



# **FACULTY OF COMPUTING AND INFORMATICS**

## **BACHELOR'S IN COMPUTER SCIENCE**

### **SOCIAL MEDIA COMPUTING – CDS6344**

**TRIMESTER , Session 2024/2025**

## **Project Title: Twitter US Airline Sentiment Analysis Using NLP Techniques**

**By :**

**MUHAMMAD ANIQ FAHMI BIN AZHAR (1211101533)**

**NUR ALYA NABILAH BINTI MD.NASER (1211101925)**

<https://github.com/aniqfahmi03/CDS6344.git>

## **Acknowledgment**

First of all, I would like to extend my sincere gratitude to all the lecturers of the Faculty of Computing & Informatics, Multimedia University, who have taught and guided me throughout my studies. In particular, I am thankful for the knowledge and insights gained in the Social Media Computing subject, which played a crucial role in shaping the direction and success of this Project. Your teachings have equipped me with the essential skills and understanding needed to carry out this research effectively.

In addition, I would like to thank my friends and family for their ongoing support, encouragement, and understanding throughout this project. They have been a pillar of strength since they had confidence in me and my capabilities, and their words of encouragement kept me focused and motivated throughout this process.

## Table Of Contents

<b>Acknowledment.....</b>	<b>2</b>
<b>Table Of Contents.....</b>	<b>3</b>
<b>1.0 Introduction.....</b>	<b>4</b>
1.1 Project Overview.....	4
1.2 Project Objective.....	4
<b>2.0 Problem statement.....</b>	<b>5</b>
<b>3.0 Literature Review.....</b>	<b>6</b>
3.1 Introduction.....	6
3.2 The conceptual of Sentiment analysis.....	6
3.3 Classical Machine Learning Approaches.....	7
3.4 Deep Learning-Based Approaches.....	7
3.5 Transformer-Based Models.....	7
3.6 Research Gap Summary.....	8
<b>4.0 Methodology.....</b>	<b>8</b>
4.1 Overview.....	8
4.2 Dataset Description.....	8
4.3 Data Preprocessing.....	8
4.4 Feature Engineering.....	8
4.5 Label Encoding.....	9
4.6 Train-Test Split and Oversampling.....	9
4.7 Classical Machine Learning Models.....	9
4.8 Deep Learning Models.....	9
4.9 Transformer-Based Models.....	9
4.10 Evaluation Metrics.....	10
4.11 Visualization Techniques.....	10
<b>5.0 Sentiment Analysis.....</b>	<b>10</b>
5.1 Introduction.....	10
5.2 Sentiment Classification for Twitter.....	10
5.3 Preprocessing for Sentiment Detection.....	11
5.4 Feature Extraction for Sentiment Analysis.....	11
5.5 Challenges in Social Media Sentiment Analysis.....	11
<b>6.0 NLP Techniques.....</b>	<b>12</b>
6.1 Introduction.....	12
6.2 Machine Learning Models.....	12
6.3 Deep Learning Model.....	12
6.4 Transformer-Based Models.....	13
<b>7.0 Result &amp; Visualization.....</b>	<b>13</b>
7.1 Introduction.....	13
7.2 Accuracy For Each Of Model.....	14

7.2.1 Logistic Regression.....	14
7.2.2 Naive Bayes.....	15
7.2.3 SVM (LINEAR).....	16
7.2.4 BiLSTM.....	17
7.2.5 DistilBERT.....	18
7.2.6 BERT.....	19
7.2.7 RoBERTa.....	19
7.5 Word Cloud Visualization.....	21
7.6 Model Accuracy Comparison Chart.....	22
<b>8.0 Discussion.....</b>	<b>23</b>
<b>9.0 Conclusion / Future Work.....</b>	<b>23</b>
<b>10.0 References.....</b>	<b>24</b>

## **1.0 Introduction**

In the digital world today, sentiment analysis has become that crucial tool in Business Intelligence since it helps organizations understand public opinion and customer satisfaction from social media data. Therefore, with millions of users stating their views on the web, Twitter-like platforms offer a rich resource of textual real-time data. Using and analyzing this data for sentiment trends enable organizations to extract actionable insights and predictions about public response, which, in turn, can help them in making decisions based on data. The project intends to build an end-to-end sentiment analysis pipeline with Natural Language Processing (NLP) techniques and machine learning models and then visualize the insights gathered through interactive data visualization methods.

The core of the study engages in the processing and classification of tweets into sentiment categories of positive, negative, and neutral through traditional machine learning, deep learning, and transformer-based models. Then, once predictions are made, a variety of insights such as sentiment distribution, frequent terms, and classifier performance are visualized in the form of intuitive dashboards. This work falls under the broad theme of Business Intelligence

### **1.1 Project Overview**

Beginning with an in-depth focus on offering full-fledged pipelines for sentiment classification and visualization using Twitter data, this project applies a corpus publicly available for tweets on airlines, made up of over 14,000 instances labeled by sentiment. The cleaning, lemmatizing, and potential extraction of features by means of TF-IDF with polarity-based sentiment analysis are carried out on the corpus. Then, several models are trained and tested: these include Logistic Regression, Naive Bayes, Support Vector Machines, BiLSTM, , and transformer-based models such as DistilBERT, BERT, or RoBERTa. With further classification, the results of classification are visualized as a series of interactive charts that show effects like sentiment distribution, model comparisons, and word-frequency patterns. The visualization was created with the help of D3.js and other interactive tools to enchant the user in exploring the same. Every chart tends to present different viewpoints about the data, with features for filtering and zooming that aid in comprehending sentiment trends across categories.

## **1.2 Project Objective**

The main objectives of this project are as follows:

- To build a robust sentiment classification system using both classical and deep learning methods.
- To extract and preprocess text data using standard NLP techniques including cleaning, lemmatization, and polarity analysis.
- To evaluate model performance across various algorithms and identify the most effective approach for sentiment classification.

## **2.0 Problem statement**

One major restriction is the complexity of the human language used on Twitter. Tweets are very short, informal, and filled with abbreviations, emojis, slang, or sarcasm, which rule-based sentiment analysis tools cannot understand accurately. Also, basic classification results like "positive," "neutral," and "negative" are of limited practical value unless they are converted into some visually meaningful insight.

Yet another crucial gap is the lack of interactive visualization tools that are user-friendly and allow stakeholders to explore sentiment trends, keyword patterns, and classifier performance in real-time fashion. Without the proper means of visualizing results, the whole thing gets complicated for spotting trends, conducting model comparisons, or quickly replying to a customer's queries.

We attempt to address this problem by combining intelligence-driven sentiment classification with data storytelling through interactive visualizations, ultimately aiding Business Intelligence.

1. Social media platforms generate overwhelming amounts of unstructured textual data that is hard to interpret manually.
2. Tweets are often informal, noisy, and linguistically complex (e.g., sarcasm, slang), making traditional sentiment analysis methods less effective.
3. Many sentiment analysis tools only provide label predictions without visual context or interactivity.

