FAKE NEWS DETECTION

A Machine Learning Project Report

Gourab Ghosh
M.Sc. in Big Data Analytics, Department of Computer Science
Ramakrishna Mission Vivekananda Educational And Research Institute
Belur Math, Howrah
Pin-711202, West Bengal
June 10, 2020

ACKNOWLEDGEMENT

This is to acknowledge the support and help that has been received while doing this Project, **FAKE NEWS DETECTION** from our Teacher, **Br. Dripta Maharaj**. It is very kind of him to help me in various stages while doing the project and to provied necessary informations which are of geart help. I am very much grateful to him and thanks him for his cordial coordination.

I would also like to thank my Institution and faculty members for helping me in my project. I want to express my gratitude to Br. Mrinmay Maharaj, Professor Sudipto Das and Swathy Prabhu Maharaj, H.O.D of Computer Science Dept, RKMVERI Belur, for extending their support.

At last but not the least I would like to express my thanks to my parents without whom this project would not have been possible.

A **Project Report** submitted by **GOURAB GHOSH**
to Ramakrishna Mission Vivekananda Educational and Research Institute for the course of **MACHINE LEARNING** in **M.Sc. in Big Data Analytics**
under the proper guidance of **Br. Dripta Maharaj**

TABLE OF CONTENT

1 ABSTRACT

Machine Learning is the systematic study of algorithms and systems that improve their knowledge or performance with experience. Machine Learning focuses on the development of computer programs that can access data and use it to learn for themselves.
This Project comes up with the application of **NLP (Natural Language Processing)** techniques for detecting the 'fake news', that is, misleading news stories that comes from the non-reputable sources. Is it possible to build a model that can differentiate between "Real" news and "Fake" news? By building a model based on a **count vectorizer** (using word tallies) or a (Term Frequency Inverse Document Frequency) **tfidf matrix**, (word tallies relative to how often they're used in other articles in your dataset) we can get the count of fake and Real News. Then applying the classifier models, such as **Multinomial NB** Classifier and **PassiveAggressive** Classifier, we can classify news class such as **Fake** and **Real**. Here I have done a Python Project to detect Fake News from a dataset using the algorithms and classifier models.

2 INTRODUCTION

These days' fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets. Fake news and lack of trust in the media are growing problems with huge ramifications in our society. Obviously, a purposely misleading story is "fake news "but lately blathering social media's discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.
The importance of disinformation within American political discourse was the subject of weighty attention, particularly following the American president election. The term **'Fake News'** became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views. In this Project, it is tried to produce model that can accurately predict the likelihood that a given article is fake news.
Facebook has been at the epicentre of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user sees it; they have also said publicly they are working on to distinguish these articles in an automated way. Certainly, it is not an easy task. A given algorithm must be politically unbiased – since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what Fake News is. Later, it is needed to look into how the techniques in the fields of machine learning, natural language processing can help us to detect fake news.

2.1 What Is Fake News?

A type of **Yellow Journalism**, that present very little or no legitimate well-researched news while instead using eye-catching headlines for increase in sales. Techniques may include exaggerations of news events, scandal-mongering, or sensationalism. Fake News encapsulates pieces of news that may be hoaxes and is generally spread through social and online media. These types of news may contain false or exaggerated claims, and may end up being viralized by algorithms.

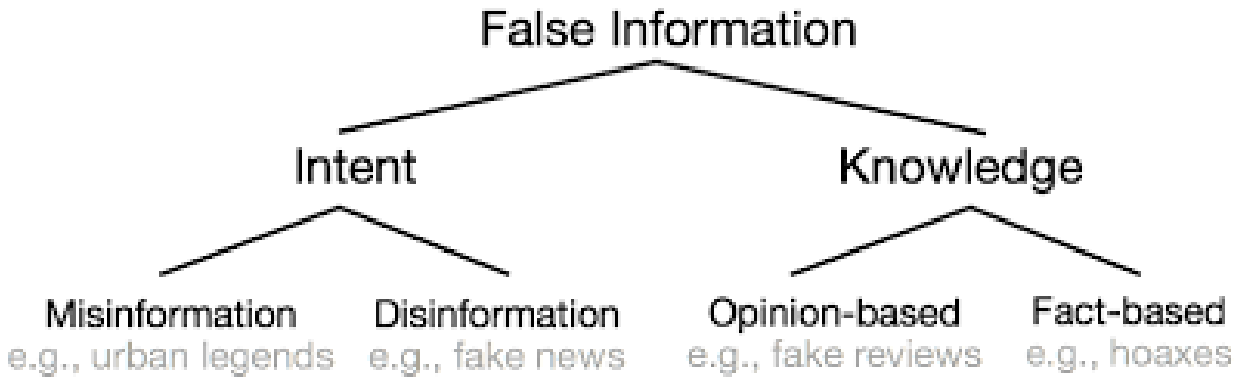

Figure 1: FAKE NEWS

3 OBJECTIVE

The main objective is to detect the fake news, which is a classic text classification problem with a straight forward proposition. It is needed to build a model that can differentiate between **"Real"** news and **"Fake"** news.

3.1 The Project Idea

In the age of social media where we continuously have different kind of news and opinions, it is indeed quite important to understand which one is true or false. In the world of Facebook, Twitter, Instagram, YouTube and many others where millions and millions of news circulate in seconds, it is required to keep track of which one is useful and which one is not.

In this Machine Learning Project, I have worked on to build a model in **Python** which can accurately detect whether a piece of news is **Real** or **Fake**.

