



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: V Month of publication: May 2023

DOI: <https://doi.org/10.22214/ijraset.2023.51637>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Fake News Detection using Machine Learning - A Working Model of Fake News Detection

E. V. Nagalakshmi¹, E. Sai Vineeth², Y. Goutham³, T. Vamshi Krishna⁴

¹Assistant Professor

Electronics And Communication Engineering, Maturi Venkats Subbs Rao(MVSR) Engineering College, Nadargul, Hyderabad

Abstract: *This project aims to address the pressing issue of fake news, which has become increasingly prevalent in today's society. With the internet and social media making news more accessible than ever, the spread of fake news can have a significant impact on social, economic, and political environments. In response to this challenge, this project investigates the use of machine learning algorithms to accurately classify news as real or fake. The project utilizes KNN, Decision Tree, and Logistic Regression algorithms to analyze large datasets of news articles and learn the patterns and characteristics of real and fake news. The primary objective of this project is to provide users with a tool that can accurately detect fake news and help prevent its spread*

I. INTRODUCTION

Fake news has become a prevalent issue in modern society due to the rapid dissemination of information through social media platforms and other digital channels. The spread of fake news can have serious consequences, leading to the manipulation of public opinion, political instability, and even violence. Therefore, it is essential to identify and prevent the spread of fake news to mitigate its negative impact. In this report, we explore the use of machine learning algorithms for fake news detection.

Machine learning algorithms can be used to automate the process of fake news detection by learning patterns and characteristics that distinguish fake news from genuine news. Machine learning algorithms use statistical methods to learn from labelled data and generalize to new, unseen data. This approach is particularly well-suited to fake news detection as it can leverage large volumes of data to identify subtle patterns and trends that might be difficult for humans to discern. By training machine learning models on large datasets of labelled news articles, it is possible to develop highly accurate models that can classify new articles as either fake or genuine with high precision. This can help to prevent the spread of fake news and protect the public from misinformation and manipulation.

A. Objectives Of The Project

- 1) Develop a machine learning model that can accurately authenticate whether a given news article is real or fake.
- 2) Collect and prepare a diverse set of training data that includes various scenarios and aspects of fake news.
- 3) Explore and use different machine learning algorithms such as logistic regression, k-nearest neighbors, and decision tree classifier to find the best algorithm for detecting fake news.
- 4) Evaluate the performance of each algorithm based on accuracy and confusion matrix.
- 5) Continuously refine and improve the model to adapt to the ever-changing landscape of fake news.

II. LITERATURE SURVEY

- 1) *Convolutional Neural Networks:* Rohit Kumar Kaliyar developed A Deep Neural Network techniques for fake news detection . He used Fake or Real News Dataset for detecting the fake news by classifying with Convolutional Neural Networks, Long Short term memory, Naïve Bayes, Decision Tree, Random Forest and K-Nearest Neighbour techniques. In this by increasing the depth of the network the accuracy is increased when using the CNN method. In this by using k-nearest neighbour algorithm the accuracy is decreased and also precision, recall, f1-score values are reduced. In this he gained maximum accuracy of 91.3% by using CNN algorithm. Belhakimi Mohamed Amine, Ahlem Drif, Silvia Giordano developed a Merging deep learning model for fake news detection
- 2) *B. K-Nearest Neighbour:* Ankit Kesarwani, Sudakar Singh Chauhan and Anil Ramachandran Nair developed a K-Nearest Neighbour Classifier technique for Fake News Detection on Social Media. In this they use Buzz Feed news. It contains the information about the Facebook news. In this the model has achieved maximum accuracy when the value of K taken between 15 to 20. In this they gain the maximum accuracy of 79% tested against Facebook news dataset.

- 3) *C. Logistic Regression:* It is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes and success) or 0 (no and failure). Uma Sharma, Sidarth Saran, Shankar M. Patil developed a Fake News Detection using Machine Learning Algorithms They used liar dataset for detecting if fake news by Naïve Bayes Classifier, Logistic Regression, Random Forest. They used Bag-Of-Words, N-Grams, TF-IDF. Logistic regression shows better results with accuracy of 65%. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf and Muhammad Ovais Ahmad implemented a model for fakenews detection on social media

III. METHODOLOGY FOR FAKE NEWS DETECTION USING ML

A. Data Set

For a project on fake news detection using machine learning, several Python libraries were utilized to process and analyze the data, as well as to train and evaluate the machine learning models. These libraries included NumPy, Pandas, Seaborn, Matplotlib, and scikit-learn (sklearn).

