ENHANCING SENTIMENT ANALYSIS OF AIRLINE TWEETS THROUGH HYBRID MACHINE LEARNING MODELS

Vinay Vaida
vvaida@albany.edu

***Abstract*—In the world of tech presence of social media is growing, airlines are increasingly using Twitter to measure customer sentiment and improve their services. However, the informal language, slang, and brevity used in tweets make it hard to analyze the emotions expressed. In this study, we are going to look at how hybrid machine learning models can help improve sentiment analysis on airline tweets. Logistic Regression, Support Vector Machines (SVM), and Random Forest are used to assess the performance of individual models. Each model has its merits and demerits revealing that depending on a single algorithm cannot be relied upon. To address these limitations, we consider two hybrid models: Voting Classifier and Stacking Ensemble Model. The Voting Classifier unites several model's forecasts to exploit their individual strengths to obtain more impressive predictions. The Stacking Ensemble Model goes further than Voting Classifier by training the individual model's predictions. Our extensive experiments show that the hybrid models perform significantly better than their individual models. These findings indicate that hybrid models have potential for considerably improving accuracy in sentiment analysis for airline tweets thus providing valuable insights for airlines seeking to enhance customer satisfaction and service quality.**

***Index Terms*—Customer sentiment analysis, SVM, Data Augmentation, NLP, Airline tweets, Sentiment classifier, TF-IDF, Tkinter.**

I. INTRODUCTION

In the world driven by data today, businesses in all industries must understand customer sentiment. The airline industry particularly is faced with a dynamic and competitive environment where customer satisfaction is the key. This research project presents a new method of improving airline tweets' sentiment analysis using a hybrid machine learning model. Leveraging the rich "Twitter US Airline Sentiment" dataset to develop a strong and accurate system for categorizing tweets into three classifications – positive, negative, and neutral sentiments. By utilizing a combination of different machine learning algorithms, this unique approach surpasses conventional methods used in sentiment analysis. The dataset employed in this research is an archive that has captured opinions of passengers regarding various features of airline services like booking experiences, in-flight features, and customer service among others. By carefully categorizing the sentiments expressed within these tweets, our hybrid model will equip airlines with invaluable data-driven insights. One of the best models for analyzing customer's sentiments about US airlines is integrated into a GUI with Tkinter.

A. Problem Statement

Current sentiment analysis models have provided valuable insights. The issue is that they are not very effective with tweets about airlines. This platform poses a lot of difficulties because of its peculiar language, complex emotions of social media users and intrinsic ambiguity. Consequently, these limitations result in misclassification, which restrict airline's ability to comprehend customer sentiment accurately and perform data-driven decision-making. Contrarily, if airlines adopt hybrid machine learning specifically for airline tweets, these barriers can be broken down thus a deeper knowledge of customer emotions helps improve customer experience, strategic decision making and ultimately becoming competitive in their industry by building a stronger model.

B. Project Background

This paper makes use of tweets for airlines of size 3.34MB mainly for analyzing the sentiments of the user. The twitter dataset present in Kaggle is a subset of twitter data uploaded for analyzing the sentimental tweets. Since the data was imbalanced we augmented the data and after data augmentation the total size of data is 5.02MB. As a part of this research, Tweets for Airlines whose Airline sentiment confidence is more than 60 percent is considered for the sentimental analysis.The project will analyze the tweets posted by the people, perform exploratory data analysis, data-preprocessing, and prepare data for models which will help classify tweets as positive or negative.Finally, Airlines can work on their business area to improve it based on the type of the tweet received.

II. RELATED WORK

Ensemble methods are those that train many learners to solve a single problem. Unlike classical learning techniques,

which construct just one learner from the training data, ensemble methods construct a set of learners and then combine them [5].This study aimed at giving the airline industry a holistic understanding of their customer's sentiments to meet all their needs. Two hundred tweets from Emirates and Jet Airways were taken for analysis while more than 1000 tweets were used to determine the recent unfortunate event involving United Airlines[1]. People employ informal words or local slang in reviews which are absent in lexicons. Consequently, researchers emphasize the utilization of other approaches to sentiment analysis in various texts.The results show that ensemble techniques outperform non-ensemble classifiers with respect to accuracy. More experiments confirmed that the use of TF-IDF as feature extraction for machine learning models gives better performance [2].Real-time feedback can assist consumers in making the best travel decision and enable airline management and personnel to assess and act quickly in enhancing services for passengers. Researchers developed a hybrid model known as HMRFLR that recommends that airlines in the United States utilize machine learning techniques (such as random forest classifier and logistic regression) to analyze their tweets [3]. Sentiment classification uses a two-step process. First, the lexicon approach scores positive, negative, and neutral opinions. The second stage involves tweets that are of a low polarity being ignored by the support vector machine (SVM) classifier in this two-step method [4].

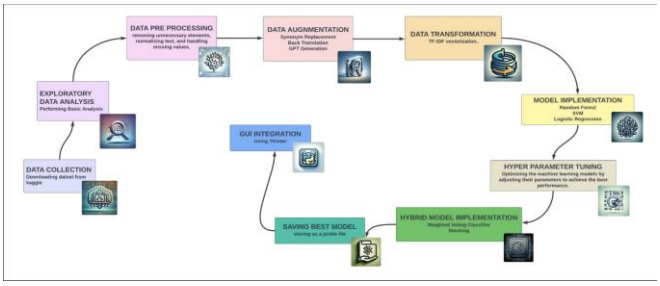


Fig. 1. Architecture of Airline Tweets Sentiment Analyzer

III. PROPOSED METHOD

Figure 1 shows the proposed architecture for Airline Tweets Sentiment Analyzer. Once EDA is performed on the Airline Tweets data, irrelevant features are removed and three NLP data augment techniques are performed to handle the imbalanced dataset. Firstly, the top tweets with the airline sentiment confidence more than 60 percent and text normalization techniques are applied to reduce the randomness and improve computation efficiency.Secondly, after data cleaning the cleaned texts are transformed into vectors by TF-IDF.Finally different classification models are implemented and the best performer after cross validation and hypertuning is deployed into GUI.The deployed model is used to find sentiment of customers traveled in a particular airline and provides output as positive, negative or neutral based on the text added in the message field.

