

NAME OF THE PROJECT :-

**Fake -News Detection Project**

Submitted by:

ABHINAV KUMAR

**ACKNOWLEDGMENT**

The following are URLs from where I have taken reference to build up my project and create this report.

* <https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/>
* <https://www.researchgate.net/publication/336436870_Fake_News_Detection_Using_Machine_Learning_approaches_A_systematic_Review>
* <https://medium.com/analytics-vidhya/fake-news-detection-using-nlp-techniques-c2dc4be05f99>
* NLTK:

[Bird, Steven, Edward Loper and Ewan Klein (2009).]

Natural Language Processing with Python. O'Reilly Media Inc.

The above are the websites that have helped and guided me for this project.

**INTRODUCTION**

* Business Problem Framing

In our modern era where internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in use of social media platforms like Facebook, Twitter etc. news spread rapidly among millions of users within a very short span of time. Falsified news

or fabricated post news is any textual or non-textual content that is fake and is generated so the readers will start believing in something which is not true. The spread of fake news has far reaching consequences like creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In the recent elections of United States, there has been much debate regarding the authenticity of various news reports favoring certain candidates and the political motives behind them. Amidst such growing concerns, the detection of fake news gains utmost importance to prevent its negative impacts on individuals and society.

* Conceptual Background of the Domain Problem

In this project, we aim to perform a binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. The most common algorithms used by fake news detection systems include machine learning algorithms such as Support Vector Machines, Random Forests, Decision trees, Logistic Regression and so on. In this project I have attempted to implement some of these algorithms to train and test our results. The main challenge throughout the project has been to build a set of uniform clean data and to tune parameters of our algorithms to attain the maximum accuracy.

* Review of Literature

The available literature has described many automatic detection techniques of fake news and deception posts. Since there are multidimensional aspects of fake news detection ranging from using chatbots for spread of misinformation to use of click baits for the rumor spreading. There are many click baits available in social media networks including Facebook which enhance sharing and liking of posts which in turn spreads falsified information. Lot of work has been done to detect falsified information. Various detection techniques have been introduced by authors the authors have introduced following Detection Methods

1. Linguistics basis Deception modelling

2. Clustering

3. Predictive modelling

4. Content cue-based methods

5. Non text cue-based methods

* Motivation for the Problem Undertaken

When detecting fake news from a knowledge-based perspective, one often uses a process known as fact-checking. Fact checking, initially developed in journalism, aims to assess news authenticity by comparing the knowledge extracted from to-be-verified news content (e.g., its claims or statements) with known facts. Expert-based fact-checking relies on domain-experts as fact-checkers to verify the given news contents. Expert-based fact-checking is often conducted by a small group of highly credible fact-checkers, is easy to manage, and leads to highly accurate results, but is costly and poorly scales with the increase in the volume of the to-be-checked news contents. Manual fact-checking does not scale with the volume of newly created information, especially on social media. **To address scalability, automatic fact-checking techniques have been developed, heavily relying on Information Retrieval (IR), Natural Language Processing (NLP), and Machine Learning (ML) techniques**.

**Analytical Problem Framing**

* Mathematical/ Analytical Modelling of the Problem

Various statistical analysis is also done on the model to check the statistical measurements such as max and min value, mean, median, variance and standard deviation. This analysis also provides the presence of outliers which can be observed by looking at 3rd quartile and the max value, if the difference between the two is more than outliers is present in the dataset. Also, these outliers are removed by calculating the z score which is another inbuilt method and a threshold value is set above which all the values having z score above threshold is considered as outliers.

1. **Logistic Regression Model: -**

Logistic Regression is a Machine Learning technique used to estimate relationships among variables using statistical methods. This algorithm is great for binary classification problems as it deals with predicting probabilities of classes, and hence our decision to choose this algorithm as our base-line run. It relies on fitting the probability of true scenarios to the proportion of actual true scenarios observed. Also, this algorithm does not require large sample sizes to start giving fairly good results.



1. **Naïve Bayes Classifier: -**

This is a simple yet powerful classification model that works remarkably well. It uses probabilities of the elements belonging to each class to form prediction. The underlying assumption in the Naïve Bayes model is that the probabilities of an attribute belonging to a class is independent of the other attributes of that class. Hence the name ‘Naive’.