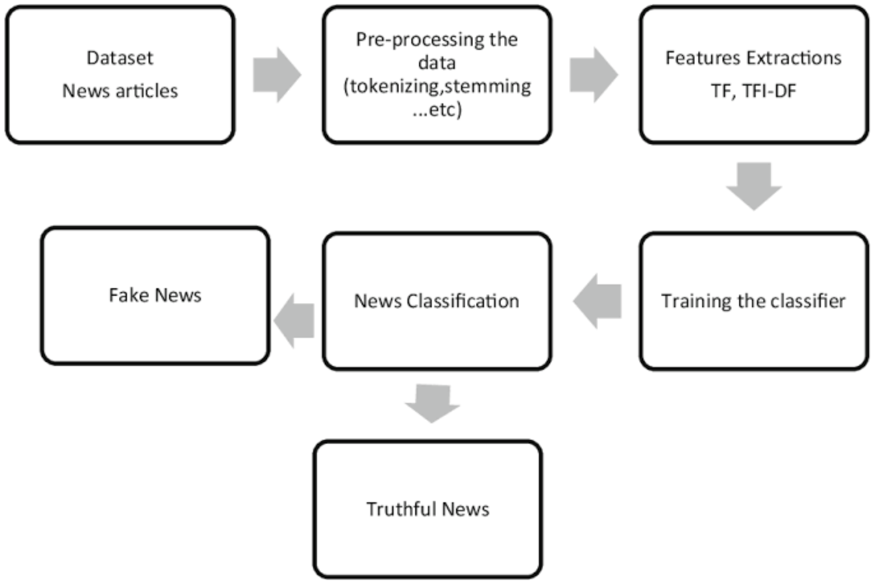

Figure 2: OBJECTIVE

4 EXISTING SYSTEM

There exists a large body of research on the topic of machine learning methods for deception detection, most of it has been focusing on classifying online reviews and publicly available social media posts. Particularly since late 2016 during the American Presidential election, the question of determining **'Fake News'** has also been the subject of particular attention within the literature.
Conroy, Rubin, and Chen outline several approaches that seem promising towards the aim of perfectly classify the misleading articles. They note that simple content-related **n-grams** and shallow **parts-of-speech (POS)** tagging have proven insufficient for the classification task, often failing to account for important context information. Rather, these methods have been shown useful only in tandem with more complex methods of analysis. Deep Syntax analysis using **Probabilistic Context Free Grammars (PCFG)** have been shown to be particularly valuable in combination with **n-gram** methods. Feng, Banerjee, and Choi are able to achieve 85-91 percent accuracy in deception related classification tasks using online review corpora. Feng and Hirst implemented a semantic analysis looking at 'object: descriptor' pairs for contradictions with the text on top of Feng's initial deep syntax model for additional improvement. Rubin, Lukoianova and Tatiana analyse rhetorical structure using a vector space model with similar success. Ciampaglia et al. employ language pattern similarity networks requiring a pre-existing knowledge base.

5 COLLECTING DATASET

The Dataset that I have used here for my Project is called **news.csv**. The dataset has a shape of **6335*4**. The first column identifies the news, the second and third one is the title and text respectively and the fourth column has labels indicating whether the news is **REAL** or **FAKE**.

Table 1: TABLE HEAD

Unnamed:0	**Title**	**Text**	**Label**
8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello	FAKE
10294	Watch the Exact Moment Paul Ryan Committed Pol	Google Pinterest Digg Linkedin Reddit Stumbleu	FAKE
3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon	REAL
10142	Bernie supporters on Twitter erupt in anger ag	Kaydee King (@KaydeeKing) November 9, 2016 T	FAKE
875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners	REAL

Table 2: TABLE FOR COUNT OF DATASET

LABELS	**COUNTS**
REAL	3171
FAKE	3164

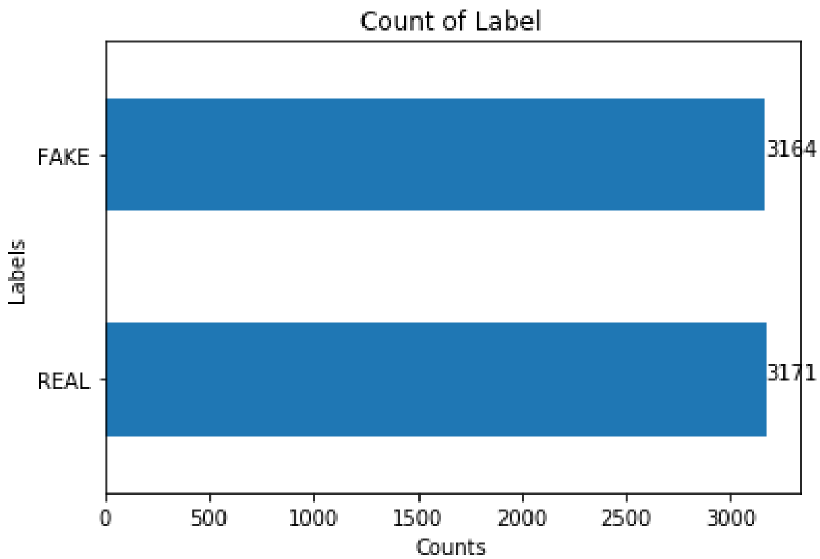

Figure 3: Count Graph

6 PROPOSED SYSTEM

In this Project, model is build based on the **count vectorizer** and **tfidf matrix** ((i.e.) word tallies relatives to how often they are used in other articles in your dataset). Since this problem is a kind of text classification. Implementing a **Naive Bayes classifier** will be best as this is standard for text-based processing. The actual goal is in developing a model which was the text transformation **(count vectorizer vs tfidf vectorizer)** and choosing which type of text to use (headlines vs full text). Now the next step is to extract the most optimal features for countvectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, and/or phrases, lower casing or not, mainly removing the stop words which are common words such as "the", "when", and "there" and only using those words that appear at least a given number of times in a given text dataset. We also have used here another Linear Classifier, **PassiveAggressive Classifier** which corrects the False news and remains same for Real news. It classifies the news set based on its word set. This Classifier also performs very well for this kind of dataset.

