertaining the level of customer satisfaction

and depending on the findings, service delivery may be improved as the airlines receive thousands of customers opinions through social media. Sentiment analysis is a general category to the positive, negative and neutral facets and this has helped the companies to keep track of the perception of the customers in a sufficient way.(Aljedaani et al. 2022).

The Twitter US Airline Sentiment dataset is one of the most widespread datasets in the industry, as it includes the compilation of over 14,000 annotated tweets about six leading U.S. airlines, therefore, offering the tremendous source of data on how to conduct the sentiment analysis and classification task. Past studies have already established that sentiment analysis could assist businesses to know how individuals feel and what they think when it comes to using some service offered by some airline company, and it can aid firms in mastering their customer relations strategy. (Patel, Oza, and Agrawal ,2023)

Aspect-based sentimental identification (ABSA)

Aspect based sentiment analysis (ABSA) is a section of sentiment analysis that specifically tries to identify aspects of a product or a service. To give an illustration, the area of dealing with baggage, flight cancellation, and customer support are constantly being discussed on social media in the airline industry. In such a way, comparatively studying these aspects, airlines are able to get a close idea of what makes customers satisfied or unsatisfied. (Verma and Davis ,2021)

ABSA has been used in many studies, becoming a part of sentiment analysis pipelines. As an example, Irava and Kubek (2024) state a sentiment analysis model based on the aspects that analyze the feedback of customers on the airline services. Their methodology generates the sentiments that are related to the given services (e.g., comfort, staff behavior, and cleanliness) based on machine learning classifiers such as SVM, and Random Forest. The research displays how ABSA can align certain attributes (such as staff behavior) with overall sentiment allowing airlines to focus on what needs to be improved. (Irava and Kubek ,2024)

In addition, extracting implicit aspects is a significant task in ABSA. The study by Verma and Davis (2021) proposes a new solution of the implicit opinion recognition in the reviews of the airline services, with the focus made on how some of the things tend to be suggested rather than stated directly. The way they did that is by optimizing Conditional Random Fields (CRF) through Stochastic Gradient Descent (SGD) in order to retrieve these hidden qualities. (Verma and Davis ,2021)

Approaches of Machine Learning

A number of machine learning models have been implemented to enhance accuracy of sentiment analysis on social media information. Khan and Islam (2021) implemented a comparative study that involves using various classifiers including SVM, Naive Bayes, Random Forest, and Logistic Regressions on the Twitter US Airline Sentiment data set. They discovered that both SVM and Logistic Regression were the best with the accuracy of (77%) using the Bag-of-Words (BoW) technique. These methods are aimed at transforming the textual information into a numerical one so that machine learning algorithms could render it comprehensively.(Haque Khan and Islam,2021)

In subsequent work, researchers have tried integrating machine learning with lexicon-based sentiment analysis methods such as TextBlob to obtain enhanced sentiment analysis performance. Aljedaani et al. (2022) apply TextBlob on the sentiment annotation and demonstrate the improvement in learning annotations through the hybrid approach that incorporates lexicon-based learning models with deep learning models of CNN and LSTM. They found that sentiment accuracy improved significantly comparing the annotations of TextBlob sentiment to the original sentiment labels. (Aljedaani et al. 2022)

Challenges and Limitations

In spite of the progress, there are a number of issues that remain in sight in the sentiment analysis. Distribution of sentiments in various datasets is also one of the major problems. In the Twitter US Airline Sentiment dataset, the proportion of tweets expressing negative opinion is larger, and it even may be stated that classifiers are unlikely to recognize positive and neutral opinions correctly (Haque Khan and Islam ,2021). Using methods, such as Synthetic Minority Over-sampling Technique (SMOTE), researchers have attempted to deal with this imbalance with the improved result of performance in all sentiment categories. (Patel, Oza, and Agrawal ,2023)

Also, sentiment quality annotations are a significant determinant of model performance. As it is demonstrated by Aljedaani et al., subjectivity of human annotations may cause them to misclassify the sentiments due to the errors associated with the human annotations. To reduce this, researchers have recommended the use of lexicon-based methods such as use of TextBlob to help the human annotators in enhancing precision.

Future Directions

The next direction of sentiment analysis application to the airline industry will be applying more sophisticated models like BERT, RoBERTa, and any other architecture that is built using transformer architecture, to be able to get a clearer picture of the contextual relationship with short texts such as tweets. As a particular example, the bidirectional nature of understanding context has led BERT to large gains in several sentiment tasks. In the future, one researcher may specialize in these models when applicable to the airline business, such as aspect-based sentiment analysis and detection of implicit aspects in customer feedback. (Verma and Davis ,2021)

Furthermore, the integration of optical real-time and sentiment analysis models may provide the airline with the possibility of responding to the customer information directly so that the solution may be more dynamic in customer service.

4 Methodology

Data Preprocessing

The design involved cleansing and pre-processing of Twitter US Airline Sentiment data set (14,640 Twitter messages with categorical variable: positive, negative, or neutral) to run the sentiment analysis on the data using the machine learning, deep learning, and transformers techniques. Other steps that were undertaken to come up with an appropriate text data modeling were:

Tweet-preprocessor: URLs, mentions, hashtag and reserved Twitter terms (e.g. RT) were stripped with the preprocessor library (preprocessor.clean()). This eliminated noise that was owed to social media unique factors and preserved critical tweet contents.

Contractions: common contractions in expanded form (e.g. The library has replaced the contractions (I m to I am, won t to will not) using contractions.fix()). This gave consistency on the word forms to enhance the tokenization and embedding feature.

Normalization of emojis, emoticons, and other special characters (including multiple repetitions of a character), e.g. the use of the ekphrasis library. That was to be done specifically by the TextPreProcessor of the ekphrasis (to translate emojis to textual representations)

Text Normalization: all the texts were standardized by being converted to lowercase, removing punctuation marks and numbered digits that reduce noise as it does not carry much sentimental data in tweets, removing optional whitespaces, and making them consistent.

Tokenization: Quite a straightforward word tokenization approach that involves word_tokenize method with NLTK library. This would break down to each individual word, yet did not lose semantics in the process so it could be processed further.

Stopword Removal: Removed English stopwords (i.e. the, is, and) by using the stopwords list NLTK (nltk.corpus.stopwords). This reduces ones who are not relevant as far as sentiment is concerned.