****

* Data Sources and their formats

The data sources that are provided us is in the form of CSV i.e., comma separated values. There are 6 columns in the dataset provided to you. The description of each of the column is given below:

“id”: Unique id of each news article

“headline”: It is the title of the news.

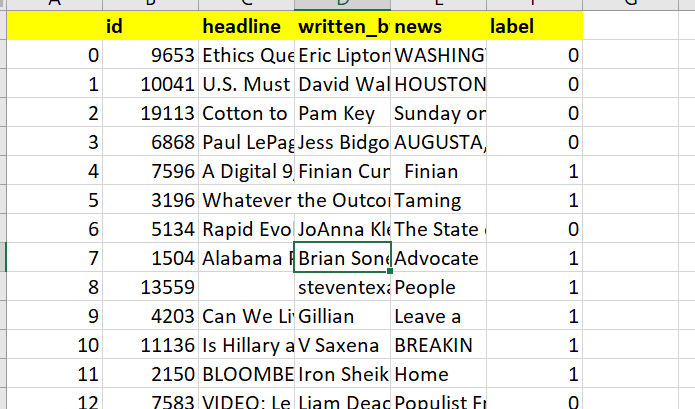
“news”: It contains the full text of the news article

“Unnamed:0”: It is a serial number

“written\_by”: It represents the author of the news article

“label”: It tells whether the news is fake (1) or not fake (0).

The below is the snip of the data.



* Data Pre-processing Done

Data pre-processing is one of the most data mining steps which deals with data preparation and transformation of the dataset and seeks at the same time to make knowledge discovery more efficient. Pre-processing includes several techniques like cleaning, integration, transformation, and reduction. Pre-processing also includes dealing with null values but fortunately this dataset does not contain any null value hence we proceed by dropping the columns that is no longer relevant for prediction.

The data was made uniform and comparable by converting it into a uniform UTF-8 encoding. There were some cases where we encountered weird symbols and letters incompatible with the character set which had to be removed. We noticed that the data from news articles were often organized into paragraphs.

Further our pre-processing of data includes removing of stop words. NLTK supports stop word **removal**, and you can find the list of **stop words** in the corpus module. To **remove stop words** from a sentence, you can divide your **text** into words and then **remove** the word if it exits in the list of **stop words** provided by NLTK.

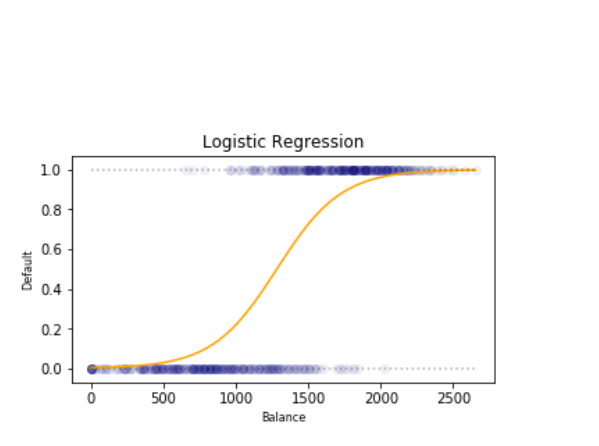
Then after removing the stop words I have used porter stemmer. The **Porter stemming algorithm** (or '**Porter stemmer**') is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.

After we have stemm the words we will now create the array and convert the words into vectors which can be done with the help of count vectorizer. **Count Vectorizer** is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (**count**) of each word that occurs in the entire text.

So, after all this filtration steps mentioned above, we finally got the pre-processed data that can be very well given to the machine learning model to build an effective fake news detector.

* Data Inputs- Logic- Output Relationships

As we have separated the dataset in x and y we can see that the target variable i.e. y is in discrete form i.e. in the form of 0 and 1 which is our dependent variable and rest all the data are our independent variables.



Logistic Regression calculates the probability, by which a sample belongs to a class, given the features in the sample. This probability is calculated for each response class. The class with the highest probability is generally taken to be the predicted class.

