

Fake News Detection

Weaam AlMutawwa

Department of Software Engineering

King Saud University

Supervised by: Dr. Sadeen AlHarbi

November 1, 2020

A project to complete the requirements of SWE485 (Selected Topics in Software Engineering)

©2020 Weaam AlMutawwa — 438201478

Contents

1	Introduction	1
2	Data Analysis	2
3	Machine Learning	8
3.1	Decision Tree	9
3.2	Logistic Regression	12
4	Conclusion	14

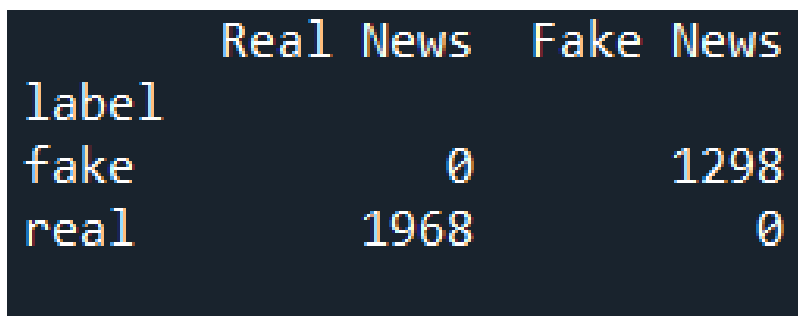
1. *Introduction*

Fake news can confuse many people in the area of politics, culture, healthcare, etc. Fake news refers to news containing misleading or fabricated contents that are actually groundless; they are intentionally exaggerated or provide false information. In this project, I will analyze 2 datasets of fake and real news headlines. Then, I will implement decision tree and logistic regression algorithms to predict whether a headline is real or fake and evaluate the performance of each, which will facilitate the detection of fake news.

2. *Data Analysis*

Dataset analysis approaches and technique differ based on the data type. Since my data type is text, I will perform text analysis. Text analysis is a machine learning technique that allows to automatically extract and classify text data. There are basic and more advanced text analysis techniques, each used for different purposes. The techniques which I found useful for my data are Word Frequency, which is a text analysis technique that measures the most frequently occurring words or concepts in a given text. Also, Text Classification, which is the process of assigning predefined tags or categories to unstructured text.

Firstly, I uploaded the datasets after adding a column name for each, Real News for real news and Fake News for fake news. Then, I used `head()` function, which returns by default the first 5 rows for the object based on position. It is useful for quickly testing if the object has the right type of data in it. After that, I used shape property to return a tuple representing the dimensionality, which result in (1,1968) for real news, and (1,1298) for fake news.



	Real News	Fake News
label		
fake	0	1298
real	1968	0

Figure 2.1: Shape property for fake and real news

Then, I added a label for each, fake for fake news and real for real news. Next, I contacted both, to help in plotting and analyzing the data in general. I plot a bar chart showing the count of real and fake news. It shows that the data are unbalanced, there is more real news than fake news.

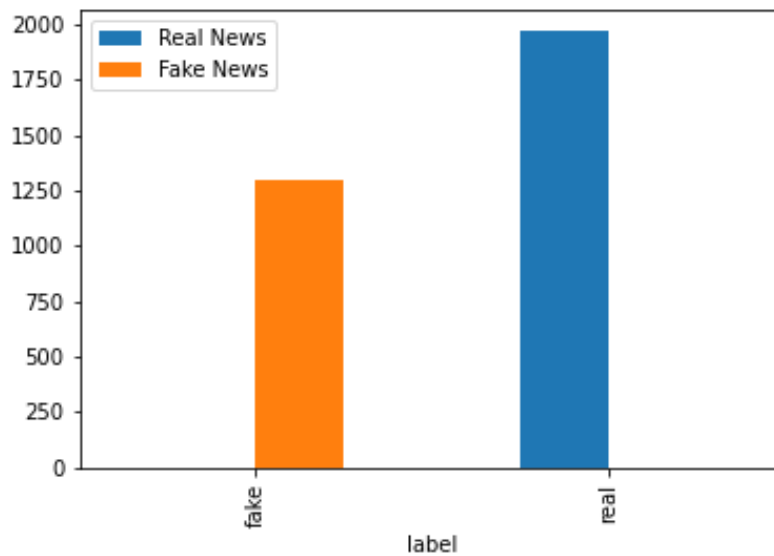


Figure 2.2: Count plot of fake and real news

Also, I used word cloud generator, which offer a basic text analysis technique to detect keyword frequency and extract phrases that often go together, to create a word cloud for real news and fake news separately and without stop words.

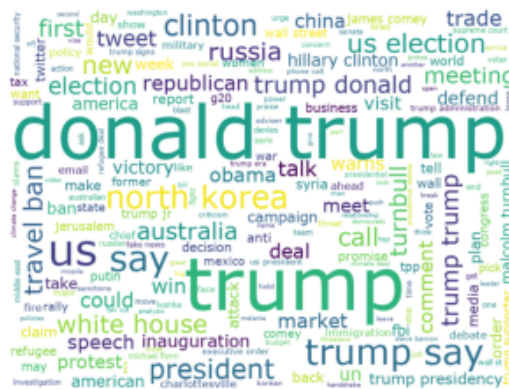


Figure 2.3: Word cloud for real news



Figure 2.4: Word cloud for fake news

Then, as requested, I wrote a function to calculate the most common 10 words in each, and then plotted them in a bar chart as well, without stop words.

```

77 # Function to analyze data by counting the most 10 frequent words
78 def wordCount(filename):
79     words = re.findall(r'\w+', open(filename).read().lower())
80     stop_words = stopwords.words('english')
81     wordsCleaned = [w for w in words if not w in stop_words]
82     l = Counter(wordsCleaned).most_common(10)
83     return l;