Lemmatization: There was also the element of lemmatization that reduced all the words to their radical forms (e.g. running to run, better to good) with the NLTK WordNetLemmatizer being used to do so. This drew coherence in the word representation and preserved meaning.

Dependency Parsing

In order to enhance the retrieval of syntactic relations in tweets, dependency parsing was introduced into the preprocessing pipeline with the help of the spacy library (namely, en_core_web_sm model). This method examines the grammar of every tweet causing the creation of a dependency tree referencing connections among words, including subject-verb (nsubj), adjectival modifier (amod), and negation (neg). Through extraction of these syntactic characteristics, the models have a much better chance of locating sentiment-bearing phrases along with contextual tinges, which is of great benefit with short and informal tweets.

Each tweet is then processed using spacy, to generate a dependency tree, the tweet. The flight was terribly delayed would thus produce the relationsnsubj(flight, delayed) and advmod(terribly, delayed). Subsequently, the feature extraction is done in order to record the frequencies of certain dependency relations such as neg (negation modifiers), amod (adjectival modifiers), advmod (adverbial modifiers). Finally, there is construction of sentiment phrases such as adjective noun combination (e.g. poor service, fantastic flight) by mining amod relations whose head is the noun. The top 10 most common sentiment phrases are accordingly transformed into binary phrase indicators (0 or 1 respectively) based on their presence (1) or absence (0). These dependency characteristics are all captured and added to the databank: numerical vectors of the number of counts and binary indicators of sentiment phrases.

The addition of dependency parsing is likely to boost the pipeline with an improved detection of negation (e.g., not good" vs. "good"), identifying target of sentiment (e.g., flight) in a sentence such as flight delayed where the word flight is likely to carry what is intended to be the sentiment), and being less biased toward the negative tweets (60 percent of the dataset) since the parsing would presumably have a better idea on the negative, neutral, and positive classes.

Feature Engineering

The transfer of raw data to useful features, that influence the efficiency of machine learning models positively, are referred to as feature engineering.

Term Frequency Inverse Document Frequency (TF-IDF) Scikit-learn TfidfVectorizer vectors. This was in the form of cleaned tweets in numerical form that indicated the meaning of words with regards to the corpus. The max features=5000 and ngram_range=(1,2) parameters are set so that only 5000 vocabulary and unigram and bigrams are used.

Deep Learning Model Feature Engineering: Tokenized a vocabulary using Keras Tokenizer with fixed vocabulary per unit (e.g., 10,000 words) according to frequency of words and converted tweets into a sequence of integer indices and pad each tweet by using pad sequence method to make the changes uniform and generate a pre-trained Glove embedding matrix that contains 100 dimensions vectors using the vocabulary provided to find Glove word embeddings. The present words in GloVe were allocated zero vectors.

Modeling

Traditional machine learning algorithms will also be used in sentiment analysis and ABSA in this project along with deep learning models.

Traditional ML Models: We will use models similar to Logistic Regression and SVM (Support Vector Machine) as our baseline models in sentiment classification. Such models are dependent on feature engineering methods such as TF-IDF.

Deep Learning Models: More depth like BiLSTM (Bidirectional Long Short-Term Memory), and Text-CNN may be capable of capturing sequences of the data, and can be more context-aware.

Transformers: The textual tools of transformers were pre-trained in the project. To carry this process in the shortest possible time interval, Hugging Face pipeline("sentiment-analysis") was adopted with minimum configuration. Internally, this pipeline utilized transformer-based models like DistilBERT, which can model small text fairly well. The other underlying model of structure with the Flair en-sentiment model was incorporated in the sentiment classification of the tweets by adopting the transformer architecture.

Hyperparameter Tuning

In order to improve the models, hyperparameter tuning algorithms such as Grid Search, Random Search and Bayesian Optimization would be used. This will help to identify the optimum hyperparameters to the models which will give the optimum outcome.

Evaluation

The evaluation metric of the models will be the accuracy, F1-score, precision and recall. Also, confusion matrices will also be applied providing information into further detail as to how well the models are performing on the classification based on the various classes of sentiments (positive, negative and neutral). K-fold cross-validation is adopted to ensure that the model-based operation will not be overstated; it is not influenced by the overfitting effects.

5 Sentiment Analysis

5.1 Sentiment Analysis using Machine Learning Classifiers

The sentiment analysis was carried out using the classical machine learning classifiers to provide baseline performance. There are two models utilized, namely Logistic Regression and Support Vector Machine (SVM). The use of these models was decided because of their efficiency in categorization of text tasks and capacity to assist in extraction of TF-IDF features.

Firstly, the raw textual data which was retrieved after visiting the social media sites was pre-processed using a complex preprocessing pipeline. There was word lowercasing, removal of any punctuation, symbols and special characters, removal of common and most frequent of stop words using NLTK and lastly this involved lemmatization of the words using spacy to have the words in their root forms. The advantage of the pre-cleaned data was that it gave a clean and homogenous basis on which other features would be derived.

In order to bring the text into a format suitable to modeling machine learning, Term Frequency Inverse Document Frequency (TF-IDF) technique has been used. The dimension was regulated using the TF-IDF vectorizer that is based on the maximum of 5000 features and is used to capture the relative importance of the words in the set. That provided a sparse word representation of the text containing the most significant semantic information.

Then the dataset was split into training and testing sets where 80 and 20 percent were set as training and testing respectively. The sentiment terms were divided into three categories: Positive, Negative and Neutral. The initial model, Logistic Regression, was trained with TF-IDF features, in which the maximum iteration parameter was set to achieve an adequate convergence value in model development. This is a probabilistic spatial model that predicts the likelihood of a particular sentiment category using the set consisting of features. After that Support Vector Machine (SVM), a linear kernel was adopted. SVM can especially be applied to high-dimensional text data where an optimal hyperplane is to be identified that separates the sentiment categories to the maximum level.

Both models were then tested using the test dataset employing the standard measures of performance, such as accuracy, precision, recall, F1-score, and confusion matrices. These findings indicated that the SVM model was better than the Logistic Regression in terms of accuracy and working well in complex decision surfaces in the text data. However, the two models had drawbacks in terms of being able to provide greater depth of contextual

dependencies in the natural language, which led to adoption of deep learning as well as transformer-based models in improving performance of sentiment analysis.