The Logistic Regression is mostly used and best suited for problems having binary response classes, for example, → 0 or 1, true or false, fake and not fake, type A or type B, etc.

* Hardware and Software Requirements and Tools Used

To build a machine learning based model a Jupyter notebook is required, the Jupyter notebook is preloaded with various packages and libraries which we have used for our model.

Below is the list of packages and that is useful for this prediction.

1. **Pandas Library**- pandas is used to perform python data analysis whose abbreviation is ‘pandas’ further it is used to read the csv files, to perform modifications in data frame such as cleaning of dataset.
2. **Numpy Library** - It is the numerical python library used to perform mathematical operations.
3. **Matplotlib and Seaborn Library** - s**eaborn** and **Matplotlib** are two of **Python's** most powerful visualization libraries. **Seaborn** uses fewer syntax and has stunning default themes and **Matplotlib** is more easily customizable through accessing the classes.
4. **Scikitlearn Library** - **Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a free software machine learning library. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Various metrices are also used under scikit learn package such as classification report, confusion metrics, accuracy score, etc.

1. **NLTK** -The Natural Language Toolkit, or more commonly **NLTK**, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

**Logistic Regression**

Logistic Regression is a Machine Learning technique used to estimate relationships among variables using statistical methods. This algorithm is great for binary classification problems as it deals with predicting probabilities of classes, and hence our decision to choose this algorithm as our base-line run. It relies on fitting the probability of true scenarios to the proportion of actual true scenarios observed. Also, this algorithm does not require large sample sizes to start giving fairly good results.

The Logistic Regression algorithm works by assigning observations to a discrete set of classes and then transforms it using a sigmoid function to give the probability value which can be mapped to a discrete class.

**Naïve Bayes Classifier**

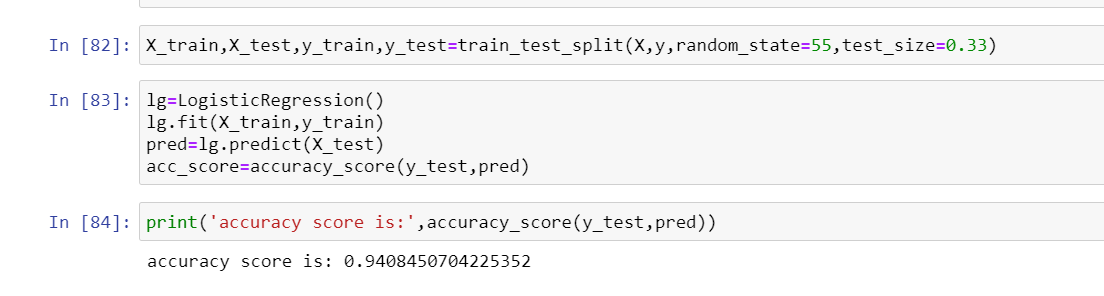
This is a simple yet powerful classification model that works remarkably well. It uses probabilities of the elements belonging to each class to form a prediction. The underlying assumption in the Naïve Bayes model is that the probabilities of an attribute belonging to a class is independent of the other attributes of that class. Hence the name ‘Naive’. In this model we multiply the conditional probabilities of each attribute given the class value, to get the probability of the test data belonging to that class. We arrive at the final prediction by selecting the class that has the highest of the probabilities for the instance belonging to that class. The advantages of using Naïve Bayes are that it is simple to compute, and it works well in categorizing data as we are using ratios for computation

