National College of Ireland

Comparative Analysis of Machine learning Algorithms using NLP Techniques in Automatic Detection of Fake News on Social Media Platforms.

MSc Internship
Cybersecurity

Manojkumar Murugesan
Student ID: 18129668

School of Computing
National College of Ireland

Supervisor: Christos Grecos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Manoj Kumar Murugesan		
Student ID:	x18129668		
Programme:	MSc. Cybersecurity	**Year:**	2019
Module:	Internship		
Supervisor:	Prof. Christos Grecos		
Submission-Due Date:	12/12/2019		
Project Title:	Comparative Analysis of Machine learning Algorithms using NLP Techniques in Automatic Detection Fake News on Social Media Platforms		
Word Count:	9207	**Page Count**	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……

Date: 12.12.2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Machine learning Algorithms using NLP Techniques in Automatic Detection of Fake News on Social Media Platforms.

Manojkumar Murugesan
x18129668

Abstract

Widening popularity of social media platforms and the increasing number of users trigger the spreading of fake news that creates chaos and tension in people's peaceful life. It is a vital interest to detect fake news, which has enormous potential to disrupts people's healthy growth. Traditional non-machine learning detection approaches like linguistic, network, and user profile analysis was deficient for dynamic and sophisticated social media network. Those conventional methods involved humans who are prone to make errors and take a lot of time. This research addresses this limitation using Natural Language Processing Techniques, along with machine learning algorithms. Our proposed system aims to detect fake news accurately, efficiently earlier, and with low false-positive rates. The first phase of the methodology involves cleaning the noises in the dataset and pre-processing to convert them into a Document-term matrix format. Literature reviews gave a spotlight on a few best performing machine learning algorithms. Such algorithms are Decision tree, Random-forest, AdaBoost, XGBoost and, LightGBM, which are well-known models for accurate and efficient text classification. The second phase of the research involved the evaluation of classification model in-terms evaluation metrics such as accuracy, precision, recall rate, f1score, and AUC. LightGBM with the bag of word technique performed magnificently with 96.3% of AUC and 93.1% accuracy rate. Lastly, an API was designed with the LightGBM model and deployed to achieve our goal of accurate detection with a low false-positive rate.

1 Introduction

The social media platform is a double-edged sword for news updates. [1] From one point of view, its easy access, economy, and one-stop platform for instant updates that lead people to follow-up and consume news from them. In another aspect of view, social media platforms pave the way to widespread fake-news which is defined as junky news crafted with false information. Fake news are written to deceive readers with three types of strategies, such as bewildering or fabrication, propaganda, and hoax, forgery information. Bewildering or fabricated news - They involve publishing irrelevant and doubtful information. Propaganda and Hoax –They deliberately mislead and deceive people by masquerading as news or with some information accidentally recognized by a traditional newspaper. Hence these biased or one-sided and deceptive news influence the target people's feelings, character, attitude, perspective, and activity for political and financial.

Nowadays, [2] the fake news has become a massive threat to freedom of speech, democracy, and journalism. Fake news will have a very bad or negative impact on the authenticity balance of the journalism ecosystem. The fact is evident from the fact that the fake news went viral through social media platforms like Twitter and Facebook during the 2016 US presidential election than any authentic news editions. Heights of reach during the month of the US election campaign was massive, where all the top fake news turned over 8,911,000 retweets and shares on Twitter. Fake news is so powerful that it persuades the consumers to believe any false information or biased. Fake news is very poisonous that could damage economic conditions and influence the results of political polls. Moreover, fake viral news manipulates people and affects the way they think and interpret the real news to create an unusual and chaotic situation in a society. Hence, to mitigate the negative effect of fake news in the community and to maintain balance in the journalism ecosystem, it is critical to build an efficient system with a feasible machine learning algorithm to automatically detect fake news on social media.

People's ignorance or reduced ability to differentiate lie and truth adds more significance for an automatic detection mechanism.Users on social media platforms are not aware of posts, tweets, news that are purposefully crafted to shape their beliefs and influence other's decisions. False information and its impact or adverse effects are different topics in anyone's mind when a piece of news shared by a dear friend. In recent times, it has become a trend that young people keep updated about politics, events, and breaking news from social media platforms. Therefore, deficiency in awareness, ignorance, improper ability to detect fake news adds up for automatic detection of fake news without any human intervention.

The existing online system of detecting fake news works based on manual verification by professionals, which have the main drawback of time latency. Most of the online fact-checking systems focus on reviewing political news, which is a limited filter for the detection of a variety of news, formats, and fails against the news spreading at lightning speed through social media networks. In addition to it, a higher volume of newsfeeds is shared, created and, commented via a social media platform that makes the detection work even more difficult.
[3] Hence, people have limited background knowledge to differentiate fake and real news. Therefore, it is desirable to build an automatic detection system to prevent the fake news from spreading so fast through complex social media like twitter, to handle a large volume of data and detect the quality of the news as soon as possible. Machine learning algorithm has shown promising performance in solving numerous complex problems like fake news detection on social media platforms.

To what extent can machine learning algorithms, along with NLP techniques like TF-IDF and bag-of-words, accurately classify fake news from real news with a low false-positive rate? A machine learning algorithm is considered to be the best when it has high accuracy and low false-positives. One such algorithm is LightGBM, which is a latest, robust classifier and highly optimized in terms of speed, memory, and accuracy rate. Other algorithms, such as XGBoost, AdaBoost, Random-forest, and Decision, which are well-known for its best performance, are used for comparison. The primary objective of this research is to evaluate classification models and design a system that detects fake news with high accuracy and efficiency.

2 Related Work

2.2 Non-Machine Learning Categories

Linguistic Approach

[4] Tan, Lee and Pang, 2014 have described the detection technique which is based on study of various communicative behaviors. The basic idea is about the psychology of truth tellers and liars, the way both communicate and share information on social media. There are few symptoms that can be noted from a liar such as their total word count would be higher than the actual count and usage of self-centered descriptive words instead of word describing public interest. [5] Hence these could be the cue for researchers to detect fake on social media. [1] Conroy, Rubin and Chen, 2015 have differentiated liars by the way or style of writing news content. There are three main sub-methods under Linguistic cues such as Data Representation, Deep Syntax, Semantic analysis, and Sentiment analysis. [4] Tan, Lee, and Pang, 2014 have tried to change a word in a tweet concerning the rate of propagation. The data consisted of tweets that were posted from the same URL but different wordings. The features that were selected is more linguistic such as length and keywords such as "share" "Pls share." He applied Logistic regression, which showed an accuracy rate of 98.8% based on the least, most retweeted, and followers count. However, when 39 different features were used, accuracy has fallen to 63%. The features included: proper nouns, length, pronoun, sentiment, adjectives, positive words. The research aimed to explore fake news by discovering features that influence propagation. Syntax analysis was another efficient approach for detection of fake news. [2] Feng and Hirst, 2013 have done an experiment based on a stylometric approach to detect fake and real news based on the writing style. The investigation used the Buzzfeed data set of mainstream and Hyper partisan news articles of which the quality was manually marked. Stylometric features contained characters, n-gram stop words, characters count per paragraph, and a bag of non-specific domain words. The dataset consisted of 1,627 news articles obtained from Buzzfeed that includes 299 fake news. Though the stylometric approach was promising to classify, it didn't efficiently detect fake news from real. Sentiment analysis comes in the row to classify fake news and real news. [3] Rubin *et al.*, 2016 have used satirical cues to differentiate fake news and real news. The approach was based on the error in punctuation, text, grammar, and showed a 90% precision rate and 87% recall rate. However, this approach is still not alright as the trustworthiness and proof-of-facts are marked with less priority.