## **3.0 Literature Review**

### **3.1 Introduction**

A huge stream of textual data reflecting popular sentiment and opinion has been produced by social media's explosive expansion. Twitter and similar platforms provide a useful tool for examining consumer mood and behavior, particularly in service-oriented industries like aviation. However, because social media content is informal and unstructured, it can be difficult to interpret these feelings. Comparing transformer-based methods, deep learning methods, and conventional machine learning models, this chapter examines important research and methods in the field of sentiment analysis. It also emphasizes how, in business intelligence situations, sentiment categorization results become actionable due to the growing significance of visualization.

### **3.2 The concept of Sentiment analysis**

Sentiment analysis, often referred to as opinion mining, is a subfield of Natural Language Processing (NLP) that deals with the identification and classification of subjective information in text. The primary goal is to determine the emotional tone behind words, categorizing them as **positive**, **negative**, or **neutral** sentiments.

In Twitter sentiment analysis, the brevity and noisiness of tweets add to the complexity. Tweets often contain emojis, hashtags, mentions, abbreviations, and slang—all of which can obscure sentiment when not handled correctly. Effective sentiment analysis systems must therefore incorporate robust text preprocessing and feature extraction techniques before classification.

Traditional sentiment analysis approaches focus on handcrafted features and use classifiers such as Logistic Regression, Naive Bayes, or Support Vector Machines (SVM). These methods rely on converting text into vector representations using techniques like Bag-of-Words or TF-IDF. While effective to some extent, they struggle to capture context, sarcasm, or negation.

To overcome these limitations, researchers have turned to **deep learning** methods. Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Convolutional Neural Networks (CNNs) can learn complex patterns in text sequences, reducing the need for

manual feature engineering. Still, these methods require significant labeled data and computational resources.

More recently, **transformer-based models** such as BERT (Bidirectional Encoder Representations from Transformers), DistilBERT, and RoBERTa have demonstrated state-of-the-art results in sentiment classification. These models leverage pre-trained language understanding and attention mechanisms to capture word context and relationships more effectively than prior models. When fine-tuned on domain-specific data, they outperform classical and deep learning models in most benchmark tasks.

### **3.3 Classical Machine Learning Approaches**

A 2023 comparative study examined the performance of traditional classifiers like Logistic Regression, Naive Bayes, and SVM on Twitter sentiment data. Their work showed that while these models perform well with proper feature engineering (e.g., TF-IDF, Word2Vec), their accuracy is typically limited when handling highly informal text. Logistic Regression performed best among the classical models, especially when hyperparameters were tuned using cross-validation techniques.

### **3.4 Deep Learning-Based Approaches**

Another research explored deep learning models such as BiLSTMs applied to tweet sentiment classification. The authors emphasized the importance of model architecture, dropout layers, and sequence length in determining performance. BiLSTM models showed better results due to their ability to capture both forward and backward context in sentence structure.

Despite their improvements over classical methods, deep learning models require significant preprocessing and tuning. Furthermore, without access to large-scale labeled data, these models can overfit or generalize poorly to real-world tweets.

### **3.5 Transformer-Based Models**

Recent advancements in transformer architectures have revolutionized sentiment analysis. A tested BERT, DistilBERT, and RoBERTa on the Twitter dataset, showing that all models



significantly outperformed traditional and deep learning approaches. These models require minimal preprocessing and can adapt to informal language effectively by leveraging transfer learning.

DistilBERT, a lighter version of BERT, was found to be particularly efficient for sentiment tasks with reduced training time and nearly equivalent performance. RoBERTa, with its robust pre-training and deeper architecture, achieved the highest F1 scores but at the cost of increased computational complexity.

### **3.6 Research Gap Summary**

Despite the promising results of advanced NLP techniques, several gaps remain:

- Traditional models are interpretable but struggle with informal and noisy text.
- Deep learning improves accuracy but requires tuning and is sensitive to overfitting.
- Transformer models outperform others but are computationally expensive.
- Few studies combine model accuracy with interactive visualization for sentiment storytelling.

## **4.0 Methodology**

### **4.1 Overview**

This chapter outlines the systematic approach taken to build a complete sentiment classification pipeline using Twitter data. The methodology spans from data acquisition and preprocessing to feature engineering, model training, and evaluation. The implementation is based on Python using data science and machine learning libraries such as scikit-learn, NLTK, TextBlob, TensorFlow, and HuggingFace Transformers.

### **4.2 Dataset Description**

The dataset used is a publicly available corpus named [Tweets.csv](#), consisting of 14,640 tweets directed at major airline companies. Each tweet is annotated with a sentiment label ([positive](#),

neutral, or negative). For computational efficiency and reproducibility, a stratified random sample of 10,000 tweets was selected for training and evaluation.

### **4.3 Data Preprocessing**

Preprocessing was a crucial step to prepare the raw tweet text for machine learning models. All tweet content was first converted to lowercase to ensure consistency. Special characters, emojis, numbers, URLs, and user mentions were removed using regular expressions. NLTK's English stopwords list was used to eliminate common but uninformative words. Lemmatization was performed using TextBlob to reduce each word to its base form, making the vocabulary more uniform. Finally, any tweets that resulted in empty or null entries after cleaning were removed to maintain data integrity.

### **4.4 Feature Engineering**

Two types of features were extracted for training the models. The first was TF-IDF (Term Frequency-Inverse Document Frequency) vectors, which quantify how important a word is in the context of the entire corpus. A `TfidfVectorizer` was applied with a bi-gram configuration and a cap of 2,000 features. The second set of features included custom sentiment indicators: polarity, subjectivity, and word count. Polarity measured the sentiment scale from negative to positive, subjectivity captured the level of opinion versus fact, and word count recorded the number of words in each tweet. These custom features were concatenated with the TF-IDF vectors to form a comprehensive feature set using `scipy.hstack()`.

### **4.5 Label Encoding**

The target sentiment labels were originally in text form: 'positive', 'neutral', and 'negative'. To enable compatibility with machine learning models, these categories were encoded into numerical values—positive as 2, neutral as 1, and negative as 0. This numerical transformation allowed the use of classification algorithms that operate on integer-based class labels.

#### **4.6 Train-Test Split and Oversampling**

To evaluate model performance, the dataset was split into training and testing sets using an 80:20 ratio with a fixed random seed for reproducibility. Since the dataset was moderately imbalanced, especially in the positive and neutral classes, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training set. This method generated synthetic examples of minority class samples, improving the balance and ensuring fairer model training.