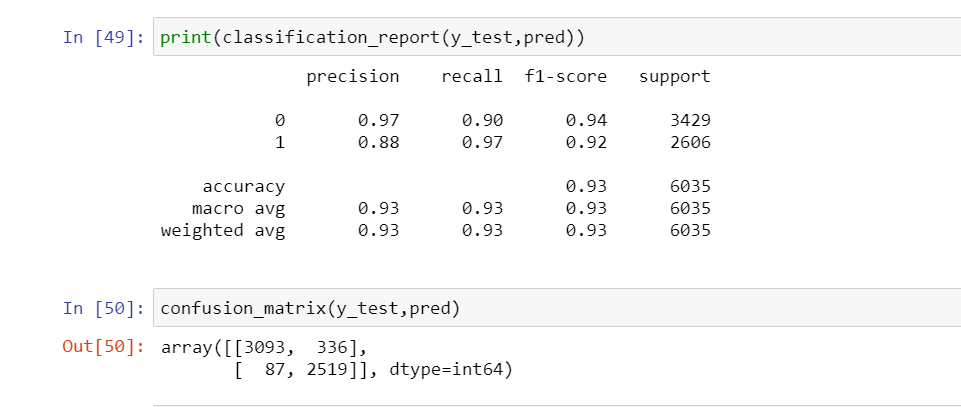
**Decision tree classifier**

**Decision trees** use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. The **decision tree** splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