Drawbacks: The primary disadvantage of using a linguistic approach is its generic nature in terms of language, topics, subject. Hand-made cue sets could show relativity for a specific situation. But the method involves humans that might have an impact on human errors and negligence. Although the n-gram word approach with in-depth syntactic features, PCFG and linguistic analysis is better than the cueing technique, they fail to exploit a semantic and syntactic functionality in the content to a greater extent.

Network Approach

[1] The difference between the linguistic approach and network approach is that the network approach involves humans and requires the right knowledge to detect fake news. This approach is about literally assessing the truthfulness of a news article manually. This process lays down a base for the further progress of fact-checking from an outside source. There are three present fact-checking such as Computational oriented, Export-oriented, and Crowdsourcing oriented. [6] Ciampaglia et al., 2015 have shown the promising methodology of seeking structured databases and derived several truth probabilities. Results of tracing an article with four different areas of subjects gave 60 % to 95% accuracy. This success rate was marked when the machine awards higher value for truth and lower value for false news. Though it is a pretty working model, the subject matter should remain in the knowledge base.

Drawbacks: The approach requires human intelligence for the fact-checking process. The methodology is not so efficient to differentiate fake news from real news as it involves human error and requires a lot of time.

User Profile Analysis

[1] User profile characteristics and behavior are very well concerned by most of today's social media platforms like Twitter, Facebook, and e-commerce websites like amazon. [8] Tacchini et al., 2017 have constructed a feature vector that contained the activities of users who have shared or liked or commented on them. The obtained feature vector is trained with logistic regression and took forward for a trial to study the user's behavior pattern to classify fake news from real.

Drawback: Feature extraction could be more specific to a particular social media platform, and features couldn't be generalized at all the platforms.

2.3 Machine learning Categories

Deep Learning Approaches

[7] Deep learning techniques have become more popular approaches among researchers in recent years. The methods have shown promising results, among other machine learning techniques. However, the feature extraction technique is more time-consuming, leading to biased predictions. Hence, this kind of drawback stands as a considerable barrier to select features or predictors.

There are two most widely used algorithms such as CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network)

Recurrent Neural Network

RNN is a sub-class of Neural network which forms a directed graph that contains nodes connected in sequential order to build a structure. [7] This structure is mainly termed as LSTM (Long Short-Term Memory). The main application of the technique is to process human language, Test-analytics, and extract relevant features from a set of information. [7] Ma, Gao, and Wong, 2017 have employed a detection technique involving a combination of temporal

text features and LSTM. The temporal features concerned about the importance of the news, elements of the users, text characteristics, and shown accuracy rate 0.89 – Temporal data feature and 0.95 – LSTM. [9] Kochkina, Liakata, and Zubiaga, 2018 have designed a multi-task training framework with the LSTM approach and crafted each layer with a specific task. Their dataset showed an accuracy rate between 0.36 and 0.64 in rumor detection.

Convolutional Neural Network

Though CNN is applied mainly for image recognisation and processing, which is in the current state of the art for vision applications, they are also applied in natural language processing applications such as fake-news detection, rumor classification. [10] Chen, Liu, and Kao, 2018 have built CNN based methodology to classify tweets based on stance and veracity that showed 0.70 and 0.53 of accuracy, respectively, during evaluation.
Drawbacks: Deep learning technique has a promising feature extraction facility to detect fake news, but the decision could be bias only with text analytics and hence demands fact-checking also. Literature reviews have shown the classification of biased political, fake news, and rumors have shown 65% to 78% of accuracy on average with CNN.

2.4 Machine Learning Classification Algorithms

Decision Tree Algorithm

Decision Tree algorithm is one of the tree-based classification methodologies.[11] The technique uses a predefined variable to work on it. The structure of the algorithm is developed in a top-down fashion, and its primary function is to divide the large volume of the dataset into small clusters based on the applied decision rules. [11] Ozbay and Alatas, 2019 have performed supervised learning on two sets of fake news and real news datasets. The authors have used a set of features along with various supervised learning algorithms such as Bayesnet, JRiP, OneR, Decision Stamp, Xero, SGD (Stochastic gradient descent), CV parameter selection method, Randomizable classifier, Logistic regression, Decision tree algorithm, and five other algorithms. Among the fourteen algorithms, the Decision Tree algorithm topped the table with 0.968, 0.963, 0.973 of accuracy, precision rate, F-Measure, respectively. Hence Decision tree algorithm outperformed the other 13 algorithms in classifying fake news with an accuracy rate of 0.968. [12] Sabbeh and Baatwah, 2018 have classified false Arabic tweets from 800 Arabic news datasets collected from twitter. Dataset was trained and tested with Five- fold cross-validations along with three different classifiers such as SVM, Naïve Bayesian, and decision tree. Results have shown that the Decision tree algorithm outperformed SVM and Bayesian with 97%. Hence, the above pieces of literature have proven that the Decision tree algorithm to be the best and outstanding classifier among SVM, Bayesian, and 13 different algorithms experimented by Ozbay and Alatas, 2019 [11].

Random Forest Algorithm

Random forest algorithm is an ensemble-based algorithm which is proposed by Leo Breiman. They use bagging methods to train different decision trees. They keep track of features and predictors for better performance and best accuracy results. They show low bias and good variance in prediction. Great randomness in the dataset might lead to overfitting for the model. [13] Kwon et al., 2013 have used twitter datasets that exploits 1.7 billion public tweets that have been posted for three years from March 2006. The classification was based on specific features such as Temporal features, structure features (User characteristics and friends network), and Linguistic features. Authors have used three classifiers such as SVM, Decision tree, and random forest algorithms with a totally of 27 features. The first round of classification was made with 11 different features where random forest algorithm stands first in the table with 0.90 accuracies and second-round performed with 27 features(combining 1st round features and another 9 features) where random forest again outperformed other with 0.89 accuracies. [14] Azab et al., 2016 have tried to detect fake accounts in twitter using various algorithms such as random forest, Decision tree, naïve Bayes, neural network, and SVM. The authors have extracted some 22 features. Each feature has been weighted and marked by GAIN measurement. Totally three rounds of classification have been performed, in which subsequent rounds were classified with a reduced number of features. Rating with all 22 features has shown a random forest algorithm to achieve higher than others. RF algorithm stood out from the crowd with TN% of 94.69.

Gradient Boosting Algorithm

Gradient boosting algorithm is widely called GBM, which is a representation of knowledge in the form of a structured tree graph where each internal node possesses their own decisions, and each branch possessing a leaf node that expresses the result of the entire tree. Boosting is one of the methodologies and an ensemble method to optimize week regression trees by iteratively train and attain efficient training objectives.

