SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I9P105

Volume 10 Issue 9, 38-51, September 2023

Original Article

Improving Sentiment Analysis on Imbalanced Airlines Twitter Data Using DSMOTE Technique

Shiramshetty Gouthami[1], Nagaratna P. Hegde[2]

[1]*Department of Computer Science and Engineering, Osmania University, Telangana, India.*
[2]*Department of Computer Science and Engineering, Vasavi College of Engineering, Telangana, India.*

[1]*Corresponding Author : gouthami.shiramshetty@gmail.com*

Received: 08 July 2023 Revised: 19 August 2023 Accepted: 09 September 2023 Published: 30 September 2023

Abstract *- Recently, social media and microblogging have gained popularity and traffic. Customer tweets to US airlines take time to analyse. A sentiment analysis model for unbalanced datasets fixes this with the help of the SMOTE method. This paper uses a new over-sampling technique to synthesise more samples near easily misclassified cases, unlike standard SMOTE, which treats all minority group samples equally. We target misclassified minority class classification to improve accuracy. The model steps through tweet sentiment classification. First, remove tweets with special characters, URLs, and stop words. It cleans tweets and extracts features to create numerical feature vectors. The Bag of Words (BoW) model uses all unique tweet terms to develop a lexicon—the presence or absence of these words numbers each tweet. We use Random Forest (RF) and Recurrent Neural Network (RNN) classification models after transforming tweets into feature vectors. A Random Forest ensemble learning system classifies using many decision trees. RNNs process sequential text using internal memory states. RF and RNN models use tweet feature vectors. Models learn feature-sentiment label patterns. They can label new tweets positive, negative, or neutral. These classification models let the installed system classify tweets by sentiment, providing valuable sentiment analysis insights. These models accurately classify tweets as positive, negative, or neutral. The density-based SMOTE results show our model's efficiency. TFIDF vectoriser Random Forest has 81% accuracy and 70% F1 score. These measures show the model can classify sentiment in imbalanced datasets, making it useful for sentiment analysis.*

Keywords *- Sentiment analysis, Class imbalance, Tweets, SMOTE, Classification.*

1. Introduction

Sentiment is the process of gathering information and automatically identifying the subjectivity of objects, and sentiment analysis plays a significant role. Social media platforms, such as Facebook, LinkedIn, and Twitter, have emerged as valuable resources for gaining knowledge of user preferences and can be utilised in various contexts [1].

These platforms offer a robust method of communication, enabling businesses to collect information about their products and services and analyse that data. Notably, the airline sector places great significance on its customers' experiences, and Twitter has become famous for allowing travellers to share their opinions. Airlines dedicate a substantial amount of time and money to increasing the loyalty of their customers. By analysing customers' comments on social media, airlines can determine the areas where they can improve and then deploy their resources appropriately. This helps the company's bottom line and positively contributes to society's advancement. In the past, before the arrival of Machine Learning (ML) and Artificial Intelligence (AI) [2], companies would have to manually annotate tweets for hours to determine whether they contained positive or negative sentiments, which would result in a delay in gaining perceptions. On the other hand, expressing opinions on social media can be difficult because of the prevalence of comparison perspectives, sarcasm, and different linguistic nuances that make it difficult to judge a situation [3, 4]. Some researchers investigated many components of opinions, such as targets, attitudes, opinion holders, and time, to unveil the complex structure of language. Sentiment analysis must additionally contend with the difficulty of binary sentiment classification (positive/negative) and fine-grained sentiment classification. Some research has been done on multi-class sentiment analysis; however, more of it needs to address class imbalance problems and online education. .uNNBag [5], RUSBoost [6], and SMOTEBoost [7] are examples of algorithms that are frequently coupled in real applications to address the imbalance problem. These techniques include ensemble learning techniques and data-level techniques. Because there is such a great need for these types of evaluations, automated approaches [8] to sentiment analysis are becoming increasingly popular.

Since air travel is presently one of the most talked about topics on Twitter, consumers of various airlines frequently use the platform to discuss their travel experiences with one another [9]. The processing of this data using methods based on machine learning allows for the extraction of valuable insights that may be used to evaluate the level of comfort experienced by passengers during their journey. In sentiment analysis, a significant amount of research has been carried out looking at air travel as a practical choice for long-distance transportation nationally and internationally. The airline sector is highly competitive because several airline service providers are located worldwide. When selecting an airline, travellers carefully consider various aspects, such as ticket prices, the length of their trip, the number of bags they are permitted to bring, and reviews left by previous passengers. Because of this, airlines are constantly working to improve the quality of their services and the facilities available to passengers in the air.

Previous research efforts had some drawbacks, even though they achieved a high level of accuracy in the sentiment analysis used for airline data classification. These studies frequently use datasets that need to be correctly balanced, which leads to the publication of accuracy metrics greater than the F1 score. Although the F1 score is recommended for datasets with uneven distribution, it is typically low in the examined research papers. When working with highly imbalanced datasets, machine learning algorithms risk becoming overfit to the class that makes up the majority. Previous research has yet to place a primary emphasis on ensuring the datasets are balanced, which may lead to bias due to model overfitting.

In addition, most of the proposed methods are based on data-intensive deep learning models, which could reduce accuracy when applied to lesser datasets. This study uses ML techniques to predict sentiment in US Airline Twitter data, overcoming the restrictions described earlier. This helps address the difficulties that have been raised. In this paper, we address the problem of several classes and the problem of imbalance class representation. We perform rigorous preprocessing on the Twitter data to make it suitable for analysis and examine strategies for dealing with class imbalance using the Density-based SMOTE method. The Density of the minority and majority sample determines its sampling weight using this method. In addition, we examine the efficiency of two different models, namely Random Forest and Recurrent Neural Network. So, this seeks to systematically analyse sentiment in US Airline Twitter data by considering the abovementioned aspects, with particular attention paid to the difficulties of multi-class classification and class imbalance.

2. Background and Related Work

Numerous methods have been suggested for preprocessing and extracting features from text to develop effective classification models. In sentiment analysis, these approaches aim to analyse domain-specific sentiments using content obtained from social media platforms.

2.1. Classical SMOTE

The SMOTE algorithm solves dataset class imbalance. It creates minority samples for each and its k closest neighbours. The SMOTE method requires initialising T (the number of minority samples), N% (the fraction of minority samples to be created), and k (the number of neighbours for producing new samples). This approach has multiple steps:

1. Select a_k, a minority sample, to produce additional samples.
2. Determine a_k k closest neighbours. Choose one arbitrarily, a_j.
3. Generate a new sample using Equation (1). In Equation (1), φ is a random value between 0 and 1. Generating the new sample involves creating a synthetic sample by linearly interpolating between a_k and a_j in the feature space. This helps to increase the representation of the minority class and balance the dataset. By repeating these steps for multiple minority samples, the SMOTE algorithm creates synthetic samples that resemble the characteristics of the minority class, effectively addressing the class imbalance issue. These newly generated samples can then be used to train machine learning models, improving their performance in the minority class.