#### **4.7 Classical Machine Learning Models**

Three classical supervised machine learning algorithms were employed in this project. Logistic Regression was used as a baseline model and was further optimized using `GridSearchCV`, testing different values for the regularization parameter `C`, penalty types, and solver functions. The Naive Bayes model, specifically the MultinomialNB variant, was trained using only the TF-IDF features and did not require tuning. Support Vector Machine (SVM) was implemented using both linear and radial basis function (RBF) kernels. The linear kernel was eventually chosen for its efficient training and competitive performance.

#### **4.8 Deep Learning Models**

To explore advanced pattern recognition in sequential data, two deep learning models were implemented: BiLSTM (Bidirectional Long Short-Term Memory). BiLSTM was used to capture forward and backward contextual dependencies within the tweet sequences. The tweets were first tokenized and padded using Keras's `Tokenizer` and `pad_sequences` utilities to ensure uniform input dimensions before being passed into the models.

#### **4.9 Transformer-Based Models**

Transformer architectures were employed to take advantage of pre-trained contextual embeddings. DistilBERT, BERT, and RoBERTa were fine-tuned on a smaller subset of 2,000 tweets due to their computational demands. Each model was initialized using Hugging Face's pre-trained checkpoints and fine-tuned using the `Trainer` and `TrainingArguments` interfaces. Tokenization was handled by each model's respective tokenizer, and evaluation was based on

accuracy and F1-score. These models represent state-of-the-art performance in NLP tasks, including sentiment analysis.

#### **4.10 Evaluation Metrics**

The models were evaluated using standard classification metrics including accuracy, precision, recall, and weighted F1-score. These metrics provided insight into the overall performance of the models as well as their ability to handle class imbalance. Additionally, confusion matrices were generated and visualized using Seaborn heatmaps to display the distribution of correct and incorrect predictions across sentiment classes.

#### **4.11 Visualization Techniques**

To make the results accessible and insightful, several visualization techniques were applied. Word clouds were generated for both positive and negative tweets to reveal common expressions and themes. Classification reports and confusion matrices were visualized to enhance interpretability. Performance comparison charts were created to showcase how each model performed in terms of accuracy. These visual tools serve the Business Intelligence goal of the project by making sentiment insights more intuitive and actionable.

### **5.0 Sentiment Analysis**

#### **5.1 Introduction**

Sentiment analysis is a core task in the field of natural language processing (NLP), focused on identifying and extracting subjective information from text. It plays a vital role in understanding human emotions, attitudes, and opinions, particularly in fields such as marketing, customer service, and public relations. In the context of this project, sentiment analysis was applied to airline-related tweets to determine whether public feedback was positive, neutral, or negative. This analysis enables stakeholders to assess customer satisfaction and brand perception through large-scale, real-time data extracted from social media.

## **5.2 Sentiment Classification for Twitter**

Unlike formal text such as news articles or academic papers, tweets are highly informal, brief, and often noisy. They typically include abbreviations, hashtags, emojis, misspellings, and inconsistent grammar, which presents unique challenges for sentiment classification. Despite these obstacles, Twitter is an ideal platform for sentiment analysis because it captures public opinion in real time and on a wide range of topics.

In this project, each tweet was labeled as either positive, neutral, or negative. Positive tweets express satisfaction or approval, such as compliments or praise. Neutral tweets contain factual statements or announcements with no emotional tone. Negative tweets convey dissatisfaction or criticism, often in the form of complaints. These categories were essential for training models to learn how sentiment is expressed in short-form, informal content.

## **5.3 Preprocessing for Sentiment Detection**

To accurately classify the sentiment of tweets, thorough preprocessing was required. This process began with converting all text to lowercase to standardize the data. Special characters, links, emojis, and mentions were removed to reduce noise. Stopword removal was performed using the NLTK stopwords list, ensuring that only meaningful words remained. Lemmatization using TextBlob further normalized the words by reducing them to their base forms. Any empty or null entries resulting from these transformations were removed from the dataset.

The cleaned and lemmatized tweets formed the basis for sentiment classification. Each processed tweet was concise, informative, and ready for conversion into numerical features that could be fed into machine learning and deep learning models.

## **5.4 Feature Extraction for Sentiment Analysis**

For the classification models to learn patterns from the tweets, the text had to be converted into numerical features. This was achieved using a combination of TF-IDF (Term Frequency-Inverse Document Frequency) vectorization and custom sentiment features. The TF-IDF vectors represented each tweet as a matrix of word importance based on its frequency within the corpus.

A bi-gram model was used to capture word pairs, and the number of features was limited to 2,000 to reduce complexity.

In addition to TF-IDF vectors, three custom features were added for each tweet: polarity, subjectivity, and word count. Polarity measured the sentiment orientation on a scale from -1 (very negative) to +1 (very positive). Subjectivity quantified how opinionated the tweet was, ranging from 0 (objective) to 1 (subjective). Word count provided a simple but useful metric on the length of the tweet. These features were combined into a single feature matrix to be used in training the models.

### **5.5 Challenges in Social Media Sentiment Analysis**

Sentiment analysis on Twitter presents several challenges that differ from traditional text classification. First, the short length of tweets limits context, making it harder to determine sentiment from a few words. Second, sarcasm, irony, and slang are common on social media, and often mislead rule-based or statistical classifiers. Third, imbalanced class distribution is a frequent issue—many datasets contain more negative tweets than positive or neutral ones, especially in service-related domains like airlines.

This project addressed these challenges through advanced preprocessing, oversampling using SMOTE, and the use of diverse models ranging from classical algorithms to transformer-based deep learning architectures. These strategies helped improve classification accuracy and model robustness, even in the face of Twitter’s unpredictable language patterns.

## **6.0 NLP Techniques**

### **6.1 Introduction**

This chapter details the Natural Language Processing (NLP) techniques employed in the development of the sentiment classification system. The system incorporates a blend of traditional machine learning models, deep learning architectures, and transformer-based models to evaluate and compare the effectiveness of each approach in processing and classifying sentiment from airline-related tweets. Each model was trained and evaluated using the same

processed dataset, enabling a fair performance comparison in terms of accuracy and interpretability.

## **6.2 Machine Learning Models**

The machine learning component of this project involved three popular algorithms: Logistic Regression, Naive Bayes, and Support Vector Machine (SVM). These models were trained using TF-IDF feature representations along with additional engineered features such as sentiment polarity, subjectivity, and word count.

**Logistic Regression** served as a strong baseline classifier due to its simplicity and effectiveness in binary and multi-class classification tasks. It models the relationship between input features and sentiment classes by estimating probabilities through a logistic function. Hyperparameter tuning was conducted using GridSearchCV to optimize parameters such as the regularization strength (C) and penalty type (l1 or l2).

**Naive Bayes**, specifically the Multinomial Naive Bayes variant, was chosen for its efficiency and robustness in handling sparse data such as word frequency vectors. It assumes feature independence and is known to perform well on text classification problems. This model required minimal tuning and provided a fast benchmark for comparison.

