

**Finding Fake News**

Prangige Shashini Randika Peiris

Supervised by: Dr. Dimitra Zografistou

and Dr. Brian Pluss

A dissertation submitted for the Degree of Master of Science

MSc in Data Science and Engineering

School of Science and Engineering

University of Dundee

17th March 2023

**Executive Summary**

**Project Summary**

The objective of this project is to develop a machine learning model to identify fake news. It has become difficult to spot and distinguish whether the news is real or fake as it has developed to a level where one can produce not just fake text but fake pictures and videos similar or indifferent from original picture / video using digital tools. It has become widespread information available online and it is becoming increasingly difficult to distinguish between real and fake news.

**The work that has been done**

The machine learning model was trained on a dataset of articles labelled as either real or fake. The model used a variety of features, including textual and metadata, to identify patterns and characteristics of fake news articles. After training, the model was tested on a separate dataset to evaluate its accuracy and performance.

**The main outcomes / results**

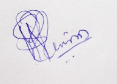
The results of the project showed that the machine learning model was able to accurately classify fake news articles with a high degree of accuracy. This approach could potentially be used by journalists, fact-checkers, and news organizations to quickly identify and flag fake news articles, enabling them to take necessary action to prevent the spread of misinformation.

**Recommendations**

Overall, this project demonstrated the potential for machine learning to be a powerful tool in the combat fake news. With continued development and refinement, machine learning models could help to reduce the impact of fake news on society and support the growth of a more informed and reliable media landscape.

**Declaration**

I declare that the special study described in this dissertation has been carried out and the dissertation composed by me, and that the dissertation has not been accepted in fulfilment of the requirements of any other degree or professional qualification.



Signed[[1]](#footnote-1):

Date: 2023-03-17

Certificate

I certify that Shashini Peiris has satisfied the conditions of the Ordinance and Regulations and is qualified to submit this dissertation in application for the degree of Master of Science.

Signed[[2]](#footnote-2):

Date: 2023-03-17

Acknowledgements

I would like to thank my project supervisors Dr. Dimitra Zografistou and Dr. Brian Pluss for the superior guidance and support given all throughout the short period of project. Mr. Andy Cobley and Dr. Craig Ramsay also must be remembered for the support given during the project.

I would also like to take this opportunity to thank all the past students who shared their valuable work for our reference and all those who supported me in evaluating my project and giving inputs where applicable.

Also, I would like to thank all the students who were generous to support me during the project as well as while I was in Sri Lanka during the Semester 2 (January 2022 - April 2022).

Table of Contents

[**Executive Summary** i](#_Toc133319835)

[**Declaration** ii](#_Toc133319836)

[Certificate iii](#_Toc133319837)

[Acknowledgements iv](#_Toc133319838)

[Table of Contents v](#_Toc133319839)

[List of Tables viii](#_Toc133319840)

[1. Introduction 1](#_Toc133319841)

[1.1 Objectives 1](#_Toc133319842)

[1.2 What is Fake News and what are the consequences of fake news? 1](#_Toc133319843)

[1.2.1 Why detecting fake news is important? 1](#_Toc133319844)

[1.2.2 How to find Fake news? 1](#_Toc133319845)

[2. Background 3](#_Toc133319846)

[2.1.1 Fake News Definition 3](#_Toc133319847)

[2.2 Approaches proposed to detect fake news 3](#_Toc133319848)

[2.3 Have others already done work in this area? 4](#_Toc133319849)

[2.4 How could that help you in your project, or not? 5](#_Toc133319850)

[2.5 Literature review 5](#_Toc133319851)

[3. Specification 6](#_Toc133319852)

[3.1 Purpose, Scope and Goal 6](#_Toc133319853)

[3.2 Data Collection 6](#_Toc133319854)

[3.2.1 Data collection sources 6](#_Toc133319855)

[3.2.2 Cleaning and pre-processing dataset 6](#_Toc133319856)

[3.3 Feature Extraction 6](#_Toc133319857)

[3.4 Modelling 7](#_Toc133319858)

[3.4.1 Why chose multiple Algorithms? 8](#_Toc133319859)

[3.4.2 Why chose these Algorithms in particular? 9](#_Toc133319860)

[3.5 Performance Metrics 9](#_Toc133319861)

[3.6 Maintenance and Updates 10](#_Toc133319862)

[3.7 Ethical Considerations 10](#_Toc133319863)

[3.8 Timeline 10](#_Toc133319864)

[3.9 Weekly Group Meetings (Summary) 11](#_Toc133319865)

[4. Design 12](#_Toc133319866)

[4.1 Environment and Programming Languages 12](#_Toc133319867)

[4.2 Frameworks and Data Structures 12](#_Toc133319868)

[4.3 Design choices that can evolve during the project 13](#_Toc133319869)

[4.4 Process flow 13](#_Toc133319870)

[5. Implementation 15](#_Toc133319871)

[5.1 Implementation 15](#_Toc133319872)

[6. Evaluation, Testing and Results 22](#_Toc133319873)

[6.1 Model Comparison 22](#_Toc133319874)

[6.2 Testing all models Pickle files 24](#_Toc133319875)

[6.3 Real World News 24](#_Toc133319876)

[6.4 Debugging 25](#_Toc133319877)

[6.5 Implementation 25](#_Toc133319878)

[6.5.1 Functional aspects 25](#_Toc133319879)

[6.5.2 Benchmarking against the work of others 25](#_Toc133319880)

[6.5.3 Non-functional aspects 26](#_Toc133319881)

[6.6 Description of the Final Product 26](#_Toc133319882)

[6.6.1 Coding 26](#_Toc133319883)

[6.6.2 Input 26](#_Toc133319884)

[6.6.3 Output 27](#_Toc133319885)

[7. Critical Appraisal, Summary and Conclusions 28](#_Toc133319886)

[7.1 Critical Appraisal 28](#_Toc133319887)

[7.2 Summary 28](#_Toc133319888)

[7.3 Conclusion 28](#_Toc133319889)

[8. Future Work 29](#_Toc133319890)

[9. References / Bibliography 30](#_Toc133319891)

[10. List of Appendices 31](#_Toc133319892)

List of Figures

[Figure 1: Bayes theorem 7](#_Toc133265274)

[Figure 2: Accuracy 8](#_Toc133265275)

[Figure 3: Precision 8](#_Toc133265276)

[Figure 4: Recall 9](#_Toc133265277)

[Figure 5: F1-Score 9](#_Toc133265278)

[Figure 6: Gantt Chart 10](#_Toc133265279)

[Figure 7: Process flow 13](#_Toc133265280)

[Figure 8: Importing Python libraries 14](#_Toc133265281)

[Figure 9: Read the dataset 14](#_Toc133265282)

[Figure 10: Pre-processed dataset 15](#_Toc133265283)

[Figure 11: Shape of training dataset 15](#_Toc133265284)

[*Figure 12: Data distribution of Training, Testing and Valid datasets* 15](#_Toc133265285)

[*Figure 13: Dataset quality check - Training, Testing and Final datasets* 15](#_Toc133265286)

[*Figure 14: Train count for CountV & TfidfV* 16](#_Toc133265287)

[*Figure 15: Word Clouds for CountV & TfidfV* 16](#_Toc133265288)

[*Figure 16: Word Count and Feature names for CountV and TfidfV* 16](#_Toc133265289)

[Figure 17: Classification Report - Model - Naive Bayes 17](#_Toc133265290)

[Figure 18: Classification Report - Model - Logistic Regression 17](#_Toc133265291)

[Figure 19: Classification Report - Model - SVM 17](#_Toc133265292)

[Figure 20: Classification Report - Model - SGD 17](#_Toc133265293)

[Figure 21: Classification Report - Model - Random Forest 17](#_Toc133265294)

[Figure 22: Classification Report - Model - Passive Aggressive 17](#_Toc133265295)

[Figure 23: Classification Report - Model - Naive Bayes 18](#_Toc133265296)

[Figure 24: Classification Report - Model - Logistic Regression 18](#_Toc133265297)

[Figure 25: Classification Report - Model - SVM 18](#_Toc133265298)

[Figure 26: Classification Report - Model - SGD 18](#_Toc133265299)

[Figure 27: Classification Report - Model - Random Forest 18](#_Toc133265300)

[Figure 28: Classification Report - Model - Passive Aggressive 18](#_Toc133265301)

[Figure 29: Confusion Matrix - Model - Naive Bayes - CountV 19](#_Toc133265302)

[Figure 30: Confusion Matrix - Model - Logistic Regression - CountV 19](#_Toc133265303)

[Figure 31: Confusion Matrix - Model - SVM - CountV 19](#_Toc133265304)

[Figure 32: Confusion Matrix - Model - SGD - CountV 19](#_Toc133265305)

[Figure 33: Confusion Matrix - Model - Random Forest - CountV 19](#_Toc133265306)

[Figure 34: Confusion Matrix - Model - Passive Aggressive - CountV 19](#_Toc133265307)

[Figure 35: Confusion Matrix - Model - Naive Bayes - TfidfV 20](#_Toc133265308)

[Figure 36: Confusion Matrix - Model - Logistic Regression - TfidfV 20](#_Toc133265309)

[Figure 37: Confusion Matrix - Model - SVM - TfidfV 20](#_Toc133265310)

[Figure 38: Confusion Matrix - Model - SGD - TfidfV 20](#_Toc133265311)

[Figure 39: Confusion Matrix - Model - Random Forest - TfidfV 20](#_Toc133265312)

[Figure 40: Confusion Matrix - Model - Passive Aggressive - TfidfV 20](#_Toc133265313)

[Figure 41: Evaluation - F1-score and Accuracy score 21](#_Toc133265314)

[Figure 42: F1-Scores of Different Models (sorted) 22](#_Toc133265315)

[Figure 43: Accuracies of Different Models (sorted) 22](#_Toc133265316)

[Figure 44: Predictions for all models 23](#_Toc133265317)

[Figure 45: A code snippet with input text field 25](#_Toc133265318)

[Figure 46: Final Output - Prediction and Analysis 26](#_Toc133265319)

List of Tables

[Table 1: Confusion Matrix 8](#_Toc133265366)

[Table 2: Timeline 10](#_Toc133265367)

[Table 3: Evaluation - Sort using F1-score 21](#_Toc133265368)

# Introduction

This section addresses the objectives and definitions of this Finding Fake News project.

## Objectives

The objective of this project is to develop machine learning models to identify fake news. It has become difficult to spot and distinguish whether the news is real or fake as it has developed to a level where one can produce not just fake text but fake pictures and videos similar or indifferent from original picture / video using digital tools. Fake news has become widespread information available online and it is becoming increasingly difficult to distinguish between real and fake news.

The other objective is to learn and master data analysis and programming skills that is pretty much required in the IT industry and Data Engineering.

## What is Fake News and what are the consequences of fake news?

Fake news is false or deceptive information presented as if it is real news, often spread through social media and other online platforms. It can be fabricated intentionally and spread for various motives, such as to deceive or manipulate public opinion, to generate web traffic, or to gain political or financial advantage. Fake news can take many forms, including misleading headlines, falsified images and videos, and fabricated stories or quotes attributed to real people or sources. The spread of fake news can have serious consequences, including influencing elections, causing social unrest, and eroding public trust in news media and other institutions.

### Why detecting fake news is important?

Detecting fake news is important because it helps to prevent the spread of misinformation and disinformation, which can have serious consequences. Here are some reasons why:

**Misleading information**: Fake news can contain misleading information that is not accurate or true. This can lead people to make decisions based on false information, which can have serious consequences.

