Fake News Detection

Abstract: The project will detect a fake news among many news. We used a dataset called 'news.csv' which contains 6335 news and 4 columns. We applied several classification algorithms like random forest, support vector machine, deep neural networks etc. We get highest 94.55% accuracy.

Motivation: With the increasing popularity of social media, people has changed the way they access news. News online has become the major source of information for people. However, much information appearing on the Internet is dubious and even intended to mislead. Some fake news are so similar to the real ones that it is difcult for human to identify them. Therefore, automated fake news detection tools like machine learning and deep learning models have become an essential requirement. In this paper, we evaluated the performance of several machine learning models and deep learning models on a fake and real news dataset.

Literature Review: Machine learning is now using in a many sectors of securities like intrusion detection system, spam email detection, anomaly detection, firewall, fake news detection, etc. **Abdullah** et. al [1] proposes a model for recognizing forged news messages from twitter posts, by figuring out how to anticipate precision appraisals, in view of computerizing forged news identification in Twitter datasets through machine learning and deep learning algorithms. After **McNemar's** test to determine if models' performance is significantly different, **JIANG** et. al[2] proposed a novel stacking model which achieved testing accuracy of 99.94% and 96.05 % respectively on the ISOT dataset and KDnugget dataset. **LI** et. al[3] proposed an unsupervised fake news detection method based on **autoencoder** (UFNDA). **Liao** et. al[4]proposed a novel fake news detection multi-task learning (FDML), which trains the fake news detection task and the news topic classification task simultaneously through multi-task learning, by leveraging the correlations among the news topics, the credibility distributions of authors, and the truthfulness of news at the same time. Moreover, we propose a novel news graph (N-Graph) method to obtain richer news representation via preserving relationships among news and design a simple dynamic weighting strategy to automatically balance the importance of multiple tasks to achieve better performance of both fake news detection task and news topic classification task at the same time.

Data: There are 4 columns. And among these, we find one column, text which is nothing but a news' body, is crucial and good enough to discriminate between fake and real news.

Data Preprocessing: We applied CountVectorizer and TF-IDF Vectorizer for the conversion from text to numerical data. And used both data transformation in each algorithm.

CountVectorizer: Countvectorizer is a method to convert text to numerical data. It is mainly used in Natural Language Processing to extract information from the documents. It is a simple method to convert a set of text documents to a matrix of token counts.

In this approach, each text document is represented as a vector of word occurrences, and the resulting matrix contains the frequency of each word in the text corpus. This method can be used to analyze the frequency of words in a document, or to compare the vocabulary used in multiple documents.

CountVectorizer works by tokenizing the text corpus and then creating a document-term matrix where the rows represent the documents in the corpus, and the columns represent the unique terms found in the corpus. The value at each cell in the matrix represents the number of times the corresponding term appears in the corresponding document. For the project, I set the range of n-gram to 1.

TF-IDF Vectorizer : TF-IDF Vectorizer (Term Frequency-Inverse Document Frequency Vectorizer) is a text feature extraction technique that aims to reflect how important a word is to a document in a text corpus. It is an extension of CountVectorizer, which counts the frequency of each word in a text corpus.

TF-IDF Vectorizer computes a numerical value for each word in a document based on the frequency of the word in the document and the inverse frequency of the word in the corpus. The idea behind this is that a word that appears frequently in a document but infrequently in the corpus is more important to that document than a word that appears frequently in both the document and the corpus.

The formula for calculating the TF-IDF value of a word is:

TF-IDF = (Term Frequency) x (Inverse Document Frequency)

where Term Frequency is the number of times a word appears in a document, and Inverse Document Frequency is a measure of how rare the word is in the corpus, defined as:

Inverse Document Frequency = log(N/df)

where N is the total number of documents in the corpus, and df is the number of documents in the corpus that contain the word.The resulting TF-IDF Vectorizer matrix is a weighted document-term matrix where the rows represent the documents in the corpus, and the columns represent the unique terms found in the corpus. The value at each cell in the matrix represents the TF-IDF value of the corresponding term in the corresponding document.