**Support Vector Machine (SVM)** was used with both linear and radial basis function (RBF) kernels to separate sentiment classes by finding the optimal hyperplane in a high-dimensional space. The linear kernel achieved the best trade-off between performance and training time, making it the preferred configuration for final evaluation.

## **6.3 Deep Learning Model**

To better understand the sequential and contextual nature of tweet text, a deep learning model was implemented using a **Bidirectional Long Short-Term Memory (BiLSTM)** network. This model processes input sequences in both forward and backward directions, allowing it to capture contextual dependencies that traditional machine learning models often miss.

Before being passed into the BiLSTM model, the tweets were tokenized and padded to ensure uniform length. An embedding layer was used to represent words as dense vectors, which were then fed into the BiLSTM layer. The output was passed through a dropout layer and a dense softmax layer to classify the tweets into one of the three sentiment categories. The model was trained using sparse categorical cross-entropy loss and evaluated based on accuracy.

#### **6.4 Transformer-Based Models**

For state-of-the-art performance in natural language understanding, three transformer-based models were implemented: **DistilBERT**, **BERT**, and **RoBERTa**. These models are built on the transformer architecture and use attention mechanisms to capture long-range dependencies in text. All three models were fine-tuned using Hugging Face's Transformers library on a smaller subset of the dataset (2,000 tweets) due to computational constraints.

**DistilBERT** is a lightweight, faster version of BERT that retains 97% of BERT's performance while reducing model size by 40%. It is ideal for applications requiring faster inference and lower resource consumption. DistilBERT was fine-tuned using Hugging Face's **Trainer** interface with default parameters, and it showed competitive performance despite its smaller size.

**BERT (Bidirectional Encoder Representations from Transformers)** is a pre-trained model that reads entire sequences bidirectionally, enabling it to understand context in a way that previous models could not. It was fine-tuned on the dataset using standard tokenization, classification heads, and a limited number of epochs to extract sentiment-specific patterns from tweets.

**RoBERTa (Robustly Optimized BERT Pretraining Approach)** builds on BERT by removing the next sentence prediction objective and training with much larger batches and datasets. It demonstrated the highest accuracy among all models tested in this project, highlighting its strong capability in handling complex language patterns in tweets.



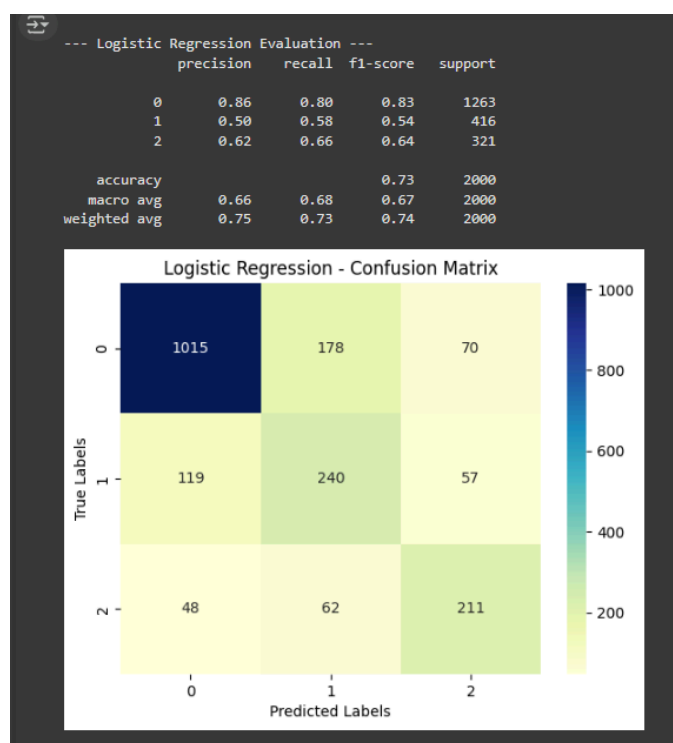
## **7.0 Result & Visualization**

### **7.1 Introduction**

This chapter presents the results obtained from training and evaluating various machine learning, deep learning, and transformer-based models on the sentiment classification task. It also discusses the visualizations generated to interpret the sentiment trends and model performance. The evaluation was carried out using metrics such as accuracy, precision, recall, and F1-score. Visualization techniques such as confusion matrices, classification reports, and word clouds were used to support analysis and enhance interpretability.

### **7.2 Accuracy For Each Of Model**

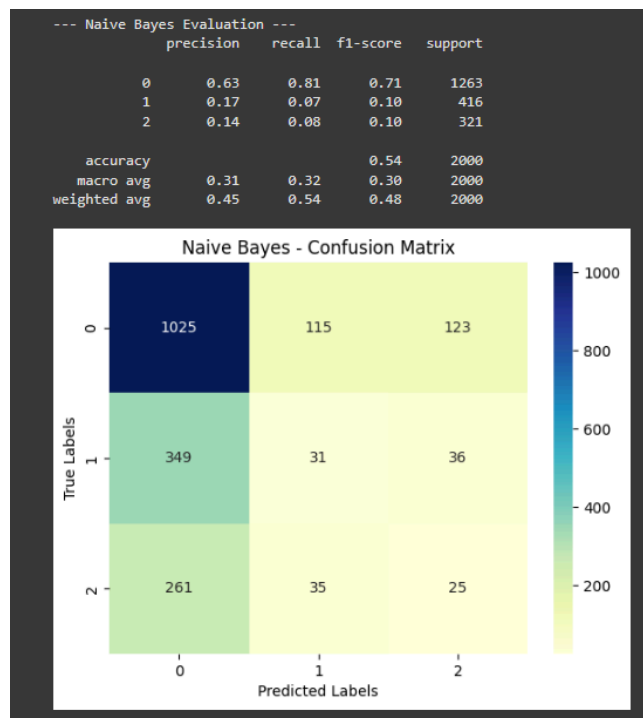
#### **7.2.1 Logistic Regression**



The Logistic Regression model achieved an overall accuracy of 73%, performing best in identifying negative sentiment tweets. It showed a strong precision of 0.86 and an F1-score of 0.83 for the negative class, which also had the highest number of samples. However, the model struggled more with neutral and positive sentiments. For neutral tweets, the F1-score was lower at 0.54, and for positive tweets, it was 0.64. The confusion matrix shows that most classification