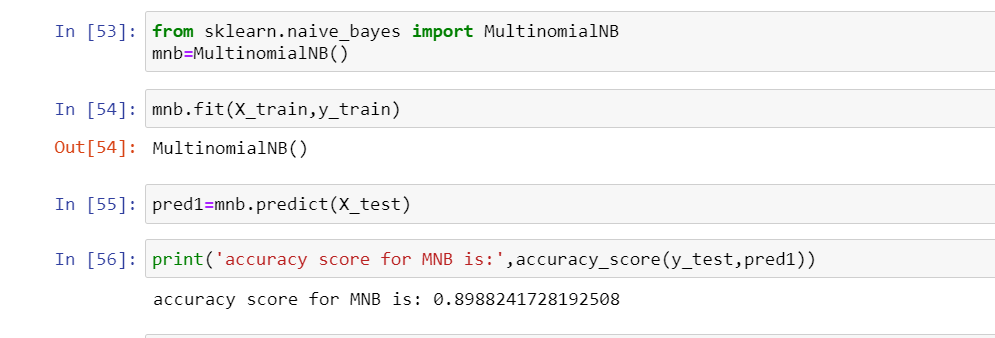
* Run and Evaluate selected models

1. **Logistic Regression Algorithm**: -



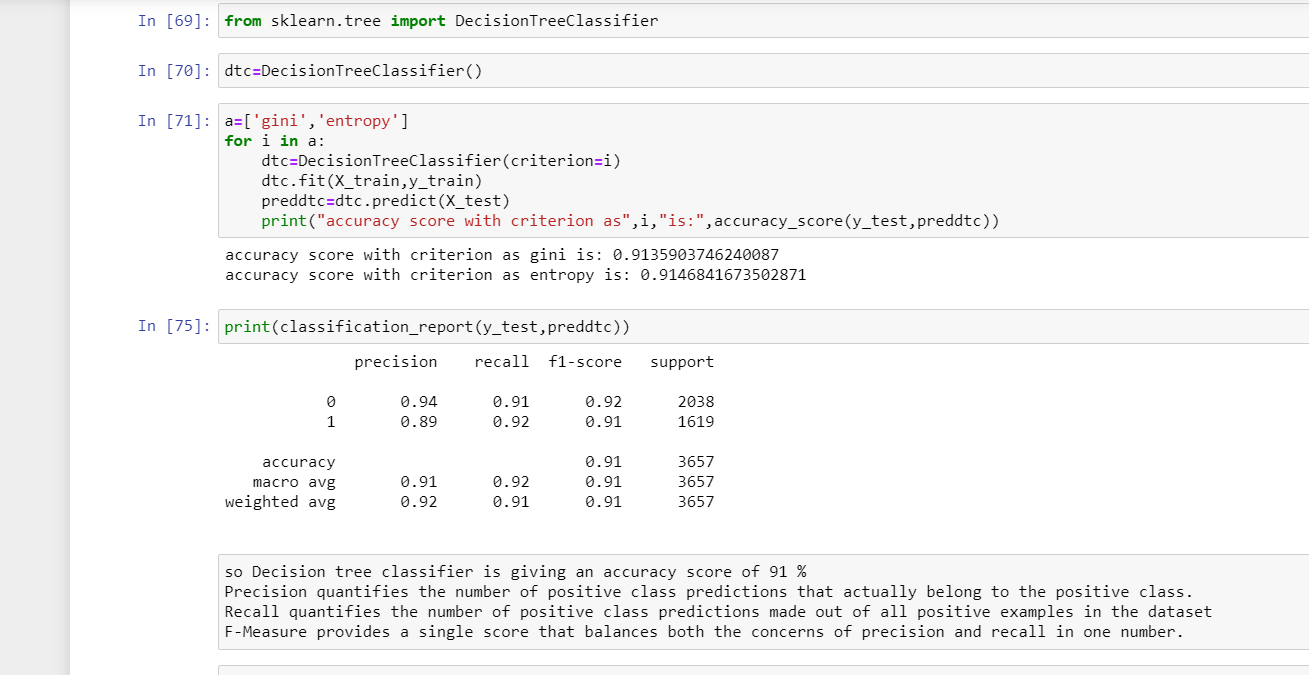


1. **MultinomialNB**: -





1. **Decision Tree Classifier: -**



* Key Metrics for success in solving problem under consideration

The key metrices used for solving the problem are as given below: -

1. **Accuracy Score** - **Classification accuracy** is simply the rate of correct **classifications**, either for an independent test set, or using some variation of the cross-validation idea. More the accuracy more is the model reliable.

2. **Precision and Recall**. **Precision** talks about all the correct predictions out of total positive predictions. **Recall** means how many individuals were classified correctly out of all the actual positive individuals.

3**. F1 score** - **F1 Score** is the weighted average of Precision and Recall. Therefore, this **score** takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but **F1** is usually more useful than accuracy, especially if you have an uneven class distribution.

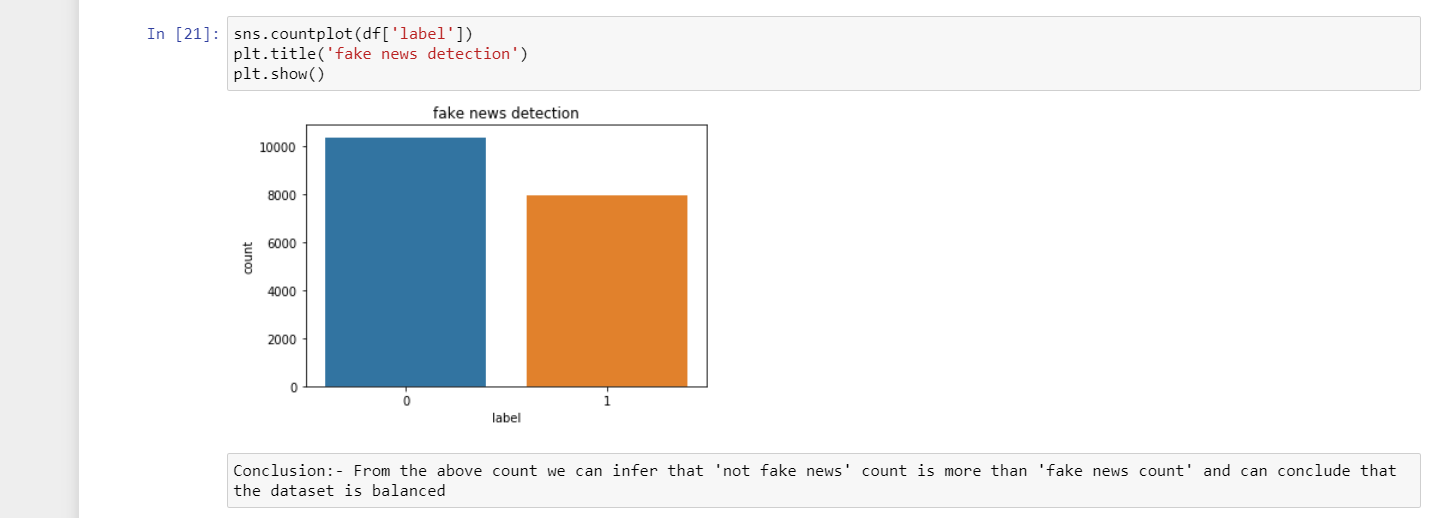
4. A **Confusion matrix** is an N x N **matrix** used for evaluating the performance of a classification model, where N is the number of target classes. The **matrix** compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.