A. Process Model

CRISP-DM is a comprehensive model, which serves as a systematic process to guide data driven initiatives. It involves six stages such as Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. We have used the CRISP-DM model as illustrated in Figure 2.The process begins with setting out business objectives, understanding the information, and progressing towards the best performing solution by utilization of modeling techniques and model evaluation. A deployment phase is important as it makes sure that a chosen model is incorporated in the operational systems.Using this approach, the team can plan each phase's duration and details. All six phases and the things covered in each respective phase are described in the following.

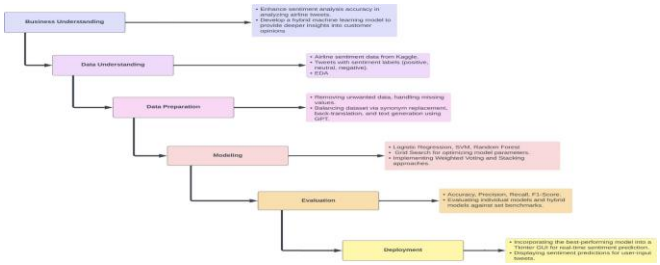


Fig. 2. Six Phases of CRISP DM methodology

1) Business Understanding: Business Understanding is the first phase of CRISP-DM methodology, it includes the project team's understanding of the business objectives, business needs and the problem. The team creates project goals, assesses resources, and establishes success criteria. Team has worked on finding appropriate research papers to understand the business requirements that support data-driven solutions aligned with project goals. The problem statement was clearly understood by the team so that each team member can proceed with subsequent phases which brings about effective data analysis and solution development within the scope of the project.

2) Data Understanding : This is the second phase, where data is gathered.It is important to gather the data for the project from a genuine and authenticated source. It helps in more accurate solutions and can be helpful for real-time applications. The team collected the data from a well-known source "Kaggle." The dataset is available on Kaggle, provided by an authenticated source. After collecting the dataset, we performed Exploratory data analysis. This step allows us to see the quality of the dataset. Once everything is done, including checking for nulls and the other data quality measures. The team planned further phases accordingly.

3) Data Preparation: It is the third step of CRISP-DM, where a team sets the final data set and uses that data set for modeling. It transforms raw data into a usable format for analysis. The team performed data cleansing steps like checking missing

values, removing errors, inconsistencies, and irrelevant information from the data. Utilized Feature selection method to select important features for achieving the project goal.

4) Data Modeling: In the Data Modeling phase of CRISP-DM, which is about analyzing sentiments in airline tweets, this project utilizes Support Vector Machines (SVM), Random Forest and Logistic Regression algorithms. SVM is a powerful supervised learning method whose purpose is to classify tweets on the basis of their feelings by finding an optimal hyperplane that separates various sentiment classes. Random Forest aggregates predictions from many trees, thereby giving robustness and accuracy as it uses an ensemble of decision trees. Logistic Regression is a statistical method, models the probability of sentiment classes given tweet features. Each algorithm is trained on labeled tweet data and fine-tuned by using GridSearchCV to achieve the best model performance.

5) Model Evaluation: The phase involved the comparison of each model's performance metrics with the others. The team executes models on test sets other than training ones, and their effectiveness is gauged through evaluation metrics such as accuracy, precision, recall, and F1-score in sentiment classification. The team implemented all models with the goal of establishing the greatest accuracy and improving it so that a very precise proposed system could be constructed. Among all models tested, the Stacking Ensemble model outperformed all other models. The Stacking Ensemble model achieved an impressive 91.15 percent accuracy, which is required for a successful solution.

6) Deployment: The data deployment phase, as it pertains to this framework, is the final stage where a data-driven solution or model is implemented and integrated within the operational environment. It involves the transformation of the deployed solution into a working version that can be used by end-users. After selecting the best model, the team has developed a GUI based on Tkinter so that users would easily interact with the model through providing input about an airline's opinion, and see sentiment predicted (Negative, Neutral, Positive).

IV. DATA PREPARATION

This phase of CRISP-DM involves performing Exploratory Data Analysis , data pre-processing, data transformation and preparing data for the model. In this project, it is one of the most essential step as well as time-consuming as it involves cleaning of the dataset, merging datasets, filtering of the data, removal of unwanted rows and columns, performing tokenization and lemmatization, stop word removal , converting textual data into vector and splitting data into testing and training. Below Figure 3 represents how data flows from gathering raw data to preparing data for the models.

A. Data Exploration

Initially we downloaded a dataset from Kaggle.The EDA was performed on the Twitter CSV file separately. The kaggle dataset includes all the airline tweets related details such as tweet id,airline sentiment confidence,airline,retweet count,

Fig. 3. Data Preparation Flowchart

text, negative reason confidence,name,tweet created, tweet location,user timezone etc.