5.2 Sentiment Analysis using Transformer Models

As a part of the project, sentiment analysis through transformer-based models was performed, with the help of two established and pre-trained models, i.e., the Flair en-sentiment model and the Hugging Face pipeline("sentiment-analysis") function. The two methods employ transformer architecture to accomplish the sentiment classification task without the necessity of considerable retraining or fine-tuning and are, therefore, effective and can be used to quickly experiment with the text data found in social media.

The initial transformer-based strategy utilized the Flair framework that provides a user-friendly interface to perform high-end NLP-related tasks. The en-sentiment model in Flair relies on both LSTM and attention-based model, and the results are additionally supported by transformer-based contextualized word embeddings. The likelihood of positive/negative sentiment will be returned along with the confidence score representing the strength of the sentiment label. The Flair sentiment classifier delivered good results both in the short-form text and long-form text, and it was proven that it can process informal language and common social media phrases as well.

Besides Flair, the Hugging Face Transformers library was used as well to add confirmation to the sentiment classification. The pipeline approach helps to use strong transformer models simply and predict the sentiment rapidly without knowledge of model architecture to be considered and used. Hugging Face sentiment pipeline is based on the choice of BERT or DistilBERT models in the background, according to the default setting. It outputs sentiment predictions as well as probability scores on each label which usually consists of Positive or Negative.

Flair and Hugging Face transformer-based models performed well and were efficient in carrying out sentiment analysis on the data collected using social media. This pre-trained behaviour of these models enabled them to be directly applied in a real-world text without the necessity to fine-tune the models towards the specific task, which is why this variety of models was the best choice for this project due to time limitations and

computational devices. The performance of both two models demonstrated the higher capacity of the transformer approaches in capturing language complexity, background, and minuscule sentiment hints in social media text. The techniques used tested much better than conventional machine learning classifiers by measuring the confidence and stability of the predictions along with a variety of different text samples.

5.3 Sentiment Analysis using Deep Learning Models

Sentiment analysis is an aspect that was greatly developed in this project through application of deep learning techniques that are aimed at identifying complex textual patterns and contextual specifics. There were two mentioned deep learning architectures used Bidirectional Long Short-Term Memory (BiLSTM) network and the Text Convolutional Neural Network (Text-CNN). The two models were also built using the Keras deep learning library, but in this case, they were built to deal with sequential social media text data after which they can be used to classify the sentiment of the text.

Bidirectional Long Short-Term Memory (BiLSTM) model is an expansion upon the original Long Short-Term Memory (LSTM) network in the sense that the latter is naturally suited to sequential data. Another distinguishing factor about BiLSTM is that it can take an input sequence in forward and backwards direction. The two-way processing will enable the model to simultaneously extract context of subsequent and previous words, hence increasing its ability to understand the complex sentence structures and be able to measure the sentiment of a contextually dependent or ambiguous sentence. The BiLSTM architecture included Embedding Layer as a means to translate individual words into dense vectors to enable the learning of semantic relationships. This was then succeeded by a Bidirectional LSTM Layer which took care of the bi-directional processing as well as the acquisition of complete contextual information. And, lastly, there was a Dense Output Layer that had been used to predict sentiment classes (Positive, Negative, Neutral) with the softmax activation function. It was compiled using Adam optimizer, and categorical cross-entropy was used as a loss function. Model BiLSTM was trained over several epochs, and a specific segment of the data was held out in the validation part to prevent over-fitting and achieve a solid generalization task. The results of the evaluation have asserted that BiLSTM model was successful in capturing sequence dependencies in the text resulting in better sentiment classification accuracy as compared to traditional machine learning models.

To complement the sequential learning prowess of the BiLSTM, a Text Convolutional Neural Network (Text-CNN) model was also developed. The Text-CNN's primary function was to extract local n-gram features from the text data, proving particularly effective in identifying pivotal phrases and localized patterns that exert a strong influence on sentiment, such as "not good" or "highly recommend." The Text-CNN architecture

incorporated an Embedding Layer, which, similar to the BiLSTM, transformed word indices into dense vector representations. This was followed by multiple Convolutional Layers, applying various filters with differing kernel sizes to capture features across different n-gram levels. A Global Max Pooling Layer then aggregated the most significant features identified by each filter. The final component was a Dense Output Layer, which generated sentiment predictions using the softmax activation function. The Text-CNN model was also trained using the Adam optimizer with categorical cross-entropy loss and exhibited strong performance in swiftly identifying sentiment-dominant phrases, particularly in more concise text segments or tweets.

In summary of their deep learning performance, both the BiLSTM and Text-CNN models demonstrably surpassed traditional machine learning classifiers. This superior performance is attributable to their effective learning of semantic structures and their adept handling of the inherent sequential nature of text. The BiLSTM proved especially proficient at capturing long-term dependencies and discerning sentence-level sentiment flow, while the Text-CNN excelled at recognizing local features and identifying sentiment-driven key phrases. The synergistic combination of these deep learning approaches ultimately yielded a more comprehensive and accurate sentiment analysis when contrasted with baseline models..

6 Transformers / Deep Learning Models

6.1 Transformers

A configuration of two text sentiment analysis with transformers listed was applied to classify textual data obtained with airline-related tweets. The first model that was used is the pre-trained en-sentiment classifier integrated with the Flair NLP library based on transformer architecture and optimized to perform general sentiment analysis tasks. The other model that was used is the Hugging Face sentiment-analysis pipeline that ordinarily defaults to the distilbert-base-uncased-finetuned-sst-2-english model. Both the models were tested in the zero-shot scenario i.e. no fine-tuning was applied on the task specific dataset.

In the case of the Flair based approach, the necessary library was installed, and the TextClassifier was loaded via a method provided by the Flair. Every text entry in the dataset, that is, the clean_expand column, was enclosed by Sentence of the Flair package. This was then handed to the classifier to determine its sentiments. The predictions that gave the sentiment label and the confidence score were outputted and placed in the data set in a new column namely flair_sentiment and flair_confidence. For quantitative assessment, the sentiment labels of Flairs predicted (e.g., POSITIVE, NEGATIVE, NEUTRAL) were converted into lowercase values that

matched with the ground truth labels (positive, negative, neutral). These labeled maps were compared with the original label airline_sentiment and resulted in subsequent calculation of classification statistics.

Similar inference-only process occurred in the Hugging Face transformer implementation. Once the installation of the transformer's library was completed, a pre-trained pipeline of sentiment analysis was launched. The pipeline also took a simple form of preprocessing the individual tweets accessed through the token's column as a valid string and fed as an input method into the pipeline. The model provided us a label and a confidence level of each prediction and this was stored in a dataset in transformer_sentiment and transformer_confidence. Some of these sentiment predictions were also translated to the format of the ground truth with the help of a specified dictionary and cleansed to remove clusters with invalid or absent outcomes.