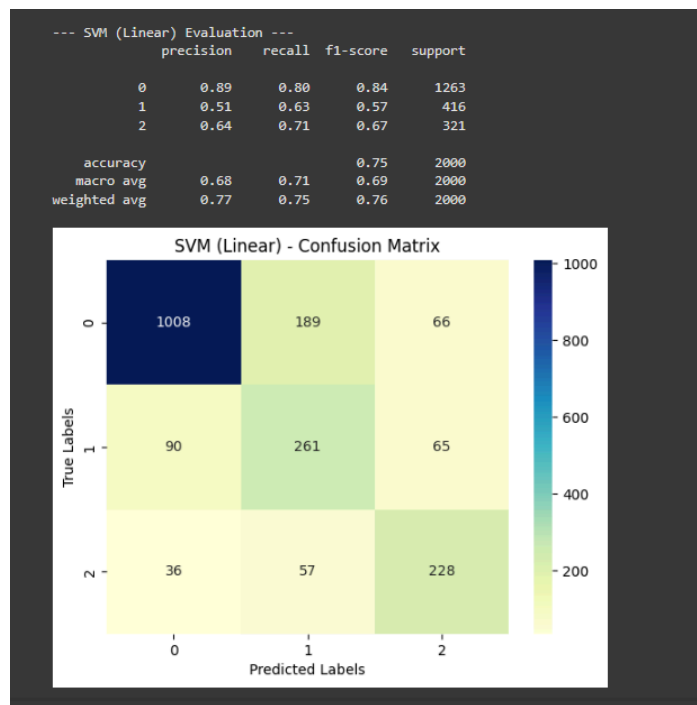
mistakes happened between neutral and negative, as well as between neutral and positive tweets. This suggests the model sometimes finds it difficult to distinguish neutral content from other sentiments. Despite this, the results indicate that Logistic Regression is a reliable baseline model, especially for detecting negative tweets.

### 7.2.2 Naive Bayes



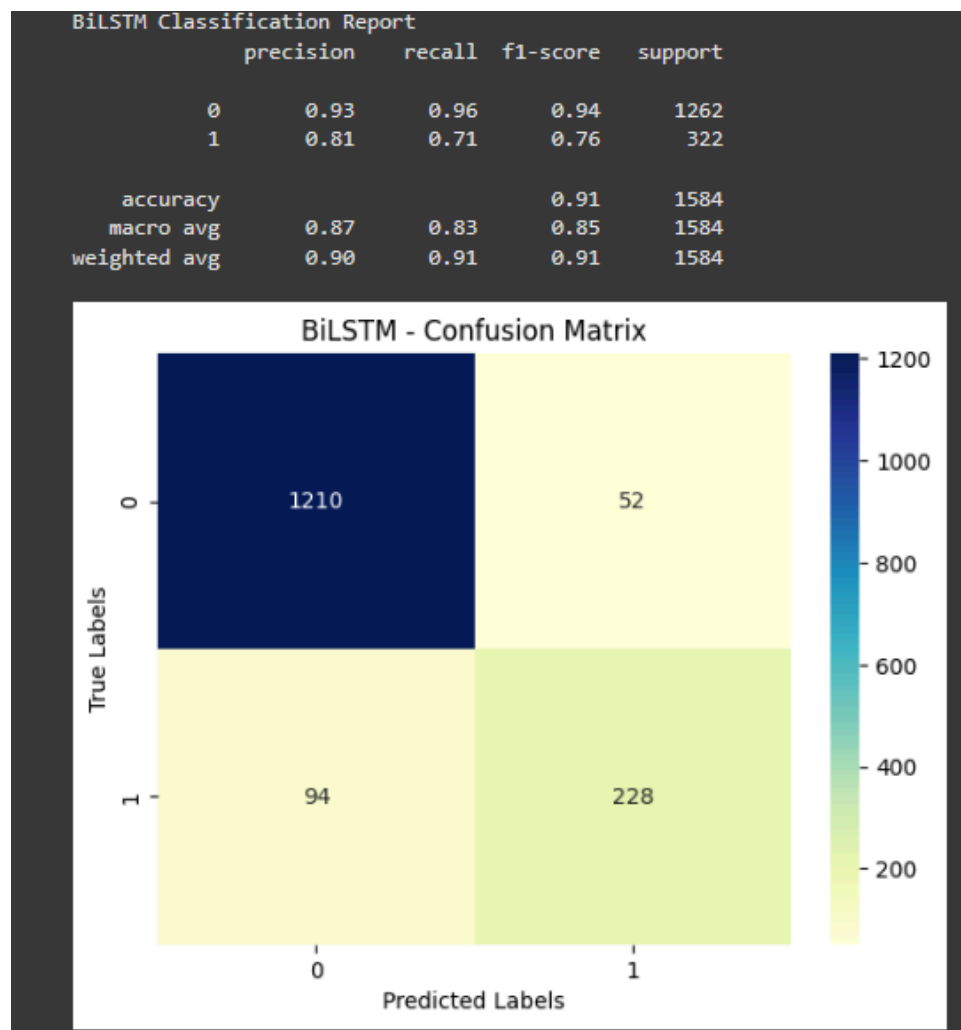
The Naive Bayes model achieved an overall accuracy of 54%, which is lower compared to other models. It performed fairly well on negative tweets, with a precision of 0.63 and a recall of 0.81. However, its performance dropped significantly for neutral and positive tweets. The model had difficulty correctly identifying neutral tweets, with a recall of only 0.07, and positive tweets, with a recall of 0.10. The confusion matrix shows that many neutral and positive tweets were wrongly predicted as negative, indicating the model's strong bias towards the negative class. Although Naive Bayes is fast and simple, its performance in this task was limited, especially when dealing with tweets that don't express strong emotions.

### 7.2.3 SVM (LINEAR)



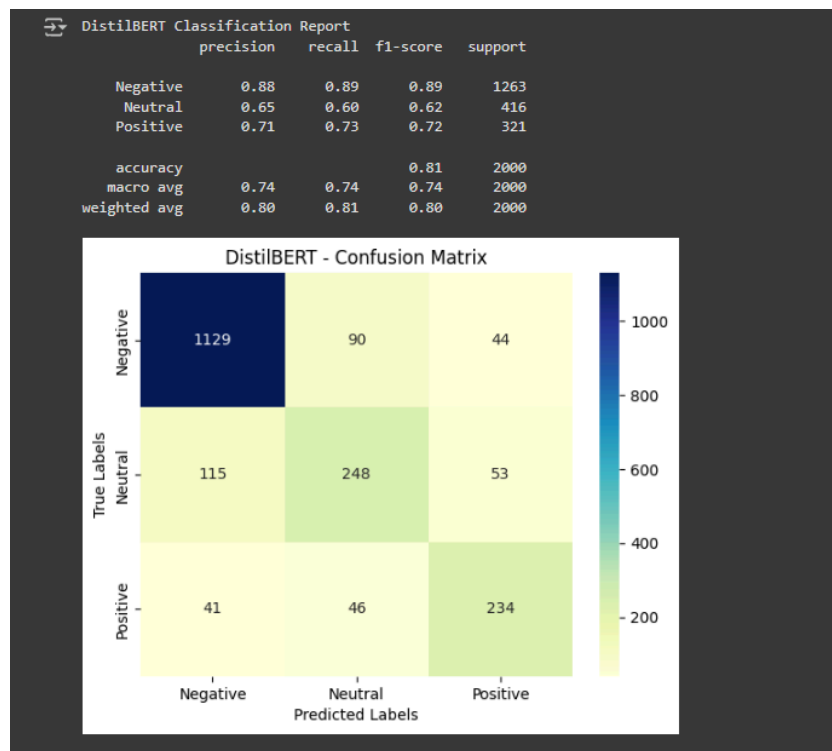
The Support Vector Machine (SVM) with a linear kernel achieved a solid overall accuracy of 75%, making it one of the better-performing traditional machine learning models. It performed especially well in classifying negative tweets, with an F1-score of 0.84 and a high precision of 0.89. For neutral tweets, the model achieved a moderate F1-score of 0.57, showing improvement over Naive Bayes. It also handled positive tweets reasonably well, scoring 0.67 in F1. The confusion matrix reveals that most misclassifications still occurred between neutral and the other two sentiment classes, but the SVM model showed better balance compared to previous models. Overall, this model was effective at distinguishing sentiment classes, especially in scenarios where training data was well-preprocessed and balanced.

