A Mini Project Report on

A Study on Fake News Detection using Machine Learning and Deep Learning Techniques

undergone at

Department of Computer science and Engineering, NITK

under the guidance of

Mahendra Pratap Singh, Assistant Professor

Submitted by

Soumyajit Bhattacharyya
192CS025
II Sem M.Tech (CS)

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science & Engineering
National Institute of Technology Karnataka, Surathkal.

June 2020

ABSTRACT

The term fake news has become a buzzword these days. It can be defined as a type of yellow journalism or propaganda that consists of purposeful misinformation or hoaxes spread via traditional print and broadcast news media or online social media . The 2016 US election was a great example of this and this is a popular research material currently. These are published usually with the intent to mislead in order to damage a community or person, create chaos, and gain financially or politically and it has the potential to mould opinions and influence decisions. Since people are often unable to spend enough time to cross-check references and be sure of the credibility of news, automated detection of fake news is indispensable. However, statistical approaches to combating fake news were dramatically limited by the lack of labeled benchmark datasets a few days ago. But after the evolution of the benchmark dataset LIAR , several datasets are formed and it is receiving great attention from the research community. This project aims to use various NLP techniques combined with several machine learning and deep learning techniques to help achieve maximum accuracy in detecting authenticity of news from a popular fake news dataset.

TABLE OF CONTENTS

1. INTRODUCTION

Nowadays social-networking systems, online news portals, and other online media have become the main sources of news through which interesting and breaking news are shared at a rapid pace. However, many news portals serve special interest by feeding with distorted, partially correct, and sometimes imaginary news that is likely to attract the attention of a target group of people. The term to describe this type of news is called fake news and it can be defined as a type of yellow journalism or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcast news media or online social media .These are published usually with the intent to mislead in order to damage a community or person, create chaos, and gain financially or politically. Fake news is increasingly becoming a menace to our society. It is typically generated for commercial interests to attract viewers and collect advertising revenue. However, people and groups with potentially malicious agendas have been known to initiate fake news in order to influence events and policies around the world. Since people are often unable to spend enough time to cross-check references and be sure of the credibility of news, quick ,accurate and automated detection of fake news is indispensable. Therefore, it is receiving great attention from the research community.

There are many instances where cleverly designed fake news had severe consequences by instigating religious or ethnic groups against innocent victims. On October 17, 2018, United States Congressman Matt Gaetz (R-FL) posted a video to Twitter and suggested, without evidence, which showed a group of people being paid by billionaire George Soros to join a migrant caravan and storm the United States border. The video was miscaptioned and the tweet contained factual inaccuracies.

On 23 June 2018, a series of horrifying images and videos began to circulate on Facebook. One showed a man's skull hacked open that was viewed more than 11,000 times. The Facebook users who posted the images claimed they showed a massacre underway in the Gashish district of Plateau State, Nigeria by Fulani Muslims who were killing Christians from the regions Berom ethnic minority. As a consequence, a massacre

did happen in Gashish that weekend and somewhere between 86 and 238 Berom people were killed, according to estimates made by the police and by local community leaders. However, some of the most incendiary images and videos were totally irrelevant to the violence in Gashish. The video showing a man's head was cut, was not even happened in Nigeria and it was recorded in Congo, in 2012.

It is also believed that circulation of fake news had a material impact on the outcome of the 2016 US Presidential Election. While Mark Zuckerberg, Facebook's CEO, made a public statement denying that Facebook had an effect on the outcome of the election, Facebook and other online media outlets have begun to develop strategies for identifying fake news and mitigating its spread. Zuckerberg admitted identifying fake news is difficult, writing, "This is an area where I believe we must proceed very carefully though. Identifying the truth is complicated."

So in this project my aim is to classify the news source as fake or real by developing some Machine Learning and Deep learning models , combining Natural Language Processing Techniques with them and comparing their performances w.r.t some well known performance evaluation metrics like accuracy, precision, recall and f1-score.

2. BASIC CONCEPTS

Some of the basic concepts used to develop the models in the project is presented below in a brief.