6.1 Detect Fake News With Python

This Python Project of Detecting fake news deals with "FAKE" and "REAL" news. Using **sklearn**, I have built a **TfidfVectorizer** and **CountVectorizer** on the dataset. Then, I have initialized a **PassiveAggressive Classifier** and **MultinomialNB Classifier** and have fitted the dataset for all models. In the end, the **Accuracy Score** tells us how good the model is fitted and the **Confusion Matrix** depicts the summary. I have also plotted the **ROC Curve** and calculate **AUC Score** to tell which classifier performs best.

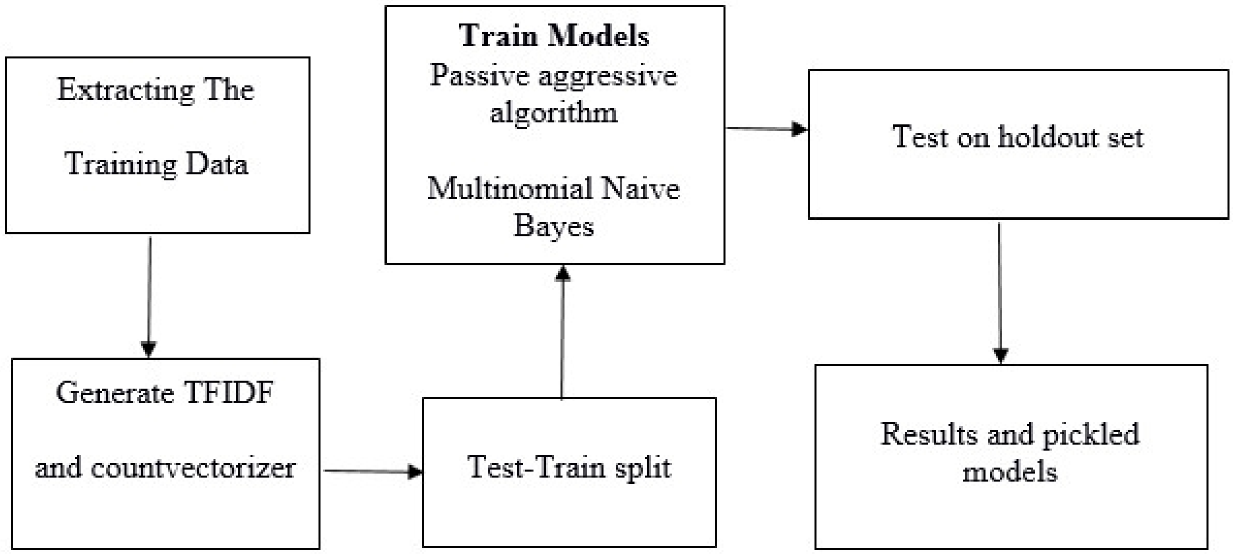

Figure 4: Model Flow Chart

6.2 What Is TfidfVectorizer?

6.2.1 TF (TERM FREQUENCY)

The number of times a word appears in a document is its **Term Frequency**. It gives us the no. of occurrence or frequency of a given word in a document under study.
A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms. If the word has higher no. of frequency in a document, then we can say that the document works as a support of word.

6.2.2 IDF (INVERSE DOCUMENT FREQUENCY)

Words that occur many times in a document, but also occur many times in many others as well, may be irrelevant. **IDF** is a measure of how significant a term is in the entire corpus. **IDF** works as a measure of significance for a given word in the document.
The **TfidfVectorizer** coverts a collection of raw documents into a matrix of **TF-IDF features**.

6.3 What Is PassiveAggressive Classifier?

PassiveAggressive Algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a misclassification, adjusting and updating. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

6.4 What Is CountVectorizer?

CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. As the name suggests **Word with Counts**. This provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

6.5 What Is Multinomial NB Classifier?

Multinomial Naive Bayes Classifier is a specialized version of **Naive Bayes Classifier**. This Classifier is suitable for classification with discrete features (such as word counts for text classification). The Multinomial distribution normally requires integer feature counts. The Multinomial NB Classifier explicitly models word counts and adjusts the underlying calculations to deal with in.

6.6 What Is ROC Curve-AUC Score?

The **Receiver Operating Characteristic Curve**, or ROC Curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system. The Curve is plotted with **TPR** on the **y-axis** against **FPR** on the **x-axis**.
Area Under Curve Score or AUC Score represents degree or measure of separability. It is known that **Higher the AUC Score, better the model** is at classifying or distinguishing. It provides an aggregate measure of performance across all possible classification models.

6.7 What Is Confusion Matrix?

A **Confusion Matrix**, also known as Error Matrix, is a specific table layout that allows visualization of the performance of an algorithm or of a classification model(classifier) on a set of test dataset for which the true values are known.
It allows us to measure Accuracy, Recall, Precision, F1 score and Specificity. It gives the terms like True Positive, True Negative, False Positive and False Negative.