**Public safety**: Fake news can also contain information that is harmful to public safety. For example, false information about natural disasters or health crises can lead people to take actions that put their lives or the lives of others at risk.

**Social division**: Fake news can also be used to fuel social division by spreading false or exaggerated information about certain groups of people. This can lead to hatred and discrimination, which can have long-lasting effects on society.

**Political manipulation**: Fake news can also be used to manipulate political outcomes by spreading false or misleading information about candidates or issues. This can undermine the democratic process and lead to the election of unfit or corrupt leaders.

### How to find Fake news?

Finding fake news is a critical skill in today's digital age, where misinformation and disinformation spread rapidly through social media and other online platforms. This project aims to equip people with the tools and knowledge they need to identify and avoid fake news.

The first step in finding fake news is to **understand** what it is. Fake news refers to deliberately false or misleading information presented as if it were real news. It can be spread through various channels, such as social media, websites, blogs, and even traditional news media.

To identify fake news, it is essential to **evaluate the source** of the information. Reputable news sources have editorial standards and fact-checking processes in place to ensure the accuracy of their reporting. On the other hand, fake news sources often lack transparency and may use misleading headlines, fabricated quotes, or doctored images to manipulate readers.

Another crucial aspect of identifying fake news is to be aware of **bias** and to look for multiple sources to verify the information. Biased sources may selectively present information to support their viewpoint, while reputable news sources strive for objectivity and balance.

# Background

Fake news has become more prevalent in the recent past due to the rapid spread of information through social media and the internet. The purpose of fake news is to deceive and mislead readers by looking and feeling like legitimate news, but it is specially crafted to deceive and mislead them. Fake news is having an increasingly significant impact on individuals, organizations, and societies. It may undermine trust in reliable sources of information, cause confusion and misunderstandings, and even result in harm.

### Fake News Definition

The fake news term originally refers to false and often sensationalist information disseminated under the guise of relevant news. (Anon., n.d.) [[3]](#footnote-3)

Methods that can detect fake news:

* The false knowledge it carries
* The writing style of the article
* The propagation pattern of articles
* The credibility of its source (Anon., n.d.)[[4]](#footnote-4)

**These 4 tips can be used to check the truthfulness of a news**:

The story: what are they trying to say? Is it an ad or a joke? Look to see if you can find the same story somewhere else

The author: is it someone’s opinion or a fact? Real news will most likely have a link to the writer’s details, but if there’s no author, dig deeper

The website: are there spelling or grammar mistakes? What’s the URL? Check the address bar at the top - most trusted URLs end with “.com”, “.co.uk”, “.net”, “.gov”, “.org”, “.mil” and “.edu”

The date: is the story recent or old? It could be outdated or a copy of something that happened years ago. Computer programs called bots post anytime & often, so be wary of this (Anon., n.d.) [[5]](#footnote-5)

## Approaches proposed to detect fake news

1. ***Language approach***: This approach focuses on the use of linguistics by a human or software program to detect fake news. The approach considers all the words in a sentence and letters in a word, how they are structured and how it fits together in a paragraph. The focus is therefore on grammar and syntax. The 3 main methods of language approach:
   1. Bag of Words (BOW)
   2. Semantic Analysis
   3. Deep Syntax
2. ***Topic-agnostic approach***: This category of approaches detects fake news by not considering the content of articles bur rather topic-agnostic features. The approach uses linguistic features and web mark-up capabilities to identify fake news. Examples of topic-agnostic features are
   1. A large number of advertisements
   2. Longer headlines with eye-catching phrases
   3. Different text patterns from mainstream news to induce emotive responses
   4. Presence of an author name
3. ***Machine learning approach***: Machine learning algorithms can be used to identify fake news.
   1. Using different types of training datasets to refine the algorithms
   2. Rumour identification framework
4. ***Knowledge-based approach***: The knowledge-based approach aims at using sources that are external to verify if the news is fake or real and to identify the news before the spread thereof becomes quicker. There are three main categories:
   1. Expert Oriented Fact Checking
   2. Computational Oriented Fact Checking
   3. Crowd Sourcing Oriented Fact Checking
5. ***Hybrid approach***: There are three generally agreed upon elements of fake news articles:
   1. The text of an article
   2. The response that the articles received
   3. The source used that motivate the news article

A recent study has been conducted that proposes a hybrid model which helps to identify fake news on social media through using a combination of human and machine learning to help identify fake news.

Another hybrid model called CSI (capture, score, integrate) has been developed and functions on the main elements

1. Capture - the process of extracting representations of articles by using a Recurrent Neutral Network (RNN)
2. Score - to create a score and representation vector
3. Integrate - to integrate the outputs of the capture and score resulting in a vector which is used for classification (Anon., n.d.) [[6]](#footnote-6)

**A machine learning approach has been used in this project to build different models to identify and compare fake news.**

## Have others already done work in this area?

Fake news is not a problem that has been identified recently. This problem has been there since ancient times and there was no solution discovered using technology until recent days.

There are many machine learning projects carried out to identify fake news and there are many sources on the web related to this issue. Some of these projects are in Github and few of such projects are given below:

1. <https://github.com/nishitpatel01/Fake_News_Detection>
2. <https://github.com/Spidy20/Fake_News_Detection>
3. <https://github.com/SaikrishnaPulipati533/Fake-News-Detection_NLP-Project>

Different projects demonstrate different approaches identified to detect fake news.

## How could that help you in your project, or not?

The following project took my attention as it uses multiple machine learning models to identify fake news and Passive Aggressive classifier has been used in this project to create the machine learning model to detect fake news.

<https://github.com/nishitpatel01/Fake_News_Detection>

Some of the codes were re-used in my project but changes were made as I use 6 different classifiers to create 12 models.

## Literature review

Detecting and preventing the spread of fake news has become a major challenge in recent years. To conduct a literature review on finding fake news, several sources were explored including academic journals, books, and online resources. The following summarizes the key findings from the literature:

* Machine Learning Approaches
* Social Media Analysis
* Fact-Checking
* User Behaviour Analysis
* Multimedia Analysis

The literature suggests that there are multiple approaches to identify fake news, including machine learning algorithms, social media analysis, fact-checking, user behavior analysis, and multimedia analysis. However, each approach has its limitations, and a combination of multiple approaches may be required to effectively identify and prevent the spread of fake news. (Anon., n.d.) [[7]](#footnote-7)

# Specification

The project specification describes the Project Overview, Data Collection, Feature Extraction, Modelling, Performance Metrics, User Interface, Deployment, Maintenance and Updates, Ethical Considerations and Timeline and Budget.

A brief description of the project, including its purpose, scope, and goals will be given in this section.

## Purpose, Scope and Goal

The purpose of this project is to develop machine learning models to identify fake news using machine learning algorithms.

The scope of Finding Fake News project includes the following:

1. Find a big dataset
2. Clean the dataset
3. Train multiple models
4. Analyse all the models
5. Run cross validation
6. Test all models on new data
7. Present results using each model

The goal of a "Finding Fake News" project is to develop a machine learning model that can accurately identify and classify news articles as either real or fake. The steps described above in the scope will be carried out to achieve this goal.

## Data Collection

The source of data and pre-processing methods used will be described briefly in this section.

### Data collection sources

Kaggle.com and Github.com was used mainly to find suitable datasets and a dataset was identified from Github use in this project.

Source of dataset: <https://github.com/nishitpatel01/Fake_News_Detection>

### Cleaning and pre-processing dataset

The dataset seems to be a clean dataset, unwanted columns and empty rows were removed already and with the time constraints, I decided to go ahead with it and pre-process using lemmatization and stemming. The purpose of combining lemmatization and stemming in this way is to normalize words and reduce the dimensionality of the textual data, making it easier to process and analyze.

Stemming and Lemmatization are text normalization techniques used in Natural Language Processing. The stemming is faster as it cuts words without knowing the context, while lemmatization is slower as it identifies the context of words before processing. (Anon., n.d.) [[8]](#footnote-8)

## Feature Extraction

The cleaned and pre-processed data then went through feature extraction using both Count Vectorizer and TFIDF Vectorizer approaches for the purpose of learning as well as to demonstrate the differences between both approaches to evaluate and identify which machine learning model gives the best results.

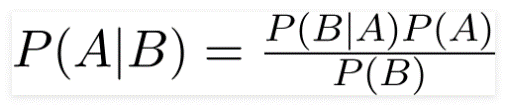
Both Count Vectorizer and TFIDF Vectorizer are methods to convert text data into numeric form so that machine can understand. Machine learning models cannot identify text hence it is required to be converted into numeric form.

Count Vectorizer counts the number of instances each word appears in a document and creates a matrix representing the frequency of each word in each document. Tfidf Vectorizer, counts the number of times each word appears in a document as well how common or rare each word is across all documents. Tfidf Vectorizer is generally considered to be more effective than Count Vectorizer for tasks like text classification. (Anon., n.d.) [[9]](#footnote-9)

## Modelling

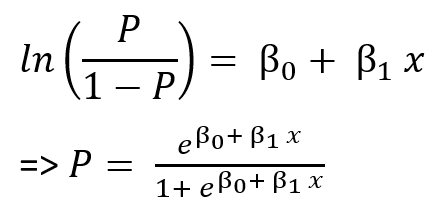
The following supervised learning models will be used in this project to identify truthfulness of news articles:

1. **Naïve bayes classifier** - A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task which is based on the Bayes theorem. (Anon., n.d.) [[10]](#footnote-10)



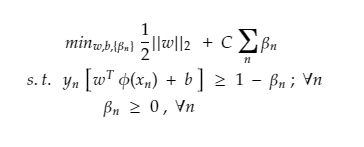
**Figure 1: Bayes theorem**

1. **Logistic Regression classifier** - Logistics regression is a statistical approach and a Machine Learning algorithm that is used for classification problems and is based on the concept of probability. It is used when the dependent variable (target) is categorical. It is widely used when the classification problem at hand is binary; true or false, yes or no, and it uses the sigmoid function to return the probability of a label. (Anon., n.d.) [[11]](#footnote-11)

 (Anon., n.d.)[[12]](#footnote-12)

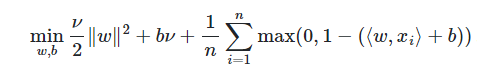
**Figure 2: Logistic Regression formula**

1. **SVM (Support vector machines) classifier** - The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. (Anon., n.d.) [[13]](#footnote-13)

 (Anon., n.d.) [[14]](#footnote-14)

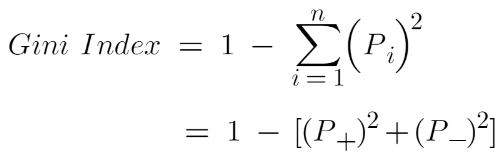
**Figure 3: SVM Classifier Formula**

1. **SGD (Stochastic Gradient Descent) classifier** - Stochastic Gradient Descent (SGD) is an efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. (Anon., n.d.) [[15]](#footnote-15)

 (Anon., n.d.) [[16]](#footnote-16)

**Figure 4: SGD Classifier Formula**

1. **Random Forest classifier** - A random forest is a meta estimator that fits multiple decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. (Anon., n.d.) [[17]](#footnote-17)

 (Anon., n.d.) [[18]](#footnote-18)

**Figure 5: Random Forest Classifier Formula**

1. **Passive Aggressive classifier** - The Passive-Aggressive algorithms are a family of Machine learning algorithms and are generally used for large-scale learning. (Anon., n.d.) [[19]](#footnote-19)

These classifiers will be used in combination with both Count Vectorizer and Tfidf Vectorizer and 12 models will be developed.