2.1. Machine Learning

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. As it is evident from the name, it gives the computer that makes it more similar to

humans: The ability to learn. Some of the popular ML algorithms are Linear Regression, SVM, Random forest, Naive Bayes, Adaptive Boosting etc.

2.2. Deep Learning

Deep learning is a specialized machine learning technique, basically a neural network. It operates using a number of levels/layers. The layers are interconnected and the output of the previous layer is fed as input to the next level,so it is called feed-forward network.Each neuron uses a linear or nonlinear activation function to systematize its output. The number of hidden layers indicates how deep a deep learning model is. Epochs are continued with some constraint to get better accuracy.In every epoch in each step,the parameters are updated to achieve better accuracy and smaller loss. The typical activation functions are like ReLU,Sigmoid and Softmax. In deep learning, a large amount of data in comparison with machine learning is needed to build the model which is ideal for big data analytics because large amounts of complex data is not properly handled by machine learning. Some typical examples are CNN, LSTM etc.

2.3. Natural Language Processing

Natural language refers to the way we, humans, communicate with each other. Namely, speech and text. Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable. Some typical examples include tokenization, stemming, Bag of Words etc.

3. DATASET DESCRIPTION

The fake news dataset used for this project was drawn from Kaggle(link). The training dataset has about 20500 rows of data from various articles on the internet. Before training the models, certain preprocessing was carried out.

A full training dataset has the following attributes:

A. id: unique id for a news article
B. title: the title of a news article
C. author: author of the news article
D. text: the text of the article; incomplete in some cases
E. label: a label that marks the article as potentially unreliable
    - 1: unreliable
    - 0: reliable

Nearly 50% of the records are labeled as unreliable or fake. There are 3 files in the dataset which are train.csv, test.csv, submission.csv. The train.csv file contains the records with labels. The test.csv file is for testing purposes , so the records of it do not have any label. The labels of it are stored in submission.csv to compare with the predicted labels by a model. I have used only the training.csv file for my models and split it into 80% for training and 20% for testing purposes.

4. PREPROCESSING AND FEATURE EXTRACTION

After collecting the data, several preprocessing needed to be done before feeding it into a model. Only the 'text' column of the data is used for preprocessing and feature extraction after that. The necessary steps are given below.

- ➢ All the stopwords are removed.
- ➢ All the records containing null or infinity values are removed.
- ➢ All the non-alphanumeric values i.e. special characters, punctuations are removed from the text.
- ➢ All the texts are converted to lowercase.
- ➢ The words in the documents are separated by comma.

After preprocessing the text is converted to a comma-separated list of words, which can be input into the Doc2Vec algorithm to produce an 300-length embedding vector representation for each article. There are a total 16,600 records after preprocessing.

4.1. Doc2Vec

Doc2Vec is a model developed in 2014 based on the existing Word2Vec model, which generates vector representations for words. Word2Vec represents documents by combining the vectors of the individual words, but in doing so it loses all word order information. Doc2Vec expands on Word2Vec by adding a ”document vector” to the output representation, which contains some information about the document as a whole, and allows the model to learn some information about word order. Preservation of word order information makes Doc2Vec useful for my application, as my aim is to detect subtle differences between text documents. While Word2Vec computes a feature vector for every word in the corpus, Doc2Vec computes a feature vector for every document in the corpus. While Word2Vec works on the intuition that the word representation should be good enough to predict the surrounding words, the underlying intuition of Doc2Vec is that the document representation should be good enough to predict the words in the document.

For example, Word2Vec’s learning strategy exploits the idea that the word mat follows the phrase the cat sat on. Doc2Vec’s learning strategy exploits the idea that the prediction of neighboring words for a given word strongly relies on the document also. Though the appearance of the phrase catch the ball is frequent in the corpus, if we know that the topic of a document is about ‘’technology’’, we can expect words such as bug or exception after the word catch (ignoring the) instead of the word ball since catch the bug/exception is more plausible under the topic ‘’technology’’. On the other hand, if the topic of the document is about ‘’sports’’, then we can expect ball after catch.