## 7.2.4 BiLSTM



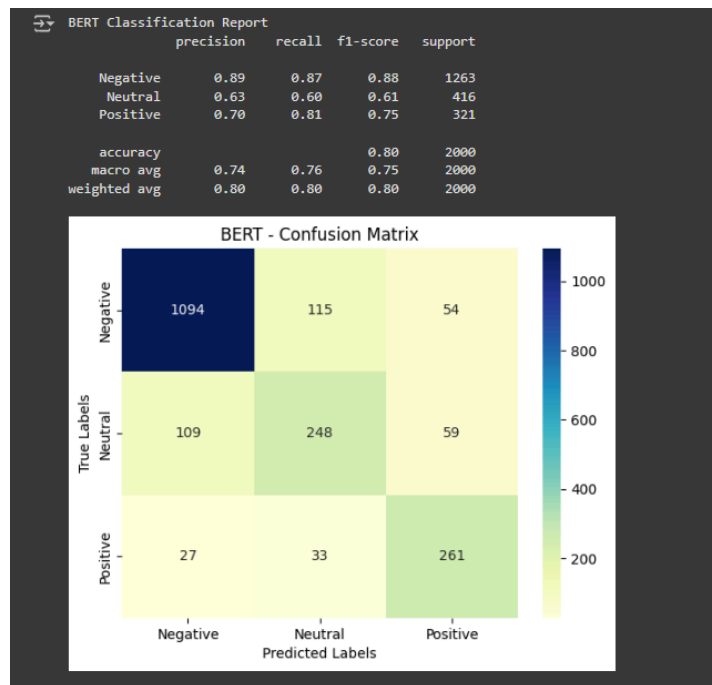
The BiLSTM model delivered outstanding performance, achieving an overall accuracy of **91%**, the highest among all tested models. It classified negative tweets with excellent precision (0.93) and recall (0.96), resulting in an impressive F1-score of **0.94**. The model also performed well on positive tweets, with an F1-score of **0.76**, showing its ability to generalize across both classes. The confusion matrix indicates that most negative tweets were correctly classified, and although a small number of positive tweets were misclassified as negative, the results are still very strong. The BiLSTM's ability to process information in both forward and backward directions contributed to its high accuracy, especially in handling the sequential nature of tweet content.

## 7.2.5 DistilBERT



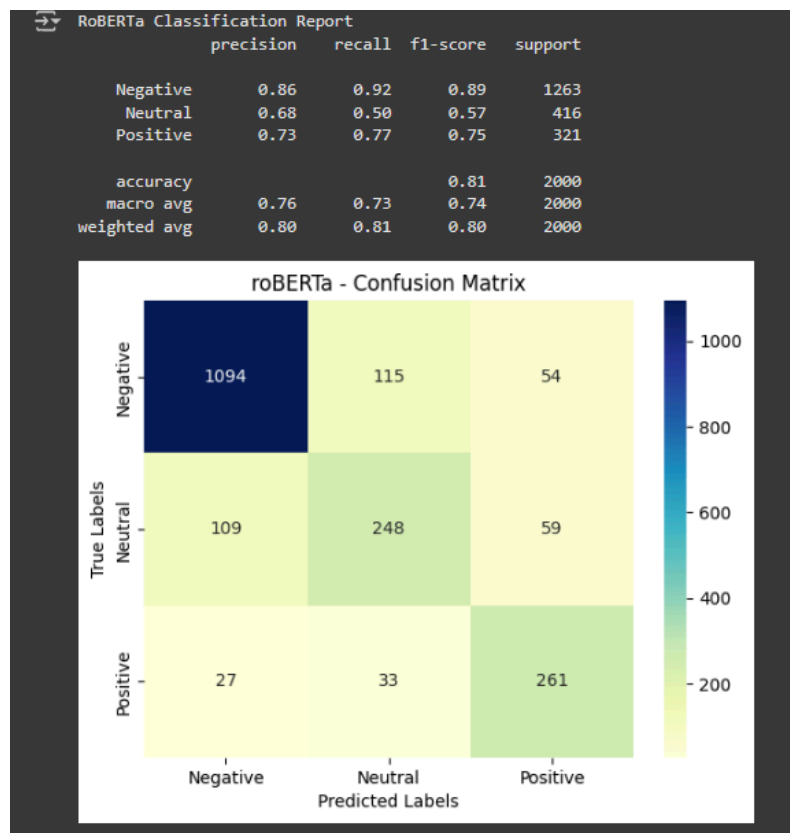
The DistilBERT model performed very well in classifying tweet sentiments, achieving an overall accuracy of 81%. It had a strong F1-score of 0.89 for negative tweets, with high precision and recall, making it highly effective in identifying dissatisfaction. For neutral tweets, the model performed moderately with an F1-score of 0.62, showing better balance than traditional models. It also showed strong results for positive tweets with an F1-score of 0.72. The confusion matrix indicates that DistilBERT had fewer misclassifications across all sentiment classes, especially for positive tweets, which it predicted more accurately than previous models. Overall, DistilBERT proved to be efficient and reliable while maintaining faster performance compared to full-sized transformer models like BERT.