Adaptive Gradient Boosting Algorithm

[16] Yuan *et al.*, 2019 Yuan et al., 2019 have proposed a semi-supervised tri-Adaboost methodology to detect network intrusions. The authors have selected a set of features using the chi-square method to improve classification efficiency. The adaptive boosting algorithm is described to be highly efficient and suitable for accurate classification. Authors have used the KDD mining dataset of 1999 rows and extracted 18 different features from them for evaluation of the algorithm. SVM was used for comparison and evaluated on the basis of detection rate, FP, Precision rate, and Detection time. Adaboost outperformed SVM with false positives of 75.4% and a precision rate of 98.54. [17] Markines, Cattuto, and Menczer, 2009 have classified spam on social media with 27,000 user's datasets in which 25,000 were manually marked as defaulters and remaining 2000 as legitimates. The classification was done based on six different features and employed two other algorithms, such as SVM and SMO, along with AdaBoost. The results have shown that AdaBoost has raised the accuracy rate than the previous works of their literature review (LogitBoost-97%). Authors have achieved 98% of accuracy and 2% of FP by iterating 1000 times. Hence, works of literature show AdaBoost to be the best classifier

among the algorithms that motivates the research to evaluate them further with a different dataset.

Extreme Gradient Boosting Algorithm

The algorithm is shortly termed as the XGBoost algorithm, which is optimized with advanced memory resources, flexibility, and efficiency. [16] Manju, Harish, and Prajwal, 2019, have built a model using the XGBoost algorithm to make the dataset fit into the model, improve efficiency and accuracy. Authors have tried to classify internet traffic to maintain the network and its quality of services. Authors have used two sets of Network traffic datasets along with eight different features. To compare and evaluate performances of classifier algorithms, authors have used the Decision tree, random forest, AdaBoost, and XGBoost for accurate classification. As expected, both the dataset has proven the XGBoost to be efficient with 96.97% accuracy (Dataset1) and 87.48% (Dataset2). [18] Kalra, Li, and Tizhoosh, 2019 have proposed a methodology to detect pathology reports and predict primary diagnosis automatically. Authors have derived features based on TF-IDF (Term Frequency-Inverse Document Frequency) from the pathology report. Authors have obtained around the 1947 pathology report, which was related to four different organs such as Thymus, testis, kidney, and lungs. For the classification and evaluation, classifier combination of SVM, XGBoost, and Logistic regression have been used. Finally, classification results have shown higher accuracy by XGBoost with 92% and outperformed SVM – 87% and logistic regression. Authors have found the result to be encouraging about classification using a machine learning algorithm, and TF-IDF feature extraction helped them to lookout suitable keywords for the early diagnosis.

LightGBM Algorithm

LightGBM works based on the histogram algorithm, and it is an open-source algorithm which was developed by Microsoft to increase efficiency, accuracy computation power, and memory resources. [19] While other GBM algorithms grow horizontally by level-wise, but lightGBM grows by leaf-wise vertically. [19] Machado, Karray, and de Sousa, 2019, have implemented a methodology to predict customer's credibility in a financial company. The authors have used the dataset from a company called ELO. Around 230 features were analyzed to predict the customer's credit card loyalty score in the company. Two classifiers, such as XGBoost and LightGBM, were used for the prediction where LightGBM has performed better than the XGBoost algorithm. Efficiency was measured in terms of RMSE (Root mean squared error), with fixed learning time and iteration number for both XGBoost and LightGBM. Measure have shown LightGBM outperforms XGBoost and poses to be the best algorithm for prediction. [20] Ke et al., 2017 compared the efficiency of LightGBM with XGboost using four different datasets. In the methodology, authors have used two variances of XGBoost and evaluated the training timings and accuracy rate of each of them. On analysis, LightGBM has shown the best accuracy rate and 9x speed rates than other algorithms. Hence the literature indicates that LightGBM significantly overrides XGBoost in the context of performance, speed, and memory.

From the above literature, machine learning methodology overcomes the drawbacks of non-machine learning such as manual fact-checking that avoids human error, deficiency in computational power, memory management, and accuracy. We have gained a variety of algorithms that performed well over the other. In work [11] [21] [12] Decision tree has outperformed around 18 algorithms, including a few prominent algorithms such as SVM, Bayesian in fake news detection. Works [13] [22] [14] random forest have outperformed SVM, Logistic regression, neural networks, and have shown promising results in fake news detection. GBM stands out with more advanced computational power, memory utilization, training time, functionality, and accuracy than the above toppers. Three most prominent GBM algorithms were reviewed and could be ranked in descending order as LightGBM, XGBoost, and AdaBoost, where LightGBM overrides XGBoost and XGBoost takes on AdaBoost. LightGBM has shown outstanding performance in the context of the financial application and tumor classification. Hence, algorithms such as Decision tree, Random Forest, AdaBoost, XGBoost, and LightGBM that have stood out in the above works of literature are used for performance evaluation and comparison.

3 Methodology

Cross-Industry Standard Process for Data Mining methodology is commonly termed as CRISP-DM. This research gets along with the CRISP-DM method. CRISP-DM methodology helps researchers to design intricate and reliable process model. The life cycle of the plan is stacked into 6 phases, as shown in Figure1. And the sequence of the process needn't be in order. The result obtained from the previous step could influence the next step in the process model. Hence the outcome of each phase would add up more benefits to the subsequent stages in the model.

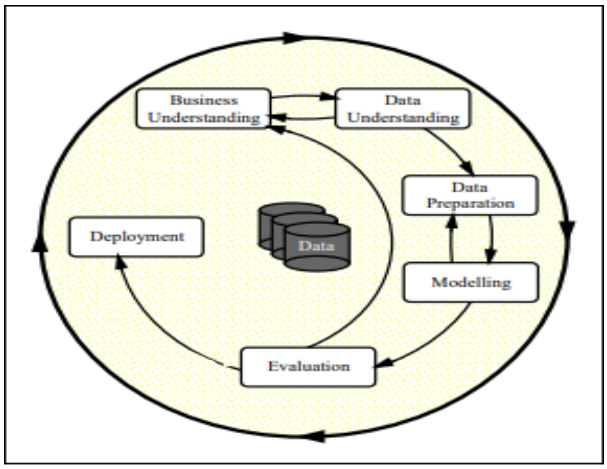


Figure1: CRISP-DM Life cycle Model

3.1 Business Objectives

Discovering the research problem from a business perspective is the first stage of the process in CRISP-DM methodology. The primary goal of this research is to detect fake news efficiently and automatically with low false positives and higher accuracy, which is deficient in previous

deduction works. The project plan is to experiment classification of fake and real news datasets with advanced gradient boosting algorithms and ensemble methods along with NLP techniques such as Bag of words and TF-IDF. News datasets are available in online dataset repositories like Kaggle. The experiment is carried out with an assumption of attaining a low false-positive score and a higher accuracy rate with LightGBM and aimed to deploy a simple API to visualize the classification model.

3.2 Data Understanding

The second phase of the methodology is to gather data and understand them in context of variable name, records, format, and cleanliness. Three sets of datasets were chosen, and their sources and description are tabulated as follows.