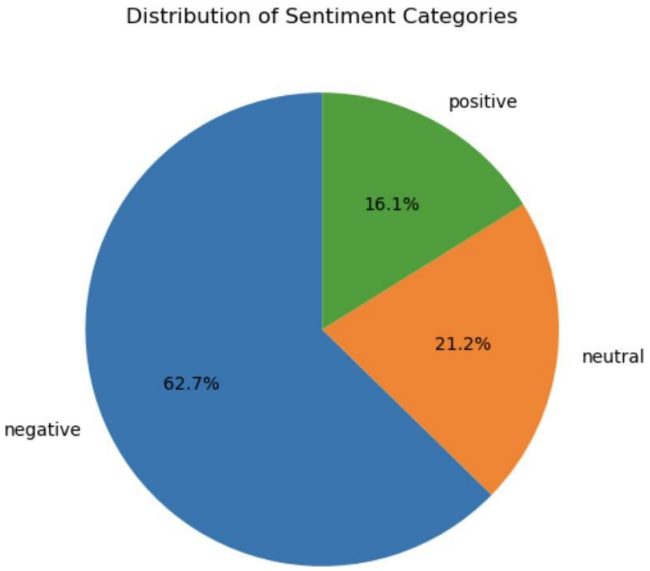


Fig. 4. Pie Chart

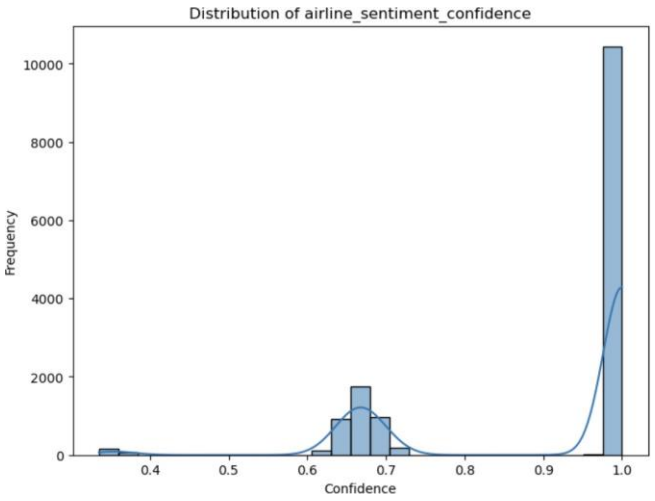


Fig. 5. Histogram showing distribution for Airline Sentiment Confidence

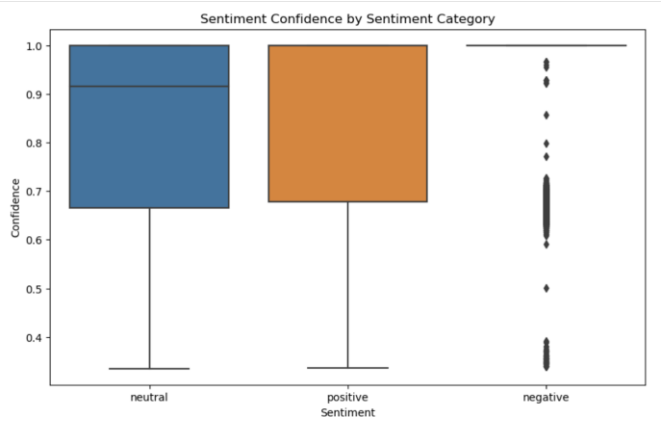


Fig. 6. Box Plot showing distribution for Airline Sentiment Confidence

After exploring the EDA, we got insights that the data is imbalanced. We have more negative tweets than positive and neutral tweets.To overcome data imbalance issue we augmented the data using three NLP data augmentation techniques.We found only about 200 tweets with airline confidence below 60 percent, indicating mostly clear sentiment expressions in the dataset.We could also see significant sentiment imbalance, with over 60 percent of tweets negatively oriented, reflecting prevalent customer dissatisfaction.

Data Augmentation: In the realm of sentiment analysis, especially when dealing with datasets extracted from social media platforms like Twitter, one often encounters the challenge of data imbalance. This issue can significantly skew the performance of machine learning models, leading to biases towards the majority class. In our case, after conducting an exploratory data analysis (EDA) on a dataset of tweets related to airline sentiment, a notable imbalance among the sentiment classes was observed as the dataset was having more negative sentiment data . To address this, various data augmentation techniques were employed to generate more data for the neutral and positive class.

Techniques Employed for Data Augmentation

Synonym Replacement:This technique involves replacing words in a sentence with their synonyms to generate a new sentence with the same meaning but different wording. This is achieved by leveraging the WordNet lexical database to find synonyms. The process involves selecting a word from the original sentence, finding its synonyms, and then randomly replacing the original word with one of its synonyms. This method is particularly useful in retaining the original sentiment of the text while introducing lexical diversity.

Back Translation: Back translation is a two-step process where a sentence is first translated from the source language (English, in this case) to a target language (such as Spanish) and then translated back to the source language. This method often introduces slight variations in phrasing and word choice, thus creating a paraphrased version of the original text. It's an effective way to augment data while maintaining the core meaning and sentiment of the text.

Text Generation Using GPT: Generative Pretrained Transformer (GPT) models can be used to generate new text samples based on a seed text. This involves providing the model with a piece of text (the seed), from which it generates a continuation. This method can introduce new and diverse textual content that shares similarities with the original dataset, helping in balancing the classes.

Final Dataset Composition After applying these augmentation techniques, the final dataset is stored and achieved a more balanced distribution of sentiment classes. The dataset, originally skewed towards negative sentiment, was transformed into a balanced one with the following composition:

Negative Sentiments: 9,113 samples Positive Sentiments: 8,929 samples Neutral Sentiments: 8,898 samples

This balanced dataset now contains a total of 26,940 tweets. It's noteworthy that tweets with an airline confidence score lower than 0.6 were excluded, ensuring the quality and relevance of the data.

	airline_sentiment	text
0	positive	@SouthwestAir the new logo is going to look am...
1	negative	@SouthwestAir it keeps saying that mobile boar...
2	negative	@united I'd really like to get off of this pla...
3	negative	@AmericanAir 4369. about 4 flights all using s...
4	positive	@JetBlue finally taking off! Las-Fll-Sju #letsgo
...	...	...
27173	neutral	go_out you in ATL! "@SouthwestAir: Congrats to...
27174	positive	@USAirways please thank Mellie at CAE, Tammy i...
27175	positive	@JetBlue Haha. Thanks. You blackguard are grea...
27176	positive	@Americanair Aww thanks AA..dfw I was in Gmail...
27177	neutral	@JetBlue Boston gate C12

26940 rows × 3 columns

Fig. 7. Final data

B. Data Preprocessing

In this project, we choose the kaggle twitter data that gives information about tweets about 7 US airlines for data-preparation and modeling purposes. The kaggle dataset has 14640 rows and 15 columns.Firstly cleaned the data, by addressing missing values, removing irrelevant features, and standardized text data. After this we have transformed the data using Data Augmentation.

Post-augmentation, our dataset comprising airline-related tweets is rich and diverse. However, to ensure its effectiveness for sentiment analysis, preprocessing is a crucial step. This process refines the text data, making it more suitable for machine learning algorithms to interpret and analyze. Preprocessing Steps Implemented:

Lowercasing The initial step involves converting all text to lowercase using python's builtin string function. This unifor-

mity ensures that the algorithm treats words like ”Plane” and ”plane” identically, avoiding unnecessary complexity.

Removing Punctuation and Numbers We strip the text of any punctuation and numbers, as these elements typically don’t contribute to sentiment analysis. This is done using regular expressions, which selectively remove characters that are not letters.

Tokenization The text is then broken down into individual words or tokens using NLTK library. This process is crucial for analyzing the text at the word level.