5. **Log Loss** - **Log loss**, aka logistic **loss** or cross-entropy **loss**. This is the **loss** function used in (multinomial) logistic regression and extensions of it such as neural networks, defined as the negative **log**-likelihood of a logistic model that returns y\_pred probabilities for its training data y\_true.

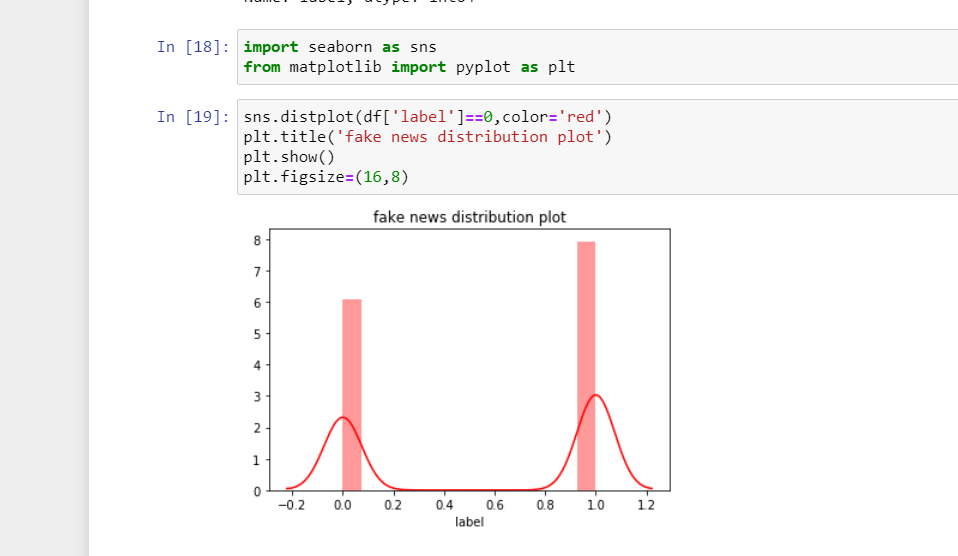
* Visualizations

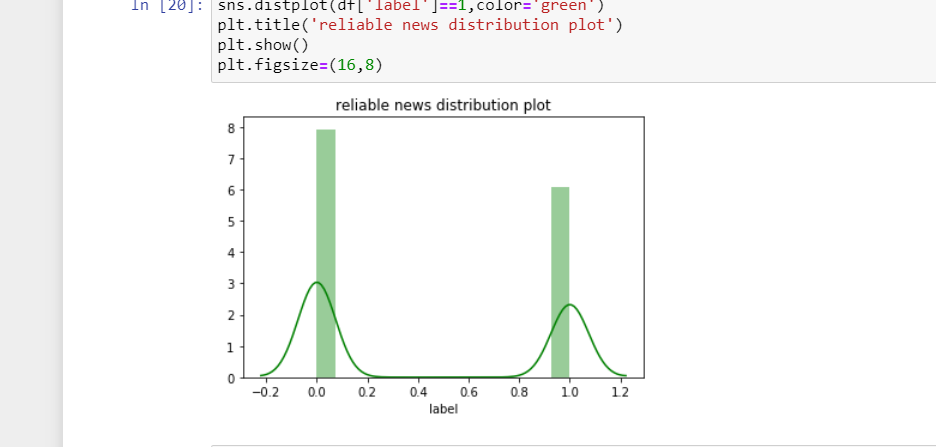
For visualization we have two inbuilt libraries, matplotlib and seaborn. With the help of this we will plot different types of plots as per the data to get some meaningful insights. The below are the plots plotted using these libraries,

1. **Countplot** – A count plot basically counts the categories and returns a count of their occurrences. It is one of the simplest plots provided by the seaborn library.



