FAKE NEWS DETECTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

RISHI RAGHAV G - 21ADR038

ROHITH S J - 21ADR040

VIBEESH N - 21ADR059

VIGNESH T - 21ADR060

VIJAYAKUMARAN S – 21ADR061

Team mentor:

Ms. D. Sathya

For

20ADT44 APPLIED MACHINE LEARNING

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

KONGU ENGINEERING COLLEGE
(Autonomous)

PERUNDURAI ERODE – 638 060

MAY 2023

ABSTRACT

In recent years, the rapid dissemination of false information through various online platforms has become a major concern, posing a significant threat to societal well-being and public discourse. The rise of fake news necessitates the development of effective automated systems for detection and mitigation. This project aims to address the challenge of fake news detection by leveraging machine learning techniques.

The proposed approach employs a comprehensive methodology that combines natural language processing (NLP) techniques, feature engineering, and machine learning algorithms. The project utilizes a large dataset consisting of labeled news articles, encompassing both legitimate and fake news samples. Preprocessing techniques such as tokenization, stop-word removal, and stemming are applied to convert the textual content into a suitable format for analysis.

The expected outcomes of this project include the development of a reliable and efficient fake news detection system capable of accurately identifying and classifying news articles as either legitimate or fake. Such a system would contribute to the ongoing efforts in combating the spread of misinformation, promoting media literacy, and safeguarding public trust in news sources.

INTRODUCTION

One of the consequences of technology is fake news. It is misinformation or misleading information offered as facts that can affect a person's opinion. This false information has several goals; organizations can use it for financial purposes (e.g., Facebook pages used it to spread fake news leading to specific ads) or for political purposes. Compared with Google, Twitter, and webmail such as Yahoo and Gmail, Facebook is the worst media platform for pervasive fake news.

The definition of fake news is the information that pushes people to the wrong road. Fake news is spreading like a wildfire these days, and people are sharing it without confirming it. Fake news can be intimidating as they attract more audience than normal.People use them because this can be a very good marketing strategy. This fake information triggers fear and panic among people. Therefore, there is a need for ways to fact check news.

In this research, the best model to find whether the news is Real or Fake. For the first stage the dataset named fake_or_real_news is taken from the Kaggle. Then the dataset is preprocessed with proper preprocessing technique using Count vectorizer, TF-idf vectorizer, then the various models are created with several classification algorithms like Decision tree, Random forest, K-Nearest neighbor, Logistic Regression, Naïve Bayes, Support Vector machine. The best model is found through analysing the different metrices of those algorithms.

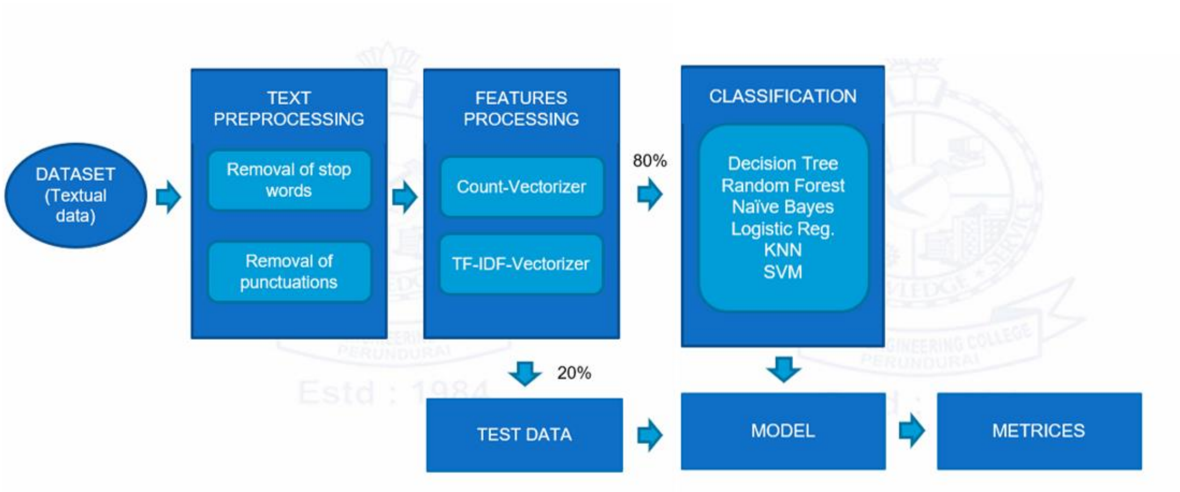
PROBLEM STATEMENT

- Now a days it is extremely difficult to decide whether the news we come across is real or not.
- This fake news had left an indelible mark on people and culture.
- Fake news play a major in election time.
- So to find whether the given information or news true or false using machine learning algorithm.
- The objective of this project is to examine the problems and reduce the spread of fake news.
- The fake news is a problem that is heavily affecting society and our perception of not only the media but also facts and opinions themselves.