Removing Stopwords Common words such as ’the’, ’is’, and ’in’, known as stopwords, are removed using the NLTK library. These words are usually irrelevant for sentiment analysis and their removal helps in focusing on the more meaningful words.

Lemmatization Finally, words are lemmatized, which means they are reduced to their base or dictionary form. For instance, ’running’ becomes ’run’. This step is important to treat different forms of the same word equally and has been implemented using NLTK library..

	airline_sentiment	text	cleaned_text
0	positive	@SouthwestAir the new logo is going to look am...	southwestair new logo going look amazing airpl...
1	negative	@SouthwestAir it keeps saying that mobile boar...	southwestair keep saying mobile boarding pass ...
2	negative	@united I'd really like to get off of this pla...	united id really like get plane
3	negative	@AmericanAir 4369. about 4 flights all using s...	americanair flight using gate wbuses plane awf...
4	positive	@JetBlue finally taking off! Las-Fll-Sju #letsgo	jetblue finally taking lasfllsju letsgo
...	...	...	...
27173	neutral	go_out you in ATL! “@SouthwestAir: Congrats to...	goout atl southwestair congrats destinationdra...
27174	positive	@USAirways please thank Mellie at CAE, Tammy i...	usairways please thank mellie cae tammy baggag...
27175	positive	@JetBlue Haha. Thanks. You blackguard are grea...	jetblue haha thanks blackguard great unlike ny...
27176	positive	@Americanair Aww thanks AA..dfw I was in Gmail...	americanair aww thanks aadfw gmail understand ...
27177	neutral	@JetBlue Boston gate C12	jetblue boston gate c

26940 rows × 3 columns

Fig. 8. Finalized dataset with the cleantext text column that we are using to extract features and implement modeling

C. Data Transformation

Vectorization in NLP is the process that involves converting textual data into vectors. In this project, TF-IDF is selected to transform tweets data, since it considers both word frequency and importance. It is available in the sklearn.feature extraction.text module of Python. It is calculated by both TF(Term Frequency) and IDF(Inverse Document Frequency). The algorithm is expressed by the following equations:

Term Frequency (TF): where ft,d is the raw count of a term in the document d, and t’dft’,j refers to the total amount of terms in the document d.

We measure TF-IDF scores using the following formula:

$$tf(t, d) = \frac{f_d(t)}{\max_{w \in d} f_d(w)}$$

Inverse Document Frequency (IDF): where N is the total number of documents in D and —dD:td— is the number of documents where t appears.

$$idf(t, D) = \ln \frac{|D|}{|\{d \in D : t \in d\}|}$$

C. TF-IDF Score:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

$$tfidf'(t, d, D) = \frac{idf(t, D)}{|D| + tfidf(t, d, D)}$$

where $f_d(t)$ is the frequency of term t in document d, and D is the corpus of documents.

For our dataset, we employed scikit-learn’s TfidfVectorizer, setting a feature limit of 5000 words. This limitation not only focuses our analysis on the most relevant words but also manages computational resources more effectively. The vectorization process involves the use of the fit transform function, which serves a dual purpose. Firstly, it learns the vocabulary of our dataset and secondly, it calculates the TF-IDF scores for each word. The resulting output is a matrix where each row corresponds to an individual tweet and each column to one of the top 5000 words, with the matrix values being the TF-IDF scores that quantify the significance of each word.Incorporating TF-IDF vectorization into our data processing pipeline offers significant advantages for sentiment analysis. It reduces the impact of common words.

```
In [31]: print(X)

  (0, 2501)     0.20955601492821382
  (0, 4462)     0.20264470289880315
  (0, 790)      0.31819823078717036
  (0, 4089)     0.3577992355343937
  (0, 3868)     0.29922422529743026
  (0, 2787)     0.30839434793817766
  (0, 1703)     0.17958681505645832
  (0, 3127)     0.23109305121303772
  (0, 4843)     0.1879842750430475
  (0, 1695)     0.17591491121766725
  (0, 112)      0.2638431486016245
  (0, 157)      0.2103347055995463
  (0, 2687)     0.20731128069560395
  (0, 1881)     0.18410931600082184
  (0, 2679)     0.3423815333594662
  (0, 3022)     0.17655669408553648
```

Fig. 9. Features Generated with TF IDF

Dataset Split: After the feature extraction phase using TF-IDF vectorization on our airline tweets dataset, the subsequent crucial step in our machine learning workflow was to divide the data into training and testing subsets. This split is a pivotal process in any machine learning project as it sets the foundation for training robust models and subsequently evaluating their performance on unseen data.

We adopted the conventional 80/20 split ratio for our dataset. This meant allocating 80 percent of the data for

training purposes and reserving 20 percent for testing. This ratio is widely accepted in the machine learning community, as it provides a substantial amount of data for training the models while still retaining a significant portion for an unbiased evaluation of their performance.

The dataset, after undergoing TF IDF transformation, resulted in each sample being represented by 5000 features. With the 80/20 split, the dimensions of our train and test sets were established as follows: The training set (X train) encompassed 21,552 samples, each with 5000 features, and the corresponding sentiment labels (y train) for these samples. The testing set (X test) comprised 5,388 samples, again with 5000 features each, and their associated sentiment labels (y test).

V. Modeling

In our endeavor to achieve accurate sentiment classification in the airline tweets dataset, we explored a range of machine learning models, each offering distinct advantages. Our primary models were Random Forest, Logistic Regression, and Support Vector Machine (SVM). To enhance these models' effectiveness, we not only explored various ensemble techniques but also conducted thorough hyperparameter tuning to optimize their performance.

The project explores various machine learning algorithms to achieve accurate sentiment classification in the airline tweets. Logistic regression, Support Vector Machine and Random Forest are chosen as the baseline models. Each model is tuned to avoid overfitting and enhance performance by adjusting hyperparameters. Moreover, hybrid models based on individuals are applied to improve the performance and stability further. Accuracy with other evaluation metrics are used to give a clear vision of their performance.