	Source	Quantity	Predictors	Categorial Variable	Predicted - Quantity
Dataset 1	https://www.kaggle.com/c/fake-news/data	11457 rows	Id, Title, Author, Text	Label- 1 – Fake news, Label - 0- Real news	1s-5811 0s-5616
Dataset 2	https://www.kaggle.com/snapcrack/all-the-news#articles1.csv	9423 rows	Id, Title, Publication, Author, Date, Year, Month, Text	Label – 0 – Real news	0s-12089
Dataset 3	https://www.kaggle.com/jruvika/fake-news-detection	12131 rows	Uuid, ord_in_thread, author, published, title, text, and 12 columns	Label – bs, bias, Conspiracy, hate, satire, state, junksci, fake.	1s -6940

Table.1: Dataset Description

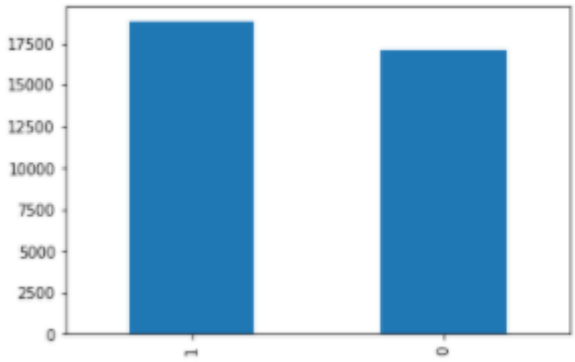
All three dataset weren't clean that contained unwanted symbols like /.",_#$%@!()^*&®Âš¯ŽÃƒ™€¢Â£Đ½Đ½Ñ‹Đµœâ€”©˜ºŸĐ²Đ°ÑˆĐ and so on. A few empty rows were often present amid of other rows in the dataset. Though base dataset dataset1 is balanced with some 0s and 1s, very few counts could lead to biased classification by the models. Hence, it is required to pre-process the chosen datasets to patch defects for accurate classification.

3.3 Data Preparation

As mentioned in the previous section, dataset 1(base dataset) possessed very few 0s and 1s that required oversampling to prevent biased classification. Unwanted noises amidst the data are blacklisted and removed by creating a module using the visual basic code base. Empty rows in between the rows are filtered and removed using =ISEMPTY command in the excel sheet. Except for the common columns such as text, author and label, other unwanted columns were removed from dataset2 and dataset3. Finally, all three datasets were merged and obtained a clean and quality version of a dataset with enough 0s and 1s for fair and accurate classification. Further, the final version of the dataset is pre-processed to fill in empty records, and NLP techniques like stemming, removal of stopwords, normalization, and tokenization were implemented to make a good fit for the machine learning algorithms. The description of the final version of dataset after final pre-processing is shown in Table 2 and visualized in graph 1.

Dataset 4	Quantity	Predictors	Categorical variable	Predicted Quantity
	35952 rows	Author, label, text, title	Label – 1 – Fake news Label - 0 – Real news	1s – 18809 0s – 17143

Table 2: Description of Pre-Processed Dataset



Graph 1: Upsampling 0s and 1s

3.4 Modelling

Machine learning algorithms cannot be directly fed with raw text that requires the data to be in numerical format. The first step of the process is to transform the string of characters in the document-term matrix format that can be efficiently interpreted by machine learning algorithms. Therefore, our research follows two NLP techniques such as TF-IDF and bag of words to transform the text into a document-term matrix format. Appropriate NLP techniques should be used for each machine learning algorithm for efficient and quick classification. For instance, bag-of-words is inefficient in Random forest and Decision Tree algorithms. Once

text data is transformed, input values suffer from high dimensional data, which is a major problem in the data mining process. Such high dimensional dataset is handled by removing redundant values and extracting required features during the data-processing phase.

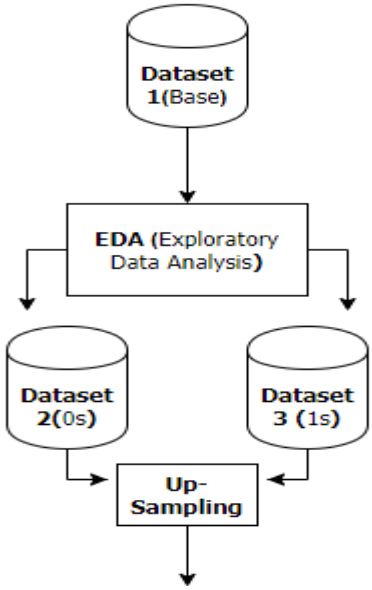
After Natural Language Processing and feature extraction processes, the dataset is fed into machine learning algorithms. Machine learning algorithms such as LightGBM, XGBoost, AdaBoost, Decision Tree, and Random forest were chosen for its outstanding performance in the previous works. Further, each classification model was fine-tuned with the best parameters through the hyper-parameter tuning technique. In the end, a simple fake news detector API is created with a top-performing classification model using a swagger tool for visualization.

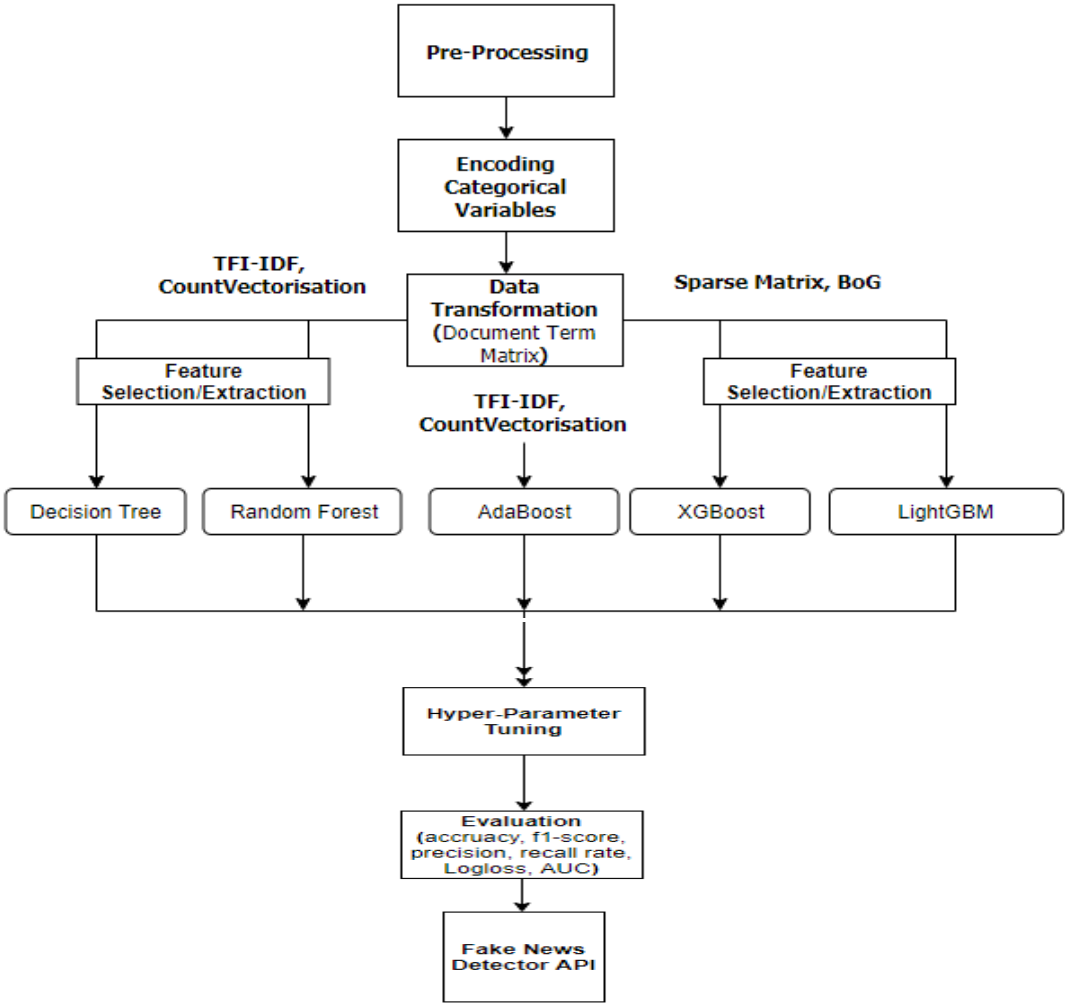
3.5 Evaluation

The fifth phase of the CRISP-DM methodology involves performance evaluation of classification models and measured with metrics such as accuracy, precision, recall, f1-score, Log loss, and AUC. The confusion matrix depicts false positives that help us to assess our objective and rank the best classification algorithms.