- By using the artificial intelligence and the machine learning, the problem can be solved .
- So, our focus is to find which machine learning algorithm is best suitable for text dataset by comparing the performance of the models.

PROPOSED SYSTEM

The suggested framework is shown below. The publicly available dataset is used to train the system. To begin, clean up the data by deleting any superfluous characters or digits. After that, the Countvectorizer and TF-IDF vectorizer are utilized to extract features from the dataset. Then this cleaned dataset is used in different algorithms. Finally, the different metrices of classification algorithms are compared and the best model is found.



FEATURE EXTRACTION :

Text classification:

we are using TF-ID (Term Frequency & Inverse Document Frequency) is a powerful feature engineering technique used to identify the important words or more precisely rare words in the text data.

ALGORITHMS USED

The following algorithms were used :

- ✓ Random Forest
- ✓ Decision Tree
- ✓ Support Vector Machine (SVM)
- ✓ Kth Nearest Neighbour (KNN)
- ✓ Logistic Regression.
- ✓ Naïve Bayes

Decision Tree classifier:

A decision tree is a supervised machine learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

```
Confusion matrix :
 [[518 120]
 [107 522]]
Accuracy : 82.08%
FOR FAKE
Precision for "FAKE" : 82.88%
Recall for "FAKE" : 81.19%
f1_score for "FAKE" : 82.03%
FOR REAL
Precision for "REAL" : 81.31%
Recall for "REAL" : 82.99%
f1_score for "REAL" : 82.14%
```

K-Nearest Neighbors:

The k-nearest neighbors algorithm, also known as KNN , is a supervised machine learning classifier, which used to make classifications or predictions.

```
Confusion matrix :
 [[554  84]
 [ 66 563]]
Accuracy : 88.16%
FOR FAKE
Precision for "FAKE" : 89.35%
Recall for "FAKE" : 86.83%
f1_score for "FAKE" : 88.08%
FOR REAL
Precision for "REAL" : 87.02%
Recall for "REAL" : 89.51%
f1_score for "REAL" : 88.24%
```

Logistic Regression:

It is a supervised machine learning method.Logistic regression is used to calculate or predict the probability of a binary (yes/no) event occurring.

```
Confusion matrix :
 [[603  35]
 [ 64 565]]
Accuracy : 92.19%
FOR FAKE
Precision for "FAKE" : 90.4%
Recall for "FAKE" : 94.51%
f1_score for "FAKE" : 92.41%
FOR REAL
Precision for "REAL" : 94.17%
Recall for "REAL" : 89.83%
f1_score for "REAL" : 91.94%
```

Naïve Bayes classifier:

The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification.

```
Confusion matrix :
 [[563  75]
 [ 97 532]]
Accuracy : 86.42%
FOR FAKE
Precision for "FAKE" : 85.3%
Recall for "FAKE" : 88.24%
f1_score for "FAKE" : 86.75%
FOR REAL
Precision for "REAL" : 87.64%
Recall for "REAL" : 84.58%
f1_score for "REAL" : 86.08%
```

Random Forest:

Random Forest Algorithm is a supervised machine learning algorithm is used for Classification and Regression problems in Machine Learning.

```
Confusion matrix :
 [[578  60]
 [ 55 574]]
Accuracy : 90.92%
FOR FAKE
Precision for "FAKE" : 91.31%
Recall for "FAKE" : 90.6%
f1_score for "FAKE" : 90.95%
FOR REAL
Precision for "REAL" : 90.54%
Recall for "REAL" : 91.26%
f1_score for "REAL" : 90.89%
```

Support Vector Machine:

SVM is a supervised machine learning algorithm that works best on smaller datasets . Support Vector Machine, can be used for both regression and classification .But is best for classification.

```
Confusion matrix :
 [[614  24]
 [ 53 576]]
Accuracy : 93.92%
FOR FAKE
Precision for "FAKE" : 92.05%
Recall for "FAKE" : 96.24%
f1_score for "FAKE" : 94.1%
FOR REAL
Precision for "REAL" : 96.0%
Recall for "REAL" : 91.57%
f1_score for "REAL" : 93.73%
```

Gradient Boosting Classifier:

Gradient Boosting is a functional gradient algorithm that repeatedly selects a function that leads in the direction of a weak hypothesis or negative gradient so that it can minimize a loss function. Gradient boosting classifier combines several weak learning models to produce a powerful predicting model.

```
Confusion matrix :
 [[57  3]
 [13 47]]
Accuracy : 86.67%
FOR REAL
Precision for "REAL" : 94.0%
Recall for "REAL" : 78.33%
f1_score for "REAL" : 85.45%
FOR FAKE
Precision for "FAKE" : 81.43%
Recall for "FAKE" : 95.0%
f1_score for "FAKE" : 87.69%
```