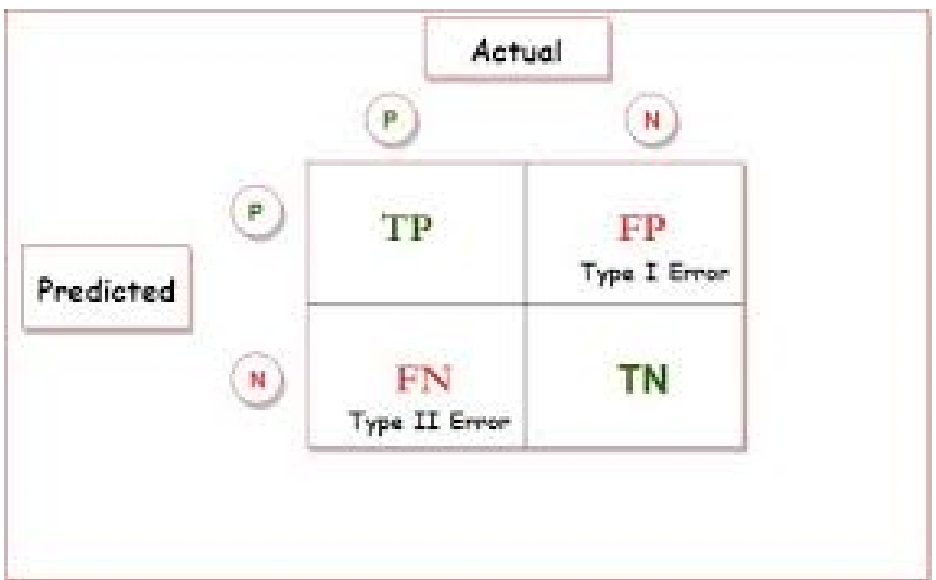

Figure 5: Confusion Matrix

6.7.1 TP (TRUE POSITIVE)

Labels which were predicted positive and were truly positive.

6.7.2 TN (TRUE NEGATIVE)

Labels which were predicted negative and were truly negative.

6.7.3 FP (FALSE POSITIVE)

Labels which were predicted positive but were truly negative, also known as type-1 error.

6.7.4 FN (FALSE NEGATIVE)

Labels which were predicted negative but were truly positive), also known as type-2 error.

6.7.5 ACCURACY

Measurement of being correct —**(TP+TN)/(TP+TN+FP+FN)**.

6.7.6 PRECISION

Measurement of consistency,minimizing **false positives -TP/(TP+FP)**.

6.7.7 RECALL

Measurement of completeness, also known as **" TRUE POSITIVE RATE/SENSITIVITY"**, least **false negative -TP/(TP+FN)**.

6.7.8 F1 SCORE

Indicates the balance between Precision and Recall-**(2*Precision*Recall)/ (Precision+ Recall)**.

6.7.9 SPECIFICITY

Proportion of true negatives that are correctly predicted, also known as**" TRUE NEGATIVE RATE" -TN/(TN+FP)**.

7 RESULTS

The Results that I have got after performing the experiment in **Python** is depicted here below. With the help of these results, it can be said that which classifier model performs best for detecting **FAKE** news.

7.1 Compute Confusion Matrix Multinomial NB Classifier for CountVectorizer

Confusion matrix, without normalization :
[[**552 86**]
[**46 583**]]