Random Forest: We started with the Random Forest classifier, valued for its robustness and versatility in handling datasets with numerous features. This model builds multiple decision trees during training and predicts the class that is the most common output of these trees. After hyperparameter tuning, which involved fitting 3 folds for each of 81 candidates (totaling 243 fits), we determined the best parameters to be max depth: 30, min samples leaf: 1, min samples split: 2, and n estimators: 200. This tuning achieved a best score of 0.7556 and led to a test accuracy of 76.76 percent, demonstrating its capacity to minimize overfitting while maintaining high accuracy.

```
Fitting 3 folds for each of 81 candidates, totalling 243 fits
Best Parameters: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Best Score: 0.755567928705122
Test Accuracy: 0.7676317743132888
```

Fig. 10. Best Parameters for Random Forest Hyperparameter tuning

Logistic Regression: Next, we employed Logistic Regression, known for its straightforwardness and effectiveness in binary classification tasks. This model excels in predicting binary outcomes based on predictor variables. Through hyperparameter tuning over 3 folds for each of 12 candidates (36 fits in total), we found the best parameters to be C: 10 and solver: liblinear. This tuning process yielded a best score of 0.8071 and a test accuracy of 83.05 percent, proving its proficiency in classifying tweets into distinct sentiment categories.

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Best Parameters: {'C': 10, 'solver': 'liblinear'}
Best Score: 0.8071640682999258
Test Accuracy: 0.8305493689680772
```

Fig. 11. Best Parameters for Logistic Regression Hyper parameter tuning

Support Vector Machine (SVM) SVM was integral to our approach, recognized for its high performance in high-dimensional spaces. It identifies the optimal hyperplane to segregate different classes within the dataset. The hyperparameter tuning for SVM involved 3 folds for each of 12 candidates (36 fits), resulting in the best parameters being C: 10, gamma: scale, and kernel: rbf. This fine-tuning achieved a best score of 0.8609 and a notable test accuracy of 90.29 percent, highlighting its ability to interpret the complex nuances of language in tweets.

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Best Parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Best Score: 0.8609409799554566
Test Accuracy: 0.9029324424647365
```

Fig. 12. Best Parameters for SVM Hyper parameter tuning

Ensemble Techniques: We enhanced our model's capabilities with ensemble techniques, firstly employing a weighted voting classifier that combined Random Forest, Logistic Regression, and SVM. This Weighted Hybrid Model (SVM, RF, LR) achieved an accuracy of 89.58 percent. We also explored a second weighted voting classifier, integrating only SVM and Logistic Regression, which attained an accuracy of 89.77 percent.

```
Weighted Hybrid Model Accuracy: 0.8958797327394209
```

Fig. 13. Accuracy from Ensemble technique (SVM, RF and LR)

```
Weighted Hybrid Model Accuracy: 0.897735708982925
```

Fig. 14. Accuracy from Ensemble technique (SVM and LR)

Stacking Ensemble Model: The pinnacle of our modeling efforts was the stacking ensemble. This sophisticated model amalgamated SVM and Logistic Regression as base models, with an additional Logistic Regression serving as the final estimator. This strategic blend exploited the unique strengths of each model, culminating in a highly refined classification system. The stacking ensemble model impressively achieved an accuracy of 91.15 percent, underscoring the efficacy of combining multiple modeling techniques, especially after meticulous hyperparameter tuning.

```
Stacking Ensemble Accuracy: 0.9114699331848553
```

Fig. 15. Accuracy from Stacking Ensemble technique

VI. Evaluation

The evaluation of our model's performance on the airline tweets sentiment classification task has been comprehensively conducted using three metrics: a confusion matrix, the Receiver Operating Characteristic (ROC) curve, and a classification report. Each of these metrics provides insight into different aspects of the model's predictive capabilities.

Random forest: The confusion matrix provides a snapshot of the model's classification accuracy, with a high count of true positives and true negatives indicative of a commendable performance. It shows the model's strength in correctly classifying the negative (class 0) and positive (class 2) sentiments with relatively high precision, though it appears to struggle slightly more with the neutral class (class 1). The matrix shows 1,389 true negatives, 1,389 true neutrals, and 1,358 true positives, with the numbers outside the main diagonal indicating instances where the model has misclassified the sentiments.

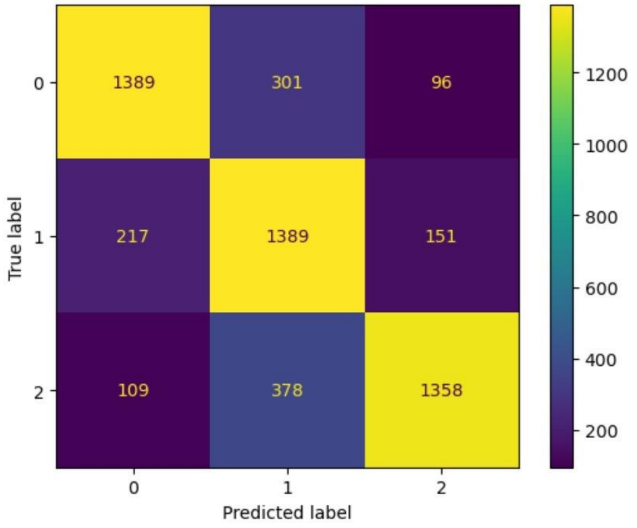


Fig. 16. Confusion Matrix for Random Forest

The ROC curve offers a graphical representation of the model's ability to distinguish between the classes at various threshold levels. The AUC values for each sentiment class are robust, lying between 0.90 and 0.93, suggesting that the model has a reliable discriminative ability. These values point to the model's competency in handling the trade-off between sensitivity and specificity across different decision thresholds.