Each model will be stored in a pickle model so that it can be used without training dataset again and again which reduces the time taken to train different models.

### Why chose multiple Algorithms?

Algorithms used:

1. Naive Bayes
2. Logistic Regression
3. SVM (Support Vector Machine)
4. SGD (Stochastic Gradient Descent)
5. Random Forest
6. Passive Aggressive

I decided to choose all the above models to demonstrate the differences between different models and how I can use these differences to explain the outcomes in my project. In general, there are multiple benefits of using multiple models and few of the benefits are:

**Improved accuracy**: it increases the accuracy of predictions and since each model has different strengths and weaknesses, a combined prediction is much healthier and accurate for a system.

**Reducing bias**: the use of one classifier may produce a bias prediction and it can be reduced when multiple classification models are used.

**Handling uncertainty**: different news articles can carry a combination of both true and fake information so use of multiple classifiers may help to handle uncertainty more effectively and avoid false positives or false negatives.

### Why chose these Algorithms in particular?

The above classifiers are Supervised Learning Algorithms and Supervised Learning Algorithms are used majorly to text classification tasks. Supervised Learning Algorithms requires labelled data to train models created using these algorithms to identify the relationship between the features and the outcome, i.e., to identify whether an article is fake or real and a model trained as above can classify and predict new and unseen articles fairly.

## Performance Metrics

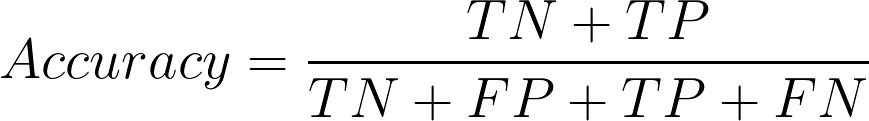
The performance metrics are derived from **Confusion matrix**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix** | | **Predicted** | |
| ***Positive*** | ***Negative*** |
| **Actual** | ***Positive*** | True positive | False Negative |
| ***Negative*** | False Positive | True Negative |

**Table 1: Confusion Matrix**

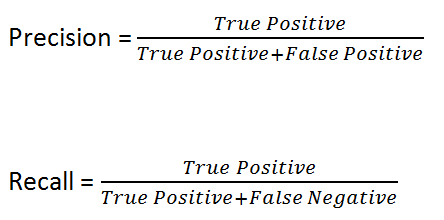
The Accuracy and F1-score are the performance metrics that will be used to evaluate the effectiveness of the models in this project.

1. **Accuracy** - Accuracy represents the number of correctly classified data instances over the total number of data instances.

 (Anon., n.d.) [[20]](#footnote-20)

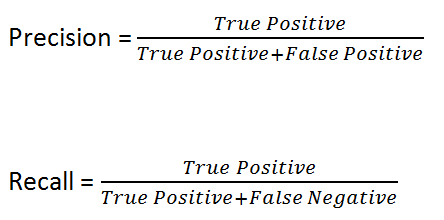
**Figure 6: Accuracy**

1. **Precision** - This measures the proportion of true positives among all the samples that the model classified as positive.



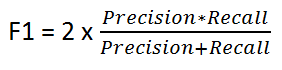
**Figure 7: Precision**

1. **Recall** - This measures the proportion of true positives among all the actual positive samples in the dataset.



**Figure 8: Recall**

1. **F1 score** - F1 Score can be used to identify the balance between Precision and Recall.

 (Anon., n.d.) [[21]](#footnote-21)

**Figure 9: F1-Score**

## Maintenance and Updates

Maintenance and update procedures for the model, with feedback mechanisms for improving its accuracy will be described in this section.

1. Different datasets also can be used to train these models to improve accuracy over time.
2. New features such as more models built using different classifiers
3. More visualizations can be introduced to ease decision making.
4. Surveys can be introduced to identify user experiences and suggestions of improvements.

These aspects will be discussed again in the Future work. (*6.7. Improvements and Future work*)

## Ethical Considerations

This section discusses any ethical considerations associated with the project, such as the potential for bias or unintended consequences, and describe strategies for mitigating these risks.

1. ***Problem***: The training dataset used to develop the machine learning models can be bias to a particular domain.

***Solution***: The news in training data is derived using news from different domains

1. ***Problem***: News can be influenced by personal beliefs, cultural aspects, and political perceptions so there is a risk of falsely identifying true news as fake and it will be controversial.

***Solution***: Provide clear guidelines and ethical principles for identifying and labelling fake news

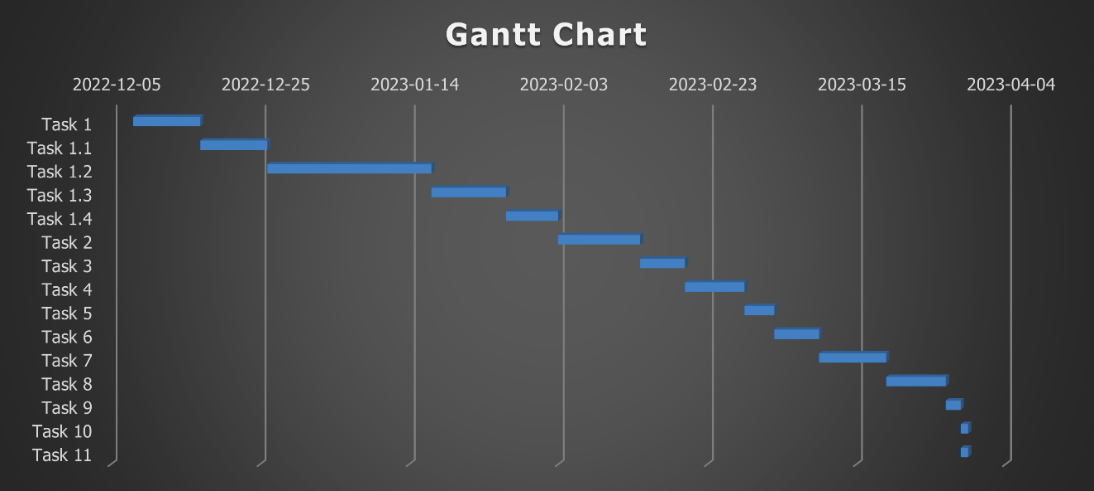
It is important to continually monitor and evaluate the performance of the machine learning models.

## Timeline

This table consists of the initial project timeline with milestones and deliverables, but it was changed due to unavoidable circumstances.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task no** | **Task** | **Duration** | **Start Date** | **End Date** |
| Task 1 | Project Initiation | 9 | 2022-12-08 | 2022-12-17 |
| Task 1.1 | Project choices | 9 | 2022-12-17 | 2022-12-26 |
| Task 1.2 | Project allocation | 22 | 2022-12-26 | 2023-01-17 |
| Task 1.3 | Risk Assessment Form | 10 | 2023-01-17 | 2023-01-27 |
| Task 1.4 | Ethics Declaration Form | 7 | 2023-01-27 | 2023-02-03 |
| Task 2 | Literature Review | 11 | 2023-02-03 | 2023-02-14 |
| Task 3 | Data Collection | 6 | 2023-02-14 | 2023-02-20 |
| Task 4 | Data Pre-processing | 8 | 2023-02-20 | 2023-02-28 |
| Task 5 | Model Development | 4 | 2023-02-28 | 2023-03-04 |
| Task 6 | Model Testing | 6 | 2023-03-04 | 2023-03-10 |
| Task 7 | Results Analysis | 9 | 2023-03-10 | 2023-03-19 |
| Task 8 | Report Writing | 8 | 2023-03-19 | 2023-03-27 |
| Task 9 | Presentation Prep | 2 | 2023-03-27 | 2023-03-29 |
| Task 10 | Final Presentation | 1 | 2023-03-29 | 2023-03-30 |
| Task 11 | Project Closure | 1 | 2023-03-29 | 2023-03-30 |

**Table 2: Timeline**



**Figure 10: Gantt Chart**

This Gantt Chart was prepared based on the project timeline as planned at the beginning. But it had to be changed due to unforeseen circumstances.

## Weekly Group Meetings (Summary)

Weekly group meetings were conducted with the supervisors with valuable inputs and guidance as well as to identify any misunderstandings and clear them. E-Mails conversations were carried out where required to clarify issues risen during the process. The points discussed mainly are:

1. What is the project and how to identify requirements with next steps
2. Analysing and designing the prototype
3. Demonstrations of developed prototypes and improvements
4. Report writing and how it should describe the system
5. Submission requirements

Due to unplanned and unavoidable personal circumstances, I had to apply for mitigation circumstances, and I obtained extension, and my submission deadline is set as 27th April 2023. During this time, my supervisors were kind enough to conduct weekly meetings personally and guide me as well as encourage me about the project to end it successfully.

# Design

The design choices made in this project is described in this section.

## Environment and Programming Languages

This section describes the resource requirements of this project.

1. Windows laptop with 16GB RAM
2. Google Chrome browser to run development tools
3. Google docs Drawings tool has been used as the tool to draw process flow (<https://docs.google.com/drawings/>)
4. Google Colab (updated to Python 3.8) is used as the development tool as well as to deploy the system (<https://colab.research.google.com/>). Google Colab allows to write and execute arbitrary python code through the browser, and it facilitates machine learning and data analysis.
5. Python programming language used in this machine learning project

## Frameworks and Data Structures

The following machine learning frameworks of Python programming language has been used in this project:

* Pandas: A library for data manipulation and analysis.
* NumPy: A library for numerical computation.
* NLTK: A library for natural language processing.
* Seaborn: A library for data visualization.
* Scikit-learn: A machine learning library that provides a wide range of algorithms and tools for data analysis and modelling.
* Wordcloud: A library for generating word clouds from text.

The following algorithms and tools from Scikit-learn is used in this project as well:

* Count Vectorizer: A tool for vectorizing text data.
* Tfidf Transformer: A tool for transforming text data using the term frequency-inverse document frequency (TF-IDF) algorithm.
* Pipeline: A tool for constructing and running machine learning pipelines.
* F1\_score: A tool for evaluating the performance of classification models.
* Confusion Matrix: A tool for visualizing the performance of classification models.

These Python libraries are also used:

* Matplotlib & Seaborn: Libraries used for data visualization.
* Multiprocessing: A library for parallel computing.
* Pickle: A library for serialization and deserialization of Python objects.
* Regular expressions (re): A library for pattern matching and string manipulation.

Efficient data structures are important for storing and manipulating large datasets in fake news detection systems. The following Data Structures are used in this project:

* Arrays
* Lists
* Dictionaries
* Graphs

## Design choices that can evolve during the project

It was decided to use one supervised learning model at the beginning of the project but later identified that comparison of multiple different models using different classifications would be ideal to make this project unique.

During the design it was decided to train multiple models and identify the best model to use in identifying fake news. While discussing with the supervisors, the idea of using all the developed models to be used in the final result was decided, i.e. train multiple models and use all of the models to predict the truthfulness of an article and predict an overall result.

## Process flow

This is the process flow for detecting real or fake news using machine learning:

**Collect a few datasets of news articles**: A couple of datasets of news articles were collected from sources such as Github.com or Kaggle.com.