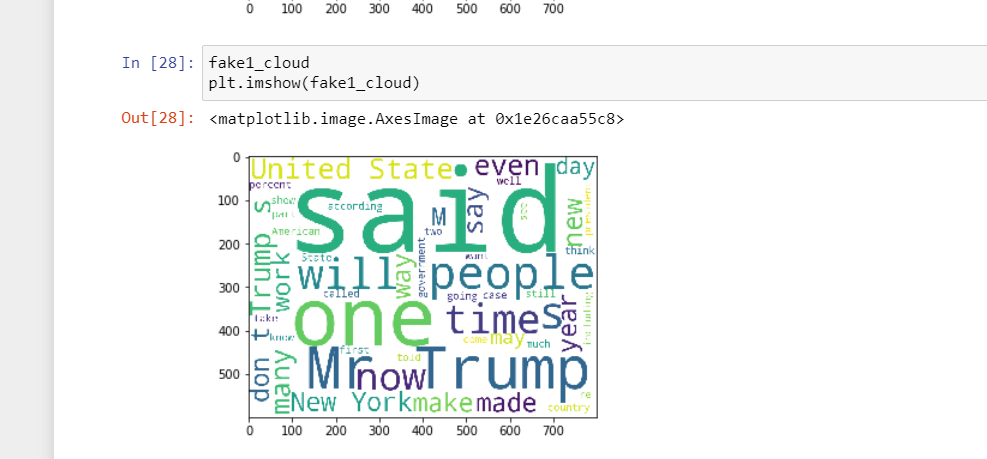
1. **Distribution plots**: - Seaborn **distplot** lets you show a histogram with a line on it. We use seaborn in combination with matplotlib, the **Python** plotting module. A **distplot** plots a univariate distribution of observations. The **distplot**() function combines the matplotlib hist function with the seaborn kdeplot() and rugplot() functions.



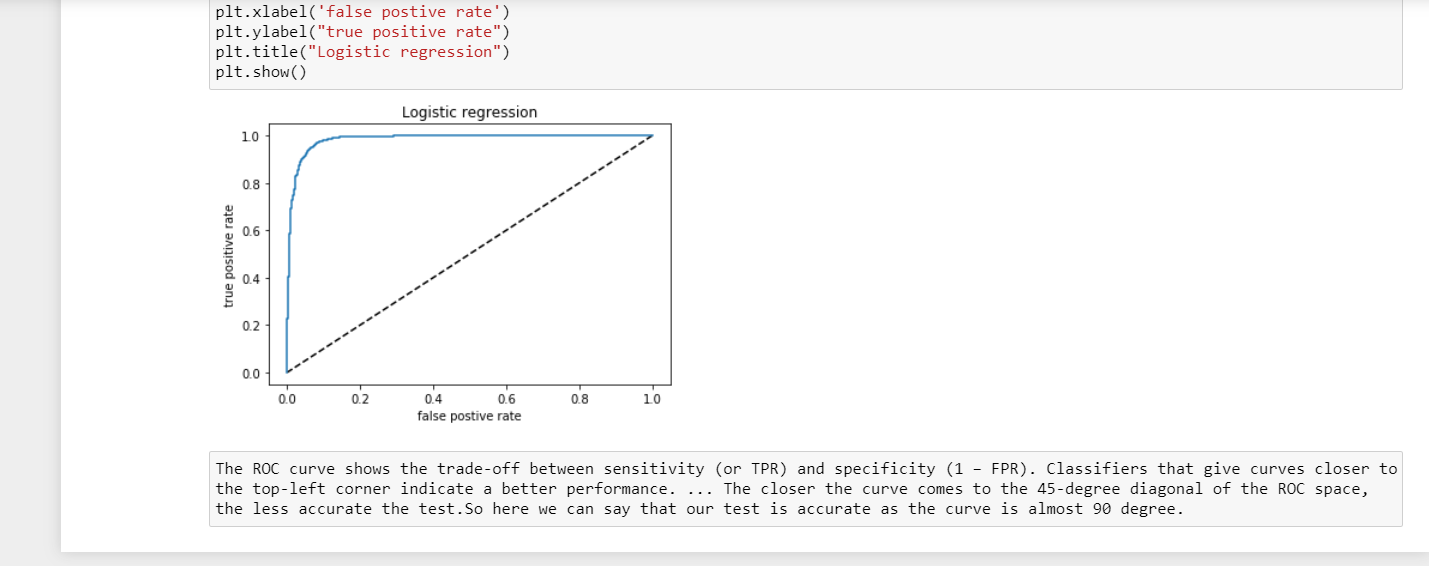


1. **Word Cloud**: - A **Wordcloud** (or **Tag cloud**) is a visual representation of text data. It displays a list of **words**, the importance of each being shown with font size or colour. This format is useful for quickly perceiving the most prominent terms.