XGBoost Classifier:

XGBoost classifier is a Machine learning algorithm that is applied for structured and tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

```
Confusion matrix :
 [[587  51]
 [ 46 583]]
Accuracy : 92.34%
FOR REAL
Precision for "REAL" : 91.96%
Recall for "REAL" : 92.69%
f1_score for "REAL" : 92.32%
FOR FAKE
Precision for "FAKE" : 92.73%
Recall for "FAKE" : 92.01%
f1_score for "FAKE" : 92.37%
```

AdaBoost Classifier:

AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

```
Confusion matrix :
 [[576  62]
 [ 35 594]]
Accuracy : 92.34%
FOR REAL
Precision for "REAL" : 90.55%
Recall for "REAL" : 94.44%
f1_score for "REAL" : 92.45%
FOR REAL
Precision for "REAL" : 90.55%
Recall for "REAL" : 94.44%
f1_score for "REAL" : 92.45%
```

Voting Classifier:

Voting Classifier is a machine-learning algorithm often used by Kagglers to boost the performance of their model and climb up the rank ladder. Voting Classifier can also be used for real-world datasets to improve performance, but it comes with some limitations.

```
Confusion matrix :
 [[55  5]
 [11 49]]
Accuracy : 86.67%
FOR REAL
Precision for "REAL" : 90.74%
Recall for "REAL" : 81.67%
f1_score for "REAL" : 85.96%
FOR FAKE
Precision for "FAKE" : 83.33%
Recall for "FAKE" : 91.67%
f1_score for "FAKE" : 87.3%
```

Performance Metrics:

Performance metrics are used to evaluate the performance of a classification model. We measured performance Accuracy, precision, recall and f1-score.

Accuracy:

The Accuracy is how we are relative to the right value.

Accuracy = number of correct predictions/total number of predictions

Accuracy equation is given by,

Accuracy = TP + TN /TP + TN + FP + FN

Precision:

Precision is how close the measurements are, which is given by

Precision = TP /(TP + FP)

Recall:

Recall (Sensitivity) is how many correctly actual positive is defined, is given by

Recall/Sensitivity/TPR = TP/ (TP + FN)

F1-Score:

The f1-score definition is that if the cost of false positives and false negatives varies, we require precision and recall, as given by

F1-Score = 2 × (Sensitivity × Precision) (Sensitivity + Precision)

where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives.

COMPARISON :

CountVectorizer							
Methods	Accuracy	Precision		Recall		F1 score	
		Fake	Real	Fake	Real	Fake	Real
Decision Tree classifier	81.45%	81.03%	80.8%	80.9%	82.24%	81.45%	81.45%
k-Nearest Neighbors	80.33%	78.27%	82.64%	83.7%	76.7%	80.72%	79.62%
Logistic Regression	91.55%	91.81%	91.0%	91.38%	91.57%	91.59%	91.28%
Naïve Bayes classifier	82.56%	88.12%	78.33%	75.55%	89.67%	81.35%	83.62%
Random Forest	90.61%	91.0%	90.22%	90.28%	90.94%	90.64%	90.58%
Support Vector Machine	87.37%	82.38%	94.33%	95.3%	79.33%	88.37%	86.18%

CountVectorizer (title+text)							
Methods	Accuracy	Precision		Recall		F1 score	
		Fake	Real	Fake	Real	Fake	Real
Decision Tree classifier	80%	81.03%	79.22%	79.0%	81.24%	80.0%	80.22%
k-Nearest Neighbors	81.22%	79.94%	82.64%	83.7%	78.7%	81.78%	80.62%
Logistic Regression	91.32%	91.64%	91.0%	91.07%	91.57%	91.35%	91.28%
Naïve Bayes classifier	83.03%	88.11%	79.07%	76.65%	89.51%	81.98%	83.97%
Random Forest	90.69%	90.62%	90.75%	90.91%	90.46%	90.77%	90.61%
Support Vector Machine	88.0%	83.2%	94.58%	95.45%	80.45%	88.91%	86.94%

TF-IDF Vectorizer							
Methods	Accuracy	Precision		Recall		F1 score	
		Fake	Real	Fake	Real	Fake	Real
Decision Tree classifier	81.14%	81.72%	80.56%	80.56%	81.72%	81.54%	81.14%
k-Nearest Neighbors	56.59%	53.72%	96.47%	99.53%	13.04%	69.78%	22.97%
Logistic Regression	92.42%	90.57%	94.49%	94.83%	89.98%	92.65%	92.18%
Naïve Bayes classifier	86.35%	86.3%	86.31%	86.52%	86.17%	86.75%	86.24%
Random Forest	91.4%	91.92%	90.88%	90.91%	91.89%	91.41%	91.38%
Support Vector Machine	93.48%	92.04%	95.84%	96.08%	91.57%	94.02%	93.66%