5. MODELS USED

I have used four classifiers on the after preprocessing and extracting features from i.e after vector representation. The classifiers/algorithms used are Naive Bayes, Support Vector Machine(SVM), Random Forest and a Feed-forward Deep Neural Network.

5.1. Naive Bayes

Naive Bayes classifier is a classifier based on Bayes Theorem. Bayes Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are hypothesis and evidence respectively. P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). P(A|B) is the posterior probability of A , i.e. probability of event after evidence is seen. P(B) is the priori of B and P(B|A) is the likelihood.

Now, with regards a dataset, Bayes Theorem can be applied in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size n) where:

$$X = (x_1, x_2, x_3, ..., x_n)$$

Now after applying Naive Bayes assumption which says all the features are independent and contribute equally to the outcome , we reach to the result :

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, as the denominator remains constant for a given input, that term can be removed and expressed as :

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

Finally, in order to create a classifier model, the probability of a given set of inputs for all possible values of the class variable y is found and the output with maximum probability is picked up as the class label. This can be expressed mathematically as:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

I have used the Gaussian Naive Bayes where the likelihood is calculated as follows :

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

5.2. Support Vector Machine

In machine learning, support-vector machines (SVMs,) are supervised learning models that analyze data used for classification and regression analysis. Given a set of training examples, each belonging to one or the other of two classes , an SVM training algorithm builds a model that assigns new instances to their respective classes. So in general it is a non-probabilistic binary linear classifier. But all the data is not linearly separable. SVMs can also perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high- dimensional feature spaces.

The main task of the support vector machine algorithm is to identify an N-dimensional hyperplane that distinguishably categorizes the data points. Here, N stands for a number of features. Between two classes of data points, there can be multiple possible hyperplanes that can be chosen. The objective of this algorithm is to find a hyperplane which maximizes the margin i.e maximally separates the closest data points on both side.The benefit associated with maximizing the margin is that it provides scope for the future data points to be more easily classified.

Suppose for 2d data the equation of hyperplane is, Y = W.X + b. Then Svm classifies the incoming samples in following way :

If Y - WX - b >= 0 it will be in one class and if Y - WX - b <= 0 it will be in another class.

5.3. Decision Tree and Random Forest

Decision tree is a supervised learning algorithm that is used in classification and regression problems. It is suitable for both categorical and continuous input and output variables. In this technique, we split the total given sample into two or more than two subpopulations based on the most significant feature or attribute in input variables.

- Root Node: It represents the entire population and this further gets divided into two or more sets based on the splitter.
- Decision Node: When a sub-node splits into further sub-nodes based on the splitter , then it is called decision node. The root node is the topmost decision node in the tree.
- Leaf/ Terminal Node: Nodes that do not split further are called Leaf or Terminal nodes.

Learned trees are a series of if-then statements which classify data by looking at the features down the tree from the root node to the leaf and after that assign a label which resides in the leaf node. Each node in a Decision Tree offers some test on a particular attribute of the data and each branch from the node is one of the possible values of that attribute. The splitter attribute which is to be tested is chosen on the basis of Information Gain.The attribute showing maximum information gain is selected as the splitter attribute at that node. There are also measures like the Gini index to split a node into subnodes.

I have used the Random Forest Classifier instead of decision trees. Random forest classifier is also a supervised learning algorithm based on ensemble classifier. It is capable of performing both regression and classification tasks with the use of multiple decision trees .This technique is called Bootstrapping or more commonly bagging. Bagging involves training each decision tree on a different data sample where sampling is done with replacement.. The basic principle of random forest classifier is to combine

multiple random subset decision trees and produce a final class or result based on the votes of the random subset of decision trees. In Random Forest, we grow multiple trees as opposed to a single tree in the CART model to determine the final output rather than relying on individual decision trees . To classify a new instance based on attributes, each tree gives a classification . The forest chooses the class based on the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees in the random forest .The parameters included in the random forest classifier are n_estimators which is total number of decision trees, and max_depth which refers to the maximum depth of the tree. Another hyper parameter is like max_features which includes the number of features for best-split.