1. **AUC- ROC Curve**: - The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR**against **FPR**at various threshold values and essentially **separates the ‘signal’ from the ‘noise’**. The **Area Under the Curve (AUC)**is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The [ROC curve](http://danielnee.com/wp-content/uploads/2014/08/ROC2.png) is plotted with TPR (True Positive Rate, which is True Positive / [True Positive + False Negative]) against the FPR (False Positive Rate, which is [False Positive / False Positive + True Negative]) where TPR is on the y-axis and FPR is on the x-axis. ROC, other than accuracy, is widely used in imbalanced data. This is the case for credit default risk prediction because it is a combination of precision and recall. And compared to the F1 Score, which is also a comprehensive representation of precision and recall, for ROC, you don’t need to manually choose one single threshold for the prediction probability to decide if the prediction output is positive or negative.



* Interpretation of the Results

The following are the inferences from the above plots: -

1. We can conclude from the plot that the data is balanced. This is an important step because there are lot of real-world datasets that are imbalanced.
2. We can conclude from the word clouds that there are lot of real and fake news about Donald Trump and United States.
3. The data is well distributed.
4. The test performed on the dataset is very accurate.

**CONCLUSION**

* Key Findings and Conclusions of the Study

We have classified our news data using three classification models. We have analysed the performance of the models using accuracy and confusion matrix.

Naïve Bayes performed very poorly. Logistic Regression surprisingly performed very well, as observed from the above results Logistic regression performed slightly better than MultinomialNB and Decision tree classifier.

The accuracy of the models used is below: -

1. Logistic Regression – 94 %
2. Multinomial NB – 90 %
3. Decision Tree Classifier – 91%

The accuracy score speaks on its own that which is the best model.

* Learning Outcomes of the Study in respect of Data Science

Although there is evident success in detection of fake news and posts using various Machine learning approaches. However, everchanging characteristics and features of fake news in social media networks is posing a challenge in categorization of fake news.

* Limitations of this work and Scope for Future Work

As explained, fake news detection itself is a binary classification problem, any machine learning method than can be used in binary classification problems is theoretically applicable. But each algorithm has its own strengths and weakness.

**Pros and Cons of Logistic Regression**

The strength of logistic regression can be summarized as follows:

* **Simple, fast and low-memory usage.** In the logistic model, for each feature variable, there is only one corresponding weight variable. So regardless of whether you update them during training or apply the model in prediction, it will be very fast and with low memory demand.
* **Interpretable.** It is easy to see the effects of each feature of the model, which is very important in finance and is also one of the reasons why the model is still widely used today.
* With good feature engineering, the **performance** can also be really good.
* It is **easy to convert** the model result to a specific strategy and deploy

However logistic regression has below weakness.

* **Easy to underfit.** Also, compared to ensembling models, the performance is not that good.
* **High demand for data,** sensitive to missing values, anomaly values, and unable to process non-linear features. This means data cleaning and feature engineering will cost quite a lot of time.
* Not good at dealing with unbalanced data, high-dimension feature set, and categorical features.

**Pros and Cons of Decision Tree**

Like logistic regression, the decision tree method has its strengths and weakness. And the pros can be described as follows:

* Easy to understand (if…then… rules-like structure) and interpretable.
* Less data pre-processing is needed compared to logistic regression. No need to do feature discretization and data normalization.
* The best existing algorithm for processing non-linear relationships.

There are also some weaknesses:

* It’s easy to generate extremely complex tree structures, which leads to overfitting.
* It’s not a good choice for high-dimension data.
* Decision trees have poor generalization capability, can’t deal with values that are not shown in the training dataset.

------------------------------END-------------------------------------