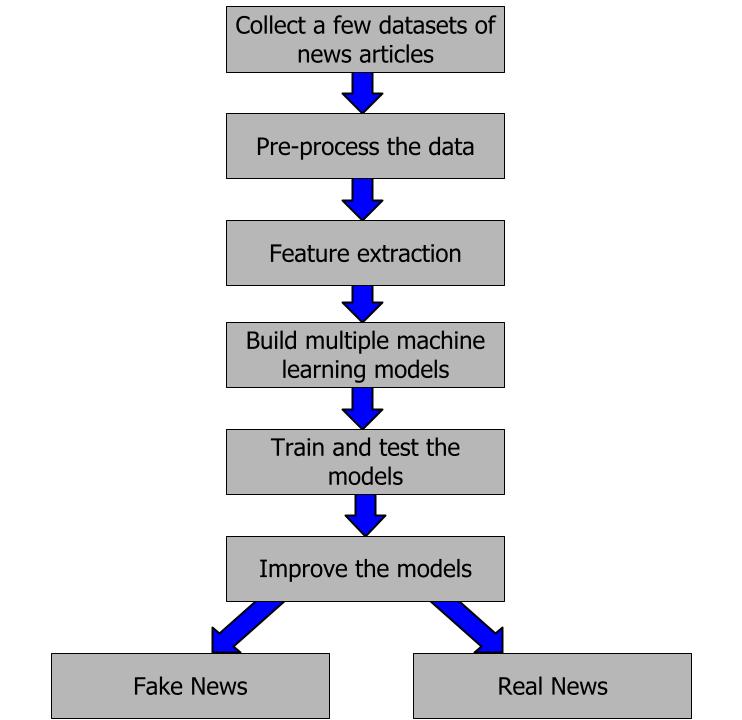
**Pre-process the data**: Then the data in the dataset were cleaned and pre-processed. This included tasks like removing special characters, removing stop words, stemming/lemmatizing the text, and converting the text into lowercase.

**Feature extraction**: Next, features extraction step was carried out from the pre-processed data using the technique TF-IDF Vectorizer and Count Vectorizer both.

**Build multiple machine learning models**: After feature extraction, I had to build machine learning models using different classification algorithms like Naive Bayes, Logistic Regression, SVM, SGD, Random Forest and Passive Aggressive to train the models.

**Train and test the models**: Once a model is developed, the dataset needs to be trained. Then the model should be tested using a separate test dataset to see how well it generalizes to new data.

**Result - Real news or Fake news**: Finally, the trained models can be used to classify new news articles as either real or fake. The output of the models will be a binary decision: **Real or Fake**.



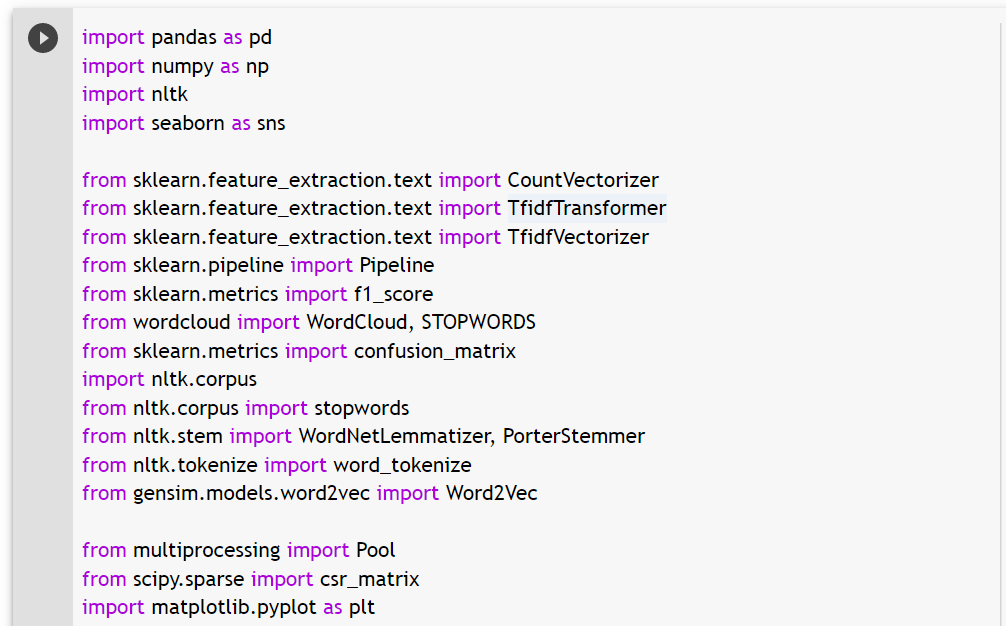
**Figure 11: Process flow[[22]](#footnote-22)**

# Implementation

This section describes the important aspects of implementation, testing and debugging that went through to produce the system: (Further information can be found in the Appendix A)

## Implementation

1. **Import Python Libraries**: The required libraries for the project are imported in this first step of the coding. More info: Refer Appendix section

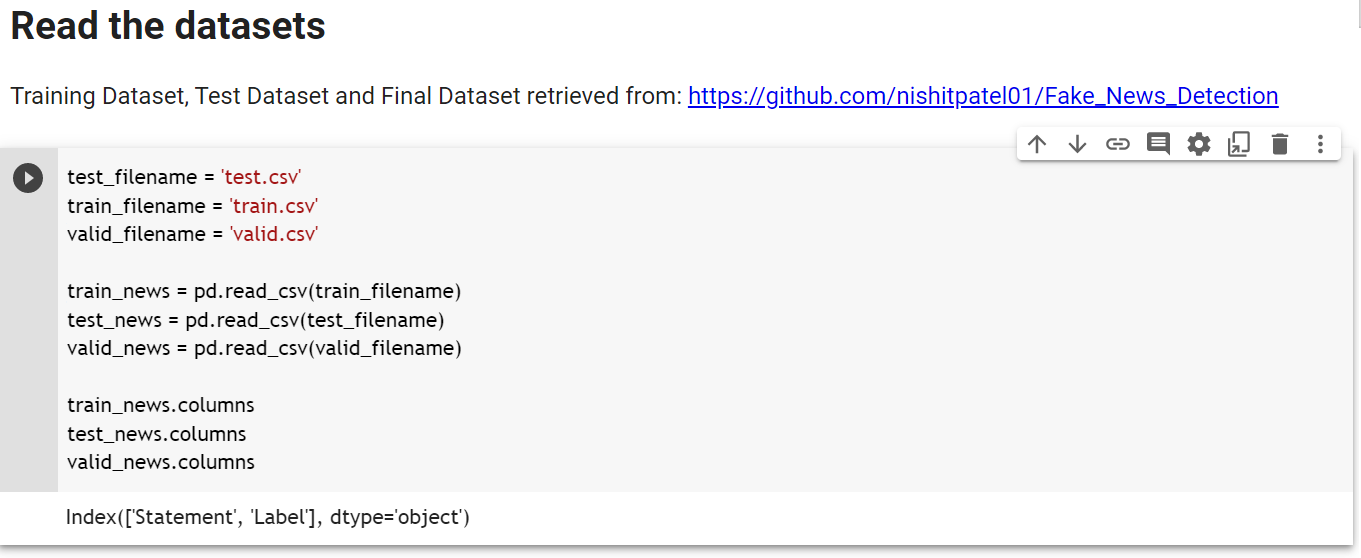


**Figure 12: Importing Python libraries**

1. **Read the datasets**: The dataset is read into the system using Pandas framework. Then it will be saved into an array.

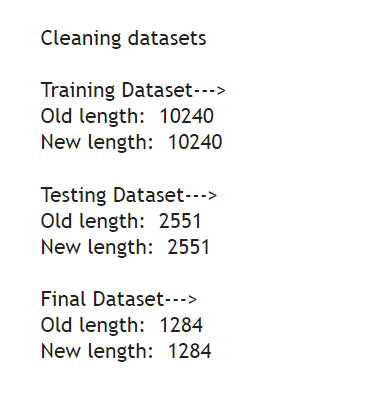
* train.csv - This csv file contains all the data that is used to train the models.
* test.csv - This csv file contains all the data that is used for testing the models.
* valid.csv - This csv file contains the data that can be used to predict using the trained models.

The columns of the datasets will be displayed.



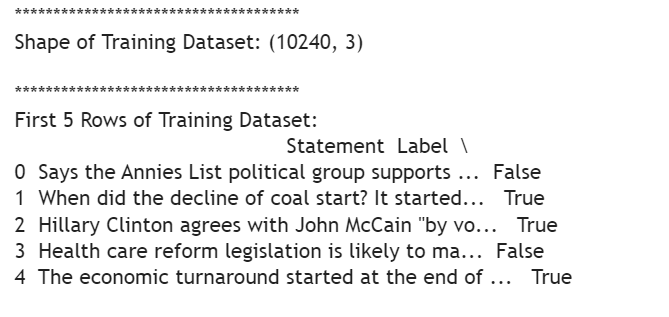
**Figure 13: Read the dataset**

1. **Data Pre-processing**: The next step is pre-processing the data in the dataset that has been read and loaded from 3 different CSV files. The techniques used in this step is described above in 3.2.2 section.



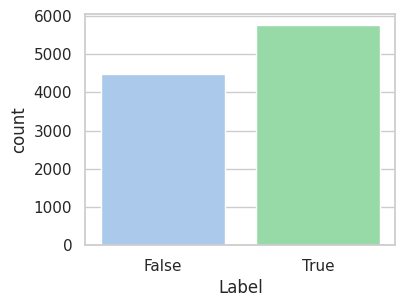
**Figure 14: Pre-processed dataset** (Refer Appendix A for more)

1. **View the Datasets**: The shapes of each of the datasets will be displayed in this step:



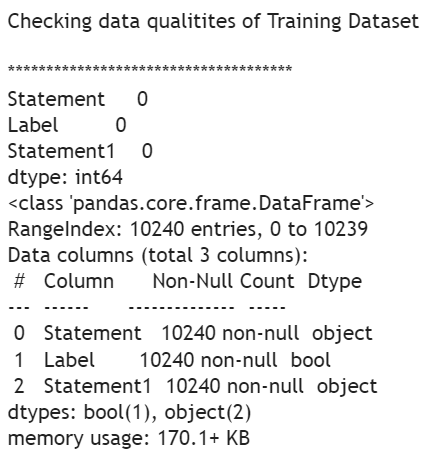
**Figure 15: Shape of training dataset** (Refer Appendix A for more)

1. **Data Distribution**: These steps are carried out to get an idea about distribution of true data and fake data in all 3 datasets. These graphs indicate that the data distribution is fair and balanced.



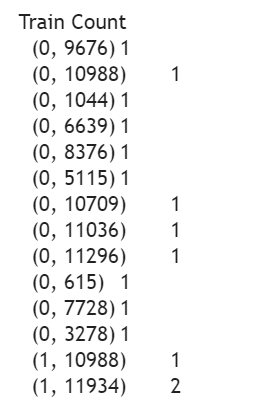
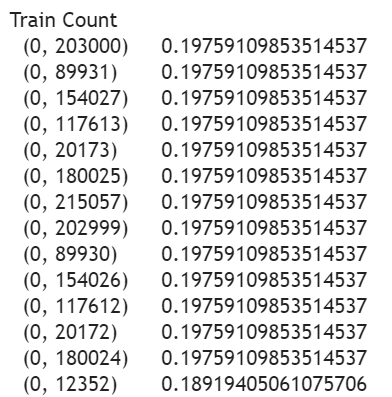
***Figure 16: Data distribution of Training*** (Refer Appendix A for more)

1. **Data integrity check**: This is to check whether there are any null values in all 3 datasets.