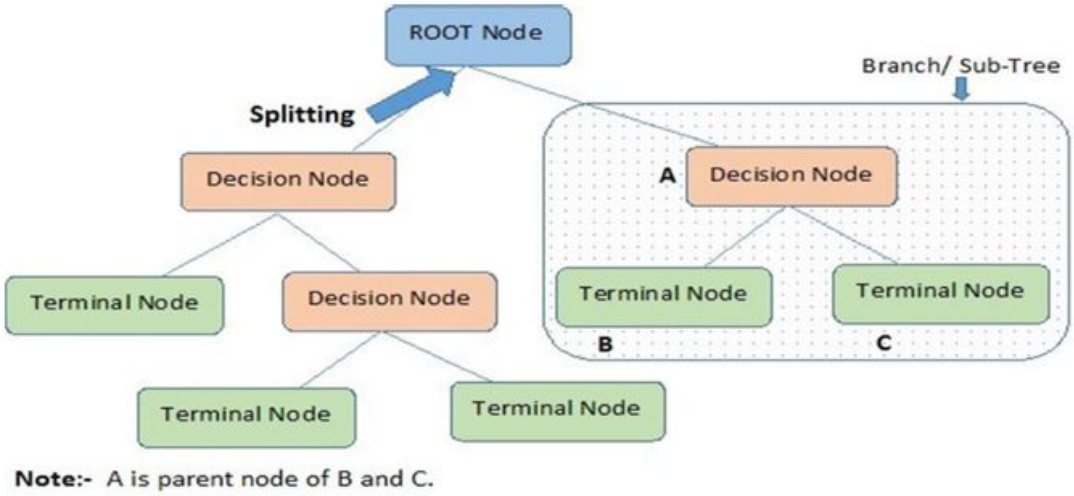

Fig 1 - *Decision Tree*

5.4. Deep Feed-forward Neural Network

The deep neural network used here is made up of 3 hidden layers between 1 input and 1 output layer which are fully connected with each other. ReLU function is used in input and hidden layers. Sigmoid function is used in the output layer as it is binary classification. Input layer contains 300 neurons as Doc2Vec converts the preprocessed

data into a 300 vector embedding. As it is binary classification the output layer contains 1 neuron. The hidden layers contain 128,64 and 32 neurons respectively from first to third. The neuron count in each hidden layer decreases by half to ensure accurate output and reduce cost. To prevent overfitting and to make the model more robust regularization technique is used. Between every two fully connected layers dropout of 0.1 and between output layer and last hidden layer dropout of 0.5 is used for the purpose of speeding up. Dropout removes neurons with their connection in a random manner to prevent overfitting and to speed up and reduce cost. Binary cross entropy function is used as loss function. The Adam optimizer is used with default learning rate 0.001. Number of epochs is set as 25 because after that accuracy is more or less stable.The batch size is set as 64.