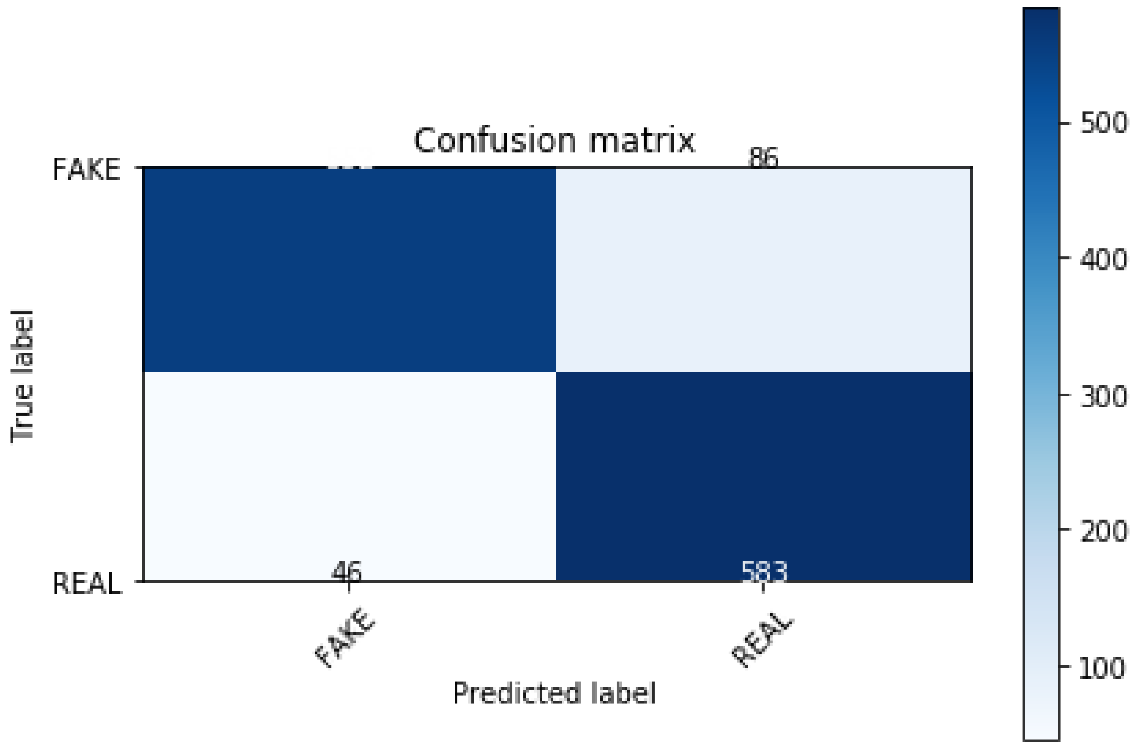

Figure 6: Confusion Matrix For Multinomial NB Classifier for CountVectorizer

True Positives: 583
True Negatives: 552
False Positives: 86
False Negatives: 46
Accuracy: 0.896
Precision: 0.8714
Recall: 0.9269
F1 Score: 0.8983
Specificity: 0.8652

7.2 Compute Confusion Matrix Multinomial NB Classifier for TfidfVectorizer

Confusion matrix, without normalization :

[[**549 89**]

[**32 597**]]

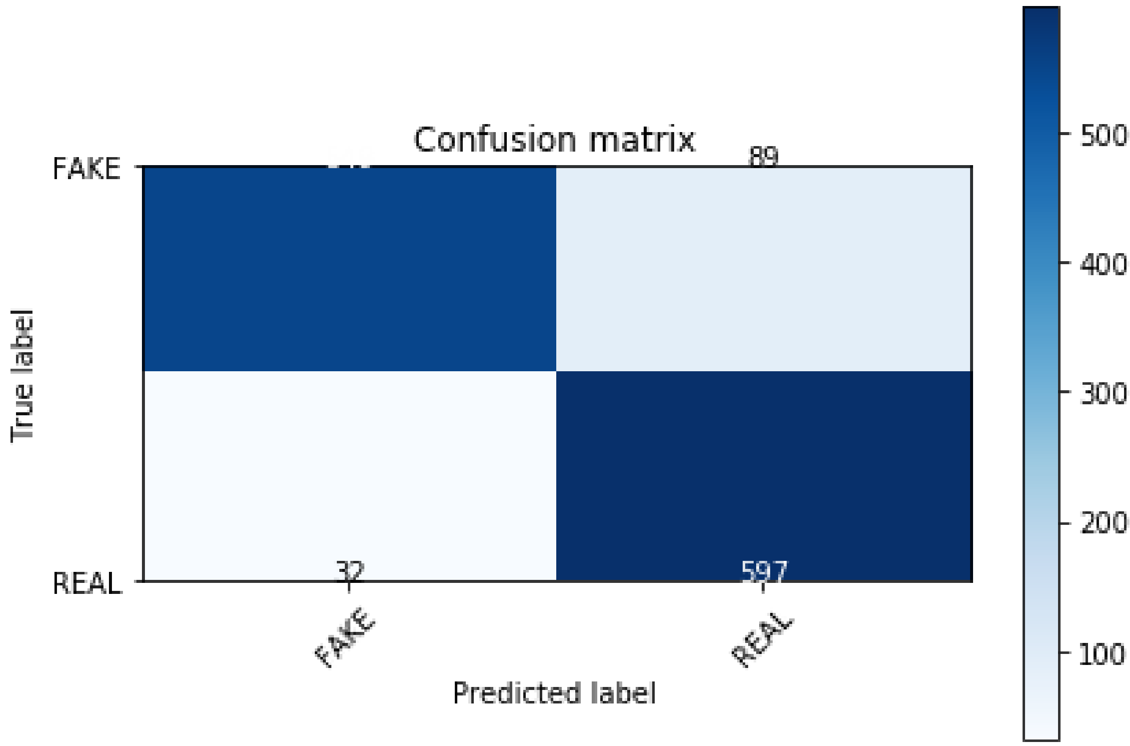

Figure 7: Confusion Matrix For Multinomial NB Classifier for TfidfVectorizer

True Positives: 597
True Negatives: 549
False Positives: 89
False Negatives: 32
Accuracy: 0.904
Precision: 0.8703
Recall: 0.9491
F1 Score: 0.9080
Specificity: 0.8605

7.3 Compute Confusion Matrix PassiveAggressive Classifier for CountVectorizer

Confusion matrix, without normalization :

[[**569 69**]
[**48 581**]]