To load and preprocess the data, NumPy and Pandas were used, which provided efficient numerical operations and powerful data manipulation and analysis capabilities, respectively. Seaborn and Matplotlib were also used to create visualizations of the data, which helped to better understand its characteristics and identify potential patterns. Finally, scikit-learn was used to train and evaluate the machine learning models. This library provided a wide range of machine learning algorithms, as well as tools for data preprocessing, feature extraction, and model selection. Common algorithms used in fake news detection include decision trees, support vector machines (SVMs), and neural networks. To prepare the data for machine learning, the data was first loaded and cleaned using Pandas. Scikit-learn was then used to extract features from the textual data, such as sentiment analysis, lexical diversity, and topic modeling. These features were used to train and evaluate the machine learning models, which were also implemented using scikit-learn. Overall, the libraries used in the project provided a powerful set of tools for processing, analyzing, and visualizing the data, as well as for training and evaluating the machine learning models. These tools were critical in enabling effective detection of fake news from textual data using machine learning

Figure 1 : Libraries used in project.

```
In [1]:
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings; warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
```

B. Data Pre-Processing

- 1) *Data Cleaning:* The first step in data preprocessing is cleaning the data, which involves removing any unnecessary or irrelevant information from the dataset, such as HTML tags, URLs, and special characters. This can be done using regular expressions and built-in Python functions.
- 2) *Text Tokenization:* After cleaning the data, the text needs to be tokenized, which involves splitting the text into individual words or tokens. This can be done using Python's NLTK library or the built-in string. Split () function
- 3) *Stop Words Removal:* Stop words are common words that do not carry much meaning in the text, such as "the", "and", and "a". Removing stop words can improve the accuracy of the machine learning model by reducing the noise in the data. This can be done using Python's NLTK library or other NLP libraries.
- 4) *Stemming/Lemmatization:* Stemming and lemmatization are techniques used to reduce words to their root forms. This can be done using Python's NLTK library or other NLP libraries.
- 5) *Feature Extraction:* After preprocessing the text, the next step is to extract features from the data. This can include features such as word frequency, sentiment analysis, lexical diversity, and topic modeling. This can be done using Python's scikit-learn library or other NLP libraries.
- 6) *Data Splitting:* Finally, the pre-processed data needs to be split into training and testing sets for machine learning. This can be done using Python's scikit-learn library or other machine learning libraries

IV. MACHINE LEARNING ALGORITHMS

A. Logistic Regression

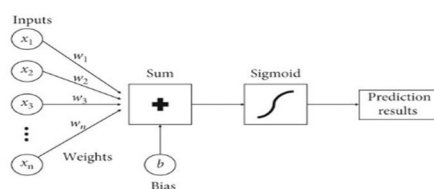
Logistic regression is a statistical algorithm used for binary classification problems, which involves predicting one of two possible outcomes. It is a supervised learning algorithm that is commonly used for predicting the probability of an event occurring, based on one or more input variables.

1) Mathematical Analysis

The logistic regression algorithm uses a logistic function to model the probability of the binary outcome. The logistic function is a sigmoid function that maps any value to a range of 0 to 1. The equation for logistic regression can be written as: $P(y = 1|X) = 1 / (1 + e^{(-z)})$

Where $P(y = 1|X)$ is the probability of the binary outcome ($y = 1$) given the input variables (X), e is the base of the natural logarithm, and z is a linear combination of the input variables and their corresponding weights.

