

Fake News Classifier Project



Submitted by:

**Pradeep Kumar Singh**

**ACKNOWLEDGMENT**

In preparation of this Project, I had to take the help and guidance of some video tutorials and NLP articles, who deserve my deepest gratitude. As the completion of this Project gave me much pleasure, I would like to show my gratitude towards codebasics and Kris Naik tutorial channel on YouTube, articles on towardsdatascience.com and medium.com for giving me a good guidelines for this Project throughout start to end using some sample data. I would also like to expand my gratitude to all those who have directly and indirectly who helped me to use wide thinking to explore here in dataset.

In addition, a thank you to my mentor Astha Mishra, who introduced me to the Methodology of work, and who support always wherever I stuck ever. I also thank FlipRobo Technologies for providing such an opportunity to work on various types of projects which gradually improve my vision to apply for the ease to solve the complexity in the data.

**INTRODUCTION**

Business Problem Framing

Nowadays, we hear so many news directly or indirectly from printed as well as digital idea. But the authenticity of the news is such issue in our society that it may be genuine or its just a rumour.Because the hype created around the city always leads our way whether we want to take it or not. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users.

Conceptual Background of the Domain Problem

It is rightly said “ Now matter how much good and best you’re, always take the advise from experienced ones to get it done in smarter way”. The same thing applies to every field. Some daily newspaper readers can pick the slang or the motive of the writer whether he wants to deliver the news or just cooking some mocking items to keep the newspaper in market.

Review of Literature

Whether or not you’re a Republican or Democrat, social-media junkie or forested hermit, you have to admit that our rapidly digitalizing, polarized world is running into a certain problem: **fake news**. Whether it’s Facebook and the whole spectrum of monolithic technology companies feeding us only information that we want to hear to increase advertising revenue, we are losing touch on diversified information, and as consumers, are facing the likelihood that whatever we are fed may not even be accurate. This is significant — given the average adult, according to eMarketer, spends 3 hours and 54 minutes per **day** on our phones. That’s a lot of time to be fed information we don’t even know is accurate.

Motivation for the Problem Undertaken

As a data scientist, my goal is to harness the good of the information age to catalyse change in society, and hopefully progress within this field and like minded researchers over time will reduce the negatives. This project, my self-dubbed “fake news classifier”, will aim to use natural language processing to classify whether a news is fake or not fake.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

Here we are dealing with two main text columns which held some importance of the data. I prefer to select on focus more on the words which has great value of importance in the context. TF-IDF is the NLP terms I am going to apply on both these columns. This converts the important words a proper vectors with some weights.

* Data Sources and their formats

I received the dataset for this project through my mentor on this project and is in the .csv file. Some of the info such as Unnamed: 0,id,written\_by are seems unnecessary with respect to the problem. So I drop it before any processing. Here is the screenshot of data

****

* Data Pre-processing Done

Removing the stop-words, punctuations and special characters from the text from both feature is done here.

* Data Inputs- Logic- Output Relationships

The input data for the processing and getting the output is converted in to the numerical forms or it data words, in to the vector form. It feeded to the model in the form of series (one by one) which analyses by the model providing the certain score through the medium of performance metrics.

* Hardware and Software Requirements and Tools Used

Hardware And Software required for this project:

* Laptop with I5 processor and 4 GB Ram
* Anaconda (Jupyter Notebook) or google colab
* Python Pandas for processing
* Scikit learn library and imbalanced-learn for model deployment

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

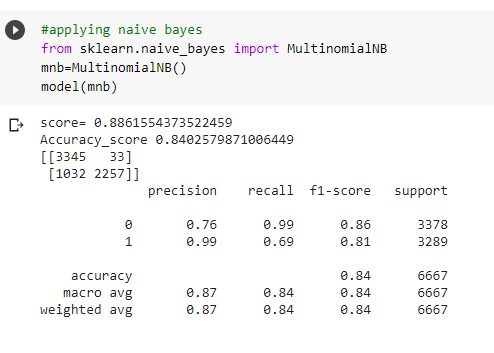
These are the following approaches I have used here:

* Importing and displaying insights of the data using pandas Dataframe.
* Analysing the data and use proper pre-processing techniques to extract out important words.
* Applying data operations such as Tf-IDF to convert text to vectorized form
* Applying algorithms one by one to check the performance using evaluation metrics.
* Testing of Identified Approaches (Algorithms)

Algorithms used in the dataset after cleaning and processing are as follows:

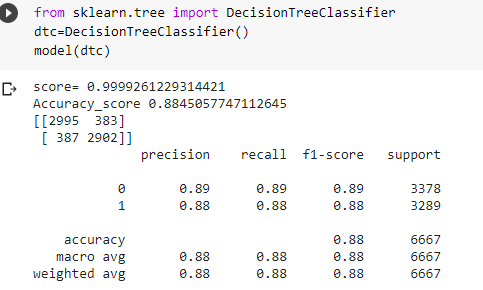
* Multinomial naive Bayes
* Decision tree Classifier
* Random Forest Classifier
* Run and Evaluate selected models

**Multinomial naive Bayes** : The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.



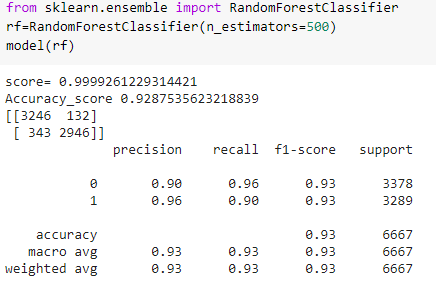
**Decision Tree**

Decision Tree is one of the earlier classification algorithm for text and data mining .Decision tree classifiers (DTC’s) are used successfully in many diverse areas of classification. The structure of this technique includes a hierarchical decomposition of the data space (only train dataset). The main idea is creating trees based on the attributes of the data points.



**Random Forest Classifier**

Random forests or random decision forests technique is an ensemble learning method for text classification. This method implies that the decision taken is depend on trees in parallel.

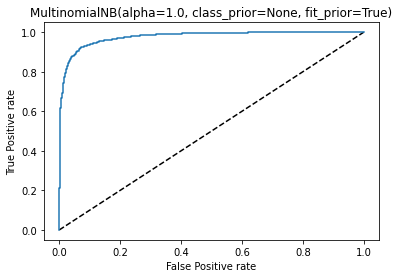
****

Key Metrics for success in solving problem under consideration

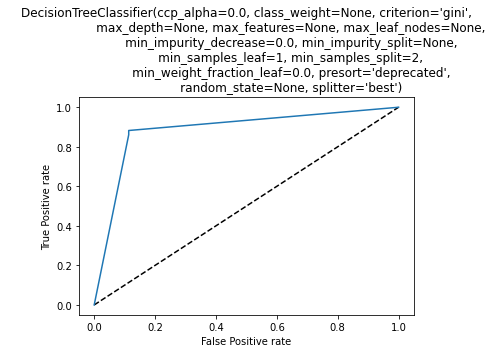
The performance metrics here mainly considered for the problem solving are:

* Accuracy score
* Confusion matrix
* Roc Curve & Auc-RoC score.
* Visualizations

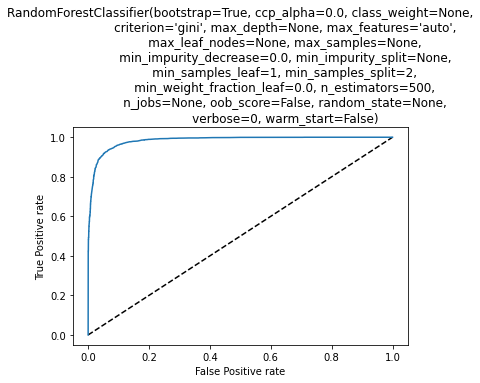
RoC curve plots for different applied algorithms are as follows:



The above graph plot is showing RoC-AuC curve is showing the fpr and tpr data points predicted using MultinomialNB. The roc-auc score for above graph is 0.8382288612615278.



This graph is showing results as per decision tree classifier. The roc-auc score for above graph is 0.8844771788049262.



ROC-AUC curve of Random Forest Classifier and score is 0.9283183030576652.

**CONCLUSION**

* Key Findings

Some of the findings from data explorations are as follows:

* From displaying the data, it seems there is lot of null values present in the data. So, it is better to proceed by dropping.
* As the above data is in text, so presence of special characters and stopwords is always there.
* The data difference ratio in target label is around 9:10.
* After proper cleaning and processing, random forest classifier gives the highest accuracy as well as roc score.
* Limitations of this work and Scope for Future Work

Some of the extention techniques that give the best results after optimization are:

* Multinomial naïve bayes using hyperparameter tuning.
* Support Vector Machine also performs well in text data but its hyperparameter tuning is very complex and takes much more time.