3.6 Deployment

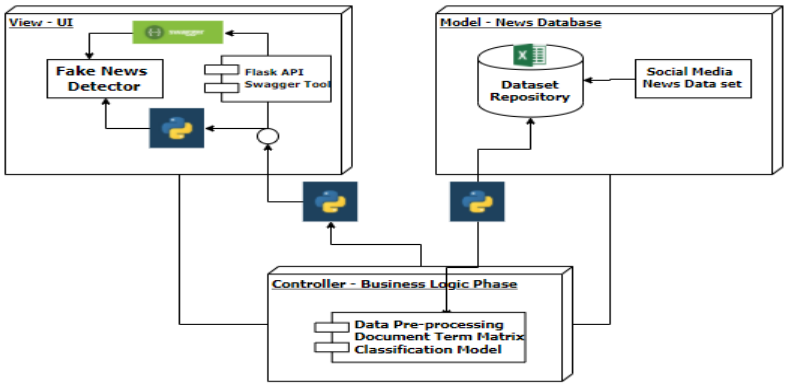
Deployment is the last phase of our methodology were in, the efficient feature selection model and classification algorithms are identified to design a real-time fake news detection API. Once an API is developed, they are required to be monitored and maintained regularly to keep a check on their performance. On the excellent performance of the model, it can be deployed on a large scale social media platform, which is the future scope of this research.Research is carried out in a flow as shown below.





4 Design Specification

A highly effective and efficient fake news detector application can be developed with a successful architecture model and a structured design. MVC architecture model is used to design our classification API as shown in fig below. MVC is a one of the most commonly used model frameworks to develop a flexible and cost-effective project. Proposed design is built with three separate components such as View – User Interface developed with python FlaskAPI and swagger tool, Model – handles data repository from where data flows between View and Controller for training and detection, Controller - which acts as an interface between View and Model that carries out business logic tasks, data manipulation, training, and testing datasets.



5 Implementation

5.1 Environmental Setup

This research is conducted in google cloud platform called colab which is collaboration of free jupyter notebook environment. They don't require any library to install and can write code and execute entirely on cloud. Each colab session is equipped with powerful computing resources such as 13GB of ram and choice of processors like GPU, TPU, and CPU. This project was executed in a session allocated with 8GB ram and chosen GPU processor for faster execution.

5.2 Data selection

The base dataset was obtained from URL: https://www.kaggle.com/c/fake-news/data. It is a competition dataset that was openly available for access and hence, no ethical issues. Since it was published for kaggle competition, the dataset was well structured and almost cleaned. It had 4 columns and 11457 rows with a class variable labeled as 1s as fake news and 0s as real news. Quantity of 1s and 0s were very well balanced, with around 5000 rows each. The lower volume of datasets might lead to biased classification at the end. Hence, the dataset requires up-sampling for the best results. WordCloud is used to visualize the dataset, as shown in fig below. The key feature of the visualization is that they exhibit the most frequently occurring words in large font size and its significance.

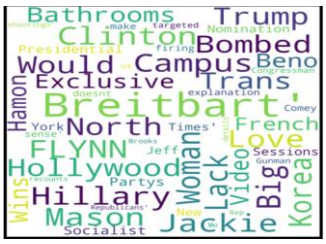


Figure 2: Dataset Visualization

5.3 Data Up-sampling

As our base dataset required up-sampling as discussed in the previous section, two sets of datasets for fake and real news were chosen and merged with the base dataset. Fakenews 1s were obtained from URL: https://www.kaggle.com/jruvika/fake-news-detection and real news 0s were obtained from URL: https://www.kaggle.com/snapcrack/all-the-news#articles1.csv. Now, the number of class variables should have increased for better training and testing datasets. The upsampled dataset has been visualized in graph 2. Hence, the random upsampling technique solves the risk of biased classification of news dataset.

5.4 Data cleaning and Pre-processing

Once a right dataset is chosen to meet our objectives, it is required to perform a set of pre-processing and cleaning tasks. Usually, social media datasets won't be in a proper structure and exists with unwanted symbols, characters and typos. Hence, Pre-processing avoids any noise in the data and helps models to run without any hassles. Cleaning process is carried out with NLP techniques such as removal of unwanted columns, stopwords, and null values, stemming and normalisation.

5.4.1 Removal of Unwanted Columns

This is a fundamental step in data cleaning process. The purpose of removing unwanted columns is to eliminate noise in the dataset. The dataset had only four columns in which three columns such as author, text, and title were obvious to affect our categorical column 'label'. Preprocessed texts are stored in a separate column. Hence, python function 'drop' was used to remove the column with raw data and irrelevant columns.

5.4.2 Removal of non-English characters

The objective of this process is to remove special symbols and characters, as shown in section 3.2. They contribute to eliminate noises in the dataset that helps to protect classification models from bad effects. A basic filter function in python is used to clean-off the non-English characters.

5.4.3 Removal of Stopwords

Process of removing stopwords is an essential task in order to optimise the performance of the classification models. Basic idea is to remove the words such as a, the, an, and other articles which are insignificant for prediction. This process prevents the addition of excessive dimensionality to the dataset and thus enhance the performance of the model. Stopword function is imported from NLTK library for the cleaning process. This function removes all those words that matches with stopword library.

5.4.4 Removal of null values

A basic cleaning process to fill in empty records with dummy values. The process is carried out with 'fillna' python function. This helps to enhance cleanliness in the dataset and performance of the model.

5.4.5 Stemming

Stemming is a process of transforming all the verb, noun, adverb and adjectives into their root word that unalters the meaning. The main objective of the process is to derive root word from the word expressed in different form. This helps us to avoid complexity in dimensionality after document metric term process. Stemming is done using 'snowballstemmer' function derived from NLTK library.

5.4.6 Normalization

This is again a method to reduce complexity in dimensionality after transforming the text data into document-term matrix. The basic idea is to standardize text in a uniform format which helps the classification model to predict more efficiently and reduce runtime.

5.6 Data Transformation – Document-Term Matrix

The biggest challenge in text classification problem is handling large volume of dataset. Hence, it is required to remove redundant and irrelevant texts to handle high dimensional datasets and improve classification accuracy. The objective of the process of transformation is to point out text that has frequency of occurrence larger than the threshold value and been weighted to represent in Vector Space Model. In VSM representation of dataset, each text is assigned with a numerical value that depicts its weight in the dataset. It also serves another purpose of converting text data into numerical terms which could be efficiently interpreted and predicted by machine learning algorithms. Following are the techniques used to transform dataset into Document-term matrix.