$$a_{new} = a_k + \varphi * \left(a_j - a_k\right) \quad (1)$$

4. Repeat steps 2 and 3.
5. For every (k= 1, 2,.., T) samples, run above process.

In recent years, sentiment analysis has become an important research area in various domains. For example, Twitter, Skytrax, and TripAdvisor have become valuable sources of customer feedback and reviews in the airline industry. Researchers have studied the sentiment toward airline services using different sentiment analysis models. In [10], the authors focused on the airline industry and collected data from social networking platforms to analyse customer sentiments and opinions. They experimented with six different sentiment analysis models and found that the BERT model achieved the highest accuracy of 86%. This model allowed them to evaluate the social status, company reputation, and brand image of Malaysian airline companies based on customer sentiments. In [11, 12], the authors employed a semi-supervised bootstrapping approach to analyse complaints related to transportation services on social media platforms. The researchers started with a small set of annotated samples and used them to identify language indicators relevant to complaints. These indicators were then used to extract additional information until they could find

no more indicators. In [13], the authors focused on measuring customer satisfaction using an ML method for sentiment analysis in tweets mentioning airlines. They gathered all the API tweet data for their research with the help of preprocessing techniques. Later, they used ML models to predict sentiments. Additionally, they conducted lexical analysis to model keyword frequencies, providing a context for interpreting sentiments. They used "Bollinger Bands" for detecting, enabling the identification of sudden shifts in passenger emotions. In [14], the authors employed the Random Forest method to develop a sentiment analysis model. Two distinct datasets were used, one consisting of tweets about airlines from GitHub and the other consisting of movie reviews from IMDb. The aim was to perform a more precise sentiment analysis about these topics. In [15], the authors focused on aspect-based sentiment analysis of Arabic tweets. They compared two-word embedding models: fastText Arabic Wikipedia and AraVec-Web. The researchers used a dataset of 5,000 Arabic tweets related to airline services. They manually labelled each tweet for aspect detection and sentiment polarity classification. Support vector machine classifiers were utilised for both aspect detection and sentiment classification. The results showed that fastText Arabic Wikipedia word embeddings performed slightly better than AraVec-Web. In [16], the authors studied sentiment analysis of Indian airlines from collected comments on social media travel websites. The VADER model was used to assign sentiment ratings based on the linguistic properties of the comments.

A hybrid model, named Hybrid Model Integrated Adaboost Approach (HMIAA), was proposed, combining Support Vector Machine (SVM) classifiers with gradient-boosted trees. The objective was to optimise the efficiency and accuracy of the sentimental classification technique. The results demonstrated that the suggested hybrid approach incorporating Adaboost outperformed other basic classifiers. The sentiment analysis performed on the datasets could provide recommendations to passengers regarding the best airlines to fly with. Overall, these studies highlight the application of various sentiment analysis techniques and models in the airline industry to understand customer sentiments and improve services based on feedback and reviews.

3. Proposed Modelling

This study integrates opinion mining theories and addresses time complexity and class imbalance in sentiment analysis. The modular architecture manipulates Twitter text and analyses sentiment. Preprocessing tweets for statistical model training begins the model. This resembles knowledge-based expert systems. The Bag of Words (BoW) model converts tweets into feature vectors for numerical representation. We split the dataset into training (75%) and testing (25%). Fitting data to machine learning models comes next. Detailing the models utilised. Tweet text preparation, lemmatisation, text embeddings, and SMOTE are task-independent and classify tweets by sentiment. Figure 1 shows the model's components and connections. This model uses a systematic and comprehensive approach to sentiment analysis to classify tweets as positive, neutral, or negative.

3.1. Dataset Description

This paper utilised the Twitter US Airline dataset, which consists of 14,640 tweets collected from various active airlines in the United States. We specifically curated this dataset for sentiment analysis tasks related to major US airlines and covered a range of customer experiences and challenges. We created the dataset in 2015 through web scraping techniques involving the participation of volunteers who classified tweets as positive, negative, or neutral. We categorised negative tweets based on specific issues, such as flight delays or poor service.

This dataset is valuable for analysing customer satisfaction and understanding their sentiments towards different airlines. In our study, we utilised this dataset to train a sentiment classifier capable of predicting the sentiment of unseen data. We split the dataset into two groups: a training set containing 75% of the data and a test set containing 25%. This division allowed us to evaluate the performance of our classifier on unseen data and assess its accuracy in sentiment prediction. Table 1 provides an overview of the attributes included in the Twitter US airline sentiment dataset, which we utilised in our analysis. These attributes offer valuable insights into the tweets' characteristics, facilitating the sentiment analysis.

3.2. Text Preprocessing of Twitter Data

The structure of Twitter data presents challenges in extracting meaningful features for analysis due to its unstructured format. Tweets often contain empty spaces, stop words, slang, special characters, hashtags, emoticons, timestamps, abbreviations, and URLs. To effectively mine this data, preprocessing is necessary. In our study, we implemented five cleaning procedures to preprocess the data.

The first step involved removing duplicate and complex phrases, such as "to," "for," and "how," which impeded information flow. In the second stage, we eliminated all punctuation marks from the tweets, including the @ sign and mentions of airline companies. This was done to streamline the text and remove unnecessary symbols. In the third phase, we removed the "@" symbol and airline company names from all tweets since they were consistently present at the beginning of each tweet. In the final stage, we converted all letters to lowercase to ensure text consistency, considering that machines are case-sensitive. Another crucial step was stemming, which involved reducing words to their base form by removing affixes. For instance, the term "flew" was transformed into "fly" through stemming.

Table 1. Feature description of US airline dataset

Attributes	Details
tweet_id	ID of tweet
airline_sentiment	Class label of tweets (+ve, neutral, −ve).
airline_sentiment_confidence	A numbered attribute which shows the trust rate of grouping the text to one of the categories.
Negative reason	The reason to consider a tweet as −ve, as per the experts.
airline	Official name of the airline
airline_sentiment_gold	Airline trust for –ve text
name	Name of the user
negativereason_gold	Trust in determining a text's -ve rationale.
retweet_count	A numerical value that represents retweets for a tweet.
text	Text of the tweet as typed by the user
tweet_coord	latitude and longitude of tweeter user
tweet_created	Created date of tweets
tweet_location	Location of the tweet
user_timezone	Timezone of a user