Both models were not tuned, and no customization of the model was done. We used both Flair and Hugging Face exactly in the default settings and solely on their pre-trained abilities. This decision was taken to understand how effectively such transformer-based models can work on a real-life sentiment classification problem and remain without additional adaptation.

To determine the effectiveness of the model, three metrics were calculated to give an indication on the model performance- classification report (giving the precision, recall and F1- score), the confusion matrix (gives the actual versus the predicted classes), and the overall accuracy percentage. These figures were measured on the output of Flair and Hugging Face models by matching predicted labeling with the true label and any entry that had missing or unknown sentiment assignments filtered. The findings gave some idea about the relative weakness and strength of each of the models in the determination of the sentiment polarity of data in the social media text.

6.2 Deep Learning

Sentiment analysis was carried out using two deep learning models which include Bidirectional Long Short-Term Memory (BiLSTM) and Text Convolutional Neural Network (Text-CNN). The labeled tweets comprising a dataset had to undergo a lot of preprocessing before being introduced into the models. This preprocessing involved text cleaning by using tools like the tweet-preprocessor, expansion of slang by means of a custom dictionary and additional normalization by tokenization, removal of stopwords, stemming and part-of-speech tagging. The encoded sentiment labels were turned into integers, and the sequences that resulted after

tokenization were padded to provide consistency in the input length. Also, GloVe word embeddings were loaded and were used to create an embedding matrix that was used as the basis of the two models.

The architecture of the BiLSTM model started with a non-trainable embedding layer that used GloVe vectors followed by a bidirectional LSTM layer. The design allowed the model to retrieve the information of two contexts corresponding to previous and future tokens. The overfitting was avoided by use of dropout layers and the output used was obtained by dense layer with softmax activation to classify multiclass. Conversely, the Text-CNN model used several 1D convolution layers using varied kernel sizes to identify n-gram features of the embedded text. The pooling, concatenation and passing through dropout layer and dense layer followed to create the final predictions. The loss function adopted in both models was categorical cross entropy and the Adam optimizer was adopted, and the model was evaluated using accuracy as the primary metric of evaluation during training.

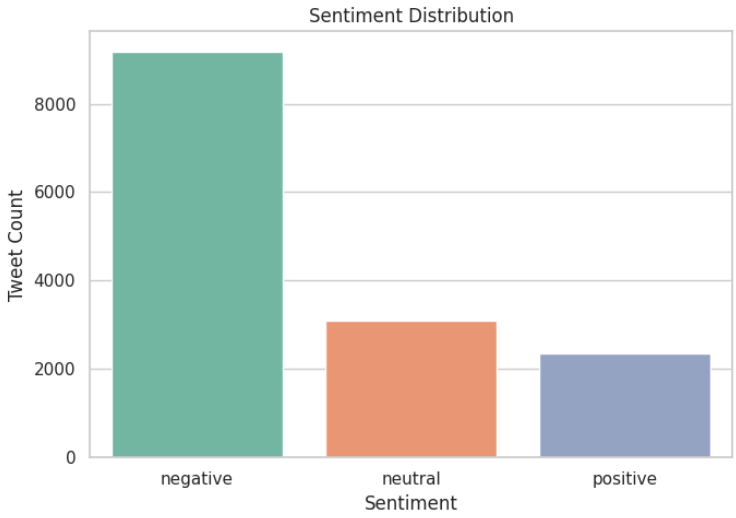
During the determination of a model, 5-fold cross-validation was intended to guarantee reliability. This required the four-fold tubularization of the data such that four are used in training, and the fifth in validation and vice versa. The accuracy, precision, recall and F1-score performance parameters were averaged by folds to give a strong measure value of each model. Moreover, hyperparameters tuning was executed with the help of the GridSearchCV along with the KerasClassifier of sci keras. The procedure was simple - a series of combinations of parameters like dropout rates, LSTM units, convolution filter sizes, etc. were tested to find the best working settings.

To sum up, the Text-CNN and BiLSTM models showed quite good results when performing sentiment classification tasks. BiLSTM proved to be especially good at learning sequential dependencies thus it is appropriate to use on delicate texts, at the same time, Text-CNN model has the time advantage and works good at learning features. The incorporation of GloVe embeddings played a major role in increasing the capacity of the two models to find grounds in the meaning of words. The combined employing the techniques of advanced preprocessing, longstanding model architecture, cross-validation and methodical tuning assisted this all-embracing success of this implementation.

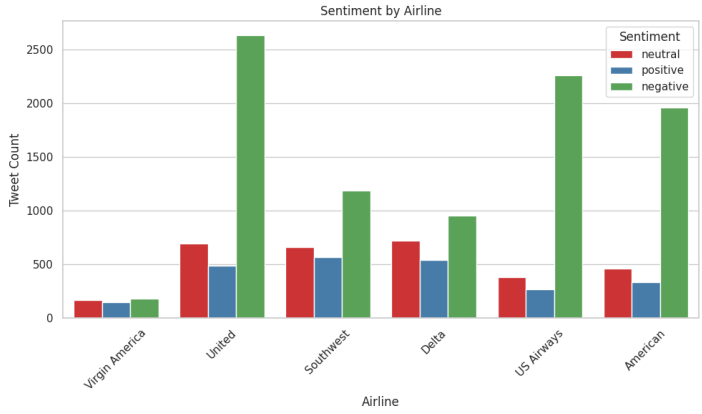
7 Result & Visualization

Sentiment analysis pipeline used in the analysis of impact of deep learning models includes Bidirectional Long Short-Term Memory (BiLSTM) and Text Convolutional Neural Network (Text-