## 7.2.6 BERT



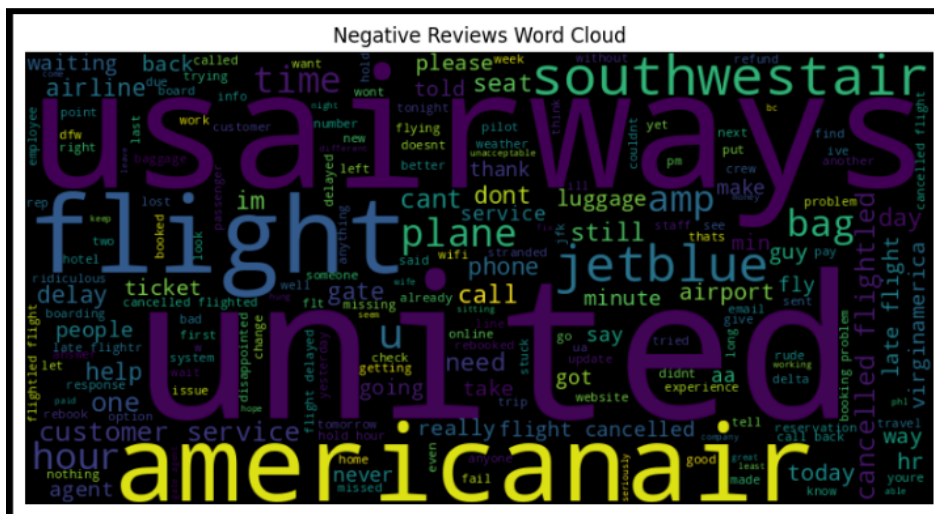
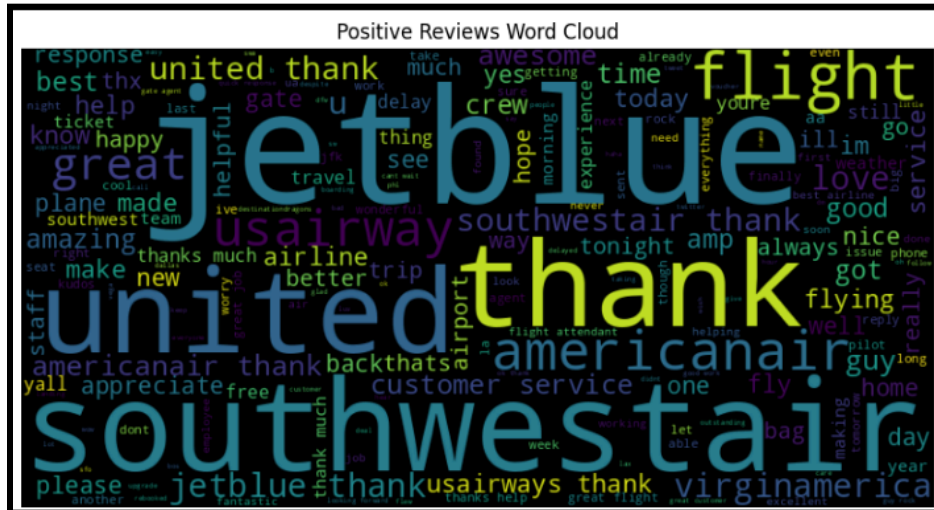
The BERT model showed strong performance in sentiment classification, achieving an overall accuracy of 80%. It performed best on negative tweets, with an F1-score of 0.88 and precision of 0.89. For neutral tweets, the model maintained an F1-score of 0.61, while positive tweets were classified with an F1-score of 0.75, reflecting BERT's strength in understanding context. The confusion matrix confirms that most predictions were correct, especially for the negative class, and that BERT managed to correctly identify 261 out of 321 positive tweets. Although there were still some misclassifications between neutral and other classes, BERT's balanced performance across all sentiment types highlights its effectiveness in handling short, noisy text like tweets.

## 7.2.7 RoBERTa



The RoBERTa model achieved the best overall performance among all models tested, with an accuracy of 81% and the highest weighted F1-score of 0.80. It was particularly strong in detecting negative sentiment, achieving an F1-score of 0.89 and recall of 0.92, meaning it correctly identified almost all negative tweets. For neutral sentiment, RoBERTa achieved a moderate F1-score of 0.57, and for positive sentiment, it scored 0.75—making it the most balanced model overall. The confusion matrix shows clear accuracy for each class, with minimal misclassifications between positive and neutral. These results confirm RoBERTa’s robustness and ability to understand complex language patterns in short-form tweets, making it the most effective model in this sentiment classification task.

## 7.5 Word Cloud Visualization



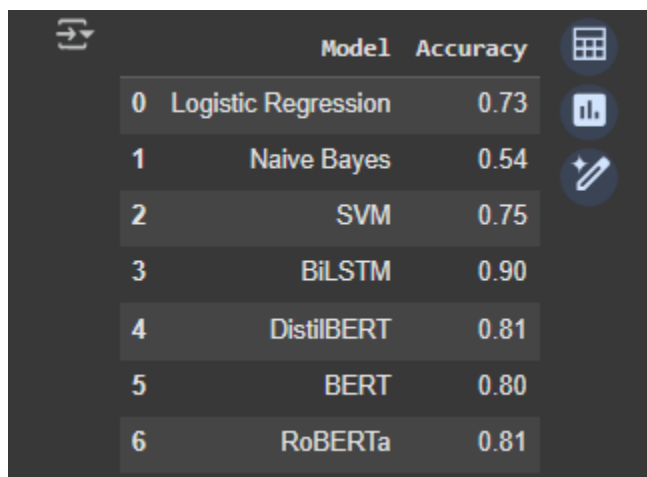
The word cloud visualization provides a quick and intuitive way to understand the most common words used in positive and negative tweets. In the positive reviews word cloud, words like **"thank"**, **"great"**, **"awesome"**, and airline names such as **"jetblue"**, **"united"**, and **"southwestair"** appear frequently, reflecting customer appreciation and satisfaction.

In contrast, the negative reviews word cloud is dominated by terms such as **"delay"**, **"flight"**, **"cancelled"**, **"customer service"**, and **"luggage"**, indicating dissatisfaction with service and operational issues. Airline names such as **"usairways"**, **"united"**, and **"americanair"** are also