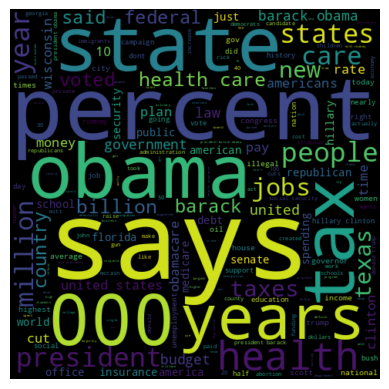
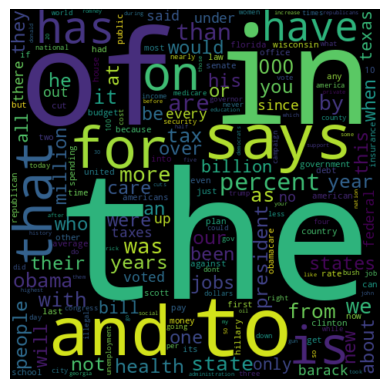


***Figure 17: Dataset quality check - Training*** (Refer Appendix A for more)

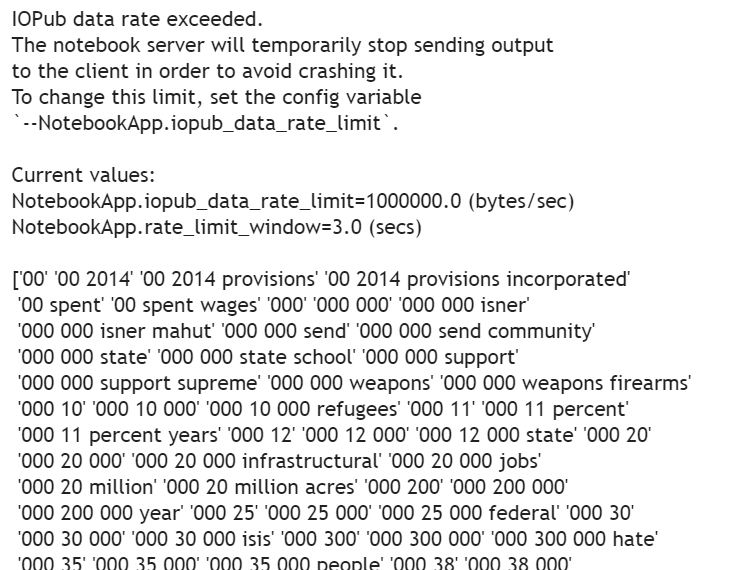
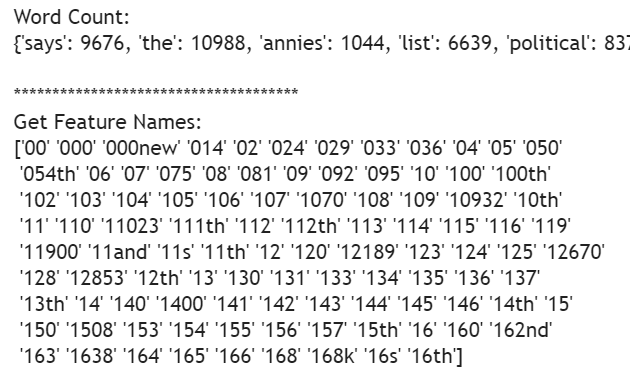
1. **Count Vectorizer and Tfidf Vectorizer**: This is the step to convert the text data into numeric data. These techniques are described in the section 3.3. above. Train count matrix and Word Clouds are used to show how the words are distributed in the datasets. Also Word count and Feature names can be viewed.

***Figure 18: Train count for CountV & TfidfV***



***Figure 19: Word Clouds for CountV & TfidfV***



***Figure 20: Word Count and Feature names for CountV and TfidfV***

***Note***: The pipeline function in Python is used to eliminate multiple steps of code and it will run multiple transformation in one go. The pipeline object is then fit on the training set using the 'fit' method, which trains the model on the training data. The fitted pipeline is then used to make predictions on the test set using the 'predict' method.

A pickle file for each model is built at the end of each function which can be re-used without training datasets again and again.

1. **Building Models using Bag of Words Technique (Count Vectorizer)**

Naïve Bayes, Logistic Regression, SVM, SGD, Random Forest and Passive Aggressive binary classifiers are being used in building the machine learning models using Count Vectorizer.

The F1-Score and Accuracy score is calculated for the purpose of comparing models in the evaluation.

A classification report is produced that shows the precision, recall, f1-score and accuracy and weighted average.

|  |  |
| --- | --- |
| **Figure 21: Classification Report - Model - Naive Bayes** | **Figure 22: Classification Report - Model - Logistic Regression** |

(Refer Appendix A for more Classification reports)

1. **Building Models using N-Grams Technique (TfidfVectorizer)**

Naïve Bayes, Logistic Regression, SVM, SGD, Random Forest and Passive Aggressive binary classifiers are being used in building the machine learning models using Tfidf Vectorizer.

The F1-Score and Accuracy score is calculated for the purpose of comparing models in the evaluation (section 5.2.).

A classification report is produced that shows the precision, recall, f1-score and accuracy and weighted average.

|  |  |
| --- | --- |
| **Figure 27: Classification Report - Model - Naive Bayes** | **Figure 28: Classification Report - Model - Logistic Regression** |

1. **K-fold Cross Validation for all classifiers**

k-fold cross-validation is a technique used to assess the performance of a model by splitting the data into k equal-sized folds and training the model on k-1 folds while testing it on the remaining fold. This process is repeated k times, with each fold being used as the test set once.

The k-fold cross-validation technique helps to address the problem of overfitting by using all the available data for both training and testing, without sacrificing the ability to evaluate the performance of the model. It also provides a more accurate estimate of the model's performance than a single train-test split, as it uses multiple splits of the data.

Here's how k-fold cross-validation works:

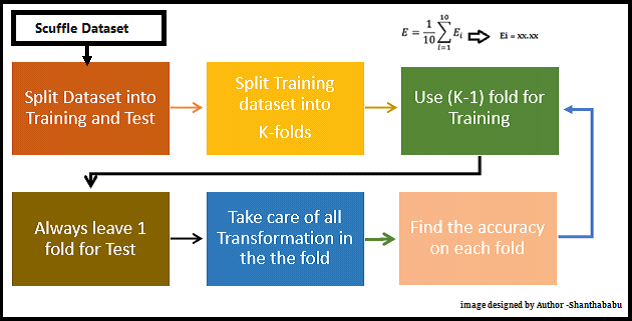
Split the data into k equal-sized folds.

For each of the k folds:

1. Use k-1 folds for training the model.
2. Use the remaining fold for testing the model.
3. Evaluate the performance of the model on the test set.

Repeat the process k times, using each fold as the test set once.

Calculate the average performance across all k folds to get an estimate of the model's performance.

[[23]](#footnote-23) (Anon., n.d.)

|  |  |
| --- | --- |
| **Graphical representation of Confusion Matrices generated for different Models with Count Vectorizer** | |
| **Figure 33: Confusion Matrix - Model - Naive Bayes - CountV** | **Figure 34: Confusion Matrix - Model - Logistic Regression - CountV** |

|  |  |
| --- | --- |
| **Graphical representation of Confusion Matrices generated for different Models with Tfidf Vectorizer** | |
| **Figure 39: Confusion Matrix - Model - Naive Bayes - TfidfV** | **Figure 40: Confusion Matrix - Model - Logistic Regression - TfidfV** |

(Refer Appendix A for more Confusion Matrices)

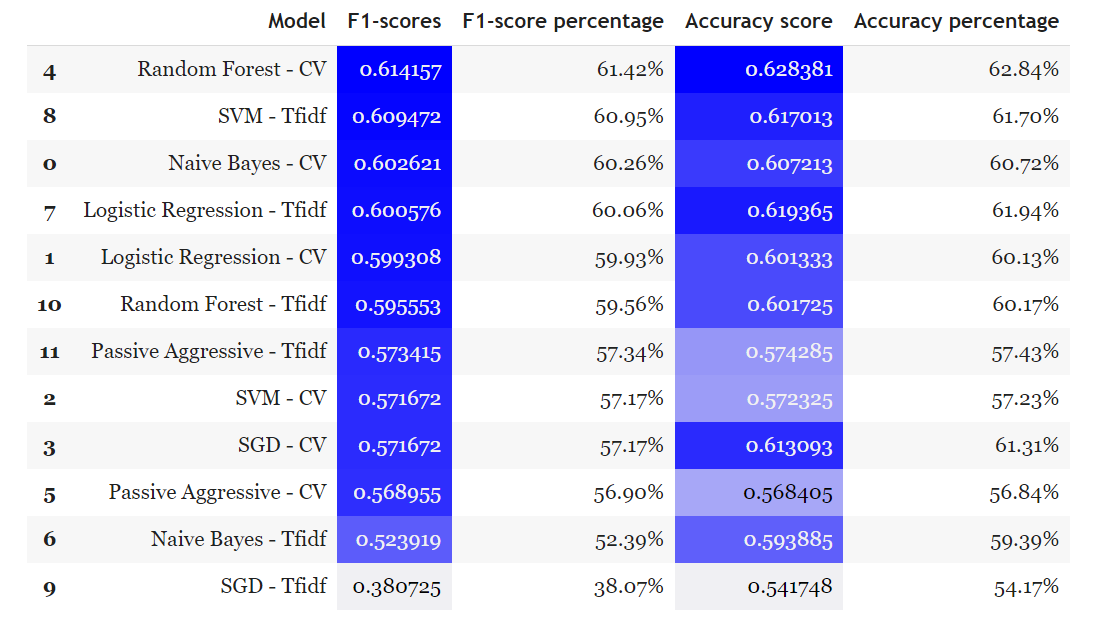
# Evaluation, Testing and Results

This section of the project explains the Evaluation, Testing and Results obtained during this project work. Evaluation is done in this project by comparing different models and visualizing the results. Testing is done using different

## Model Comparison

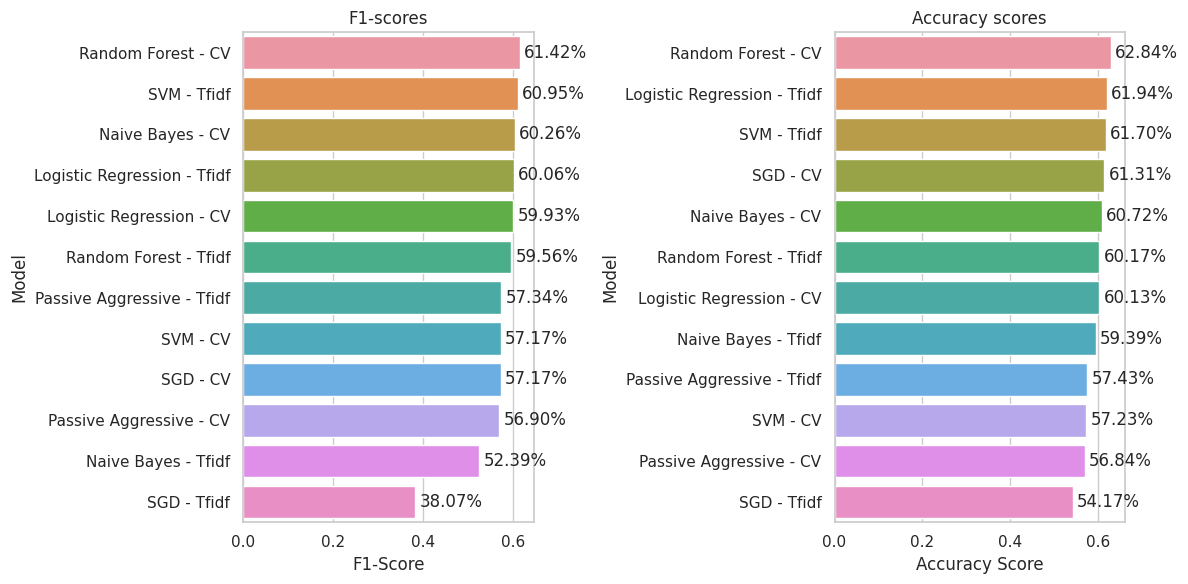
Each model is compared using a table, graphs and bar charts based on F1-Score and Accuracy.