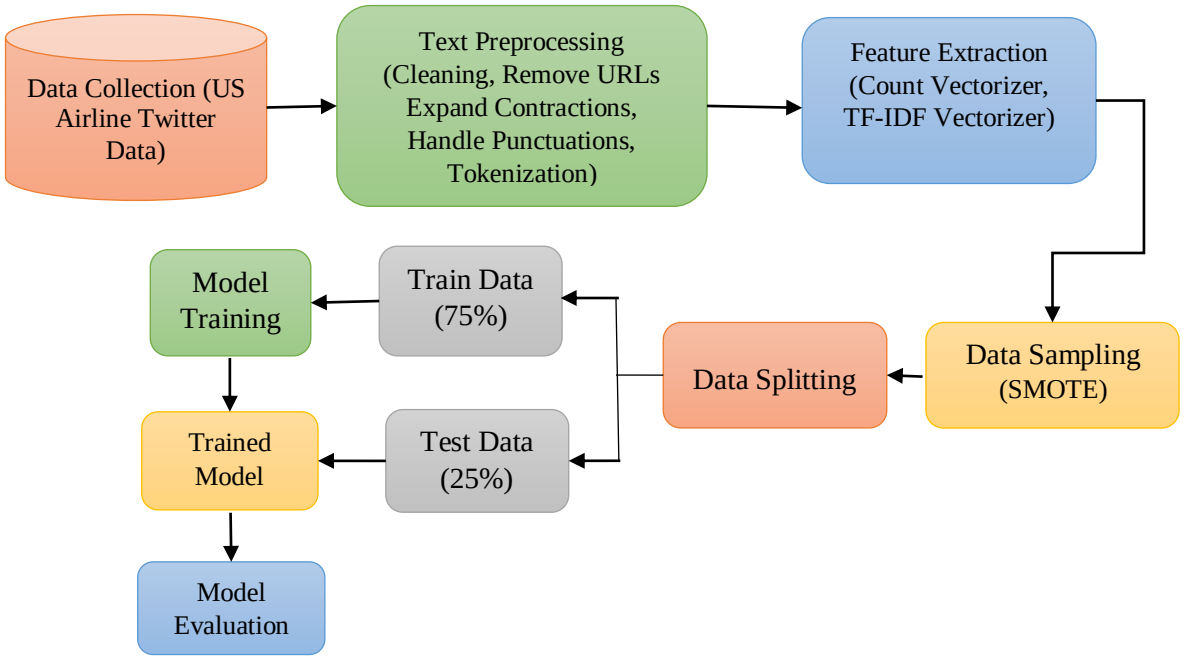

Fig. 1 The flow of the proposed methodology

Applying stemming improved the system's effectiveness by simplifying word representation. After completing the preprocessing steps, we constructed a corpus, a collection of text representing the cleaned tweets. We employed the Bag of Words (BoW) technique to encode the tweets as feature vectors. BoW represents each tweet as a vector based on the occurrence of words in the text. These feature vectors were then used in machine-learning models for prediction tasks. We also eliminated punctuation and numeric values from the data during the cleaning process.

Punctuation was removed as it contributed little to text analysis and could confuse the models in distinguishing punctuation marks from other characters. Numeric values were disregarded as they had little impact on text analysis, and simplifying the data improved the training process of the models. The preprocessing steps involved cleaning, standardising, stemming, constructing the corpus, and tokenising, breaking down the text into individual words or tokens. These steps were essential in preparing the data for further analysis and machine learning tasks.

3.3. Feature Extraction

Machine learning algorithms expect a two-dimensional table with rows representing instances or documents and columns describing attributes or features. We need vectors to represent textual data. Count Vectorizer and TF-IDF Vectorizer transform text into vectors [17]. We may need feature selection to apply machine learning algorithms to text segments. In large datasets with many words, this step reduces input data dimensionality. The dataset may contain a dictionary of terms after preprocessing and cleaning. We prioritise these words using a Count Vectorizer or TF-IDF Vectorizer.

3.3.1. Count Vectorizer

It is imple and effective. It weights words by their frequency in a document. The Count Vectorizer tokenises text and develops a vocabulary. Count Vectorizer produces a word count matrix with each cell representing a document's word frequency with matrix values as integers.

3.3.2. TF-IDF Vectorizer

It assigns a weight to each word based on its frequency in each document and across the collection. TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure of a word's relevance in a document compared to its presence in the complete collection [18-20]. It weights words that are more common in one document. Count Vectorizer and TF-IDF Vectorizer are commonly used to convert text into numerical representations for machine learning algorithms. The dataset's needs and properties determine the option.

$$TF - IDF = TF_{t,d} * log \frac{N}{D_f} \quad (2)$$

3.4. Data Re-Sampling Technique

The DSMOTE algorithm is a data re-sampling technique used for class imbalance problems. Imbalancing contains an unequal distribution of target classes, which results in biased model performance when the majority class dominates the dataset [21]. In our study, we employed the DSMOTE algorithm to rebalance the dataset and give more importance to the minority class. The DSMOTE algorithm considers the Density of "minority and majority samples" in determining their weights. It identifies minority samples located in areas with sparse minority and dense majority samples as more prone to misclassification. Therefore, these samples are assigned larger sample weights. On the other hand, samples located in areas with denser minority samples and sparse majority samples are considered less prone to misclassification and are assigned smaller sample weights. This approach ensures that the oversampling process is performed to preserve the underlying distribution and structure of the data.

Using the DSMOTE algorithm, we aim to balance the distribution of samples across classes, particularly giving more weight and importance to the minority class. This allows the classifier to learn from a more balanced representation of the dataset, improving its ability to classify instances from the minority class accurately. The algorithm considers the local Density of both samples, ensuring that the oversampling process effectively captures the data's characteristics. Before applying the DSMOTE algorithm, several parameters need to be initialised, including the number of minority samples (*l*), the Density of the minority class (*ld*), and a parameter (α) that determines the number of minority samples to be generated. These parameters help customise the oversampling process based on the specific characteristics of the dataset. In summary, the DSMOTE rebalances the distribution of samples, giving more weight to the minority class. Considering the Density of samples from different classes, we can effectively balance the dataset and improve the classifier's performance, particularly for the minority class.

$$\xi_{min} = \{a_1^{min}, a_2^{min}, \dots, a_{m_1}^{min}\} \quad (3)$$

and the majority samples of the training set as

$$\xi_{maj} = \{a_1^{maj}, a_2^{maj}, \dots, a_{m_2}^{maj}\} \quad (4)$$

In equations (3) and (4), m_1 represents the number of minority samples, while m_2 represents the number of majority samples. We use these equations in the SMOTE algorithm to calculate the distances between a given minority sample (a_k^{min},) and its id-th nearest neighbours in both the minority class (ξ_{min}) and the majority class (ξ_{maj}). The resulting distances are denoted as d_{dk}^{min} and d_{dk}^{maj}, respectively.

The SMOTE algorithm can be summarised into the following steps:

1. For each minority sample a_k^{min}, k=1,2,…,m_1 where k ranges from 1 to the total number of minority samples ($m1$), calculate the distances between a_k^{min} and its id-th nearest neighbours in the minority class $\xi_{min}$and ξ_{maj}, and denote them as d_{dk}^{min} and d_{dk}^{maj} respectively.
2. Among the i_d the nearest neighbor distances of the minority samples, identify the maximum distance. This maximum distance indicates the farthest neighbor among the i_d th nearest neighbors of the minority samples. These steps are essential in the SMOTE algorithm as they determine the distances between each minority sample and its neighbours in both the minority and majority classes. By considering these distances, the algorithm generates synthetic samples located in regions of the feature space where the minority class is underrepresented. This helps to balance the class distribution and improve the performance of classifiers on imbalanced datasets.

$$d_{max}^{min} = max\left(d_{d1}^{min}.d_{d2}^{min},\dots,d_{dm_1}^{min}\right) \quad (5)$$

Find the maximum distance to the majority samples for each i_d th.

$$d_{max}^{maj} = max\left(d_{d1}^{maj}.d_{d2}^{maj},\dots,d_{dm_1}^{maj}\right) \quad (6)$$

3. To evaluate samples density in the SMOTE, we utilise equations (7) and (8). These equations are applied to each minority sample in the dataset. Let's expand on the steps involved a_k^{min}, k=1, 2,…,m_1 and get D_k^{min} and D_k^{maj}.