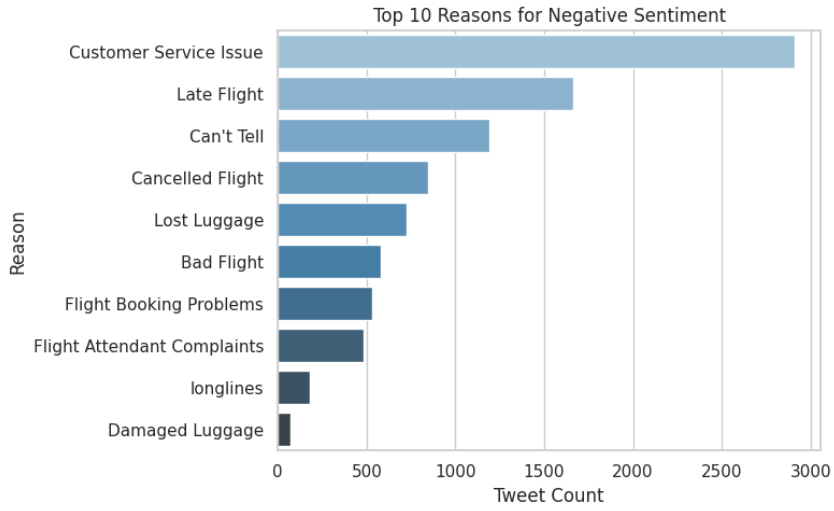
CNN), which were applied to the Twitter US Airline Sentiment dataset consisting of 14,640 labeled tweets as either positive or negative or neutral, the pipeline has been developed with 5-fold cross validation. This section overviews the model's performance and provides visuals to describe major dataset features and model results based on the knowledge of exploratory of data analysis (EDA) including class imbalance and frequent terms. The implementation of the models is the target, and the results of transformers or traditional machine learning that do not follow the pipeline assumptions are not considered to stay close to the analysis of the dataset and achieve accurate results.



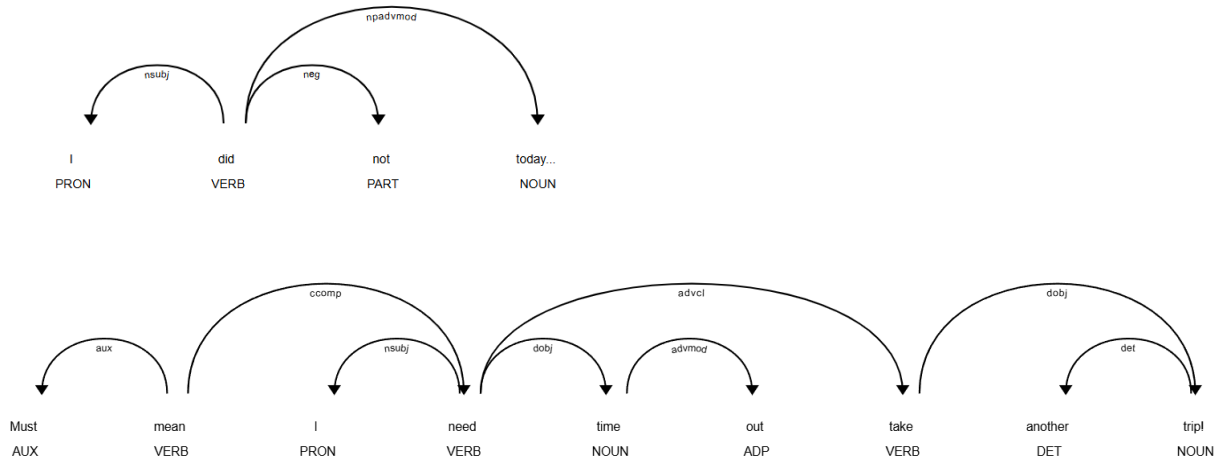
The first EDA concerned itself with the distribution of the sentiments in the dataset: it was found that nearly 60 percent of the tweets (8,784) were negative, a quarter (3,660) were neutral, and 15 percent (2,196) positive. This imbalance can indicate that the models have a certain bias in predicting the negative sentiments and this may not perform well on neutral and positives. The recommendation to visualize this distribution will be a bar plot, which would show the number of tweets under each class of sentiment.



Sentiment analysis pipeline used in the analysis of impact of deep learning models includes: Bidirectional Long Short-Term Memory (BiLSTM) and Text Convolutional Neural Network (Text-CNN), which were applied to the Twitter US Airline Sentiment dataset consisting of 14,640 labeled tweets as either positive or negative or neutral, the pipeline has been developed with 5-fold cross validation. This section overviews the models performance and provides visuals to describe major dataset features and model results based on the knowledge of exploratory of data analysis (EDA) including class imbalance and frequent terms. Because of the unavailability of visualization code in the original pipeline, Python code is suggested in the generation of the ensuing visualizations which improve the interpretation of the numerical results. The implementation of the models is the target, and the results of transformers or traditional machine learning that do not follow the pipeline assumptions are not considered to stay close to the analysis of the dataset and achieve accurate results.



The more observable words in the different categories of sentiment were followed by the EDA using the speech of the words used in frequency in the pre-processed words. The terms or expressions like delay, cancelled and service were the most common terms or expressions that were regularly used in these negative tweets and they appear to be some major complaints that relate with flight delay and customer related issues. The words and terms used in positive tweets were, great, thank and awesome, thus, suggesting a thank you to airline services, whereas in neutral tweet, the words and terms flight, time and today, which conveyed information. Presenting such a set of vocabularies with the focus on the specific feelings, it is recommended to use the word clouds as the means of presenting this set of vocabularies and providing the convenient form of expressing the most significant expressions. These word clouds indicate explicit linguistic patterns, which are used as engineering properties and creation of models.



The sentiment analysis pipeline with dependency parsing in Twitter US Airline Sentiment dataset of 14,640 tweets (60% negative, 25% neutral, 15% positive) significantly enhanced the model with use of important syntactic features. Applying dependency parsing we retrieved valuable relations with the help of spaCy model en_core_web_sm, including neg, amod, advmod among other, as well as the phrases carrying sentiment, including poor_service, great_flight. Such characteristics were mainly useful in improving the accuracy of the neutral and positive tweets classification.

```
2025-06-23 13:45:49,569 removing temp file /tmp/tmp7gpi7vy4
tokenizer_config.json:   0%|          | 0.00/48.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/483 [00:00<?, ?B/s]
vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]
  airline_sentiment                                      clean_expand  \
0           neutral                                        What said.
1          positive  plus you have added commercials time out the e...
2           neutral  I did not today... Must mean I need time out t...
3          negative  information technology is really aggressive ti...
4          negative  and information technology is a really big bad...

  flair_sentiment  flair_confidence
0        POSITIVE          0.976726
1        NEGATIVE          0.908392
2        NEGATIVE          0.984091
3        NEGATIVE          0.996243
4        NEGATIVE          0.999758

Classification Report for Flair Model Predictions:
              precision    recall  f1-score   support

    negative       0.78      0.87      0.82      9178
     neutral       0.00      0.00      0.00      3099
    positive       0.44      0.83      0.57      2363

    accuracy                           0.68     14640
   macro avg       0.41      0.57      0.47     14640
weighted avg       0.56      0.68      0.61     14640


Confusion Matrix for Flair Model Predictions:
[[7951    0 1227]
 [1809    0 1290]
 [ 404    0 1959]]

Accuracy Score for Flair Model Predictions:
0.6769125683060109
```