Figure 2 : Flow Chart of Logistic Regression



2) Steps Included

- Step1: Data preparation: Obtain a labelled data set with a binary dependent variable and one or more independent variables
- Step2: Model formulation: Formulate a logistic regression model by specifying the relationship between the dependent variable and independent variables as a logistic function.
- Step3: Parameter estimation: Estimate the parameters of the logistic regression model using maximum likelihood estimation.
- Step4: Model evaluation: Evaluate the performance of the logistic regression model using metrics such as accuracy, precision, recall, and F1 score.
- Step5: Model improvement: Refine the logistic regression model by adding or removing independent variables, transforming variables, or using a different model formulation to improve its performance

3) Merits

- Logistic regression is a simple and interpretable algorithm, making it easy to understand and explain to non-technical stakeholders.
- It is a powerful algorithm for binary classification problems, with high accuracy and good performance on large datasets
- Logistic regression can handle both continuous and categorical input variables, making it versatile for a wide range of applications.

4) Demerits

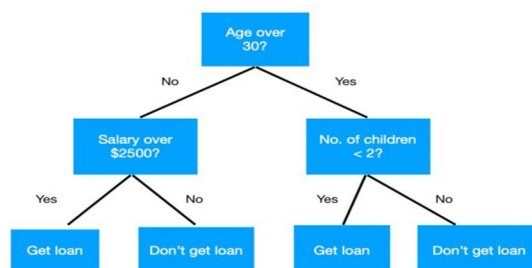
- Logistic regression assumes a linear relationship between the input variables and the output, which may not always be the case in real-world scenarios.
- It is a parametric algorithm, which means it requires a specific functional form for the model, and may not work well on data that does not conform
- Logistic regression may suffer from overfitting, especially when the number of input variables is large compared to the size of the training dataset.

B. Decision Tree

Decision tree is a popular machine learning algorithm used for classification and regression tasks. It is a supervised learning algorithm that creates a tree-like model of decisions and their possible consequences, allowing for predictions based on the input variables.

In the following example, we've to approve a loan on the basis of the age, salary, and no. of children the person has. We ask a conditional question at each node and make splits accordingly, till we reach a decision at the leaf node (i.e., get loan/don't get loan).

Figure 3: Example for Decision tree



1) Mathematical Analysis

The decision tree algorithm works by recursively partitioning the data into subsets based on the values of the input variables, using a tree-like structure to represent the decision-making process.

At each node of the tree, a decision is made based on the values of one of the input variables, with each possible outcome leading to a new branch in the tree.

The algorithm continues to split the data into smaller and smaller subsets until a stopping criterion is met.

2) Steps Included

Step1: Data preparation: Obtain a labelled data set with a categorical or continuous dependent variable and one or more independent variables.

Step2: Tree formulation: Formulate a decision tree by recursively splitting the data into subsets based on the independent variables.

Step3: Split selection: Select the best split for each node by maximizing a split criterion, such as information gain or Gini index.

Step4: Model evaluation: Evaluate the performance of the decision tree by comparing the predictions to the actual values of the dependent variable

3) Merits

- Decision trees are easy to interpret and explain, as the resulting tree structure can be visualized and easily understood by non-technical stakeholders.
- Decision trees can handle both continuous and categorical input variables, making it versatile for a wide range of applications.
- Decision trees are capable of capturing complex nonlinear relationships between the input variables and the output, making it a powerful algorithm for data with complex patterns.

4) Demerits

- Decision trees can be prone to overfitting, especially when the number of input variables is large compared to the size of the training dataset.
- Decision trees are sensitive to the choice of hyperparameters, which can significantly affect the performance of the model.
- Decision trees are not robust to small changes in the data, as the resulting tree structure can be highly sensitive to the exact values and order of the input variables.

C. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification.

KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is close to the test data.

The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and class holds the highest probability will be selected. In the case of regression, the value is the mean of the 'K' selected training points.

Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure.'

Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.

Figure 4: KNN Algorithm Plot

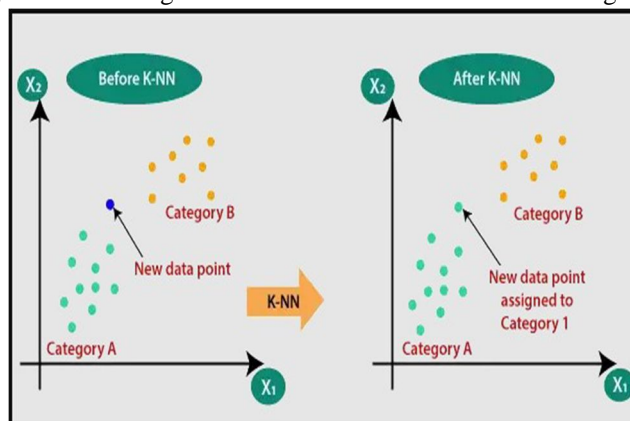
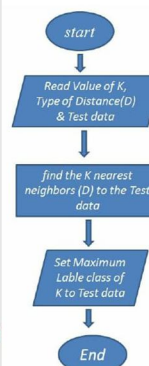