5.6.1 TF-IDF

TF-IDF is text mining and feature extraction technique, proposed by Spark Jones. TF-IDF is abbreviated as Term frequency – Inverse document frequency. The main of the techniques is to calculate the significance of a word to a document in a dataset. The weight of a word is directly proportional to the count of occurrences of a particular word in the document. Classical formula used by TF-IDF to calculate weight of a word is shown below.

$$W_{ij}=tf_{ij}*\log(N/df_i)$$

Where, W_{ij} is the weight of the word I in document j,

N is the number of documents,

tf_{ij} is the term frequency in document j,

df_i is the document frquency.

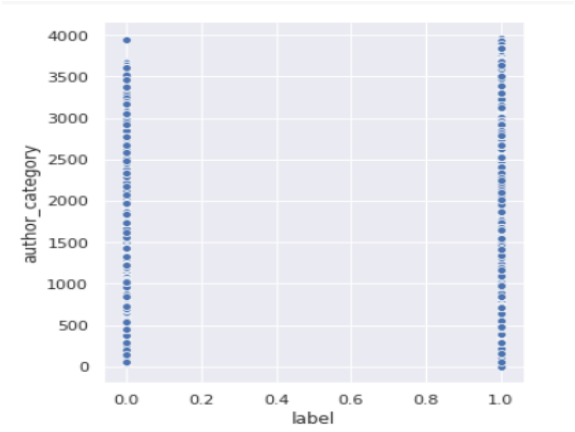
Tf - that is, term frequency is a measure of the frequency of a particular word in the document. A word can occur several imes in a lengthy document or short document. Hence Term frequency is calculated by a number of occurances of the word divided by length of the document.
IDF – stands for Inverse Document Frequency. It is a measure of the importance of a word in a document rather than frequency measure as in TF, where every words were given equal importance. The basic principle is that they weigh down the most frequently occurring terms in the document. For instance, articles like is, the, was, and so on were weighed down as they often occur in the document.
In this project, TF-IDF is implemented using 'TfidfTranformer' function imported from sklearn.feature_extraction library. Initially, the text is tokenized using the Countvectorizer function to encode text with frequency value and fed into 'TfidfTranformer' to calculate tfidf score for text. Transformed text is given as input into machine learning algorithms.

5.6.2 Word Bag

Bag of words is another technique of converting text data into vector matrix that the machine learning algorithms would easily interpret and processes them quickly. They help machine learning algorithms to extract features from the given dataset. The basic idea of the technique is to encode text data based on its occurrence in the dataset. WordBatch library is used to import 'wordbag' function to transform text data in this project. One of the special features about the wordbag is that they have n_grams parameter where necessary features such as uni-grams or bi-grams of text can be derived. Additionally, they have the feature of calculating the weight of the n-grams using tf and idf parameters. The processed dataset is finally fed into the machine learning algorithm for classification.

5.7 Label Encoding

Label encoding is a process of converting class variables into a numerical value. Encoding categorical variables is essential to train machine learning algorithms on how to interpret class variables in a data set. For instance, we have column 'author' with authors' names who have composed the news articles. Encoding column 'author' helps the classification model to identify a news article easily and consider it as one of the factors to predict fakeness in text. Hence, column 'author' is encoded using labelEncoder() function. The final dataset generated 3994 different authors assigned with unique ID. Similarly, column 'label' is encoded with 1s for fake news and 0s for real news for efficient training and accurate prediction. Correlation statistics between 'author' and 'label' is visualized in graph.2



Graph 2: Correlation between 'author' and 'label'

5.8 Fake News Classification Using Machine Learning Algorithms

5.8.1 Decision-Tree Algorithm

The decision tree algorithm is a data mining approach that iteratively split dataset based on a depth-first greedy approach or breadth-first approach until all the text data comes under a single class (root node). Decision algorithm process dataset in two phases. The first phase is tree building, and the other one is tree pruning. Tree building is a phase where the dataset is split in a top-down fashion until all text comes under a particular class. Fundamentally, decision tree implementation requires choosing of right input variables such as target variable and one or more predictors and making a perfect split on predictor variable until an efficient tree is built. A decision tree structure is made of a root node, internal node, branch, and leaf. Tree pruning is done in bottom-top fashion in order to reduce overfitting and improve accuracy rate. An initial experiment with default parameters performed well with the TF-IDF matrix and showed relatively low false positives with an accuracy rate of 0.8713.

5.8.2 Random-Forest Algorithm

Random forest is an ensemble-based machine learning algorithm that handles data complexity by grouping them under a class. Breiman and Cutler proposed the algorithm. Building the model involves training datasets and predicts the target variable by considering the collective votes of the various trained sub-trees. The maximum number of votes is considered as output. An initial experiment with a combination of TF-IDF and default parameters didn't perform well compared to other classifiers. The algorithm predicted target values at an accuracy rate of 0.8068 and classified with low false positives. The algorithm required Hyper-parameter tuning to boost its performance.

5.8.3 Ada-Boost Algorithm

Ada-Boost is the best classier for predicting two or more classes. They are much efficient that enhances week classifiers using – {hm(x)}. Boosting feature helps week classifiers to perform better than its actual potential. Boosting mechanism involves combining week classifiers into a strong H{x} using scalar weights {am} on every round. The overall performance of AdaBoost is good, with a combination of default parameters and TF-IDF text format. It classified with an accuracy of 0.8918 and gave very low false positives.

5.8.4 XGBoost Algorithm

eXtreme Gradient Boosting algorithm is shortly called as XGBoost. They are an efficient and scalable version of the gradient boosting algorithm, which was proposed by Friedman in 2001. The package has compatibility of solving linear and tree-training problems. They are also capable of supporting regression, classification, and ranking functionalities. The following are features of XGBoost.

Computation Speed: Algorithms work fast with the 'GPUhist' parameter under the tree_method attribute. They utilize GPU resources for fast training and testing datasets. Hence the computational performance of the XGBosst algorithm is enhanced with GPUhist settings.

Input Data: XGBoost worked well with sparse matrix input data as the bag of words technique involves the process of transforming the dataset into a sparse matrix in a horizontal fashion. The sparse matrix has shown optimal computational performance and a good fit for testing and training datasets using XGBoost than TF-IDF. The transformation is done using the 'hstack' function imported from the 'sparse' library. XGBoost is a well-known machine learning algorithm for efficient text classification. As literature showed, XGBoost performed excellently in classification and gave very low false positives. It has given 90.59% of accuracy in classifying fake news from real news.

5.8.5 LightGBM Algorithm

LightGBM is a variant of the Gradient Boosting algorithm like XGBoost. They are relatively new GBM algorithms and works based on tree learning methodology. They are much efficient in terms of speed, memory consumption, and accuracy rate. They deal with continuous values as discrete terms, which in return makes training much faster and reduces memory consumption. GPU resource utilization boosts up the speed further higher. A significant volume of the dataset is interpreted quickly with parallel learning support. The mechanism of producing complex tree by leaf-wise splitting tends to outperform other machine learning algorithms in terms of accuracy rate. LightGBM algorithm with a combination of default parameters and bag of words in sparse matrix input data has outperformed different best classifiers with the highest accuracy rate of 0.926 and very low false positives. It has shown the best precision, f1 score, and recall rate with an average score of 91%.