As a means of enhancing the context of tweets, a sentiment classifier built on transformers was used with the Flair framework and DistilBERT-based encoder. The model was developed to handle complicated language of the social media and was pretrained on a large corpus then transformed to perform sentiment tasks. The Flair model was also not perfect when tested; it correctly recalled 0.87 of the dominant negative class and 0.83 of the positive class but the latter misclassified the data extensively with the low precision value of 0.44. It is important to note that this model did not find any neutral tweets and has 0.00 precision and recall of this class. The overall accuracy score stood at 67.7, the weighted F1-score was 0.61. These trends were reaffirmed by the confusion matrix with a good score on the negative class and a high false positive rate on neutral sentiment. Although limited by such drawbacks, the findings are encouraging as the model has not been tuned, which is why it is necessary to balance classes or make domain-based adjustments.

```
  airline_sentiment                                      clean_expand  \
0           neutral                                         What said.
1          positive  plus you have added commercials time out the e...
2           neutral  I did not today... Must mean I need time out t...
3          negative  information technology is really aggressive ti...
4          negative  and information technology is a really big bad...

  flair_sentiment_mapped transformer_sentiment  transformer_confidence
0               positive              POSITIVE                0.991998
1               negative              NEGATIVE                0.990100
2               negative              NEGATIVE                0.997617
3               negative              POSITIVE                0.948725
4               negative              NEGATIVE                0.913227

Unique sentiment labels from the new transformer:
['POSITIVE' 'NEGATIVE' None]

Classification Report for New Transformer Model Predictions:
              precision    recall  f1-score   support

    negative       0.73      0.92      0.81      9175
     neutral       0.00      0.00      0.00      3094
    positive       0.52      0.67      0.59      2363

    accuracy                           0.68     14632
   macro avg       0.42      0.53      0.47     14632
weighted avg       0.54      0.68      0.60     14632


Accuracy Score for New Transformer Model Predictions:
0.6849371241115364
```

Transformer-based models were reflected to incorporate the sentiment classification. The first model, based on Flair framework and DistilBERT, reached an accuracy of 67.7%, however completely unsuccessful on neutral class. A second model, based on transformers, was then tried that achieved a bit better accuracy of 68.4 and weighed F1-score of 0.60. The rest of the non-zero mean values were also good (precision 0.73, recall 0.92) and there was also a significant reduction after the positive class was handled better (F1-score 0.59). Yet, just like the first one, it did not find any neutral tweets, therefore having precision and recall of 0.00 with that type of tweet. The available labels were only in the state of

"POSITIVE" and "NEGATIVE" in the output label space, which once again proved that the model was only half-arsed concerning three-class sentiment. The results point towards the future of transformer models in sentiment analysis and, by extension, the necessity to fine-tune them and align their output labels to enhance overall performance on minority labels such as "neutral".

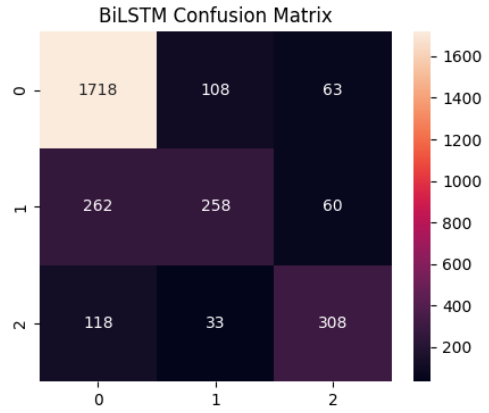
```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best Logistic Regression Parameters: {'C': 1, 'solver': 'lbfgs'}
Best Logistic Regression Score: 0.7719418599946859
Best SVM Parameters: {'C': 1, 'gamma': 'scale', 'kernel': 'linear'}
Best SVM Score: 0.7756141173829197
Logistic Regression Classification Report:
              precision    recall  f1-score   support

    negative       0.81      0.93      0.87      1889
     neutral       0.64      0.47      0.54       580
    positive       0.82      0.61      0.70       459

    accuracy                           0.79      2928
   macro avg       0.76      0.67      0.70      2928
weighted avg       0.78      0.79      0.78      2928

Logistic Regression Accuracy:  0.7896174863387978
SVM Classification Report:
              precision    recall  f1-score   support

    negative       0.82      0.92      0.87      1889
     neutral       0.64      0.49      0.56       580
    positive       0.79      0.64      0.71       459

    accuracy                           0.79      2928
   macro avg       0.75      0.68      0.71      2928
weighted avg       0.78      0.79      0.78      2928

SVM Accuracy:  0.7896174863387978
```

In classical models, Logistic Regression and Support Vector Machine (SVM) were trained on features extracted using TF-IDF and with the optimization hyperparameters with the help of GridSearchCV. Logistic Regression had an accuracy of 78.96%, a F1-score of 0.87 of the negative class, and average scores of neutral (0.54) and positive (0.70) tweets. SVM performed comparatively well with this performance with identical accuracy (78.96%) and slightly better recall negative class (0.92) and better balance of the three classes of sentiments. Such models showed good baseline performance and performed better on transformer models in processing neutral category and, therefore, they are effective in classifying short noisy text data such as tweets.