* 1. Tabular comparison - All the models are listed in the below table, and it is sorted using the F1-Score. Accuracy score is also given in it.



**Table 3: Evaluation - Sort using F1-score**

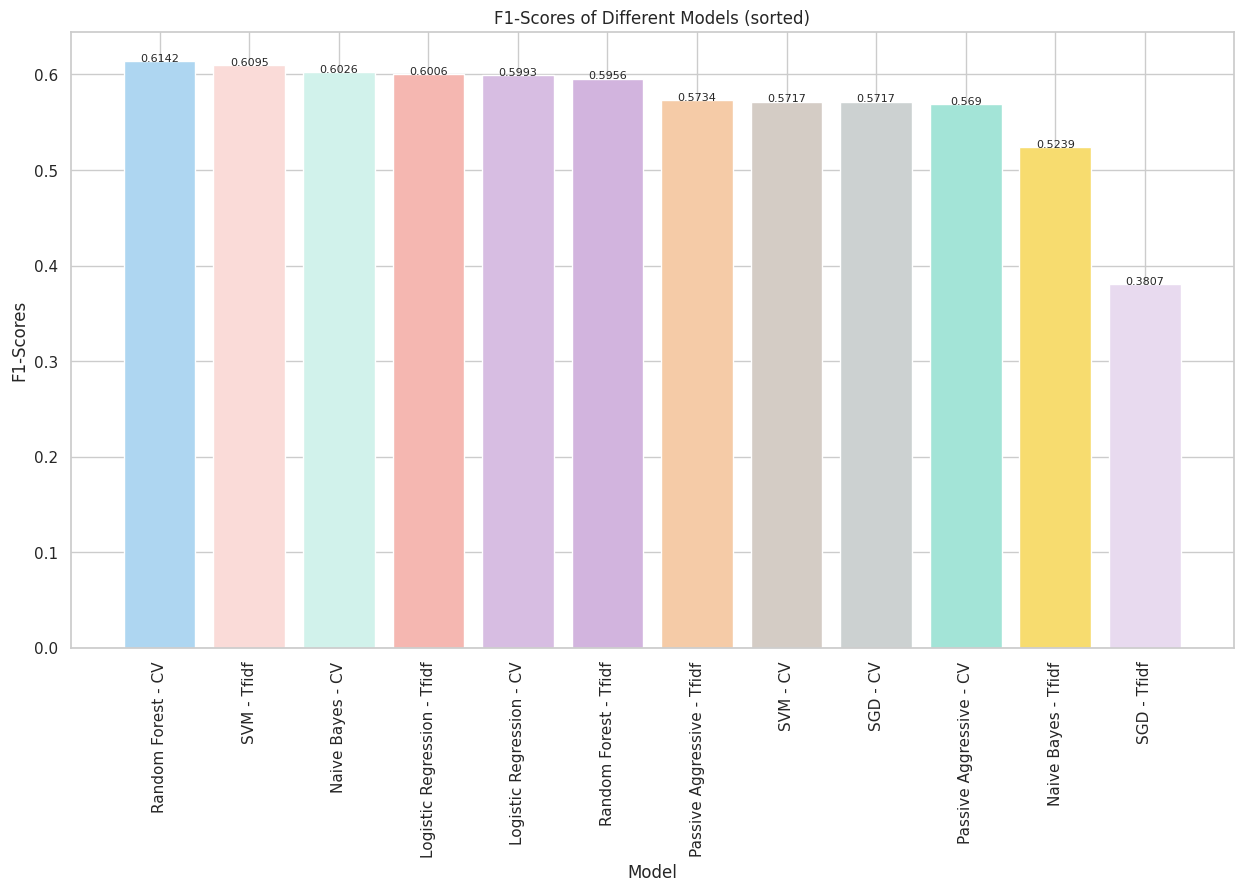
* 1. Graphical comparison



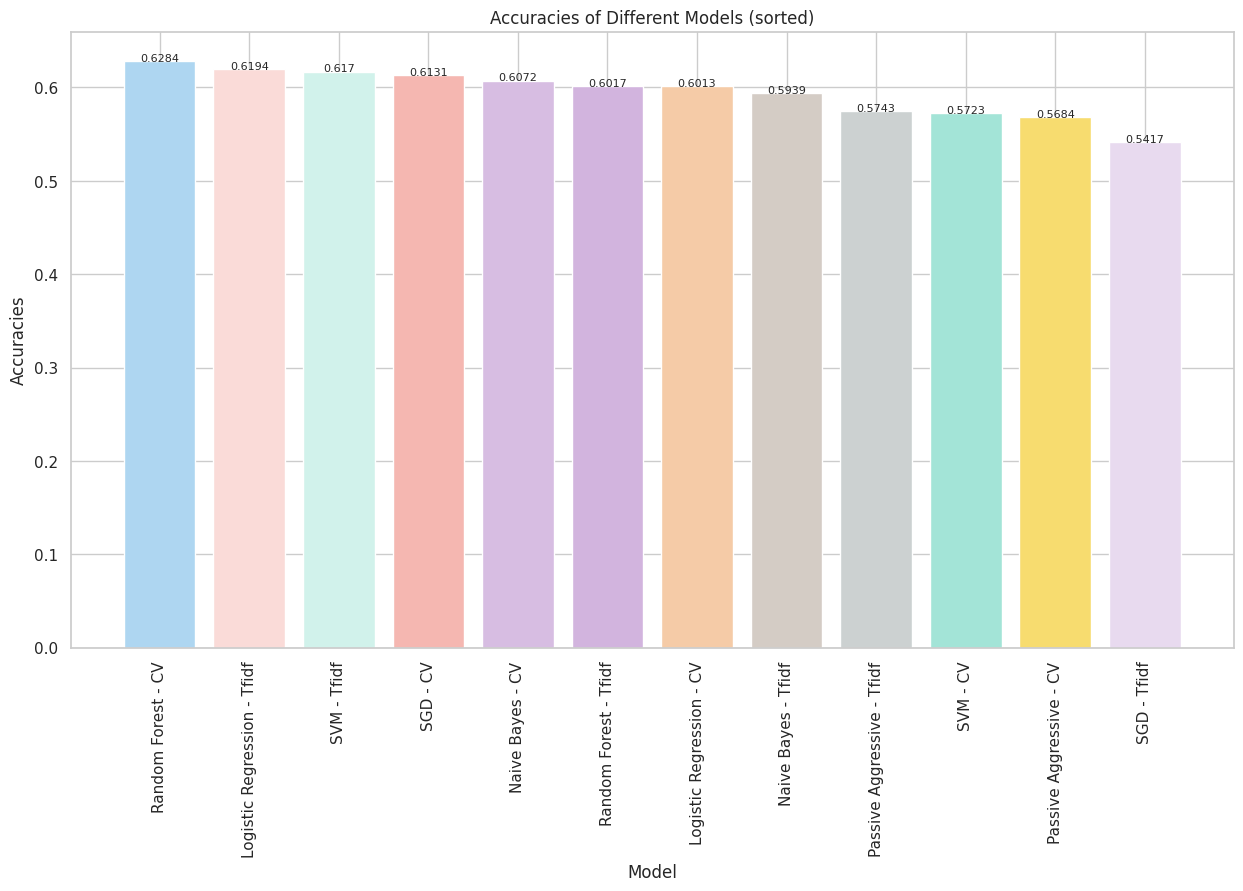
**Figure 45: Evaluation - F1-score and Accuracy score**

The Random Forest model developed using Count Vectorizer tops all models in this comparison and SGD model developed using Tfidf Vectorizer scores last.

* 1. Bar chart comparison



**Figure 46: F1-Scores of Different Models (sorted)**



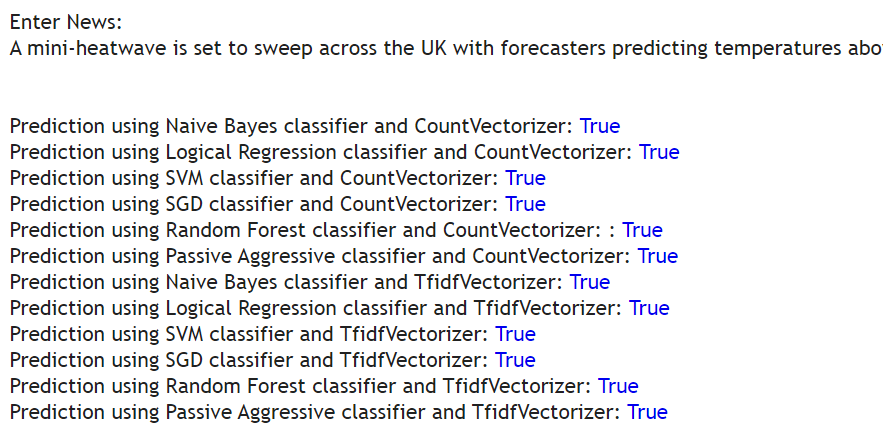
**Figure 47: Accuracies of Different Models (sorted)**

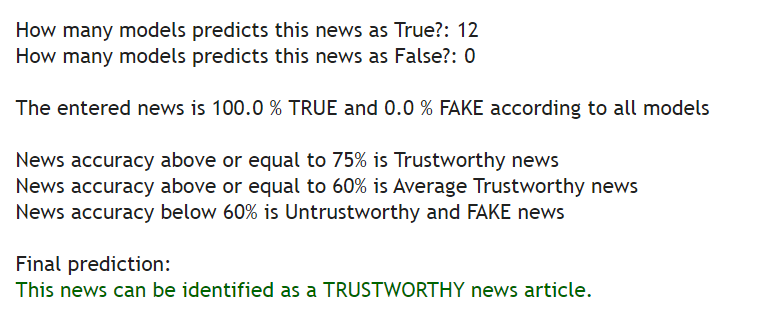
## Testing all models Pickle files

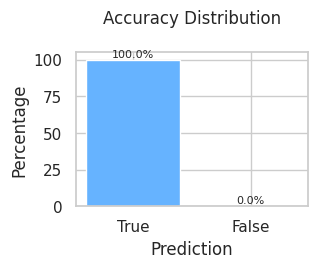
The pickle files created when training models will be loaded and unpickled in this step. The preprocess\_text() and classify\_news() functions are defined and these functions will be called in the next step.