Completing the model's performance picture, the classification report summarizes key metrics, including precision, recall, and F1-scores for each class. The Random Forest model presents a balanced profile with precision scores from 0.67 to 0.85 and recall scores ranging from 0.74 to 0.79. The

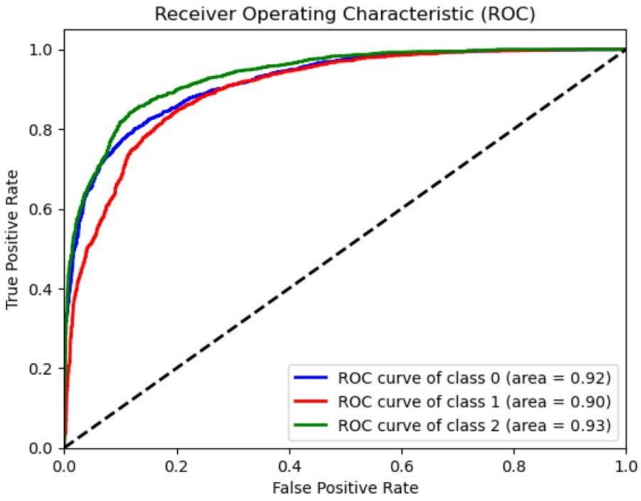


Fig. 17. ROC-AUC curve for Random Forest model

F1-scores, which provide a harmonic mean of precision and recall, are fairly consistent, affirming the model's balanced predictive performance. The overall accuracy of approximately 77 percent,further underscores the effectiveness of the Random Forest in classifying sentiments, despite the inherently challenging nature of the task.

```
Fitting 3 folds for each of 81 candidates, totalling 243 fits
Best Parameters: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Best Score: 0.7555679287305122
Test Accuracy: 0.7676317743132888
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.78      0.79      1786
           1       0.67      0.79      0.73      1757
           2       0.85      0.74      0.79      1845

    accuracy                           0.77      5388
   macro avg       0.78      0.77      0.77      5388
weighted avg       0.78      0.77      0.77      5388
```

Fig. 18. Performance Metrics for Random Forest model

SVM: The confusion matrix reveals the model's accuracy in classifying tweets into the correct sentiment categories. The matrix showcases a substantial number of true positives and true negatives, indicating a high level of accuracy. For instance, the model correctly identified 1,547 tweets as negative (class 0), 1,602 as neutral (class 1), and 1,716 as positive (class 2). The off-diagonal numbers, which are relatively low, represent the misclassifications, demonstrating the model's precision in distinguishing between the sentiments.

The ROC curve complements the confusion matrix by illustrating the model's performance across different threshold levels. The curve plots the true positive rate against the false positive rate, with the area under the curve (AUC) indicating the model's ability to differentiate between the sentiment classes. For all classes, the AUC is close to 1 (0.98 for negative, 0.97 for neutral, and 0.98 for positive), suggesting that the model has an excellent discriminative power for all sentiment categories.

Lastly, the classification report provides a summary of the precision, recall, and F1-score for each class. These scores are crucial for understanding the balance between the model's

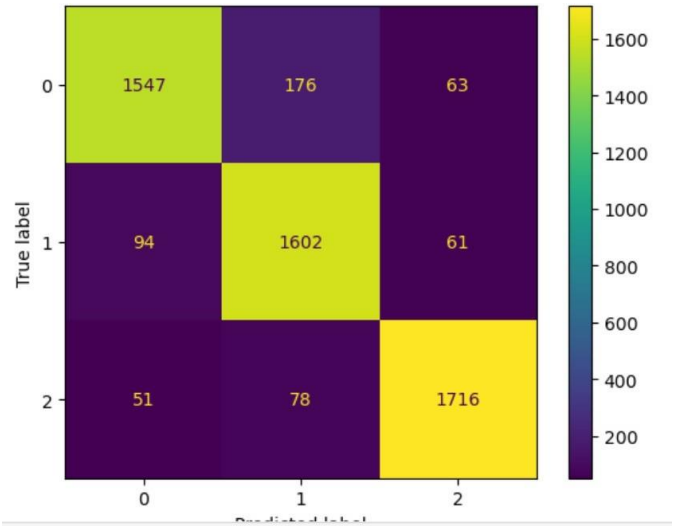


Fig. 19. Confusion Matrix for SVM Model

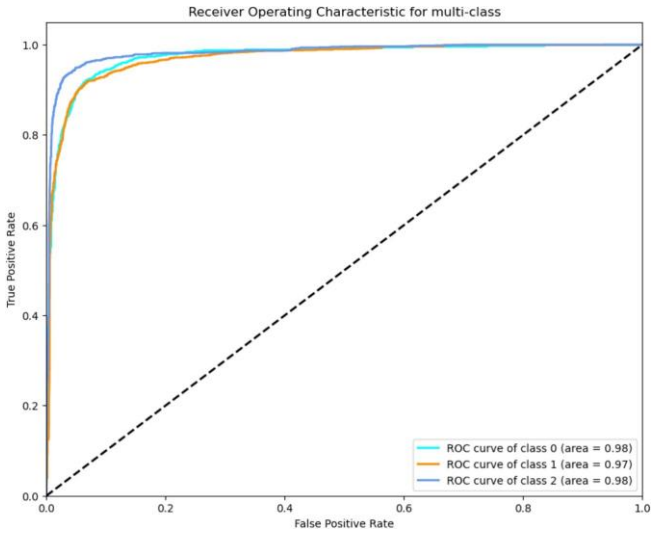


Fig. 20. ROC-AUC curve for SVM Model

sensitivity (recall) and its precision. With precision scores ranging from 0.86 to 0.93 and recall scores similarly high, the model demonstrates an exceptional balance in its predictive accuracy across all classes. The F1-scores, which are the harmonic mean of precision and recall, further confirm the model's robustness with scores close to 0.90 for all classes, indicating a high degree of accuracy and consistency.

Logistic Regression: The confusion matrix displays a significant count of true positives and true negatives, signaling accurate classifications. Specifically, the model correctly predicted 1,449 tweets as negative, 1,417 as neutral, and 1,609 as positive, which are substantial figures demonstrating the model's ability to correctly discern sentiments. The elements outside the matrix's main diagonal indicate misclassifications, which are comparatively low, showing the model's adeptness