Backpropagation algorithm is used where based on the loss in the previous epoch the weights of the connections and biases of the neurons are modified based on some rule. The algorithm works by calculating the gradient of the loss function w.r.t every weight using chain rule one layer at a time.The algorithm propagates backward from the last layer to avoid unnecessary calculations and is an example of dynamic programming.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               38528
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 48,897
Trainable params: 48,897
Non-trainable params: 0
_________________________________________________________________
```

Fig 2 - *Model Summary*

6. EXPERIMENTAL DETAILS AND RESULTS

The entire code is hosted using Google Colaboratory and is written using Python language. The dataset from Kaggle is stored in Google Drive and used after mounting the drive. At first the preprocessing is done on the dataset. Then the documents are converted into labeled sentences i.e each word having a token. After that Doc2Vec model is applied to convert each document into 300-length feature vector embedding. After that the data is split into train and test data as 80% and 20% respectively. Then Naive Bayes, SVM, Random Forest and Deep Feed-forward Neural Network classifiers are applied on the data and their performances are evaluated using performance metrics like accuracy, precision, recall and f1-score.

A confusion matrix is a representation of predicted results for a classification problem by providing the prediction of the samples separated as shown in Table-1. Suppose class 0 is positive and class 1 is negative.

	Class 0(Predicted)	Class 1(Predicted)
Class 0(Actual)	TP	FN
Class 1(Actual)	FP	TN