In Eq. (7) and (8), γ is a small for $D_k^{min} > 0$ and $D_k^{maj} > 0$.

$$D_k^{min} = 1 - \frac{d_k^{min}}{d_{max}^{min}} + \gamma \quad (7)$$

$$D_k^{maj} = 1 - \frac{d_k^{maj}}{d_{max}^{maj}} + \gamma \quad (8)$$

4. Equation (9) aims to assess the probability of misclassification for each minority sample. It quantifies how easily a minority sample can be misclassified based on its Density compared to the majority samples. The sample weight (x_k) for each minority sample, where k ranges from 1 to the total number of minority samples, g_k, k=1, 2…, m_1 is then determined using Equation (10).

$$g_k = \frac{D_k^{maj}}{D_k^{min}} \quad (9)$$

$$x_k = \frac{g_k}{\sum_{k=1}^{m_1} g_k} \quad (10)$$

5. To determine the number of minority samples that need to be generated (H) using the parameter α, Equation (11) is utilised. H is then assigned to each individual minority sample. Next, p_k (where k ranges from 1 to the total number of minority samples) is obtained using Equation (12).

$$H = \alpha(m_2 - m_1) \quad (11)$$

$$p_k = \frac{x_k}{\sum_{k=1}^{m_1} x_k} * H \quad (12)$$

6. In step 6 of the SMOTE algorithm, we find the l nearest neighbours of the current minority sample (a_k). From these neighbours, we randomly select one sample, denoted as a_j.
7. In step 7, we generate a new sample (a_{new}) using Equation (1), which combines the attributes of a_k and a_j.
8. Step 8 is repeated for round (p_k) times, where round (p_k) is the rounded value of p_k. This results in the generation of round (p_k) new minority samples.
9. Step 9 is applied to every minority sample (k=1, 2..., m_1), repeating steps 6, 7, and 8.

Through this process, the newly generated minority samples are located close to the existing minority samples, which are more prone to misclassification. This helps the classifier concentrate on the minority class and improves its ability to classify minority instances accurately.

3.5. Classification and Comparison

Lexicon-based and machine-learning algorithms can be used for tweet sentiment analysis. This study uses neural network classification to analyse Twitter sentiment. Machine learning algorithms can automatically learn and classify meaningful patterns from large datasets, making them useful in sentiment analysis. Text classification and sentiment analysis function well using neural network classification to train a model to associate tweet textual elements with sentiment labels. Random Forests: Random forest-based text classification model flowchart.

$$h(x\Theta k), k = 1, \dots n \quad (13)$$

The Θk random vectors are independent and identically distributed. Each vector contributes a single vote for the most popular class based on the input X. Random Forests is a system consisting of multiple classifiers. It creates numerous trees, known as sub-classifiers or internal classifiers, by combining the tree generation concept of CART and the bagging predictor. By aggregating the votes from multiple trees, Random Forests can achieve more accurate classification results on the test data set. The system's

accuracy depends on the strength of the individual trees and the weak correlation among them.

The final vote will be more accurate if the trees are strong and the correlations are weak. Random Forests is highly regarded as one of the most accurate classifiers developed. It utilises the Random Forests algorithm, which incorporates Bagging Sampling and CART. Through a voting mechanism, it becomes a highly accurate predictor. We present a flow chart depicting the construction of a text classification model based on the random forest algorithm in Figure 2.

Step 1: Prepare the text vector set. The text dataset undergoes preprocessing to create a collection of text vectors in the Vector Space Model (VSM), which can be used with the random forest algorithm.

Step 2: Building the random forest text classifier.

1. Implement the Bagging method to generate multiple training sets for constructing a *stree*. Given an initial training set *T* of size *S*, Bagging creates *stree* new training sets, each of size *S*.
2. For each training set, construct a CART classification tree without pruning following these steps:
   a) Assume there are primitive attributes. Choose a positive integer, *atry*, meet *atry*<<*A*. Empirical results suggest setting *atry* =A1/2 for better classification performance.
   b) At each internal node, randomly select *atry* attributes from the original *A* attributes as candidate attributes for splitting the node. This value remains the same across the entire forest.
   c) Use the Gini index to determine the best attribute for splitting the node among the *atry* candidate attributes.
   d) Grow each tree to its fullest extent, resulting in the maximum tree, t_{max}. Nodes with few samples below a specified threshold are considered small or pure nodes. These nodes may no longer have attributes and can be treated as branches. A small node contains a limited number of samples.
   e) No pruning is performed on the maximum tree t_{max}.

Step 3: Utilising the classifier. Classification output is determined by majority vote.

Recurrent Neural Network (RNN): This algorithm is provided with training and test data.

Traditional neural networks are unsuitable for specific natural language processing (NLP) tasks requiring sequential inputs, such as sentence word prediction. RNN sentiment analysis is particularly effective in these cases.

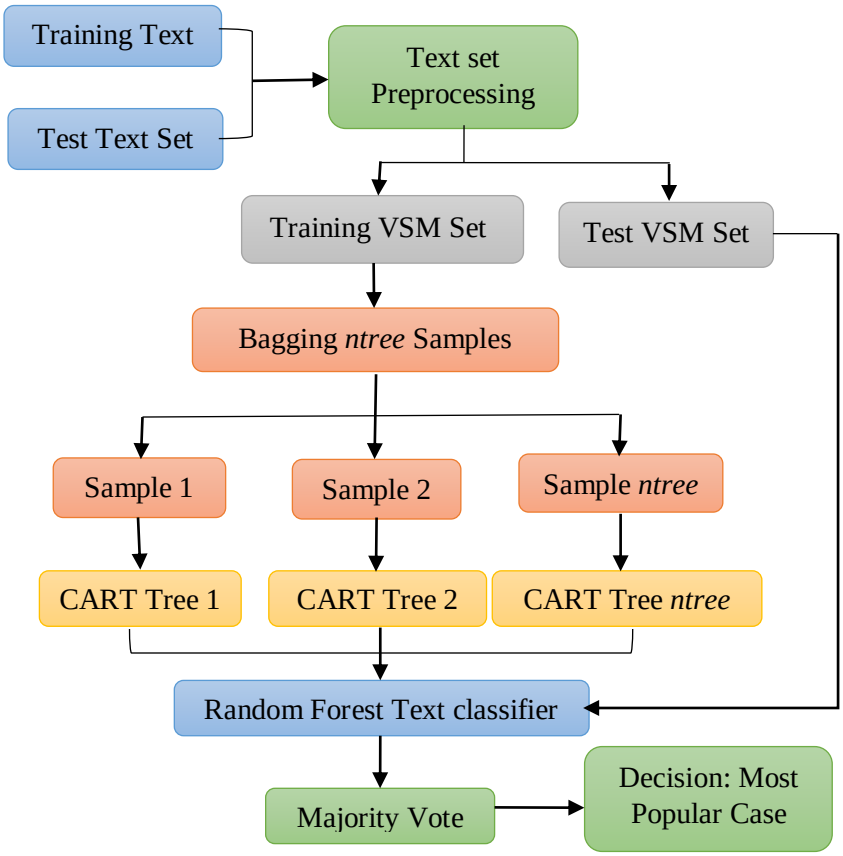

Fig. 2 Random forest algorithm-based text classification model flowchart

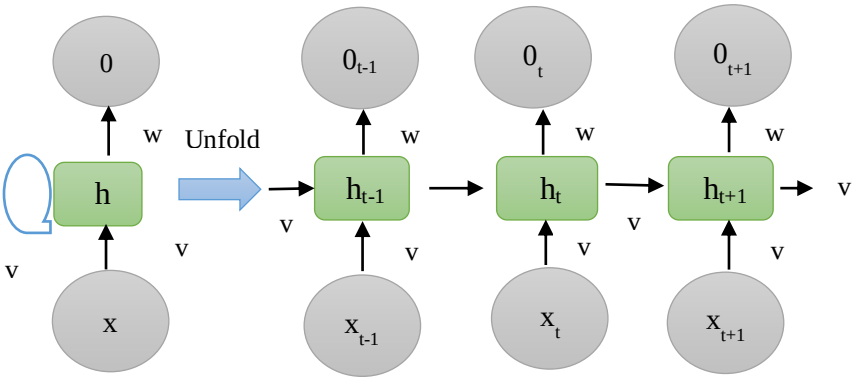

Fig. 3 RNN framework

Figure 3 illustrates RNN memory cells capable of storing extended sequences. The Equation shows RNN's basic formula:

$$a_t = f(h_{t-1}, x_t) \tag{14}$$

In Recursive Neural Networks (RNNs), the output at a given node is represented by "a_t" and is obtained from the previous node in the network. The activation function "f" used in RNNs is typically the hyperbolic tangent (*tanh*) function. The input sequences, denoted as$(x_0, x_1, \ldots, x_t)$, are fed into the network.