Figure 5 : KNN Flowchart



1) Mathematical Analysis

The first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are — Euclidian, Manhattan (for continuous) and Hamming distance (for categorical)

Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

Manhattan Distance: This is the distance between real vectors using the sum of their absolute difference.

Distance functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $

2) Steps Included

The K-NN working can be explained on the basis of the below algorithm:

- a) Step-1: Select the number K of the neighbors
- b) Step-2: Calculate the Euclidean distance of K number of neighbors
- c) Step-3: Take K nearest neighbors as per calculated Euclidean distance.
- d) Step-4: Among these k neighbors, count the number of the data points in each category.
- e) Step-5: Assign the new data points to that category for which the number of the neighbors is maximum.

3) Merits:

Simple to implement: KNN is a simple algorithm that requires little data pre-processing and can be implemented with a few lines of code.

Fast for small data sets: KNN is fast for small data sets, as the algorithm only needs to store the training data and calculate distances during prediction.

Can handle multi-class problems: KNN can be used for both binary and multi-class classification problems.

4) Demerits

Slow for large data sets: KNN can be slow for large data sets, as the prediction time increases with the size of the training data set.

Can be affected by noisy data: KNN can be affected by noisy data, as a small number of outliers can have a large impact on the prediction.

D. Machine Learning Algorithm Implementation

The following image represents the data preprocessing steps that were performed for the fake news detection project. These steps were necessary to transform the raw data into a format that could be used for machine learning. The preprocessing steps included data cleaning, tokenization, stop word removal, stemming, lemmatise.

Figure 6 Data Pre-processing code

```
In [14]: def clean_text(text):
         remove = ''
         for w in text:
             if w not in string.punctuation:
                 remove = remove + w
             else:
                 remove = remove + ' '
         return remove

In [38]: def preprocess(text):
         return ' '.join([word for word in word_tokenize(text) if word not in stopwords.words('english') and not word.isdigit() and word not in string.punctuation])

In [39]: stemmer = PorterStemmer()
         def stem_words(text):
             return ' '.join([stemmer.stem(word) for word in text.split()])
         df['text_'] = df['text_'].apply(lambda x: stem_words(x))

In [41]: lemmatizer = WordNetLemmatizer()
         def lemmatize_words(text):
             return ' '.join([lemmatizer.lemmatize(word) for word in text.split()])
         df['text_'] = df['text_'].apply(lambda text: lemmatize_words(text))
```

The following image represents two text representation techniques - TF-IDF and BoW - which were used in the fake news detection project. These techniques were used to transform the raw text data into numerical features that could be used for machine learning