Table 1 - *Confusion Matrix*

- True Positive (TP) : Actually the sample is positive, and prediction is also positive.
- False Negative (FN) : Actually the sample is positive, but prediction is negative.
- True Negative (TN) : Actually the sample is negative, and prediction is also negative.
- False Positive (FP) : Actually the sample is negative, but prediction is positive.

In terms of TP, FP, TN, FN the accuracy, f1-score, recall and precision is defined as :

- Accuracy = (TP + TN) / (TP + FP + TN + FN) ; Implies the fraction of correctly classified samples.
- Precision = (TP) / (TP + FP) ; Implies the probability that a sample classified as positive is indeed positive.
- Recall = (TP) / (TP + FN) ; Implies the probability that the class is correctly recognized.
- F1-Score = 2 * Recall * Precision / (Recall + Precision) ; Which uses harmonic mean instead of arithmetic mean.

	Accuracy	Precision	Recall	F1-score
Naive Bayes	72.14%	68.08%	85.95%	75.98%
SVM	91.55%	90.72%	92.83%	91.77%
Random Forest	88.35%	89.68%	87.04%	88.34%
Deep Feed-forward Neural Network	**92.29%**	**91.97%**	**92.93%**	**92.45%**

Table 2 - *Performance comparison of the classifiers*

The results and comparisons of the classifiers in terms of different evaluation metrics is given in table 2. It can be seen from the table that the Deep Feed-forward Neural Network gives best performance in terms of all metrics among the four. Naive Bayes classifier gives the worst performance among the four. The snapshots of the confusion matrix along with the performance metrics are given in Fig 3, Fig 4, Fig 5 and Fig 6. The code can be found here.

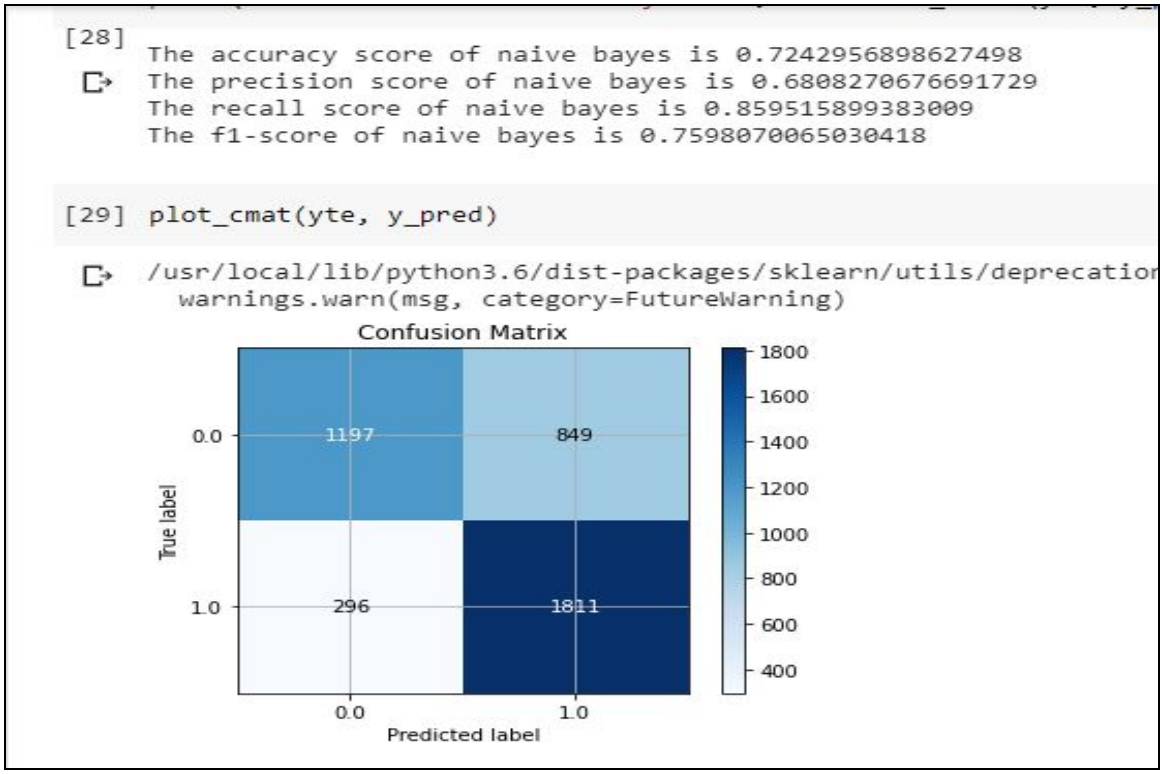

Fig 3 - *Naive Bayes*

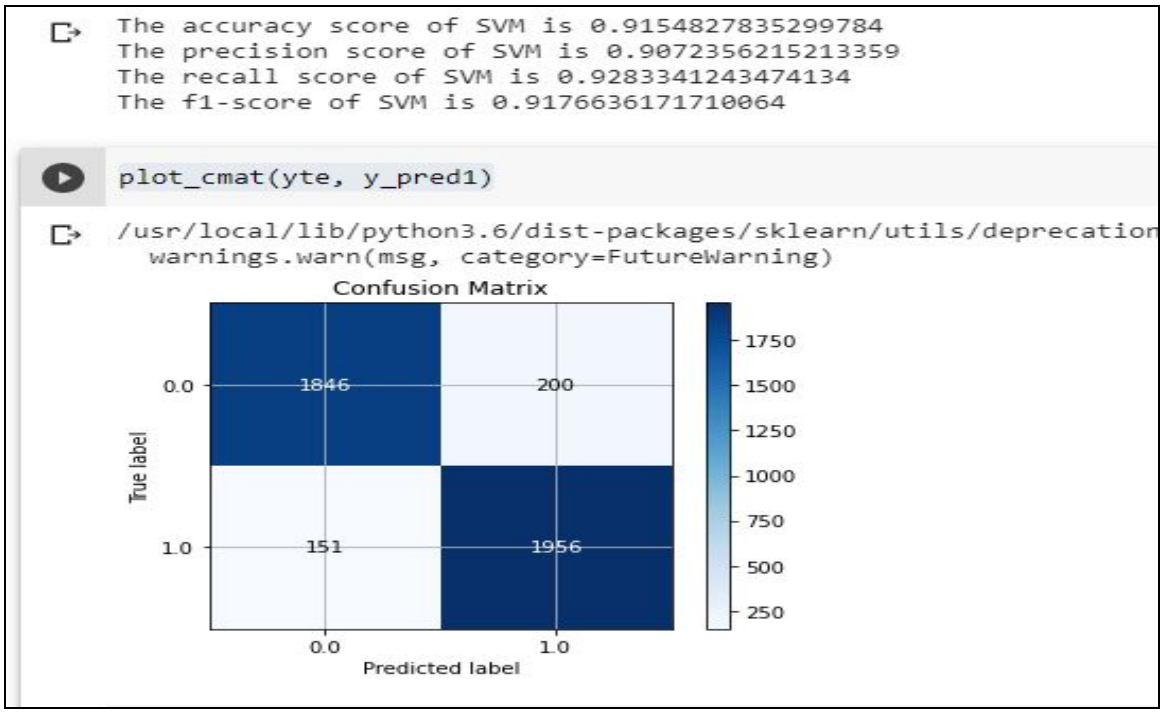

Fig 4 - *Support Vector Machine*