5.9 Optimization using Hyperparameter Tuning

Hyper-parameter tuning is a process of optimizing the performance of machine learning algorithms with the right combination of hyper-parameters. Though it is a tedious and time-

consuming process, they are required by classification models for maximum accuracy rate. The manual way of setting hyperparameter to a model requires knowledge of optimal model architecture and explore a range of combination of parameters. But, there are automated ways of obtaining hyperparameters like grid search and randomized search techniques for the best performance of a model.

5.9.1 Randomized search of Hyperparameter

Randomized search technique is the best way of choosing the right combination of parameters for a model than a grid search technique. The basic methodology behind the method is that they select random samples of hyperparameters based on statistical distribution. Since not all parameters are significant for a model, randomized search of hyperparameter is the best technique than a grid search.

Hyperparameter tuning has optimized random-forest algorithm with an improvement of accuracy up to 10% than with default parameters. XGBoost, LightGBM, AdaBoost showed only 1% improvement in accuracy rate with tuning.

5.10 Evaluation Metrics

Evaluating the performance of classification models is a crucial phase of a research project. This phase gives out the performance progress of the classification model with specific metrics, which are defined as follows.
TP – True positives
TN – True Negatives
FP – False Positives
FN – False Negatives

5.10.1 Accuracy

Accuracy is a standard evaluation metric that depicts how good a machine learning algorithm has performed in solving a problem. Typically, it is measured by the ratio of the number of correct fake news predictions to the total number of input news samples.

Accuracy: (TP + TN) / (P + N)

5.10.2 Precision

Precision rate is the ratio of the number of correct fake news detection to the number of actual fake news. The expression is as shown below to calculate the precision.

Precision: TP / (TP + FP)

5.10.3 Recall

Recall rate is otherwise called a Sensitivity rate, which is calculated as correct fake news detection divided by the total number of fake news. It is the number of true positives. The formula is as shown below.

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{\mathrm{TP}}{\mathrm{P}}$$

5.10.4 F1-Score

F1- score is a calculation of harmonic mean of precision and recall rate.

$$\mathrm{F}_1 = \frac{2 \cdot \mathrm{PREC} \cdot \mathrm{REC}}{\mathrm{PREC} + \mathrm{REC}}$$

5.10.5 AUC

Area under curve is shortly called as AUC. It portrays the result of an aggregated measure of a machine learning algorithm in-context of classification threshold. They depict information on which class is classified best, by a machine learning algorithm. AUC results are considered significant than the accuracy rate, as they don't be biased with the size of the input data. AUC is calculated based on recall and precision rate. AUC score ranges from 0-1.

6 Experiments and Result

Experiment 1: Decision Tree algorithm – TF-IDF Document matrix

```
confusion matrix before tuning
[[4486  628]
 [ 599 5073]]
```

```
confusion matrix after tuning
[[4576  538]
 [ 631 5041]]
```

Algorithm	Accuracy	f1-score	Recall-Rate	Precision	AUC
Decision Tree without Hyper-parameter Tuning	0.886241	0.892113	0.894394	0.889844	0.885797
Decision Tree with Hyper-parameter Tuning	0.891619	0.896098	0.888752	0.903567	0.891775

Data were transformed into Document-term matrix using TF-IDF, as bag of words weren't efficiently working with the Decision tree algorithm. It has given better results with tuning than default parameters. Overall performance was good with an AUC score of 89% and a precision score of 90%.

Experiment 2: Random Forest Algorithm – TF-IDF Document matrix

```
confusion matrix without tuning
[[4385  729]
 [1459 4213]]
```

```
confusion matrix with tuning
[[4825  289]
 [ 701 4971]]
```

Algorithm	Accuracy	f1-score	Recall-Rate	Precision	AUC
Random forest without Hyper-parameter Tuning	0.797144	0.793857	0.742772	0.852489	0.800111
Random forest with Hyper-parameter Tuning	0.908214	0.90944	0.87641	0.945057	0.909949

As in the decision tree experiment, TF-IDF processed dataset was used for testing and training. Hyper-parameter tuning has shown up to 10% of improvement. It has performed well with an AUC score of 90.9% and a good precision score of 94%.

Experiment 3: AdaBoost – TF-IDF Document Matrix

```
confusion matrix without tuning
[[4439  675]
 [ 642 5030]]
```

```
confusion matrix with tuning
[[4692  422]
 [ 619 5053]]
```

Algorithm	Accuracy	f1-score	Recall-Rate	Precision	AUC
AdaBoost without Hyper-parameter Tuning	0.877897	0.88424	0.886812	0.881683	0.877411
AdaBoosst Tree with Hyper-parameter Tuning	0.903486	0.906612	0.890867	0.922922	0.904174

AdaBoost has performed well than random-forest and decision tree. In terms of performance metrics, it has given a good accuracy rate and AUC score of around 90%.

Experiment 4: XGBoost – Bag of words Sparse Matrix Data

```
confusion matrix before tuning
[[4637  532]
 [ 482 5135]]
```

```
confusion matrix after tuning
[[4780  389]
 [ 501 5116]]
```

Algorithm	Accuracy	f1-score	Recall-Rate	Precision	AUC
XGBoost without Hyper-parameter Tuning	0.905989	0.910138	0.914189	0.906123	0.905634
XGBoost with Hyper-parameter Tuning	0.917486	0.919978	0.910806	0.929337	0.917775

XGBoost was efficient in run time with bag of words data transformation. The model was trained with an 'error' evaluation metric to validate split-test data. XGBoost has performed excellently with an AUC score of 91.7% with tuning and best recall rate of 91.4% among the other algorithms.

Experiment 5: LightGBM – Bag of words Sparse Matrix Data

Algorithm	Accuracy	f1-score	Recall-Rate	Precision	AUC
LightGBM without Hyper-parameter Tuning	0.926585	0.926748	0.900755	0.954286	0.962087
LightGBM with Hyper-parameter Tuning	0.931591	0.931931	0.908306	0.956818	0.963168

LightGBM has shown phenomenal performance with a highest accuracy score of 93.1,f1-score with 93.1%, excellent precision rate of 95.6%, and best AUC score of 96.3%, which signifies the amount of correct prediction of fake news within threshold cut point. 'Binary_logloss' metric was used for validation on the split-test dataset.

6.1 Result

Algorithm	Accuracy	f1-score	Recall	Precision	AUC
Decision Tree without Hyper-parameter Tuning	0.886241	0.892113	0.894394	0.889844	0.885797
Decision Tree with Hyper-parameter Tuning	0.891619	0.896098	0.888752	0.903567	0.891775
Random forest without Hyper-parameter Tuning	0.797144	0.793857	0.742772	0.852489	0.800111
Random forest with Hyper-parameter Tuning	0.908214	0.90944	0.87641	0.945057	0.909949
AdaBoost without Hyper-parameter Tuning	0.877897	0.88424	0.886812	0.881683	0.877411
AdaBoosst Tree with Hyper-parameter Tuning	0.903486	0.906612	0.890867	0.922922	0.904174
XGBoost without Hyper-parameter Tuning	0.905989	0.910138	0.914189	0.906123	0.905634
XGBoost with Hyper-parameter Tuning	0.917486	0.919978	0.910806	0.929337	0.917775
LightGBM without Hyper-parameter Tuning	0.926585	0.926748	0.900755	0.954286	0.962087
LightGBM with Hyper-parameter Tuning	0.931591	0.931931	0.908306	0.956818	0.963168

Table 3: Overall Result

LightGBM was found to be the best performing classification in terms of all evaluation metrics with the best accuracy, f-1 score, and AUC score. XGBoost and AdaBoost have also performed quite well with a good accuracy rate and AUC score of 90%. XGBoost was the best in terms of recall rate of 91.4%. Random forest performed very poor with default parameters with the least accuracy rate and AUC score of around 80%. The decision tree algorithm performed equally good as boosting algorithms with a reasonable accuracy rate and an AUC score of 89%. But random-forest algorithm with hyperparameter tuning had outperformed decision tree in-terms of every metric. Therefore, the performance of LightGBM was phenomenal in all the way and outperformed other boosting algorithms with the best results.