Figure 7 : TF-IDF, Count Vectorizer Code

```
In [22]: tfidf_transformer = TfidfTransformer().fit(bow_reviews)
         tfidf_rev4 = tfidf_transformer.transform(bow_msg4)
         print(bow_msg4)

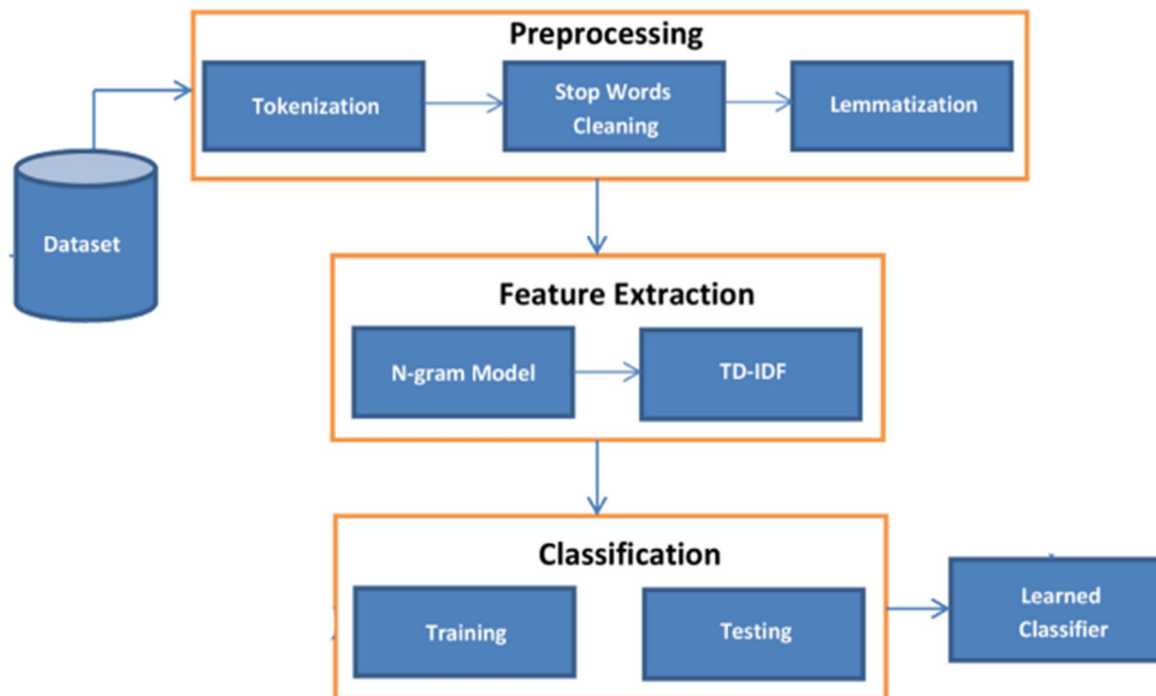
In [14]: bow_transformer = CountVectorizer(analyzer=text_process)
         bow_transformer
```

The following images represent the implementation of the three machine learning algorithms - Logistic Regression, Decision Tree, and KNN