Recursive neural networks can capture the deep tree structure of texts, allowing them to grasp the semantics of the input data. However, it is worth noting that analysing the tree structure can be computationally intensive and time-consuming, which is considered a disadvantage of Recursive Neural Networks [22].

Recurrent Neural Networks (RNNs) have been developed to address this drawback, which has improved the time complexity compared to Recursive Neural Networks. RNNs have optimised mechanisms that make them more efficient in processing sequential data, such as text, without sacrificing their ability to capture semantic information.

4. Results and Discussions

Experiments and results are presented with an Intel 5-core personal computer for all testing. Anaconda, a Python scientific computing platform, built and implemented the model. These tests investigate algorithm performance and generalisation to different sentiment recognition tasks. Accuracy and F1 measure assesses the model. Previous research has utilised these indicators to evaluate model performance. Accuracy in binary classification issues is the model's correct and total prediction ratio. However, the F1 measure balances model performance by considering precision and recall. We can quantify how well the model classifies approaches by computing accuracy and F1. These metrics show how well the model predicts positive, negative, and neutral attitudes. We can compare the model's performance to other methods by comparing the results to past studies.

This study tests the model's performance and generalizability of sentiment recognition tasks. We may evaluate the model's strengths and limitations, find opportunities for improvement, and assess its applicability in real-world sentiment analysis tasks by analysing the accuracy and F1 measure.

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN+FP+TP+FN}} \tag{15}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{17}$$

$$\text{F1 score} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \tag{18}$$

To generalise to multi-class situations, we define precision and recall differently while keeping F1 the same. In the equations below:

$$\text{Precision}_c = \frac{S_{ii}}{S_{ii}} + \sum_{j=1 \text{ to } n; i \neq j} S_{ij} \tag{19}$$

$$\text{Recall}_c = \frac{S_{ii}}{S_{ii}} + \sum_{j=1 \text{ to } n; i \neq j} S_{ji} \tag{20}$$

Figure 4 demonstrates that negative tweets outnumber good tweets 4 to 1 and neutral tweets 3 to 1. Class imbalance biases our classification model(s). Using the proposed approach shown in Figure 5 indicates that US Airways performed the worst, with ~9x as many negative and positive tweets. Virgin America performed the best, with only 1.2x as many negative and positive tweets.

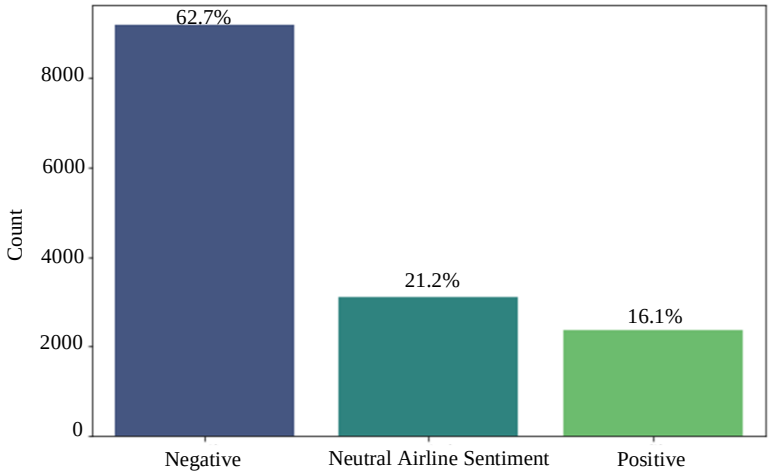

Fig. 4 Distribution of sentiment across all the tweets

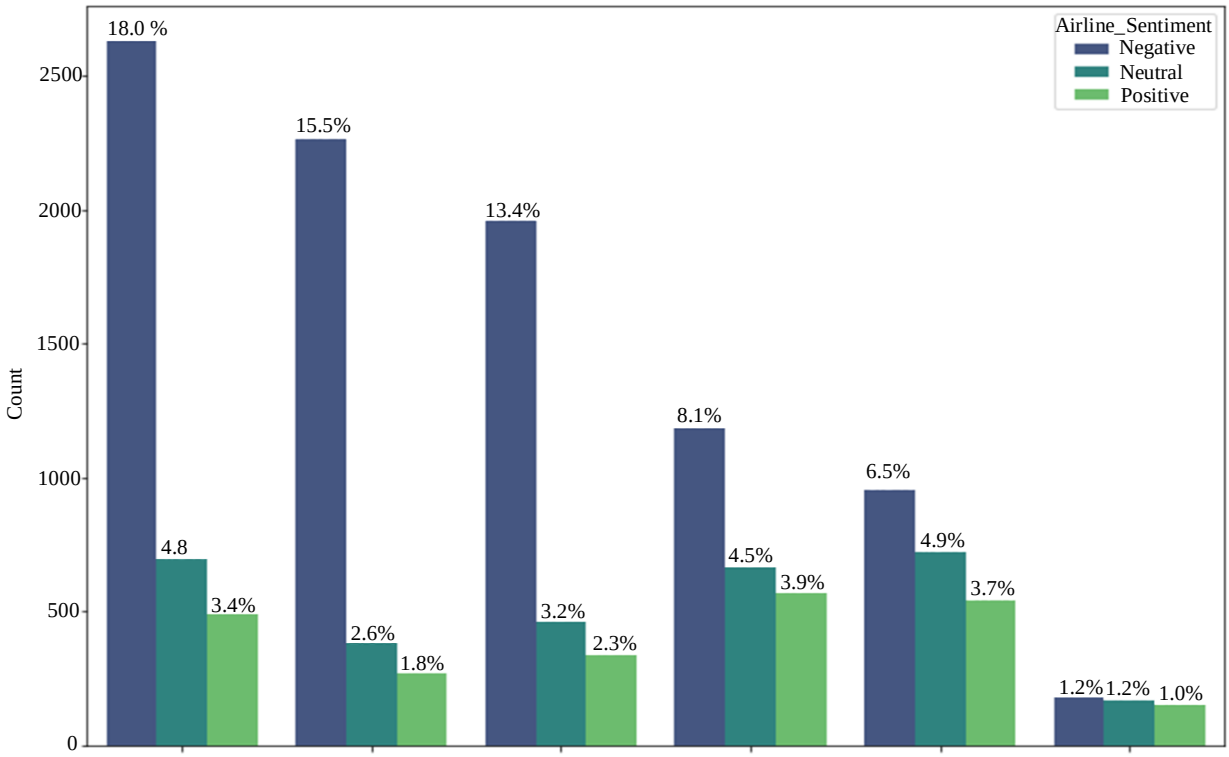

Fig. 5 Distribution of sentiment of tweets for each airline

Figure 6 displays the misclassification error 0.2661 when using the Count Vectorizer with a base learner count 200. However, our results indicate that as the number of base learners increases, the majority tends to agree, resulting in improved discrimination.