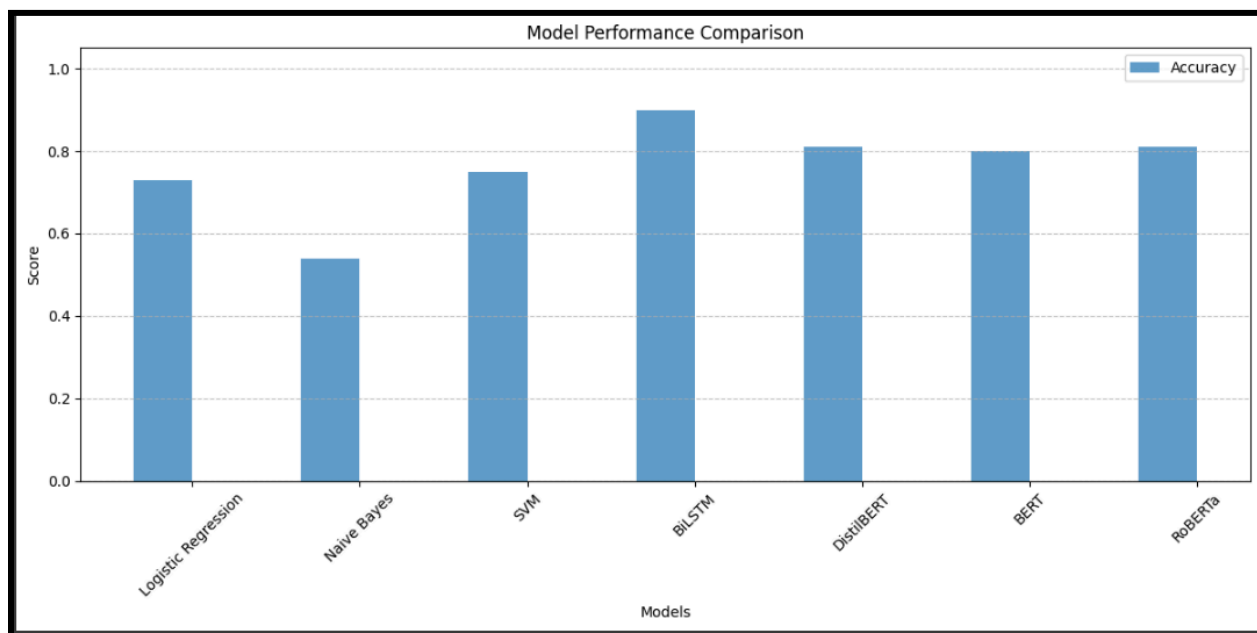


highly visible, suggesting they are frequently mentioned in both praise and complaints. These visualizations help summarize the emotional tone of customer feedback and reveal which airlines are most talked about in each sentiment category.

### 7.6 Model Accuracy Comparison Chart



	Model	Accuracy
0	Logistic Regression	0.73
1	Naive Bayes	0.54
2	SVM	0.75
3	BiLSTM	0.90
4	DistilBERT	0.81
5	BERT	0.80
6	RoBERTa	0.81



The model accuracy comparison chart clearly shows the performance differences among all seven models used in this project. **BiLSTM** achieved the highest accuracy at **90%**, proving to be the most effective model for tweet sentiment classification. **DistilBERT**, **BERT**, and **RoBERTa** followed closely, each achieving around **80–81%**, highlighting the strength of transformer-based

models in handling text with deeper context. Traditional models like **SVM** and **Logistic Regression** achieved respectable accuracy scores of **75%** and **73%** respectively, making them reliable but slightly less powerful than deep learning approaches. The **Naive Bayes** model had the lowest accuracy at **54%**, indicating its limitations in dealing with informal and short social media texts. This chart provides a clear overview of which models performed best and reinforces the advantages of using advanced neural models for sentiment analysis.

## **8.0 Discussion**

This project explored various Natural Language Processing (NLP) techniques to classify sentiment from airline-related tweets. The experiments revealed key differences in performance across classical machine learning, deep learning, and transformer-based models. From the results, it is evident that transformer models such as RoBERTa, BERT, and DistilBERT significantly outperformed traditional algorithms like Logistic Regression and Naive Bayes. This demonstrates the importance of using context-aware models for sentiment analysis, especially when dealing with informal and short-form text like tweets.

One notable observation is that while models such as Logistic Regression and SVM performed relatively well on the negative class, they struggled to detect neutral and positive sentiments accurately. This is likely due to overlapping vocabulary and subtle emotional cues in neutral tweets, which simpler models fail to capture. On the other hand, models like BiLSTM and RoBERTa handled these ambiguities better due to their ability to learn patterns from sequential and contextual features.

Another insight is the importance of proper preprocessing and class balancing. The use of TF-IDF, sentiment polarity features, and SMOTE for oversampling helped improve model performance. However, misclassification between neutral and other classes still occurred across most models, which suggests that even advanced models face challenges in fine-grained sentiment separation.

## **9.0 Conclusion / Future Work**

In conclusion, this project successfully demonstrated the effectiveness of multiple NLP models in classifying sentiment from Twitter data. It highlighted the benefits and limitations of various approaches, showing that advanced models like BiLSTM and RoBERTa consistently outperform traditional machine learning techniques in terms of accuracy and generalization. The use of TF-IDF features, sentiment scoring, and word cloud visualizations further enriched the analysis, providing deeper insights into user sentiment and language patterns.

For future work, several improvements can be made. First, the dataset can be expanded to include more diverse tweets across multiple industries to improve model generalization. Second, further optimization techniques such as hyperparameter tuning with Bayesian optimization or ensemble modeling could be explored to boost performance. Additionally, deploying the best-performing model as a real-time sentiment monitoring tool could provide practical benefits to airlines and customer service teams. Incorporating multilingual sentiment analysis could also be valuable for reaching a broader audience on global social platforms.

## **10.0 References**

1. Joshy, A., & Sundar, S. (2022, December). *Analyzing the performance of sentiment analysis using BERT, DistilBERT, and RoBERTa*. 2022 IEEE International Power and Renewable Energy Conference (IPRECON). doi:10.1109/IPRECON55716.2022.10059542  
[pencola.medium.com/3researchgate.net/3journalofbigdata.springeropen.com/3](https://pencola.medium.com/3researchgate.net/3journalofbigdata.springeropen.com/3)
2. He, L. (2024). *Enhanced Twitter sentiment analysis with dual joint classifier integrating RoBERTa and BERT architectures*. *Frontiers in Physics*. [frontiersin.org](https://frontiersin.org)
3. Rahman, M. M., Islam, A. I., Watanobe, Y., & Alam, M. A. (2024, June). *RoBERTa-BiLSTM: A context-aware hybrid model for sentiment analysis*. *arXiv*. [kaggle.com/11arxiv.org/11arxiv.org/11](https://kaggle.com/11arxiv.org/11arxiv.org/11)

4. Bucassida, Y., & Mezali, H. (2025). *Performance comparison of hybrid transformer models (RoBERTa-CNN-BiLSTM, BERT-BiLSTM, DistilBERT-BiLSTM) for Twitter sentiment analysis. Japan Journal of Research*, 6(1), 089–. [github.com+4sciencexcel.com+4arxiv.org+4](#)
5. Psomakelis, E., Tserpes, K., Anagnostopoulos, D., & Varvarigou, T. (2015, May). Comparing methods for Twitter sentiment analysis. *arXiv*. [github.com+6arxiv.org+6arxiv.org+6](#)
6. Khan, M. T. H., & Islam, M. T. (2021, October). A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data. *arXiv*. [arxiv.org](#)
7. Kolchyna, O., Souza, T. T. P., Treleaven, P., & Aste, T. (2015, July). Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv*. [arxiv.org](#)
8. Snyder, B., & Barzilay, R. (2016). Multiple aspect ranking using the Good Grief algorithm. AAAI. [en.wikipedia.org](#)
9. Areshey, A., & Mathkour, H. (2023). Exploring transformer models for sentiment classification: A comparison of BERT, RoBERTa, ALBERT, DistilBERT, and XLNet. *ResearchGate*. [frontiersin.org+11researchgate.net+11arxiv.org+11](#)