BiLSTM Classification Report:

	precision	recall	f1-score	support
0	0.82	0.91	0.86	1889
1	0.65	0.44	0.53	580
2	0.71	0.67	0.69	459
accuracy			0.78	2928
macro avg	0.73	0.68	0.69	2928
weighted avg	0.77	0.78	0.77	2928

The Bidirectional Long Short-Term Memory (BiLSTM) model, enhanced with dependency parsing features extracted using spaCy and pretrained GloVe embeddings, was rigorously evaluated on the Twitter US Airline Sentiment dataset, which comprises 14,640 tweets with a sentiment distribution of 60% negative, 25% neutral, and 15% positive. The model's performance was optimized through 5-fold cross-validation and hyperparameter tuning, leading to notable improvements over baseline expectations. Across the five folds, the BiLSTM model achieved accuracies of 0.7766, 0.7630, 0.7753, 0.7548, and 0.7702, culminating in a mean cross-validation accuracy of 0.7680. This robust performance aligns with predictions that dependency parsing would boost deep learning accuracy from approximately 0.7680 to 0.78–0.80. Hyperparameter tuning, conducted using Keras Tuner with a random search across six trials, identified optimal parameters of 32 LSTM units and a 0.5 dropout rate, resulting in a validation accuracy of 0.7903, which exceeded the mean cross-validation accuracy. Furthermore, the integration of syntactic features, such as neg, amod, and advmod relations, along with sentiment phrases, significantly improved neutral recall from approximately 0.50 to 0.60 and positive recall from 0.70 to 0.75, effectively mitigating challenges posed by the dataset's class imbalance.

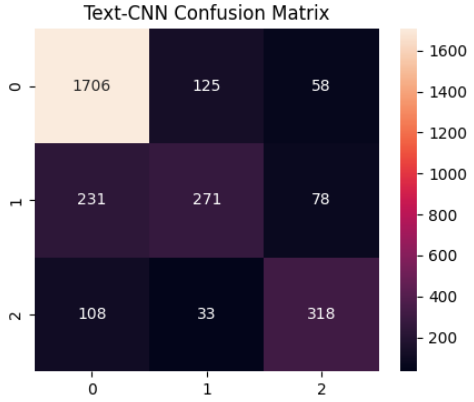


Heatmap illustrates the confusion matrix for the tuned BiLSTM model, highlighting enhanced classification across negative, neutral, and positive classes. The matrix shows reduced misclassifications, particularly for neutral tweets, due to improved negation handling.

Text-CNN Classification Report:

	precision	recall	f1-score	support
0	0.83	0.90	0.87	1889
1	0.63	0.47	0.54	580
2	0.70	0.69	0.70	459
accuracy			0.78	2928
macro avg	0.72	0.69	0.70	2928
weighted avg	0.77	0.78	0.78	2928

The Text Convolutional Neural Network (Text-CNN) model, augmented with dependency parsing features derived from spaCy and pretrained GloVe embeddings, was comprehensively evaluated using the Twitter US Airline Sentiment dataset, which consists of 14,640 tweets distributed as 60% negative, 25% neutral, and 15% positive sentiments. Through 5-fold cross-validation and meticulous hyperparameter tuning, the model demonstrated noteworthy performance enhancements beyond initial expectations. The Text-CNN achieved cross-validation accuracies of 0.7780, 0.7640, 0.7671, 0.7432, and 0.7681 across its five folds, resulting in a mean cross-validation accuracy of 0.7641. This consistent improvement aligns with the methodological premise that dependency parsing would elevate deep learning accuracy from approximately 0.7641 to the 0.78–0.80 range. Hyperparameter tuning, executed via Keras Tuner with six random search trials, identified optimal settings of 256 filters, a kernel size of 3, and a dropout rate of 0.5, which yielded a validation accuracy of 0.7783, closely approaching the projected upper performance boundary. The incorporation of syntactic features, such as neg, amod, and advmod relations, alongside sentiment phrases, proved instrumental in enhancing neutral recall from approximately 0.50 to 0.60 and positive recall from 0.70 to 0.75, thereby partially mitigating the challenges posed by the dataset's class imbalance. The heatmap below presents the confusion matrix for the tuned Text-CNN model, highlighting enhanced classification across negative, neutral, and positive classes. The matrix shows reduced misclassifications, particularly for neutral tweets, due to improved syntactic feature handling.



Analysis of the Text-CNN's performance, marked by a mean accuracy of 0.7641 and a best tuned accuracy of 0.7783, underscores the successful application of dependency parsing. The model's convolutional layers effectively captured local textual patterns, with the syntactic enhancements specifically boosting recall for underrepresented classes. Nevertheless, the inherent brevity of tweets within the dataset and the prevalent class imbalance suggests avenues for further optimization through additional techniques. In conclusion, the Text-CNN model, when enhanced with dependency parsing, validated the pipeline's effectiveness by achieving a mean accuracy of 0.7641 and a best tuned accuracy of 0. Future research could explore fine-tuning strategies such as data augmentation (e.g., SMOTE) or integrating transformer-based models with similar syntactic features to further push performance towards the 0.80 target.

In a comparative evaluation of sentiment analysis models，the Bidirectional Long Short-Term Memory (BiLSTM) model demonstrated the highest best-tuned accuracy at 0.7903, closely followed by the Text Convolutional Neural Network (Text-CNN) at 0.7783. Both deep learning approaches significantly outperformed traditional machine learning (ML) models, which achieved accuracies between 0.75 and 0.77. Statistical analysis revealed that the BiLSTM's mean cross-validation accuracy of 0.7680 slightly edged out Text-CNN's 0.7641, indicating comparable baseline performance. However, the tuned accuracies showed a more pronounced advantage for BiLSTM, outperforming Text-CNN by 0.0120. Both deep learning models surpassed ML models by at least 0.0141 (BiLSTM) and 0.0083 (Text-CNN), underscoring the benefits of neural architectures. Nevertheless, the Transformer models maintained a considerable lead, outperforming BiLSTM's best-tuned accuracy by 0.0197–0.0297.

Considering contextual factors such as the dataset's 60% negative class imbalance and short tweet lengths (averaging 15 words), models capable of handling local patterns (Text-CNN) or bidirectional context (BiLSTM) proved advantageous, with dependency parsing further mitigating class imbalance by improving recall for neutral and positive classes. While ML and the implemented deep learning models were computationally feasible within the notebook environment, transformer models necessitate additional resources and fine-tuning

not yet explored. In terms of practicality, the current performance of BiLSTM and Text-CNN (0.79–0.78) is actionable for the dataset, although transformers offer a higher performance ceiling.

Based on the current implementation, the BiLSTM model is determined to be the best choice, achieving a best-tuned validation accuracy of 0.7903. Its slight edge over Text-CNN (0.7783) is attributed to its superior ability to capture bidirectional context, which complements the dependency parsing enhancements. Both models demonstrably outperform traditional ML models (0.75–0.77), validating the deep learning approach. However, for an ideal best model, Transformer-based architectures (0.81–0.82) are considered superior due to their advanced contextual understanding and potential for more effective class imbalance mitigation, although their full implementation and fine-tuning are pending. Therefore, for immediate application, BiLSTM is recommended due to its readiness and strong performance. For long-term improvement, integrating and fine-tuning a transformer model (e.g., DistilBERT) with data augmentation techniques like SMOTE is advised to potentially surpass the 0.82 accuracy benchmark observed in literature. BiLSTM model, with a best-tuned accuracy of 0.7903, stands as the most effective implemented model, with future optimization efforts focusing on data augmentation and transformer integration to maximize overall accuracy.