V. SYSTEM ARCHITECTURE AND WORKFLOW

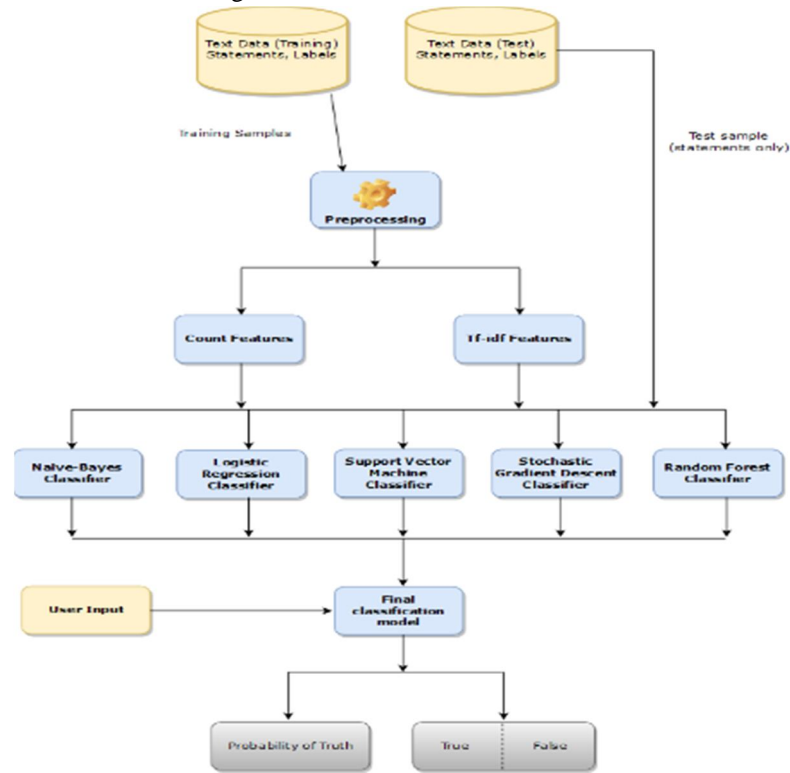
A. Block Diagram Description

Figure 8: Block Diagram of Fake News Detection



B. Flowchart

Figure 9 : Flow chart of fake news detection



VI. RESULTS

K Nearest Neighbor Algorithm Results

```

In [33]: pipeline = Pipeline([
        ('vect', CountVecorizer(analyzer=text_preprocess)),
        ('tfidf', TfidfTransformer()),
        ('classifier', KNeighborsClassifier(n_neighbors=2))
    ])

In [34]: pipeline.fit(review_train, label_train)

Out[34]: Pipeline
  • CountVecorizer
  • TfidfTransformer
  • KNeighborsClassifier

In [35]: knn_pred = pipeline.predict(review_test)
knn_pred
Out[35]: array(['CG', 'CG', 'CG', ..., 'CG', 'CG', 'CG'], dtype=object)

In [36]: pipeline.predict(['"Mamendra Modi is the PM of India"'])
Out[36]: array(['CR'], dtype=object)

In [37]: print('Classification Report:', classification_report(label_test, knn_pred))
print('Confusion Matrix:', confusion_matrix(label_test, knn_pred))
print('Accuracy Score:', accuracy_score(label_test, knn_pred))
print('Model Prediction Accuracy:', str(np.round(accuracy_score(label_test, knn_pred)*100, 2)) + '%')

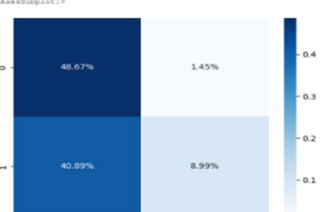
Classification Report:
              precision    recall  f1-score   support

    CG      0.54      0.97      0.70      1092
    CR      0.86      0.18      0.30      7059
 accuracy
macro avg      0.70      0.58      0.50      14151
weighted avg      0.70      0.58      0.50      14151

Confusion Matrix: [[4887 205]
 [1787 1272]]
Accuracy Score: 0.5745670270452225
Model Prediction Accuracy: 57.45%

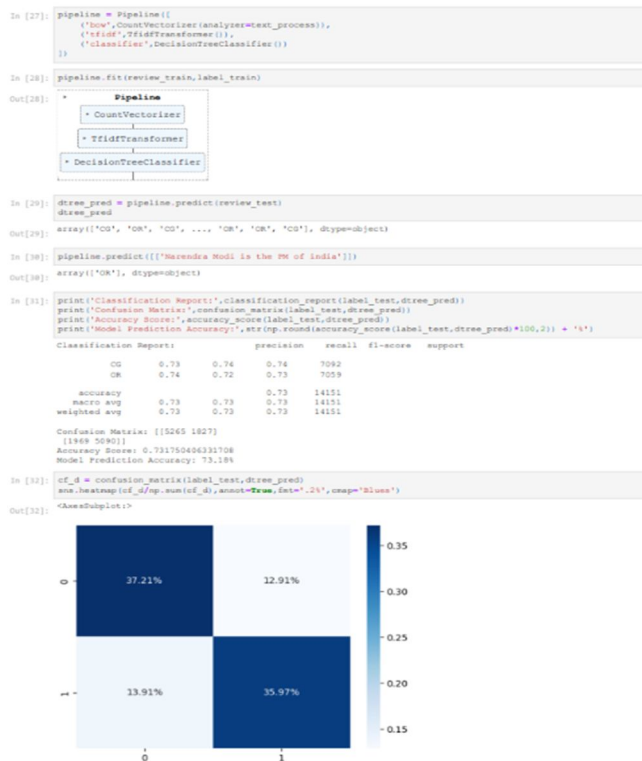
In [38]: cf_k = confusion_matrix(label_test, knn_pred)
sns.heatmap(cf_k, annot=True, fmt='.2%', cmap='Blues')

Out[38]: <AxesSubplot>
  