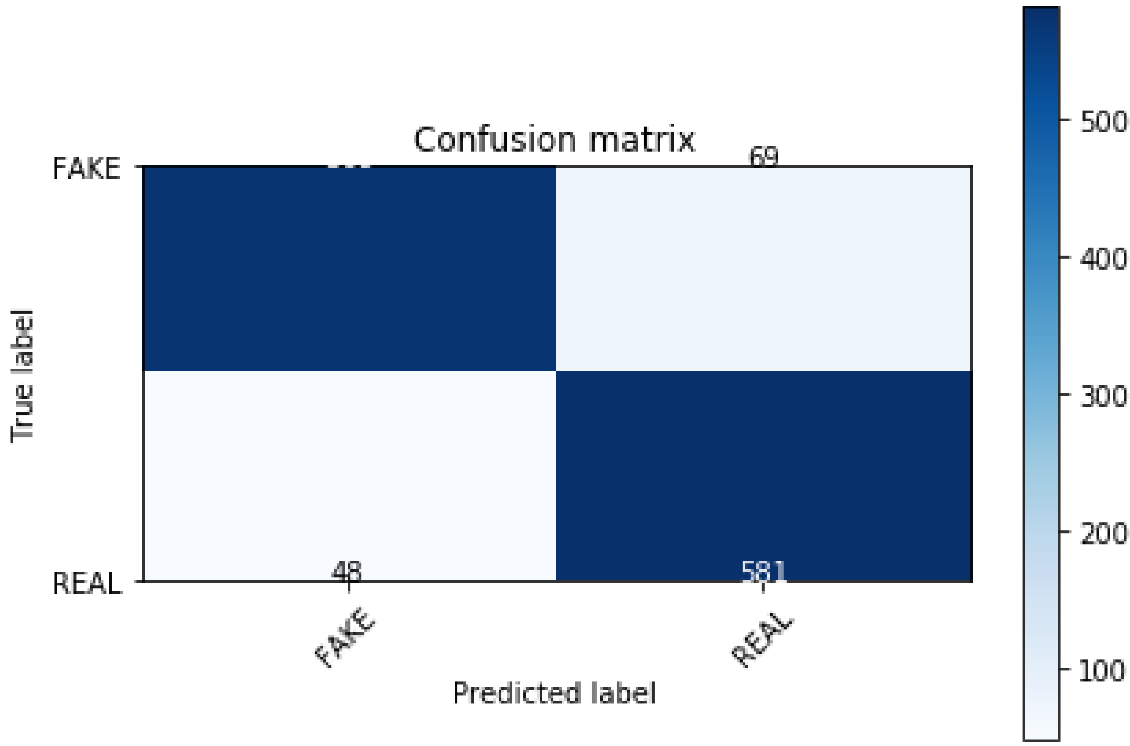

Figure 8: Confusion Matrix For PassiveAggressive Classifier for CountVectorizer

True Positives: 581
True Negatives: 569
False Positives: 69
False Negatives: 48
Accuracy: 0.908
Precision: 0.8938
Recall: 0.9237
F1 Score: 0.9085
Specificity: 0.8918

7.4 Compute Confusion Matrix PassiveAggressive Classifier for TfidfVectorizer

Confusion matrix, without normalization :

[[**589** **49**]
 [**43** **586**]]

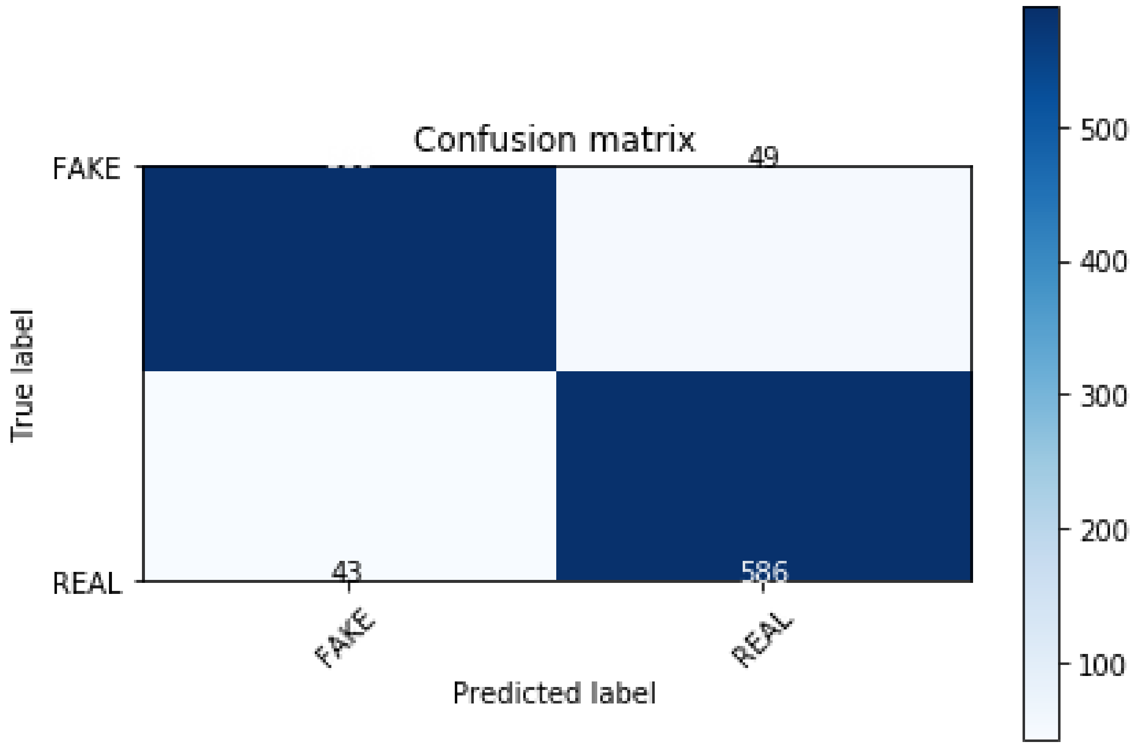

Figure 9: Confusion Matrix For PassiveAggressive Classifier for TfidfVectorizer

True Positives: 586
True Negatives: 589
False Positives: 49
False Negatives: 43
Accuracy: 0.927
Precision: 0.9228
Recall: 0.9316
F1 Score: 0.9272
Specificity: 0.9232