```
The accuracy score of random forest is 0.883457741391765
The precision score of random forest is 0.8968215158924205
The recall score of random forest is 0.8704318936877077
The f1-score of random forest is 0.8834296724470135

[37] plot_cmat(yte, y_pred2)

/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.
  warnings.warn(msg, category=FutureWarning)
```

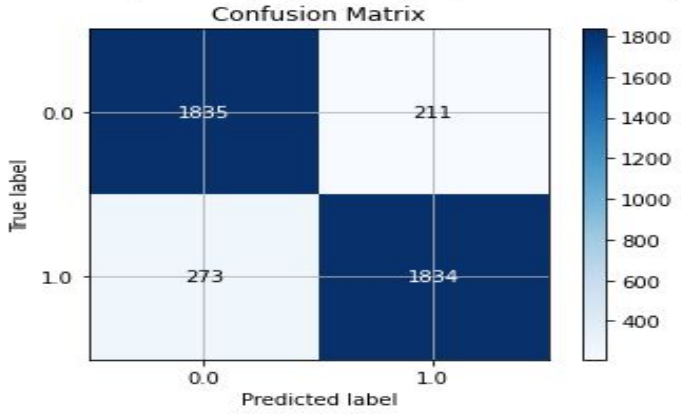

Fig 5 - *Random Forest*

```
The accuracy score of deep neural network is 0.9229472670358777
The precision score of deep neural network is 0.9196806012212306
The recall score of deep neural network is 0.9292833412434741
The f1-score of deep neural network is 0.9244570349386213

26] plot_cmat(y_test, y_pred3.round())

/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:8
  warnings.warn(msg, category=FutureWarning)
```

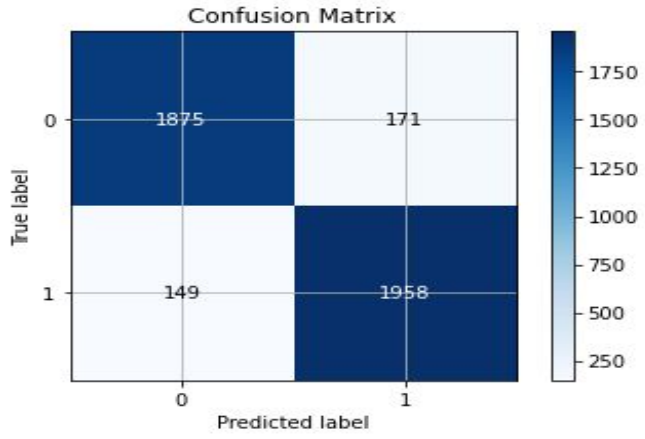

Fig 6 - *Deep Feed-forward Neural Network*

7. CONCLUSIONS AND FUTURE WORKS

This study of fake news detection combining NLP and ML techniques is a generic one where the Deep Feed-forward Neural Network gives the best result. But to achieve maximum out of it several advanced modifications can be done. Along with text other metadata like author, title can be used to achieve higher accuracy. For fake news detection, we can also add as features the source of the news, including any associated URLs, the topic (e.g., science, politics, sports, etc.), publishing medium (blog, print, social media), country or geographic region of origin, publication year etc. The semantic and syntax analysis of the text can be done using EMPATH and POS tags respectively. Also GloVe embedding can be used for better results. Advanced NLP techniques like topic based modelling can provide a better result. More advanced deep learning models like CNN, LSTM can be used for achieving higher accuracy. Larger dataset can be used to make the most of deep learning. More closed to real life dataset can be formed by using APIs like twitter, facebook to collect the news from social media itself. In this way a complete, robust, production-quality fake news detection tool can be produced.

8. REFERENCES

[1] "Liar, Liar Pants on Fire : A New Benchmark Dataset for Fake News Detection" - William Yang Wang , 2017. https://arxiv.org/pdf/1705.00648.pdf

[2] "A Benchmark Study on Machine Learning Methods for Fake News Detection" - Junaed Younus Khan, Md. Tawkat Islam Khondaker, Anindya Iqbal, Sadia Afroz, 2019 . https://arxiv.org/pdf/1905.04749.pdf

[3] "Fake News Detection on Social Media:A Data Mining Perspective" - Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang and Huan Liu, 2017. https://arxiv.org/pdf/1708.01967.pdf