Similarly, Figure 7 illustrates the misclassification error of 0.2570 when using the TF-IDF vectoriser with a base learner count of 200. Again, as the number of base learners increases, the majority agreement leads to enhanced discrimination, as our findings demonstrate. Furthermore, Furthermore, Figure 8 presents the performance of the RF classifier with the vectoriser model. It correctly classifies 1,751 out of 2,627 negative tweets, 308 out of 577 neutral tweets, and 209 out of 456 positive tweets.

These results highlight the effectiveness of the RF classifier in accurately classifying different sentiment classes. These results highlight the effectiveness of the RF classifier in accurately organising other sentiment classes.

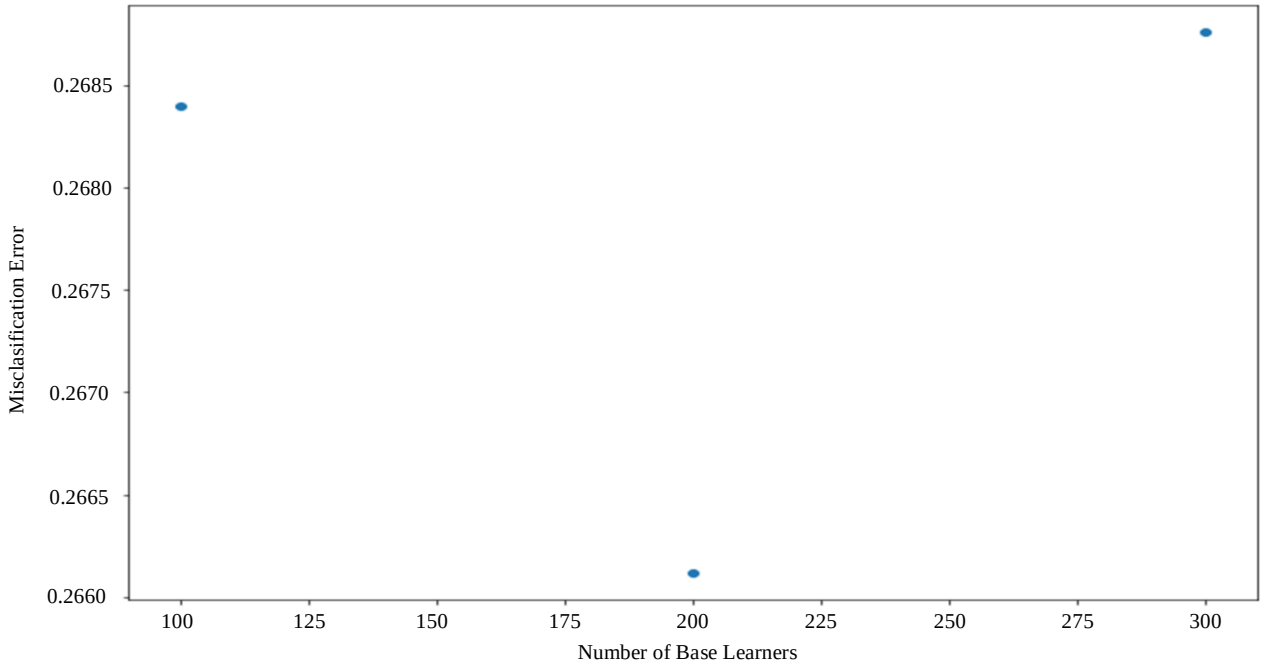

Fig. 6 Count vectorizer misclassification error

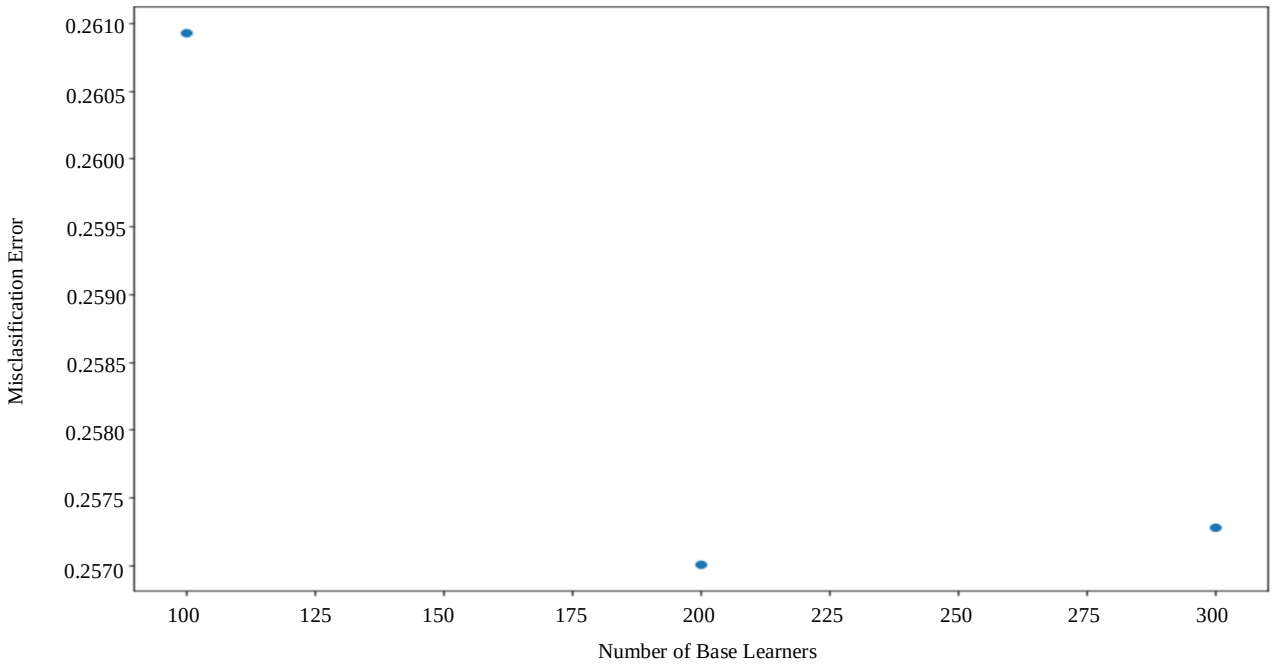

Fig. 7 TF-IDF vectorizer misclassification error

Figure 9 displays the performance of the RF classifier with the TF-IDF model. It correctly classifies 1,751 out of 2,627 negative tweets, 364 out of 577 neutral tweets, and 249 out of 456 positive tweets. These results demonstrate the effectiveness of the RF classifier with the TF-IDF model in accurately classifying different sentiment categories. The classifier accurately identifies negative tweets, achieving a notably high success rate of 91%. However, it performs relatively lower in classifying neutral and positive tweets, with 44% and 64% accuracy, respectively.Table 2 presents each model's precision, recall, and F1-score based on the classification results obtained from the US Airline dataset. These evaluation metrics provide insights into the performance of the models in classifying different sentiment categories. It is evident from the table that all models struggle to accurately organise tweets with a neutral sentiment, as indicated by the lower evaluation metric scores for this category.