7 Fake News Detector API

Fake news detector is a user API that emulates fake news detection if ever implemented in real-time application. The API communicates with the user through a POST request and enables users to input author, text and title of the news for validation. Client-side events are handled using REACT API which is a commonly used function to build a simple API in python. WSGI server is used to process post requests sent from the client to the server-side for pre-processing and rate trustworthiness in the text. The fake rate value is used as a parameter to predict news as either reliable or unreliable, as shown in the output. The WSGI server hosts the API at port 5000, and error codes 200, 400, 500 were used to respond to incidents. Detailed implementation of API is given in the configuration manual.

Input:

24	22	Alexandri	1	Now nun	of the Best Kinds of Milk That Arent Dairy

Output:

200

Response body


```
{
  "prediction": "Unreliable News",
  "fake_rate": 0.94
}
```


8 Discussion

Table. 3 shows comparison metrics of all proposed classification model. It is evident from the table that LightGBM has outperformed all other algorithms with the best results of f1-score, accuracy rate, AUC, and precision score. Correctly, it can be observed that XGBoost and LightGBM had a better AUC score of 91.77% and 96% than other classification algorithms. The computation time of the model was way different from each other. LightGBM was so fast, and XGBoost required more time. XGBoost Exhibited the best recall core of 91.% that signifies correct fake news prediction from the actual count of fake news in the dataset. However, LightGBM stood next with a 90% recall rate, and the random forest was the least with 74%. AdaBoost and Decision tree algorithm performed equally with a recall rate of around 88%.

Data up sampling improved the performance significantly of all the classification models. The data processed with word of bags failed to work with the decision, random forest, and AdaBoost algorithms due to inefficiency and which in turn didn't allow us to compare evaluation with NLP techniques. N-grams were used in both TF-IDF and bag of words DTM techniques, but we could only able to give a range of values from which algorithms choose the best feature such as Unigram or trigram or Trigram. Lastly, it was hard to find the right documentation for relatively new algorithms such as XGBoost and LightGBM to understand parameters and internal architecture.

9 Conclusion

The primary objective of this project is to classify fake news from real news data with a high accuracy rate and low false positives. We had gone through a series of processes to achieve the goal. Firstly, the base dataset was obtained from the Kaggle competition forum. The small quantity of fake news (1) and real news (0) puts the credibility of the project in question. The problem was handled with random upsampling. Datasets were pre-processed to reduce the space complexity of the machine learning algorithm and to improve efficiency. A literature review was done over various research papers that dealt with fake news detection and text classifications. Decision tree, random forest, AdaBoost, XGBoost, and LightGBM were the proposed combination of algorithms which very well performed in previous researches.

The research involved data transformation with NLP techniques such as TF-IDF and Word bag into a document-term matrix format. Dataset was trained and tested with N-gram value of a range of 1-3. Concerning the result table display above, LightGBM and XGBoost performed exceptionally with the bag of words NLP technique. LightGBM topped the table with an accuracy rate of 93.1% and the best AUC score of 96.3. XGboost achieved the highest recall rate of 91.4%. Hence, the hypothesis of fake news classification with high accuracy and low false-positive with LightGBM has been proved through the research. Finally, fake news classification API with the best-performing machine learning has been deployed and tested.

Future Work:

1) Developed API prototype could be further developed and incorporated into social media platforms to aware people of the fake rate of news.

2) Accuracy of the classification model can be improved with feature selection techniques such as the Wrapper method that randomly selects a set of features.

3) The current system of detection can be enhanced with any new existing algorithms or with a hybrid of algorithms for better performance.

4) The proposed method is limited to a bag of word techniques that narrows the application towards social media platforms. Hence it could be widened by training with a range of datasets from various online news sources.

References

[1] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proc. Assoc. Inf. Sci. Technol.*, vol. 52, no. 1, pp. 1–4, 2015.

[2] M. Potthast, "A Stylometric Inquiry into Hyperpartisan and Fake News."

[3] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News," 2016, pp. 7–17.

[4] C. Tan, L. Lee, and B. Pang, "The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter," *52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf.*, vol. 1, pp. 175–185, 2014.

[5] V. L. Rubin, "Scholarship @ Western Deception Detection and Rumor Debunking for Social Media DECEPTION DETECTION AND RUMOR DEBUNKING," 2017.

[6] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PLoS One*, vol. 10, no. 6, pp. 1–13, 2015.

[7] J. Ma, W. Gao, and K. F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*, vol. 1, pp. 708–717, 2017.

[8] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, "Some like it Hoax: Automated fake news detection in social networks," *CEUR Workshop Proc.*, vol. 1960, pp. 1–12, 2017.

[9] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task Learning for Rumour Verification," 2018.

[10] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao, "IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification," no. 2011, pp. 465–469, 2018.

[11] F. A. Ozbay and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Phys. A Stat. Mech. its Appl.*, no. xxxx, p. 123174, 2019.

[12] S. F. Sabbeh and S. Y. Baatwah, "Arabic news credibility on twitter: An enhanced model using hybrid features," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 8, pp. 2327–2338, 2018.

[13] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 1103–1108, 2013.

[14] A. El Azab, A. M. Idrees, M. A. Mahmoud, and H. Hefny, "Fake Account Detection in Twitter Based on Minimum Weighted Feature set," *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 10, no. 1, pp. 13–18, 2016.

[15] N. Manju, B. S. Harish, and V. Prajwal, "Ensemble Feature Selection and Classification of Internet Traffic using XGBoost Classifier," no. July, pp. 37–44, 2019.

[16] Y. Yuan, L. Huo, Y. Yuan, and Z. Wang, "Semi-supervised tri-Adaboost algorithm for network intrusion detection," *Int. J. Distrib. Sens. Networks*, vol. 15, no. 6, 2019.

[17] B. Markines, C. Cattuto, and F. Menczer, "Social spam detection," *ACM Int. Conf. Proceeding Ser.*, pp. 41–48, 2009.

[18] S. Kalra, L. Li, and H. R. Tizhoosh, “Automatic Classification of Pathology Reports using TF-IDF Features,” pp. 1–4, 2019.
[19] M. R. Machado, S. Karray, and I. T. de Sousa, “LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry,” *2019 14th Int. Conf. Comput. Sci. Educ.*, no. Iccse, pp. 1111–1116, 2019.
[20] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3147–3155, 2017.
[21] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, “Faking Sandy,” pp. 729–736, 2013.
[22] E. Beğenilmiş and S. Uskudarli, “Organized behavior classification of tweet sets using supervised learning methods,” *ACM Int. Conf. Proceeding Ser.*, 2018.