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Test Accuracy: 0.9029324424647365
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.87      0.89      1786
           1       0.86      0.91      0.89      1757
           2       0.93      0.93      0.93      1845

    accuracy                           0.90      5388
   macro avg       0.90      0.90      0.90      5388
weighted avg       0.90      0.90      0.90      5388
```

Fig. 21. Performance Metrics for SVM Model

at distinguishing between different sentiment categories.

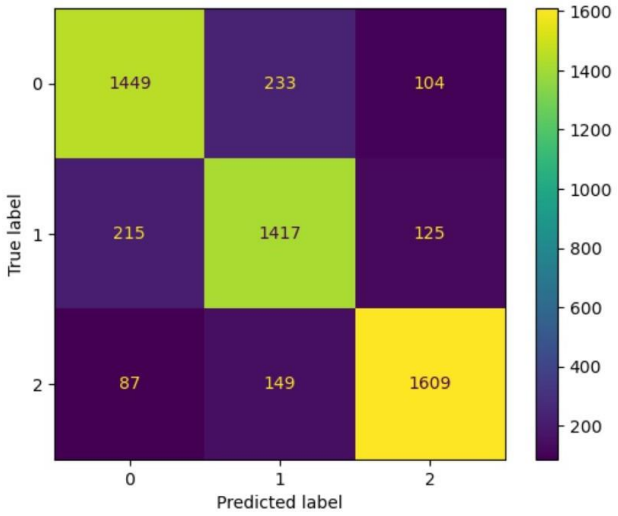
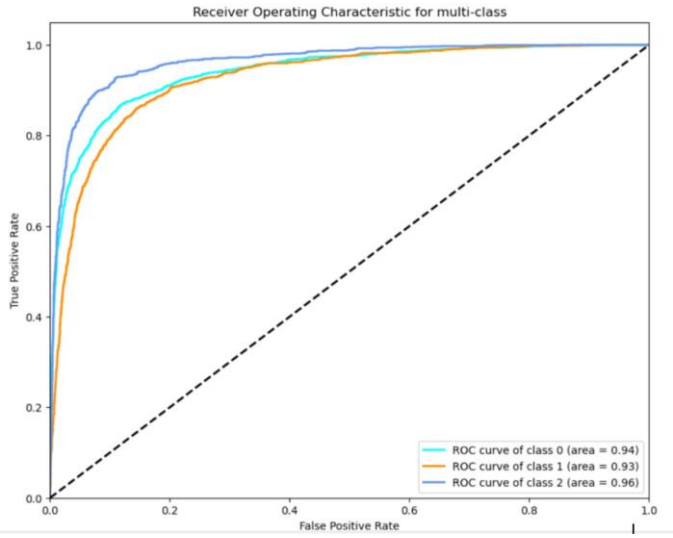


Fig. 22. Confusion Matrix for Logistic Regression Model

Complementing the confusion matrix, the ROC curve graphically represents the model's capability to classify sentiments at various threshold levels. The high area under the curve (AUC) values for each class, with 0.94 for negative, 0.93 for neutral, and 0.96 for positive sentiments, suggest the model's strong discriminative power, reinforcing its effectiveness in separating the classes across varied decision thresholds.

The classification report offers a quantitative summary of the model's performance, including precision, recall, and F1-scores. These metrics provide a deeper understanding of the model's balance between correctly identifying relevant instances (precision) and the proportion of actual positives correctly identified (recall). With precision and recall scores closely aligned, ranging from 0.79 to 0.88, the model shows a commendable balance in its classification ability. The F1-scores, which harmonize precision and recall, further affirm the model's consistent performance across all classes, with scores hovering around the 0.80 to 0.87 mark. An overall accuracy of 83 percent showcases the model's competency in accurately classifying the sentiment of tweets.

Stacking Ensemble : The confusion matrix depicts a high level of accuracy, with the majority of predictions concentrated




```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Best Parameters: {'C': 10, 'solver': 'liblinear'}
Best Score: 0.8071640682999258
Test Accuracy: 0.8305493689680772
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.81      0.82      1786
           1       0.79      0.81      0.80      1757
           2       0.88      0.87      0.87      1845

    accuracy                           0.83      5388
   macro avg       0.83      0.83      0.83      5388
weighted avg       0.83      0.83      0.83      5388
```

Fig. 24. Performance Metrics for Logistic Regression

on the matrix's main diagonal, highlighting the model's capability in correctly identifying the sentiments. The matrix shows that the model correctly classified 1,611 negative (class 0), 1,587 neutral (class 1), and 1,713 positive (class 2) sentiments. The relatively low numbers in the off-diagonal cells point to fewer instances of misclassification, suggesting that the model is adept at distinguishing between the sentiment categories effectively.

Complementing the insights from the confusion matrix, the ROC curve illustrates the model's exceptional performance across different thresholds. With AUC values nearing the ideal score of 1.0 for all sentiment classes (0.97 for both negative and neutral, and 0.98 for positive), the model demonstrates an outstanding ability to balance the true positive rate and false positive rate, which is critical for models dealing with multi-class classification problems.

The classification report provides a more detailed perspective of the model's performance, showing precision, recall, and F1-scores that underscore its balanced prediction capability across all classes. With each score around the 0.90 mark, the report confirms the model's precision and its ability to recall instances across sentiment labels accurately. An overall accuracy rate of over 91 percent further establishes this model

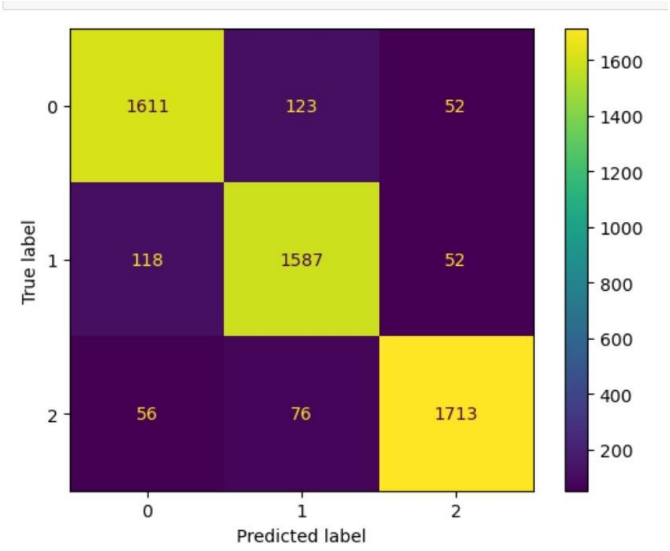


Fig. 25. Confusion Matrix for Stacking Ensemble Model

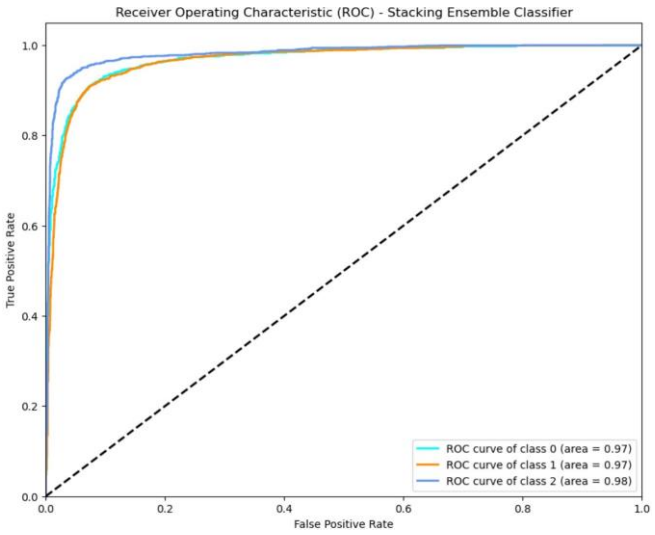


Fig. 26. ROC-AUC curve for Stacking Ensemble Model

as a robust classifier.