## Real World News

When the user enters a news article into the input text field, preprocess\_text() and classify\_news() functions will be called and it will predict True or False for each of the models.







**Figure 48: Predictions for all models**

## Debugging

It is advised to use multi-step approach to debug fake news project and following are few steps that can be carried out and ensure that information is accurate and credible:

**Check the source**: The source of the news article can be checked. The reputability of source, history of publishing accurate news, basic check on the website such as domain name, about us section, and the contact information can be checked to ensure whether the source is legitimate or not.

**Check the author**: The author of the news also can be used to validate a source of news; author name, background, history of writing credible news, expertise in the field can be checked and if the author information is not available, it can be googled for further assurance.

**Fact-check the story**: Fact-checking websites like FactCheck.org or Snopes.com to verify the story of the news article.

**Check the date**: Some news can be out-dated and there could be updated version of the same news so such news can be identified as fake news.

**Analyze the content**: News articles that seem to be bias, misinformation, or propaganda in the story so such articles cannot be either fake or real, which can be difficult to decide as fake or real.

**Use tools**: NewsGuard, Factmata, and Media Bias / Fact Check are some tools that can detect potentially fake news and provide additional information about the sources.

**Consult with experts**: It is always best to consult an expert in the field or journalists who cover the topic to make an informed decision. (Anon., n.d.) [[24]](#footnote-24)

## Implementation

The functional aspects and non-functional aspects of the Finding Fake News project is described below.

### Functional aspects

These are some of the functional aspects to consider when implementing a Finding fake news project:

**Data collection**: The system uses datasets from Github as well as Kaggle that has reliable and comprehensive datasets of news articles.

**Data visualization**: The system uses ***Matlab*** and ***Seaborn*** to generate visualizations in a way that is easy to understand and helps users identify trends and patterns.

**User feedback**: A simple survey prepared using Google Form or Microsoft Form can be used to collect user feedback. This will be implemented in the future. (*6.7. Improvements and Future work*)

### Benchmarking against the work of others

This project uses multiple classification models to predict whether some news article is True or Fake while the work done by others uses only one classification model. This is a significant difference in this system although some re-uses it’s features and coding with significant improvements.

### Non-functional aspects

When implementing a project to find fake news, there are several non-functional aspects to consider, such as Security, Scalability, Performance, Usability, Accessibility and these aspects are critical to ensuring the effectiveness, reliability, and usability of a fake news detection system

**Security**: The security requirements should be researched, identified, and evaluated to protect it against cyber threats and prevent unauthorized access. (*6.7. Improvements and Future work*)

**Accessibility**: The accessible requirements also must be identified and addressed so that system is available to users with disabilities and provide support for multiple languages and cultural contexts. (*6.7. Improvements and Future work*)

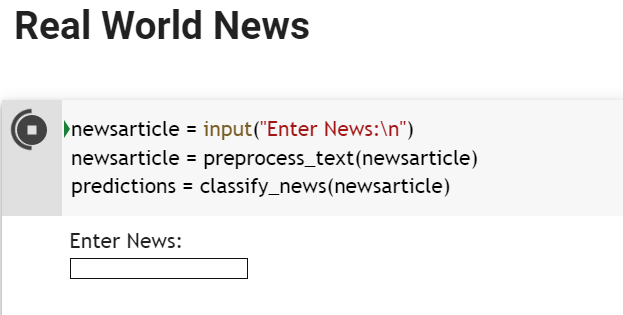
## Description of the Final Product

Finding fake news project was completely created and tested in a Google Colab notebook although I had the initial requirement to create a small system using Flask app and run on it. Due to some limitations in the IDE (Pycharm), the final product is developed, run, and tested within Google Colab notebook.

* The user should enter the desired news article into the text field
* The system pre-processes the article and runs the article through unpickled models. Then the system produces the prediction based on each of the models.
* Then it shows how many models have predicted the news article as True and as False.
* Then the system produces the final prediction based on the above criteria
* Then the system produces a visualization of the prediction in percentages.

### Coding

While the coding is saved in Github, the code snippet is shown:



**Figure 49: A code snippet with input text field**

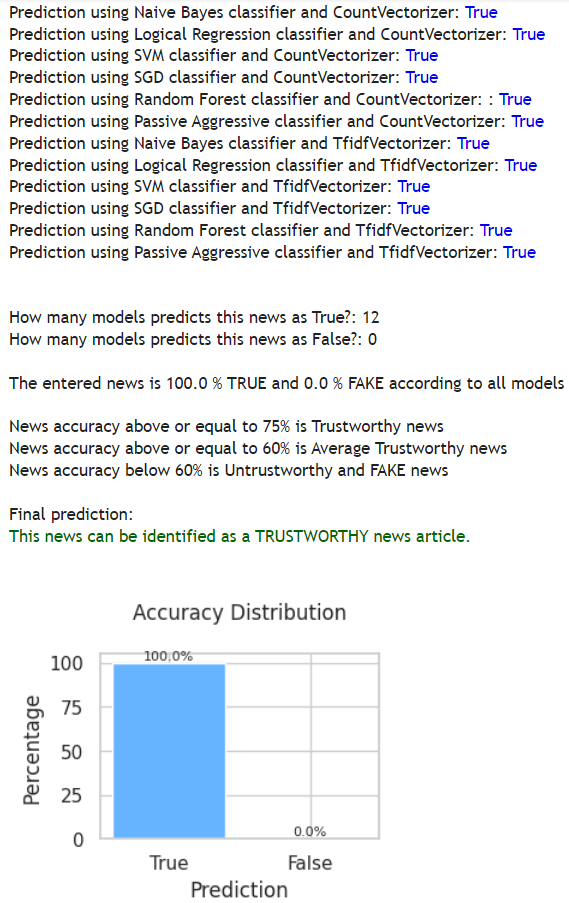
### Input

***News article URL***: <https://www.bbc.co.uk/news/uk-england-65371512>

***News article:*** *The sky turned vivid shades of purple and was spotted from Cumbria to as far south as the Isles of Scilly. The phenomenon is caused by the interaction of the solar wind and Earth's magnetic field and upper atmosphere, said the Met Office. Clearer skies tonight may mean there will be a greater chance for some good views of the aurora, it said. Aurorawatch, UK. a service run by Lancaster University, issued a red alert indicating it was likely that aurora would be visible by eye and camera from anywhere in the UK.*

### Output

This code snippet generates the following output:



**Figure 50: Final Output - Prediction and Analysis**

# Critical Appraisal, Summary and Conclusions

## Critical Appraisal

I wish I could start this project by going through the concepts first rather than during the middle of the project which is something I would do if I were to do this project all over again.

If I were to develop this project again, I would also consider learn Visual Studio Code and develop this project in it and develop a Flask based app to run the final product.

Also, I would develop or use Webscraper to generate datasets and use it in this project.

## Summary

This project has been a challenging but interesting experience for me from the beginning till the end, since I had to experience so many obstacles, overcome fears and finally learn and master a productive and well recognized skill and that will paint my future as a Data Analyst or Data Engineer. I am glad that I ended up picking MSc in Data Science and Engineering which filled my life with so many different life experiences including migration.

Although I am a quick learner, it took some time for me to refresh my memory and understand Machine Learning concepts and techniques again, since we had the Introduction to Machine Learning module in the previous semester and I wish I could do this project right after learning that module. Also, the understanding of each project topic was quite low and unsure at the time of the selection of the project, so it took some time to understand the concepts of the selected project and what needs to be done was bit hesitant at the start.

But the guidance and encouragement of the generous supervisors made this learning curve pretty much less stressful and interesting, and their feedback was always right there every time I needed.

Despite the challenges, obstacles (sicknesses and accidents), and delays during the project, as well as learning technologies that are not familiar to me before, I am delighted to receive satisfying feedback on my project work as well as the project report and I find this state as success and contentment.

Also, I am happy to say that I made so many mistakes during this project and managed to work through again and correct them with guidance of supervisors and course coordinators, who were away from just an e-Mail.

## Conclusion

Finally, the models developed can determine whether a given news article is True or Fake and generate a detailed report with a visualization. Since the datasets used in the project are balanced, a satisfactory conclusion can be given that the models created are highly accurate so the project is a success.

# Future Work

This section describes the suggested future work to the project Finding Fake News so it can ignore such news and proceed:

1. Hyper-parameter tuning will be introduced into the system to cross validate models created.
2. Use of different datasets from different domains to train the models.
3. Create a Webscrape app and download news articles.
4. Integrate the model into a web application or other platform that can help users identify fake news. This can be done using APIs, browser extensions, or other tools that will be researched further.
5. Continuously update the model: Monitor and update the model regularly to keep up with new trends and tactics used by fake news publishers.
6. A simple survey prepared using Google Form or Microsoft Form can be used to collect user feedback. This will be implemented in the future.
7. The security requirements should be researched, identified, and evaluated to protect it against cyber threats and prevent unauthorized access.
8. The accessible requirements also must be identified and addressed so that system is available to users with disabilities and provide support for multiple languages and cultural contexts.
9. Further going, identifying news articles in text form is comparatively easy so it would be big challenge to identify news in the form of images and videos. Going forward, the concept of building a system to identify these challenges will be highly recognized if one become able to develop.

# References / Bibliography

**General information and reading material**

DataFlair. (2019). Advanced Python Project - Detecting Fake News with Python. [online] Available at: https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/

www.mindtools.com. (n.d.). MindTools | Home. [online] Available at: https://www.mindtools.com/a0g6bjj/how-to-spot-real-and-fake-news

BBC Bitesize. (n.d.). Fact or fake. [online] Available at: https://www.bbc.co.uk/bitesize/tags/zr2yscw/fact-or-fake/1

www.learndatasci.com. (n.d.). Binary Classification. [online] Available at: https://www.learndatasci.com/glossary/binary-classification/.

de Beer, D. and Matthee, M. (2020). Approaches to Identify Fake News: A Systematic Literature Review. Integrated Science in Digital Age 2020, [online] 136, pp.13–22. doi:https://doi.org/10.1007/978-3-030-49264-9\_2.

de Oliveira, N.R., Pisa, P.S., Lopez, M.A., de Medeiros, D.S.V. and Mattos, D.M.F. (2021). Identifying Fake News on Social Networks Based on Natural Language Processing: Trends and Challenges. Information, [online] 12(1), p.38. doi:https://doi.org/10.3390/info12010038.

Yufeng G (2017). The 7 Steps of Machine Learning. [online] Medium. Available at: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e.

Alokesh (2020). *Passive Aggressive Classifiers*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/passive-aggressive-classifiers/>.

Arthanayake, N. (2021). *Tips for spotting fake news online*. [online] BBC Bitesize. Available at: <https://www.bbc.co.uk/bitesize/articles/zrprrj6>.

Arya, N. (2022). *Logistic Regression for Classification*. [online] KDnuggets. Available at: <https://www.kdnuggets.com/2022/04/logistic-regression-classification.html>.

B, H.N. (2020). *Confusion Matrix, Accuracy, Precision, Recall, F1 Score*. [online] Medium. Available at: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>.

de Beer, D. and Matthee, M. (2020). Approaches to Identify Fake News: A Systematic Literature Review. *Integrated Science in Digital Age 2020*, [online] 136, pp.13–22. doi:<https://doi.org/10.1007/978-3-030-49264-9_2>.

de Oliveira, N.R., Pisa, P.S., Lopez, M.A., de Medeiros, D.S.V. and Mattos, D.M.F. (2021). Identifying Fake News on Social Networks Based on Natural Language Processing: Trends and Challenges. *Information*, [online] 12(1), p.38. doi:<https://doi.org/10.3390/info12010038>.