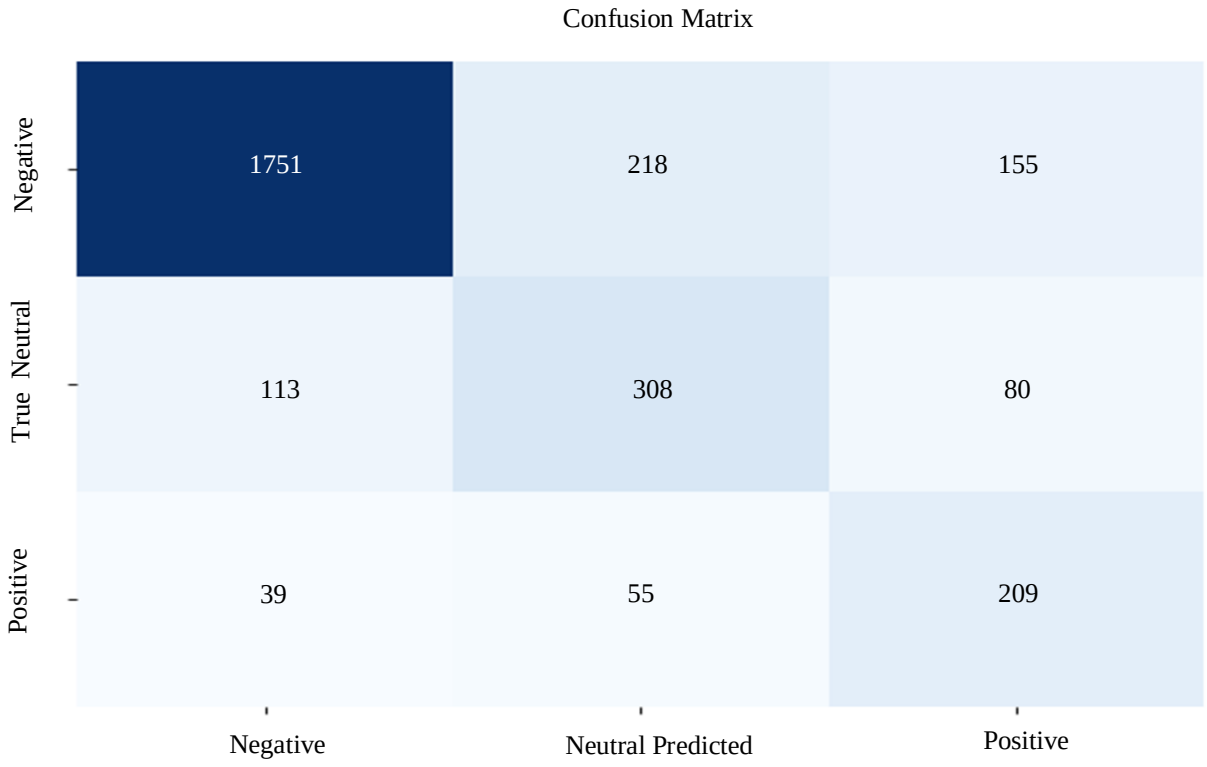

Fig. 8 Confusion matrix–RF classifier with vectoriser

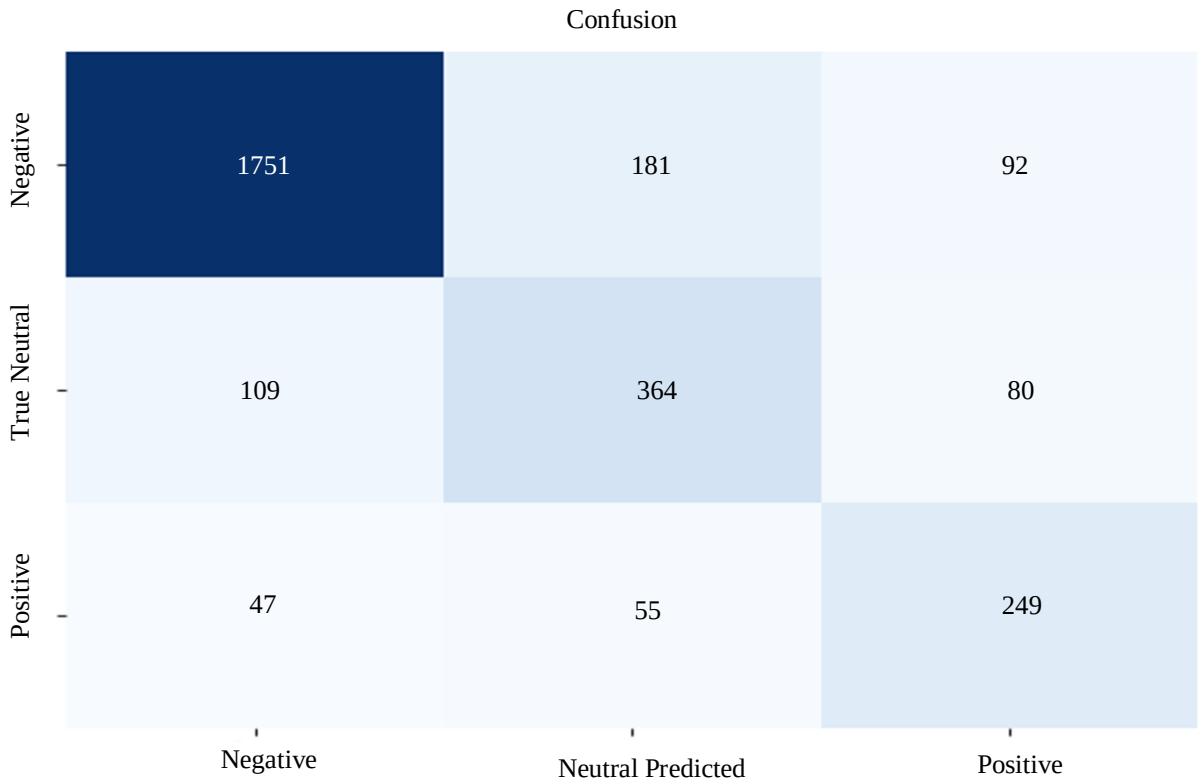

Fig. 9 Confusion matrix–RF classifier with TFIDF

Table 2. Classification performance of the framework with different classifiers on the tweet sentiment extraction dataset

Classifier/Metric	RNN			RF with Count Vectorizer			RF with TF-IDF Vectorizer		
	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive
Precision	0.85	0.53	0.68	0.87	0.49	0.62	0.86	0.57	0.72
Recall	0.84	0.56	0.68	0.86	0.49	0.64	0.91	0.44	0.64
F1-score	0.84	0.55	0.68	0.87	0.49	0.63	0.88	0.50	0.68

Among the models, the Random Forest model with TF-IDF Vectorizer exhibits the most difficulty correctly classifying neutral tweets, with a precision score of 0.44. On the other hand, the evaluation metric scores for tweets with negative sentiment are notably high, indicating that the models perform well in recognising negative sentiments. It is worth noting that the Random Forest model with TF-IDF Vectorizer achieves an impressive F1-score of 0.88, indicating significant improvement compared to other models. Interestingly, the performance of the RNN classifier slightly falls below this value.

4.1. Experiment Results of Models Using DSMOTE Data

DSMOTE balanced dataset experiments use TF-IDF and BoW features. Table 3 illustrates the performance of all TF-IDF and BoW models.

Table 3. TF-IDF and BoW performance of all models utilising DSMOTE data

Classifier	Accuracy	Precision	Recall	F1-Score
RNN	0.76	0.69	0.69	0.69
RF with Count Vectorizer	0.78	0.66	0.67	0.66
RF with TF-IDF Vectorizer	0.81	0.71	0.67	0.70

The performance of the Random Forest model has shown significant improvement when trained on TF-IDF features derived from the DSMOTE over-sampled dataset. The over-sampling technique increases the dataset's size, increasing the number of features available for qualifying the models. This augmentation of the dataset helps to achieve a better fit of the models and enhances their overall performance.