TF-IDF Vectorizer(title+text)							
Methods	Accuracy	Precision		Recall		F1 score	
		Fake	Real	Fake	Real	Fake	Real
Decision Tree classifier	81.37%	82.63%	80.18%	79.78%	82.99%	81.18%	81.56%
k-Nearest Neighbors	88.16%	89.35%	87.02%	86.83%	89.51%	88.08%	88.24%
Logistic Regression	92.19%	90.4%	94.17%	94.51%	89.83%	92.41%	91.94%%
Naïve Bayes classifier	86.42%	85.3%	87.64%	88.24%	84.58%	86.75%	86.08%
Random Forest	91.4%	92.32%	90.5%	90.44%	92.37%	91.37%	91.42%
Support Vector Machine	94.32%	92.05%	96.01%	96.24%	91.57%	94.1%	93.73%
Gradient Boosting classifier	86.67%	81.43%	94.0%	95.0%	78.33%	87.69%	85.45%
Xgboost classifier	92.34%	92.73%	91.96%	92.01%	92.69%	92.37%	92.32%
Adaboost Classifier	92.34%	94.27%	90.55%	90.28%	94.44%	92.23%	92.45%
Voting Classifier	86.67%	83.33%	90.74%	91.67%	81.67%	87.3%	85.96%

CONCLUSION :

As the result, the Support Vector Machine has the highest accuracy of 94.32% by using the TF-IDF vectorizer in which the input feature has both the title and the text content of the specific new. So we are selecting the SVM(Support Vector Machine) as our base model. Using this model we can predict the fake news with maximum Accuracy.

REFERENCE :

1. https://towardsdatascience.com/predicting-fake-news-using-nlp-and-machine-learning-scikit-learn-glove-keras-lstm-7bbd557c3443
2. https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/b.e-cse-batchno-214.pdf
3. https://www.hindawi.com/journals/complexity/2020/8885861/
4. D. Keskar, S. Palwe, and A. Gupta, Fake News Classification on Twitter Using Flume, N-Gram Analysis, and Decision Tree Machine Learning Technique. Singapore: Springer, 2020, pp. 139–147, doi: 10.1007/978-981-15-0790-8_15.
5. A. Bali, M. Fernandes, S. Choubey, and M. Goel, ''Comparative performance of machine learning algorithms for fake news detection,'' 2019, doi: 10.1007/978-981-13-9942-8_40.
6. K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, ''FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media,'' Big Data, vol. 8, no. 3, pp. 171–188, Jun. 2020, doi: 10.1089/big.2020.0062.
7. L. Wu and H. Liu, ''Tracing fake-news footprints: Characterizing social media messages by how they propagate,'' in Proc. 11th ACM Int. Conf. Web Search Data Mining, Feb. 2018, pp. 637–645, doi: 10.1145/3159652.3159677.
8. J. Zhang, B. Dong, and P. S. Yu, ''FAKEDETECTOR: Effective fake news detection with deep diffusive neural network,'' 2018, arXiv:1805.08751.
9. S. Gilda, ''Notice of violation of IEEE publication principles: Evaluating machine learning algorithms for fake news detection,'' in Proc. IEEE 15th Student Conf. Res. Develop. (SCOReD), Dec. 2017, pp. 110–115, doi: 10.1109/SCORED.2017.8305411.
10. H. Ko, J. Y. Hong, S. Kim, L. Mesicek, and I. S. Na, ''Human-machine interaction: A case study on fake news detection using a backtracking based on a cognitive system,'' Cognit. Syst. Res., vol. 55, pp. 77–81, Jun. 2019, doi: 10.1016/j.cogsys.2018.12.018.

11. C.-S. Atodiresei, A. Tănăselea, and A. Iftene, ''Identifying fake news and fake users on Twitter,'' Proc. Comput. Sci., vol. 126, pp. 451–461, Jan. 2018, doi: 10.1016/j.procS.2018.07.279. [17] J. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, ''Content based fake news detection using knowledge graphs,'' in Proc. 17th Int. Semantic Web Conf., Monterey, CA, USA, Oct. 2018, doi: 10.1007/978-3-030-00671-6_39.

12. M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, ''A stylometric inquiry into hyperpartisan and fake news,'' in Proc. 56th Annu. Meeting Assoc. Comput. Linguist. Proc. Conf. (ACL), vol. 1, 2018, pp. 231–240, doi: 10.18653/v1/p18-1022