```

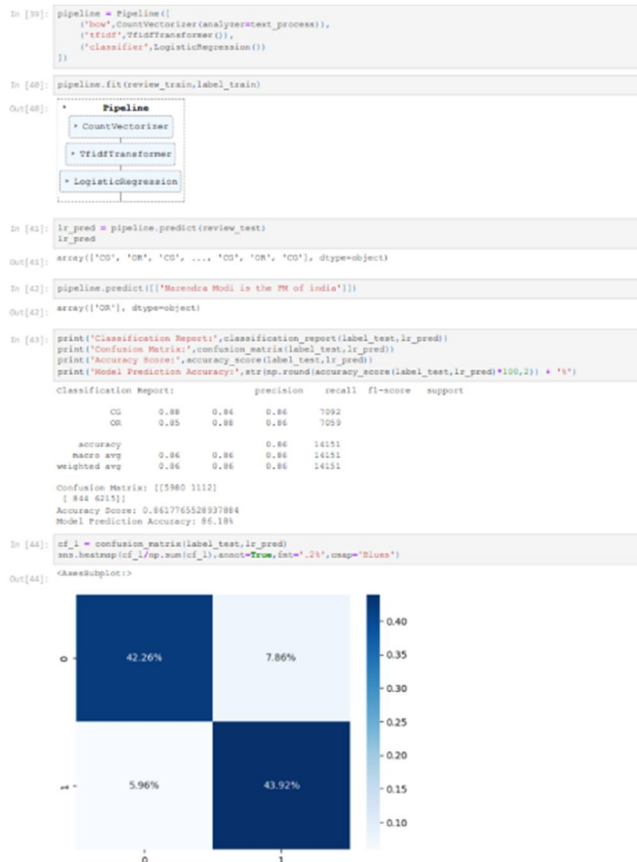


	Actual CG (0)	Actual CR (1)
Predicted CG (0)	48.67%	1.45%
Predicted CR (1)	40.99%	8.99%

Decision Tree Algorithm Results



Logistic Regression Algorithm Results



From the above figures , which are the visualisation of confusion matrix using the density of data set that has been classified into 4 sections representing the 4 possible outputs the model , TRUE_TRUE , TRUE-FALSE , FALSE-FALSE , FALSE-TRUE . The dataset has been trained and tested and the accuracy of the models and the performance is clearly visualised .

Decision Tree Confusion Matrix shows that the model has achieved an accuracy of 74.27%

KNN Confusion Matrix shows that the model has achieved an accuracy of 57.83%

Logistic Regression Confusion Matrix shows that the model has achieved an accuracy of 86.39 %

S.NO		LOGISTIC REGRESSION ALGORITHM	DECISION TREE ALGORITHM	K NEAREST NEIGHBOUR ALGORITHM
1	ACCURACY	86.39%	74.27 %	57.83%
2	CONFUSION MATRIX	[[5979 1038] [887 6247]]	[[5339 1678] [1963 5171]]	[[6830 187] [5700 1354]]

Table 1: Results of Fake News Detection

VII. CONCLUSION & FUTURE SCOPE

A. Conclusion

- 1) In conclusion, the use of machine learning for fake news detection has proven to be a promising approach in addressing the issue of misinformation.
- 2) Through the application of various techniques such as natural language processing, feature engineering, and classification algorithms, we are developing a model that can effectively distinguish between fake and real news articles.
- 3) Accuracy of the model can be improved with the use of more advanced algorithms and larger datasets; our results indicate that machine learning can play a valuable role in combating the spread of fake news.
- 4) It is important to continue refining and validating these models to help promote the dissemination of accurate information and combat the negative effects of misinformation on society.

B. Future Scope

Improving the accuracy of the model by using more advanced algorithms, such as deep learning techniques, to further enhance the ability to differentiate between fake and real news articles.

- 1) Expanding the scope of the project to include detection of more types of misinformation, such as propaganda, disinformation, and conspiracy theories.
- 2) Developing a real-time system that can analyze news articles as they are published to quickly identify and flag potentially fake news stories.
- 3) Integrating the fake news detection model with existing social media platforms to automatically flag potentially fake news stories, helping to prevent the spread of misinformation

REFERENCES

Base Paper: Survey on fake news detection, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org ICACT – 2021

- [1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3 Sep 2017
- [2] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3 Sep 2017
- [3] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903.
- [4] Fake news websites. (n.d.) Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Fake_news_website. Accessed Feb. 6, 2017
- [5] Cade Metz. (2016, Dec. 16). The bittersweet sweepstakes to build an AI that destroys fake news



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)