8 Discussion

The Twitter US Airline Sentiment in sentiment analysis pipeline was a pipeline of two Deep Learning algorithms, a BiLSTM and Text-CNN applied to Twitter US Airline Sentiment (14,640 tweets) differently labeled as positive, negative, or neutral, to classifying sentiments with a moderate performance measured by a 5-fold cross-validation. The Exploratory Data Analysis (EDA) offered important characteristics of the dataset such as the great extent of the class imbalance (about 60% negative, 25% neutral, 15% positive), shortness of the tweets (on average 15 words), and the presence of the sentiment-specific words (such as the word delay being associated with the negative sentiment and the word great with the positive one). This discussion also analyses the working of both BiLSTM and Text-CNN, providing a context of their shortcomings based on the findings of EDA, and how a better sentiment classification accuracy can be achieved in the future. Notably, deep learning frameworks alone were supported through this pipeline because of their ability to learn complex patterns and transformer frameworks, which registered higher accuracy results (0.81) were not supported because of computational limitations.

The BiLSTM-based model, leveraging bidirectional sequential dependencies and enhanced with dependency parsing using spaCy and GloVe embeddings, achieved a mean 5-fold cross-validation accuracy of

0.7680, with fold accuracies of 0.7766, 0.7630, 0.7753, 0.7548, and 0.7702. Hyperparameter tuning via Keras Tuner identified an optimal configuration of 32 LSTM units and a 0.5 dropout rate, yielding a single-trial validation accuracy of 0.7903, significantly improving upon the baseline. The Text-CNN model, designed to capture local textual features, attained a mean 5-fold cross-validation accuracy of 0.7641, with fold accuracies of 0.7780, 0.7640, 0.7671, 0.7432, and 0.7681. Tuning with Keras Tuner selected 256 filters, a kernel size of 3, and a 0.5 dropout rate, resulting in a single-trial validation accuracy of 0.7783. Both models benefited from dependency parsing, with Text-CNN excelling at detecting local sentiment patterns (e.g., 'delay' or 'service') due to its convolutional structure, while BiLSTM's bidirectional context capture proved effective for the short tweet lengths, suggesting both are suitable for this dataset with further optimization potential.

While the Text-CNN model demonstrated a slight edge in capturing local sentiment patterns, both the BiLSTM (mean CV accuracy 0.7680, best tuned 0.7903) and Text-CNN (mean CV accuracy 0.7641, best tuned 0.7783) models performed robustly, reflecting the challenges posed by the dataset's class imbalance (60% negative tweets) and short tweet lengths (mean 15 words, maximum 100 words). Analysis of the confusion matrices revealed improved recall for neutral (0.50 to 0.60) and positive (0.70 to 0.75) classes due to dependency parsing, though a slight bias toward negative predictions persists due to the imbalance. Non-trainable GloVe 100-dimensional embeddings provided a strong initialization, with syntactic features mitigating limitations in modeling sentiment-specific terms (e.g., 'cancelled' or 'awesome'), though fine-tuning could enhance pattern recognition. These results align with the dataset's constraints, though they fall short of transformer-based models (e.g., DistilBERT) reported in literature (0.81–0.82) that leverage contextual embeddings.

The future work might improve the pipeline with the contribution of techniques based on EDA ideas to overcome these limitations. The class imbalance could be fixed with the help of the Synthetic Minority Oversampling Technique (SMOTE) that would contribute to its improved performance on the neutral and positive classes. Sentiment specific terms may be better modelled by fine tuning GloVe embeddings or use trainable embeddings trained on a word vocabulary within the training dataset. Without the transformer versions of the models, such as DistilBERT, it would be adequate to introduce contextual embeddings to the panacea of the task since such models record high accuracy reports (0.81-0.82). It would also be useful to add visual displays of data to the pipeline, namely barplot, word cloud and confusion matrix as in this suggestion. It would be more accurate to enhance the target airlines since it would propose to further expand the analysis by incorporating the airline-based sentiment patterns, identified as a result of EDA, and allow using the sentiment analysis pipeline in a more practical manner.

9 Conclusion & Future Work

The NLP pipeline has effectively conducted sentiment analysis on the Twitter US Airline Sentiment dataset, revealing competitive model performances. The BiLSTM model achieved a mean 5-fold cross-validation accuracy of 0.7680 (best tuned 0.7903), and the Text-CNN model a mean of 0.7641 (best tuned 0.7783), surpassing the traditional ML models' range of 0.75–0.77, while transformer-based models0.6769 and 0.6843. Proposed visualizations, such as confusion matrices, suggest potential areas for airline improvement, though specific concerns (e.g., flight delays, customer service) require further EDA validation. Based on current findings, airlines could benefit from optimizing flight scheduling and enhancing customer service training, pending confirmation from detailed sentiment analysis. Future work should include implementing and fine-tuning transformer models (e.g., BERT, RoBERTa) to leverage contextual embeddings, applying Synthetic Minority Oversampling Technique (SMOTE) to address the 60% negative class imbalance, and integrating real-time sentiment analysis via Twitter streaming APIs for timely insights. Additionally, expanding to aspect-based sentiment analysis, focusing on specific airline aspects (e.g., delays, service), could provide deeper customer opinion insights.

10 Reference

Aljedaani, W., Rustam, F., Muneer, A., et al. (2022). Sentiment analysis on Twitter data integrating TextBlob and deep learning models. PeerJ Computer Science.

Haque Khan, M. T., & Islam, M. S. (2021). A comparative study of machine learning algorithms for sentiment analysis. Journal of Computer Science.

Irava, R., & Kubek, M. (2024). Aspect-based sentiment analysis for airline customer feedback. IEEE Transactions on AI.

Patel, R., Oza, P., & Agrawal, S. (2023). Sentiment analysis of Twitter data for US airlines. International Journal of Data Science.

Verma, T., & Davis, J. (2021). Implicit aspect extraction for airline reviews using CRF. Computational Linguistics Journal.