TF-IDF Vectorizer can be used to identify the most important words in a document, compare the vocabulary used in multiple documents, or perform text classification and clustering tasks.

Random Forest Classification: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. In my project, I used both 'gini' and 'entropy' based trees and used both data transformation in each classification.

Support Vector Machine: The objective of the support vector machine algorithm is to find a hyperplane in an N dimensional space(N — the number of features) that distinctly classifies the data points. I applied 'rbf' kernel and got a decent accuracy 94% in TF-IDF data.

Multilayer Perceptron Classification: Multilayer Perceptron falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer.

Deep Neural Network: Deep neural networks are a powerful category of machine learning algorithms implemented by stacking layers of neural networks along the depth and width of smaller architectures. We got highest accuracy(94.55%) applying DNN in TF-IDF data.

Results:

Naïve Bayes: Multinomial

CountVectorization						**TF-IDF**					
Accuracy		**88.40%**				**Accuracy**		**81.93%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	84	94	89			0	73	99	84		
1	93	84	88	6.5	16.4	1	98	66	79	1.3	33.9

Naïve Bayes: Gaussian

CountVectorization						**TF-IDF**					
Accuracy		**81.22%**				**Accuracy**		**80.19%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	76	89	82			0	77	85	81		
1	88	74	80	11.0	26.1	1	84	75	80	14.8	24.6

Random Forest Classification: Gini

CountVectorization						**TF-IDF**					
Accuracy		**89.27%**				**Accuracy**		**91.87%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	87	91	89			0	90	94	92		
1	91	87	89	8.6	12.7	1	94	90	92	5.8	10.3

Random Forest Classification: Entropy

CountVectorization						TF-IDF					
Accuracy		**89.58%**				**Accuracy**		**91.79%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	88	91	89			0	91	93	92		
1	91	89	90	9.4	11.4	1	93	91	92	7.3	9.1

Support Vector Machine: kernel='rbf', gamma=0.5, C=10

CountVectorization						TF-IDF					
Accuracy		54.14%				**Accuracy**		**94.40%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	100	6	11			0	95	93	94		
1	53	100	69	94.3	N/A	1	94	95	95	6.7	4.6

Multilayer Perceptron Classification: activation: relu ; solver: adam

CountVectorization						TF-IDF					
Accuracy		**94.16%**				**Accuracy**		**94.24%**			
	Precision %	Recall %	F1 %	F/P %	F/N %		Precision %	Recall %	F1 %	F/P %	F/N %
0	92	96	94			0	93	95	94		
1	96	92	94	3.7	7.8	1	95	94	94	5.2	6.3

Deep Neural Network:

TF-IDF					
Accuracy		**94.95%**			
	Precision %	Recall %	F1 %	F/P %	F/N %
0	95	95	95		
1	95	95	95	5.4	4.8

Conclusion: In most of the algorithms, TF-IDF transformed data are well suited in accuracy. And, after tuning the hyperparameters of the Random Forest, there was no significant change in accuracy. We got the highest accuracy in deep neural network, about 95%. So, in my opinion deep neural network is better in detecting fake news.

References:

[1]. Abdullah-All-Tanvir, Ehesas Mia Mahir, Saima Akhter, Mohammad Rezwanul Huq, " Detecting Fake News using Machine Learning and Deep Learning Algorithms", 2019.

[2]. TAO JIANG, JIAN PING LI, AMIN UL HAQ, ABDUS SABOOR, AND AMJAD ALI, " A Novel Stacking Approach for Accurate Detection of Fake News", 2021.

[3]. DUN LI, HAIMEI GUO , ZHENFEI WANG, AND ZHIYUN ZHENG, “Unsupervised Fake News Detection Based on Autoencoder", 2021.

[4]. Qing Liao, Heyan Chai, Hao Han, Xiang Zhang, Xuan Wang,Wen Xia,Ye Ding,"An Integrated Multi-Task Model for Fake News Detection" 2021.