Figure 10 illustrates that among all the models evaluated, the Random Forest model with TF-IDF Vectorizer achieves the highest accuracy score of 0.81, surpassing the performance of all other models when trained on Bag of Words (BoW) features. This outcome indicates the effectiveness of TF-IDF features and highlights the advantages of the Random Forest model in sentiment analysis tasks.

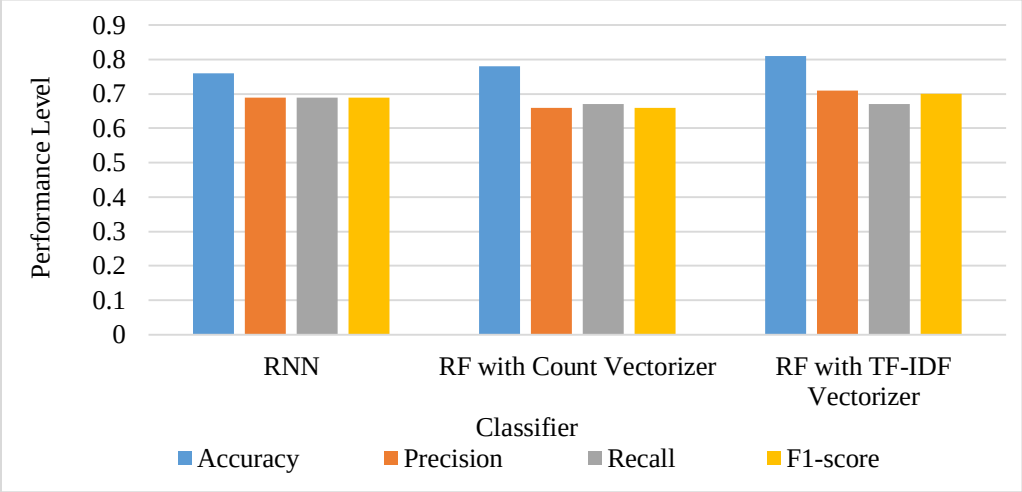

Fig. 10 Impact of SMOTE on performance metrics for ML classifiers

5. Conclusion

This paper classified tweets using machine learning. A model to analyse and categorise tweets was needed because Twitter generates so many tweets daily. This paper used two feature extraction techniques for model training: TF-IDF and the Bag of Words (Count Vectorizer).

The performance of the models could have improved when trained on the imbalanced dataset. However, we observed a significant performance improvement when the models were trained on the balanced dataset. Resulting in greater classification accuracy. Three classification methods predicted six airline tweet emotions. The Random Forest classification model trained on TF-IDF vectorised data performed best. The DSMOTE over-sampled dataset yielded an 81% accuracy and 70% F1 score across all three sentiment classifications. According to the data, airlines must identify the customer experience aspects most likely to affect positive or bad outcomes. These models help airlines respond to unfavourable tweets faster than traditional survey methods. This helps them avoid or minimise daily business disruptions of such events.

References

[1] Bing Liu, *The Problem of Sentiment Analysis, In Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, pp. 18-54, 2020. [CrossRef] [Publisher Link]

[2] Hamed Nozari, Javid Ghahremani-Nahr, and Agnieszka Szmelter-Jarosz, "AI and Machine Learning for Real-World Problems," *Advances in Computers*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Bing Liu, *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*, Data-Centric Systems and Applications, Springer Science & Business Media, pp. 1-532, 2007. [Google Scholar] [Publisher Link]

[4] Richard Socher et al., "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631-1642, 2013. [Google Scholar] [Publisher Link]

[5] Jerzy BÃlaszczy´nski, Jerzy Stefanowski, and Marcin Szajek, "Local Neighbourhood in Generalizing Bagging for Imbalanced Data, *Conference: The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 1-15, 2013. [Publisher Link]

[6] Chris Seiffert et al., "RUSBoost: Improving Classification Performance When Training Data is Skewed," *2008 19th International Conference on Pattern Recognition*, pp. 1-4, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nitesh V. Chawla et al., *SMOTEBoost: Improving Prediction of the Minority Class in Boosting*, Knowledge Discovery in Databases: PKDD, vol. 2838, pp. 107-119, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[8] Akash Yadav et al., "Sentiment Analysis Using Twitter Data," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 5, pp. 5833-5837, 2023. [CrossRef] [Publisher Link]

[9] S. Celine, M. Maria Dominic, and M. Savitha Devi, "Logistic Regression for Employability Prediction," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 2471-2478, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Huay Wen Kang et al., "Sentiment Analysis on Malaysian Airlines with BERT," *Journal of the Institution of Engineers, Malaysia*, vol. 82, no. 3, pp. 47-52, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Akash Gautam et al., "Semi-Supervised Iterative Approach for Domain-Specific Complaint Detection in Social Media," *Proceedings of the 3rd Workshop on E-Commerce and NLP*, pp. 46-53, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Arijit Dey, Jitendra Nath Shrivastava, and Chandan Kumar, "Transformer Based Knowledge Graph Construction in Adverse Drug Reactions Prediction from Social Media Reviews," *International Journal of Engineering Trends and Technology*, vol. 70, no. 10, pp. 402-407, 2022. [CrossRef] [Publisher Link]

[13] Shengyang Wu, and Yi Gao, "Machine Learning Approach to Analyze the Sentiment of Airline Passengers' Tweets," *Transportation Research Record: Journal of the Transportation Research Board*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Umer Hanif, Safiullah Khan, and Muhammad Hassan, "Sentiment Analysis Through Machine Learning Approach by Applying Random Forest Algorithm on Airline & IMDB Tweets," *International Journal of Computational and Innovative Sciences*, vol. 1, no. 3, pp. 1-11, 2023. [Publisher Link]

[15] Mohammed Matuq Ashi, Muazzam Ahmed Siddiqui, and Farrukh Nadeem, "Pre-Trained Word Embeddings for Arabic Aspect-Based Sentiment Analysis of Airline Tweets," *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, vol. 845, pp. 241-251, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Rajat Yadu, and Ragini Shukla, "A Hybrid Model Integrating Adaboost Approach for Sentimental Analysis of Airline Tweets," *Revue d'Intelligence Artificielle*, vol. 36, no. 4, pp. 519-528, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] M. Avinash, and E. Sivasankar, "A Study of Feature Extraction Techniques for Sentiment Analysis," *Emerging Technologies in Data Mining and Information Security*, vol. 814, pp. 475-486, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Prajwal Madhusudhana Reddy, "Conducting Sentiment Analysis on Twitter Tweets to Predict the Outcomes of the Upcoming Karnataka State Elections," *SSRG International Journal of Computer Science and Engineering*, vol. 10, no. 6, pp. 22-35, 2023. [CrossRef] [Publisher Link]

[19] İlhan Tarımer, Adil Çoban, and Arif Emre Kocaman, "Sentiment Analysis on IMDB Movie Comments and Twitter Data by Machine Learning and Vector Space Techniques," *arXiv*, pp. 1-8, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ubaid Mohamed Dahir, and Faisal Kevin Alkindy, "Utilising Machine Learning for Sentiment Analysis of IMDB Movie Review Data," *International Journal of Engineering Trends and Technology*, vol. 71, no. 5, pp. 18-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] El Barakaz Fatima et al., "Minimizing the Overlapping Degree to Improve Class Imbalanced Learning under Sparse Feature Selection: Application to Fraud Detection," *IEEE Access*, vol. 9, pp. 28101-28110, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Richard Socher et al., "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 129-136, 2011. [Google Scholar] [Publisher Link]