Dinesh (2021). *කෘතීම බුද්ධිය | Artificial Intelligence in Sinhala*. [online] www.youtube.com. Available at: <https://www.youtube.com/watch?v=A7d_yQ4ECNE&list=PL495mke12zYDHN9ONfcal1eQfo8VqmOgu> [Accessed 25 Apr. 2023].

Gandhi, R. (2018a). *Naive Bayes Classifier*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>.

Gandhi, R. (2018b). *Support Vector Machine — Introduction to Machine Learning Algorithms*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.

Johnson, D. (2020). *Stemming and Lemmatization in Python NLTK with Examples*. [online] www.guru99.com. Available at: <https://www.guru99.com/stemming-lemmatization-python-nltk.html>.

Kapl, D., coding, ylan K. has years of experience as a S.D.S.H. enjoys, teaching and everyone, has created this website to make M.L. accessible to (2022). *Machine Learning 101: CountVectorizer Vs TFIDFVectorizer» EML*. [online] enjoymachinelearning.com. Available at: <https://enjoymachinelearning.com/blog/countvectorizer-vs-tfidfvectorizer> [Accessed 25 Apr. 2023].

Kiely, E. and Robertson, L. (2016). *How to Spot Fake News - FactCheck.org*. [online] FactCheck.org. Available at: <https://www.factcheck.org/2016/11/how-to-spot-fake-news/>.

Koo Ping Shung (2018). *Accuracy, Precision, Recall or F1?* [online] Towards Data Science. Available at: <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>.

Krasadakis, G. (2021). *How to stop Fake News and misinformation using digital technologies*. [online] The Innovation Mode. Available at: <https://www.theinnovationmode.com/the-innovation-blog/misinformation-online-a-solution-powered-by-state-of-the-art-tech>.

Pandian, S. (2022). *K-Fold Cross Validation Technique and its Essentials*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/>.

Peiris, S. (2023). *MSc Project - Finding Fake News - Process*. [online] Google Docs. Available at: <https://docs.google.com/drawings/d/1weZghLELVhQCpKcIhwlqKIovowDQsG19WNy7l_dSWy0/edit> [Accessed 25 Apr. 2023].

Radhika (2020). *Mathematics Behind SVM | Math Behind Support Vector Machine*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/>.

Saini, A. (2021). *Random Forest Algorithm for Absolute Beginners in Data Science*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/>.

Scikit-learn (2018). *3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation*. [online] Scikit-learn.org. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

scikit-learn.org. (n.d.). *1.5. Stochastic Gradient Descent — scikit-learn 0.23.2 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/sgd.html>.

Tate, S. (2023). *Tackling Fake News with Machine Learning*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2023/02/tackling-fake-news-with-machine-learning/> [Accessed 25 Apr. 2023].

www.tutorialspoint.com. (n.d.). *Scikit Learn - Stochastic Gradient Descent - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/scikit_learn/scikit_learn_stochastic_gradient_descent.htm>.

Zhou, X. and Zafarani, R. (2020). A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys*, 53(5). doi:<https://doi.org/10.1145/3395046>.

**Fake News Detection Analysis**

www.youtube.com. (n.d.). Project 4. Fake News Prediction using Machine Learning with Python | Machine Learning Projects. [online] Available at: https://www.youtube.com/watch?v=nacLBdyG6jE&t=1980s [Accessed 25 Apr. 2023].

www.youtube.com. (n.d.). Fake News Detection Project | Data Science with Python | Machine Learning session 2. [online] Available at: https://www.youtube.com/watch?v=xyq-zYr1cnI&t=6330s [Accessed 25 Apr. 2023].

**Dataset & Coding references**

nishitpatel01 (2018). nishitpatel01/Fake\_News\_Detection. [online] GitHub. Available at: https://github.com/nishitpatel01/Fake\_News\_Detection [Accessed 29 May 2019].

**Pipeline function**

scikit-learn.org. (n.d.). sklearn.pipeline.Pipeline — scikit-learn 0.24.1 documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html.

**Word Cloud**

Holtz, Y. (n.d.). Custom python wordcloud. [online] The Python Graph Gallery. Available at: https://www.python-graph-gallery.com/261-custom-python-wordcloud?utm\_content=cmp-true [Accessed 25 Apr. 2023].

**Documentation reference**

Shoemaker, E. (n.d.). JMU Scholarly Commons Using data science to detect fake news. [online] Available at: https://commons.lib.jmu.edu/cgi/viewcontent.cgi?article=1754&context=honors201019.

FAKE NEWS DETECTION WITH REAL NEWS GENERATION A PROJECT REPORT BACHELOR OF TECHNOLOGY DEPARTMENT OF INFORMATION TECHNOLOGY FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR -603203. (2021). Available at: http://dspace.srmist.edu.in/jspui/bitstream/123456789/44561/1/P11595.pdf [Accessed 25 Apr. 2023].

**Coding**

Badole, Mayur. “Difference between Fit(), Transform(), Fit\_transform() Methods in Scikit-Learn (with Python Code).” *Analytics Vidhya*, 30 Apr. 2021, www.analyticsvidhya.com/blog/2021/04/difference-between-fit-transform-fit\_transform-methods-in-scikit-learn-with-python-code/.

BSD. “Sklearn.feature\_extraction.text.CountVectorizer — Scikit-Learn 0.20.3 Documentation.” *Scikit-Learn.org*, 2018, scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html.

---. “Sklearn.pipeline.Pipeline — Scikit-Learn 0.24.1 Documentation.” *Scikit-Learn.org*, scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html.

Dickinson, Bryan. “Generate Meaningful Word Clouds in Python.” *Medium*, 12 Feb. 2021, towardsdatascience.com/generate-meaningful-word-clouds-in-python-5b85f5668eeb.

Dinesh. “Python Tutorial - 55 | Pickle Module (Serializing) | Sinhala.” *Www.youtube.com*, 2021, www.youtube.com/watch?v=D-zT4tqbebQ&t=2s. Accessed 25 Apr. 2023.

Harrison. “Python Programming Tutorials.” *Pythonprogramming.net*, pythonprogramming.net/python-pickle-module-save-objects-serialization/.

Khanna, Chetna. “Text Pre-Processing: Stop Words Removal Using Different Libraries.” *Medium*, 10 Feb. 2021, towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a.

---. “What and Why behind Fit\_transform() vs Transform() in Scikit-Learn !” *Medium*, 25 Dec. 2020, towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe.

Kharwal, Aman. “Classification Report in Machine Learning.” *Data Science | Machine Learning | Python | C++ | Coding | Programming | JavaScript*, 7 July 2021, thecleverprogrammer.com/2021/07/07/classification-report-in-machine-learning/.

Nantasenamat, Chanin. “Google Colaboratory.” *Colab.research.google.com*, 29 Mar. 2023, colab.research.google.com/drive/1Tmqs4934VEE2xMdQlCiYwY-GhQsakL\_n#scrollTo=-JbHzHy6oEix. Accessed 25 Apr. 2023.

Nishit Patel. “Fake News Detection.” *GitHub*, 23 Apr. 2023, github.com/nishitpatel01/Fake\_News\_Detection/blob/master/final-fnd.ipynb. Accessed 25 Apr. 2023.

Rene D. “ANSI Color Codes in Python.” *Gist*, 2020, gist.github.com/rene-d/9e584a7dd2935d0f461904b9f2950007.

SciKit-Learn. “3.1. Cross-Validation: Evaluating Estimator Performance — Scikit-Learn 0.21.3 Documentation.” *Scikit-Learn.org*, 2009, scikit-learn.org/stable/modules/cross\_validation.html.

Sphinx. “Pandas.DataFrame.dropna — Pandas 1.3.1 Documentation.” *Pandas.pydata.org*, pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html.

---. “Seaborn.color\_palette — Seaborn 0.12.2 Documentation.” *Seaborn.pydata.org*, seaborn.pydata.org/generated/seaborn.color\_palette.html.

---. “Styling — Pandas 1.1.2 Documentation.” *Pandas.pydata.org*, pandas.pydata.org/pandas-docs/stable/user\_guide/style.html.

StudyTonight. “How to Print Colored Text in Python - Studytonight.” *Www.studytonight.com*, www.studytonight.com/python-howtos/how-to-print-colored-text-in-python?utm\_content=cmp-true. Accessed 25 Apr. 2023.

Vasanth. “How to Combine Single Pkl File? · Issue #1431 · Open-Mmlab/Mmaction2.” *GitHub*, 2022, github.com/open-mmlab/mmaction2/issues/1431. Accessed 25 Apr. 2023.

Waskom, Michael. “Seaborn.countplot — Seaborn 0.9.0 Documentation.” *Pydata.org*, 2012, seaborn.pydata.org/generated/seaborn.countplot.html.

# List of Appendices

These are the appendices, and it will be attached as separate files.

1. : Source Code
2. : Project Plan
3. : Datasets downloaded from github

1. Electronic signatures are acceptable [↑](#footnote-ref-1)
2. Electronic signatures are acceptable [↑](#footnote-ref-2)
3. <https://www.mdpi.com/2078-2489/12/1/38> [↑](#footnote-ref-3)
4. <https://dl.acm.org/doi/abs/10.1145/3395046> [↑](#footnote-ref-4)
5. <https://www.bbc.co.uk/bitesize/articles/zrprrj6> [↑](#footnote-ref-5)
6. <https://link.springer.com/chapter/10.1007/978-3-030-49264-9_2> [↑](#footnote-ref-6)
7. <https://www.theinnovationmode.com/the-innovation-blog/misinformation-online-a-solution-powered-by-state-of-the-art-tech> [↑](#footnote-ref-7)
8. <https://www.guru99.com/stemming-lemmatization-python-nltk.html> [↑](#footnote-ref-8)
9. <https://enjoymachinelearning.com/blog/countvectorizer-vs-tfidfvectorizer> [↑](#footnote-ref-9)
10. <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c> [↑](#footnote-ref-10)
11. <https://www.kdnuggets.com/2022/04/logistic-regression-classification.html> [↑](#footnote-ref-11)
12. <https://www.dotnetlovers.com/article/225/logistic-regression-explained> [↑](#footnote-ref-12)
13. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> [↑](#footnote-ref-13)
14. <https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/> [↑](#footnote-ref-14)
15. <https://www.tutorialspoint.com/scikit_learn/scikit_learn_stochastic_gradient_descent.htm> [↑](#footnote-ref-15)
16. <https://scikit-learn.org/stable/modules/sgd.html> [↑](#footnote-ref-16)
17. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> [↑](#footnote-ref-17)
18. <https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/> [↑](#footnote-ref-18)
19. <https://www.geeksforgeeks.org/passive-aggressive-classifiers/> [↑](#footnote-ref-19)
20. <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd> [↑](#footnote-ref-20)
21. <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9> [↑](#footnote-ref-21)
22. <https://docs.google.com/drawings/d/1weZghLELVhQCpKcIhwlqKIovowDQsG19WNy7l_dSWy0/edit> [↑](#footnote-ref-22)
23. <https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/> [↑](#footnote-ref-23)
24. <https://www.factcheck.org/2016/11/how-to-spot-fake-news/> [↑](#footnote-ref-24)