```
Stacking Ensemble Accuracy: 0.9114699331848553
Stacking Ensemble Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.90      0.90      1786
           1       0.89      0.90      0.90      1757
           2       0.94      0.93      0.94      1845

    accuracy                           0.91      5388
   macro avg       0.91      0.91      0.91      5388
weighted avg       0.91      0.91      0.91      5388
```

Fig. 27. Performance Metrics for Stacking Ensemble Model

This ensemble model stands out as the best among those tested for several reasons. It combines the strengths of individual models to create a more powerful aggregated model, capturing a broader range of features and relationships within the data. The ensemble approach also reduces the likelihood of overfitting, a common challenge in machine learning models, by blending the decision-making processes of various models. Furthermore, the high accuracy and consistent scores across various evaluation metrics signify that the Stacking Ensemble model is not only reliable but also provides consistent performance across different aspects of sentiment classification. In conclusion, the Stacking Ensemble model emerges as the superior choice due to its high accuracy, robustness, and ability to leverage the strengths of multiple models. The evaluation metrics collectively affirm its top-tier status in sentiment classification tasks, making it an excellent tool for interpreting the nuanced expressions of sentiment in airline tweets.

The results are summarized in Table I:

Table I: Table comparing model performance of 6 classifiers

TABLE I
TABLE COMPARING MODEL PERFORMANCE OF 6 CLASSIFIERS

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	83.05%	0.83	0.83	0.83
Support Vector Machine	90.29%	0.90	0.90	0.90
Random Forest	76.76%	0.78	0.77	0.77
Weighted Hybrid Model (SVM, LR)	89.77%	0.90	0.90	0.90
Weighted Hybrid Model (SVM, RF, LR)	89.58%	0.90	0.90	0.90
Stacking Ensemble Model	91.15%	0.91	0.91	0.91

VII. DEPLOYMENT

The deployment phase of our project was crucial in showcasing the real-world applicability and performance of our sentiment analysis model. For this purpose, we chose the Stacking Ensemble Model, which had shown impressive accuracy in classifying sentiments of airline tweets. To make the model and the accompanying TF-IDF Vectorizer readily usable, we serialized them into pickle files – a standard practice for preserving and deploying Python objects.

Our primary focus during deployment was to ensure that the model was not only functional but also accessible to users without technical backgrounds. To achieve this, we developed a Graphical User Interface (GUI) using Tkinter, a Python library renowned for its simplicity and effectiveness in creating GUI applications. This interface was designed to be intuitive and straightforward, catering to a wide range of users.

The centerpiece of the GUI is a text input field where users can type or paste the text of an airline tweet. This field is adaptable to different lengths of text, accommodating the varied nature of tweets. Once the text is entered, users can initiate the sentiment analysis by clicking on a prominently displayed 'Predict' button. This action triggers the model to process the input text.

Upon clicking the 'Predict' button, the GUI activates the serialized model and vectorizer. The text entered by the user is first passed through the TfidfVectorizer for feature extraction. Subsequently, the processed text is fed into the Stacking Ensemble Model, which then predicts the sentiment of the tweet. This predicted sentiment, categorized as Negative, Neutral, or Positive, is displayed in a small pop-up window. The design of this result display is clean and simple, ensuring that users can easily understand the output of the sentiment analysis.

Fig. 28. Airline Tweets analysis GUI window

VIII. CONCLUSION

In conclusion, we achieved both data balance and great model stability were achieved through innovative augmentation methods. The hybrid model outperformed its individual components, with superlative accuracy and remarkable generalization. Furthermore, a practical progress has been made through the development of the Tkinter app which is user friendly for real-time sentiment analysis that stresses our concern for its accessibility and usability. On the other hand, the adaptability and efficiency of our approach are demonstrated by the model's strong performance in dealing with various sentiment expressions, especially those seen in airline tweets. Sentiment analysis is an area where the purpose of this project is not only to enhance analytical skills but also to demonstrate our dedication to providing impactful solutions.

IX. FUTURE SCOPE

The future research will focus on further improving performance and precision of our hybrid model by exploring deep learning methods. Adaptive learning will be included to enable the model changes to suit the changing patterns in data while moving towards real time social media analysis with interactive user interfaces. In addition, the model would be adjusted so as to accommodate for cross-domain applications in a multitude of industries and languages thereby opening up new opportunities for valuable insights and better decision-making across sectors that are different.

REFERENCES

[1] D. Dutta Das, S. Sharma, S. Natani, N. Khare, and B. Singh, Sentimental Analysis for airline Twitter data," IOP Conference Series: Materials Science and Engineering, vol. 263, p. 042067, 2017. doi:10.1088/1757-899x/263/4/042067.

[2] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah, and G. Choi, “Tweets classification on the base of sentiments for US airline companies,” Entropy, vol. 21, no. 11, p. 1078, 2019. doi:10.3390/e21111078
[3] N. Singh and M. Upreti, HMRFLR: A hybrid model for sentiment analysis of social media surveillance on airlines, 2022. doi:10.21203/rs.3.rs-2012451/v1
[4] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, “Tweet sentiment analysis with classifier ensembles,” Decision Support Systems, vol. 66, pp. 170–179, 2014. doi:10.1016/j.dss.2014.07.003
[5] D. Tiwari and N. Singh, “Ensemble approach for twitter sentiment analysis,” International Journal of Information Technology and Computer Science, vol. 11, no. 8, pp. 20–26, 2019. doi:10.5815/ijitcs.2019.08.03