7.5 Summary Table

Table 3: DATA FOR VECTORIZERS UNDER CLASSIFIERS

CATAGORY	**(MNB)COUNT**	**(MNB)TFIDF**	**(PA)COUNT**	**(PA)TFIDF**
TRUE POSITIVE	583	597	581	**586**
TRUE NEGATIVE	552	549	569	**589**
FALSE POSITIVE	86	89	69	**49**
FALSE NEGATIVE	46	32	48	**43**
ACCURACY	0.896	0.904	0.908	**0.927**
PRECISION	0.8714	0.8703	0.8938	**0.9228**
RECALL	0.9269	0.9491	0.9237	**0.9316**
F1 SCORE	0.8983	0.9080	0.9085	**0.9272**
SPECIFICITY	0.8652	0.8605	0.8918	**0.9232**

***MNB=MULTINOMIAL NAIVE BAYES
**PA=PASSIVE AGGRESSIVE

7.6 Remark

From the result that I have got in the **Python Experiment**, it can be concluded that **TfidfVectorizer** under **PassiveAggressive Classifier** has performed better than all other models (the **other 3**—CountVectorizer and TfidfVectorizer under Multinomial Naive Bayes Classifier and CountVectorizer under PassiveAggressive Classifier) and has been marked **bold** in the above table. So, it can be concluded that for detecting fake news TfidfVectorizer under PassiveAggressive Classifier is very good as a classifier. The other 3 models have also performed very well, but TfidfVectorizer under PassiveAggressive Classifier is the best among all.
In case of detecting false news, **Precision** is of most importance as, **FALSE POSITIVE** is to be minimized. We can take any real news as fake one but it will be very harmful if we consider any fake news as real one as it can restless people and run havoc in the society. In fake news detection, we also want to have high **Recall**, TRUE POSITIVE RATE. As, **high Recall** will have **least FALSE NEGATIVE** and we will get most of the true news as true news not fake. And TfidfVectorizer under PassiveAggressive Classifier satisfies all the conditions. So, we will work on the **TfidfVectorizer** under **PassiveAggressive Classifier** and get other required results.

7.7 ROC Curve-AUC Score

The **ROC Curve** for all the fitted models have been drawn and from there We calculate the **AUC Score**.

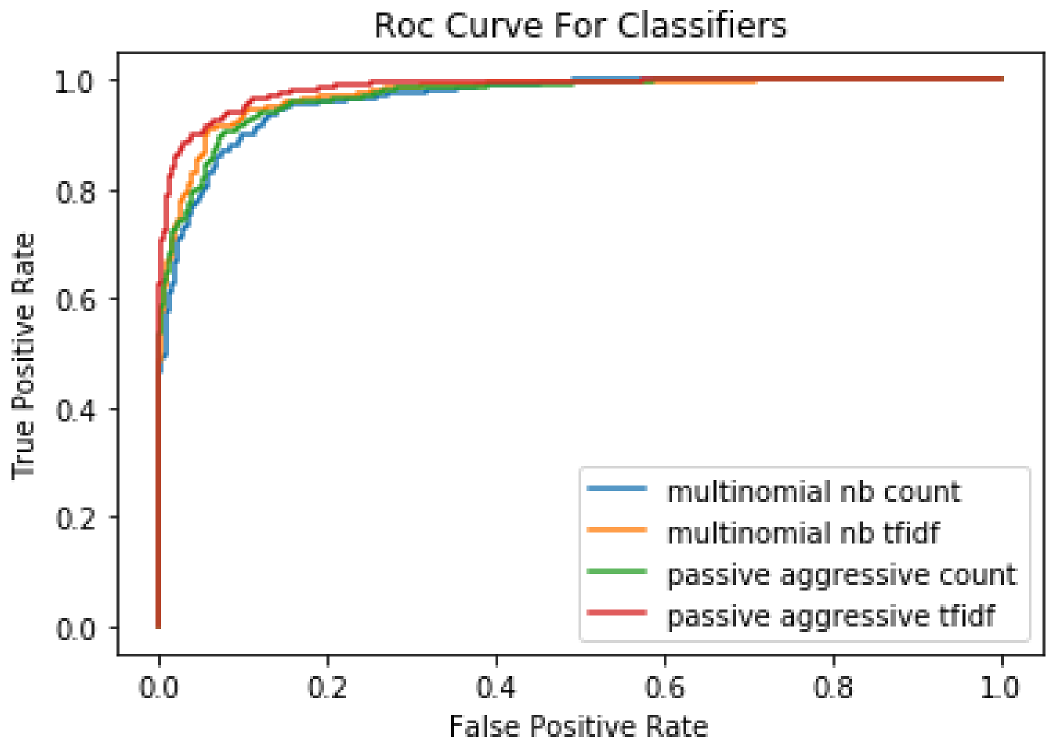

Figure 10: ROC Curve-AUC Score

multinomial nb count: ROC AUC =0.943
multinomial nb tfidf: ROC AUC =0.974
passive aggressive count: ROC AUC =0.965
passive aggressive tfidf: **ROC AUC =0.984**

From **ROC Curve-AOC Score**, it is also clear that **TfidfVectorizer** under **PassiveAggressive Classifier** has more area than any other has and it is almost **equal to 1**. So, it can be said that TfidfVectorizer under PassiveAggressive Classifier can distinguish the Real and Fake news properly and can alsoclassify them correctly.So it is best to use TfidfVectorizer under PassiveAggressive Classifier to classify Real and Fake News.ROC AUC Score of Passive Aggressive Tfidf has been marked **bold** above.

8 EXPERIMENTAL RESULTS

8.1 Top 10 FAKE and REAL News

I have calculated **Top 10 Fake** and **Real News** from the dataset using TfidfVectorizer under PassiveAggressive Classifier.

'FAKE': [(-5.75262481957928, '2016'),(-5.013395767858511, 'october'),(-3.3317257993521103, 'share'),(-3.18200894322618
'november'),(-3.1797842442085833, 'hillary'),(-3.166170786600595, 'election'),(-2.98624689630123, 'article'),(-2.76602634451195
'establishment'),(-2.622800113119332, 'source'),(-2.6008732183439367, 'advertisement')]

'REAL': [(2.2081239496198077, 'tuesday'),(2.2393404470471086, 'march'),(2.3409065304458325, 'rush'),(2.5316800802138
'candidates'),(2.774128444835958, 'friday'),(2.795466506249762, 'gop'),(2.859657787244346, 'conservative'),(3.03237366210724
'marriage'),(3.2988305717585407, 'says'),(4.885489173724793, 'said')]

The above is the list of words with **Classifier Coefficient** for top 10 FAKE and REAL News. It shows us that which word is classified in which catagory with a certain probability or likelihood.
So, we can say from now that when and where the words will occur, we can classify them as **FAKE** or **REAL** accordingly.

8.2 Most 20 REAL and FAKE News

I have also calculated **Most 20 Real** and **Fake News** from the dataset using TfidfVectorizer under PassiveAggressive Classifier.

*** **Most Real**:
[(-4.425109600312677, 'said'),(-4.522246319612897, 'trump'),(-4.913489679506659, 'clinton'),(-5.433999076102811,
'people'),(-5.43734250839567, 'state'),(-5.474468531540541, 'president'),(-5.518993714015171, 'obama'),(-5.522634286821445,
'new'),(-5.54871648515088, 'campaign'),(-5.6934753665840425, 'republican'),(-5.790413196932821, 'party'),(-5.9043444024079
'time'),(-5.910333199226768, 'like'),(-5.927230878590333, 'states'),(-5.938548337550461, 'sanders'),(-5.938873583113727,
'just'),(-6.0240501866528735, 'house'),(-6.045894498625599, 'percent'),(-6.0649002240255605, 'political'),(-6.121058523886313
'cruz')]

*** **Most Fake**:
[(-16.272226130581554, '000035'),(-16.272226130581554, '0001'),(-16.272226130581554, '0001pt'),(-16.272226130581554,
'0002'),(-16.272226130581554, '000billion'),(-16.272226130581554,'0011'),(-16.272226130581554,'004s'),(-16.272226130581554,
(-16.272226130581554,'005s'),(-16.272226130581554,'00684'),(-16.272226130581554,'006s'),(-16.272226130581554,'007'),
,(-16.272226130581554,'007s'),(-16.272226130581554,'008s'),(-16.272226130581554,'0099'),(-16.272226130581554,'00am'),
(-16.272226130581554,'00p'),(-16.272226130581554,'00pm'),(-16.272226130581554,'013c2812c9'),(-16.272226130581554,
'014')]

The above is the list of words for most 20 REAL and FAKE News with **Classifier Coefficient**. It gives us a certain probability or likelihood of classifiying a certain word as a **REAL** or **FALSE** news.
So, we can say from now that when and where the words will occur, we can classify them as **REAL** or **FAKE** accordingly.

8.3 Aggregated Rank of REAL and FAKE Dataset

I have also calculated **Aggregated Rank** of occurring of words combining all the model classifiers in **REAL** and **FAKE** news and ordered them in ascending order.
It depicts the picture of occurance of words when combined from all model classifiers and classify them as **REAL** and **FAKE** ones accordingly.

Table 4: TABLE FOR AGGREGATED RANK OF WORDS

WORDS	**LABEL**	**AGG RANK**	**COUNT**
said	**Real**	9.667	3
republican	Real	1.5	2
rush	Real	6.5	2
march	Real	5	2
gop	Real	6	2
friday	Real	5	2
candidates	Real	3.5	2
campaign	Real	3.5	2
marriage	Real	4.5	2
trump	Real	9.5	2
000035	**Fake**	1	2
005s	Fake	9	2
2016	Fake	4.5	2
share	Fake	4	2
0001	Fake	2	2
article	Fake	5	2
00684	Fake	10	2
october	Fake	3	2
005	Fake	8	2
000billion	Fake	5	2

9 CONCLUSION

In this project, we explore different aspects of incorporating neural, statistical and external features to deep neural networks on the task of fake news stance detection.We also presented in-depth analysis of several state-of-the-art recurrent and convolution architectures. The presented idea leverages features extracted using count vectorizer and tfidf vectorizer under multinomial nb classifier and passiveaggressive classifier. We found that **Tfidf Vectorizer** under **PassiveAggressive Classifier** performs the best in determinig **Fake News** and in clasfing them as **Real** and **Fake** one. Though there is still scope for further improvement in the topic and hope to be explored in future.

10 REFERENCES

10.1 Websites

1. **Google** 'www.google.com'
2. **Machine Learning Wikipedia** 'en.wikipedia.org/wiki/Machine learning'
3. **NLP for Machine Learning** 'towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d
4. **Fake News Detection as Python Project** 'data-flair.training/blogs/advanced-python-project-detecting-fake-news'
5. **Fake News Detection Using Machine Learning** 'www.pantechsolutions.net/fake-news-detection-using-machine-learni

10.2 Books

6. **Building Machine Learning Systems with Python** by Willi Richert,Luis Pedro Coelho (PACKT PUBLISHING;BIRMINGHAM - MUMBAI)

7. **Deep Learning with Applications Using Python Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras** by Navin Kumar Manaswi,Foreword by Tarry Singh (Apress)

8. **Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems** by Aurélien Géron (O'Reilly)