```

Figure 2.5: *wordCount(filename)* function to count the most frequent 10 words

```

Most frequent words in fake news are:
[('trump', 1328), ('donald', 228), ('hillary', 150), ('clinton', 132), ('election', 74), ('new', 67),
 ('president', 64), ('obama', 60), ('america', 54), ('win', 50)]
Most frequent words in real news are:
[('trump', 1744), ('donald', 829), ('us', 230), ('trumps', 219), ('says', 178), ('election', 87),
 ('north', 83), ('clinton', 83), ('korea', 79), ('ban', 75)]

```

Figure 2.6: Output of *wordCount(filename)* function

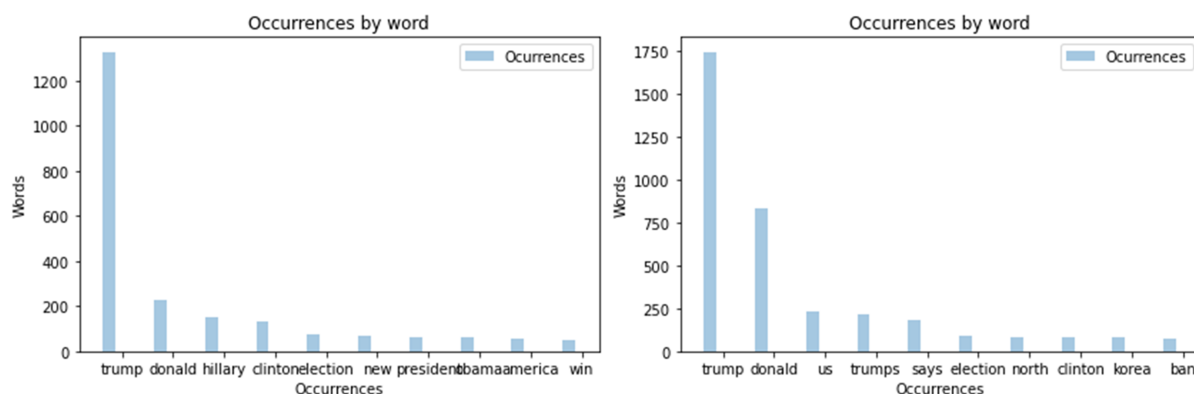


Figure 2.7: Frequency of common words in fake and real news

After analyzing data, I found out that these 3 specific keywords are useful:

1. Trump, in real news it appears 1744 times and 1328 in fake news.
2. Donald, in real news it appears 829 times and 228 in fake news.
3. Clinton, in real news it appears 83 times and 132 in fake news.

3. *Machine Learning*

Machine learning is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. In this part, I will report describe and report briefly the code I wrote and the execution of the classifiers I used, which are decision tree and logistic regression.

Firstly, I wrote *get_data(faketxt, realtxt)* which open the text files, read them and split data for the real and fake news separately, taking and return from each 70% for training, 15% for validation and 15% for testing as well as the total sets of each. I call it in main and then added a label for the real and fake news. After that, I wrote *get_features(faketxt, realtxt)* which open the text files and count the number of occurrence of each word, and the words which occurred 5 times or more are appended to the features array which the function will return, without considering stop words. Then, I wrote *make_matrix(fake, real)* which will concatenate the fake and real news for the training, validation and test sets and generate the label for each. After that, I wrote *convertNumeric(data)* which use CountVectorizer to convert a collection of text documents to a matrix of token counts to use in machine learning. Last but not least, I shuffled the data using *shuffle()* function which is implemented in sklearn library and used to reorganize the order of the items..

3.1 Decision Tree

Now, my data is ready to use by the classifiers. I wrote a function for each. Firstly, for the decision tree I wrote *select_modelDT(train, train_label, test, test_label, val, val_label, features)* function, I trained the model using 5 different values of max_depth and randomly chose different splitting criteria. The model performs the best when split criteria was entropy with max depth of 30. After testing, the accuracy became 59.9%.

Here are the results of all models, including the testing:

-----Decision Tree Training-----					4th Model Accuracy: 41.836735 %				
1st Model Accuracy: 60.204082 %					[[187 8]				
[[0 195]					[277 18]]				
[0 295]]					precision	recall	f1-score	support	
fake	0.00	0.00	0.00	195	fake	0.40	0.96	0.57	195
real	0.60	1.00	0.75	295	real	0.69	0.06	0.11	295
accuracy			0.60	490	accuracy			0.42	490
macro avg	0.30	0.50	0.38	490	macro avg	0.55	0.51	0.34	490
weighted avg	0.36	0.60	0.45	490	weighted avg	0.58	0.42	0.29	490
2nd Model Accuracy: 61.224490 %					5th Model Accuracy: 64.489796 %				
[[5 190]					[[29 166]				
[0 295]]					[8 287]]				
precision	recall	f1-score	support		precision	recall	f1-score	support	
fake	1.00	0.03	0.05	195	fake	0.78	0.15	0.25	195
real	0.61	1.00	0.76	295	real	0.63	0.97	0.77	295
accuracy			0.61	490	accuracy			0.64	490
macro avg	0.80	0.51	0.40	490	macro avg	0.71	0.56	0.51	490
weighted avg	0.76	0.61	0.48	490	weighted avg	0.69	0.64	0.56	490
3rd Model Accuracy: 63.061224 %					Testing Model Accuracy: 59.713701 %				
[[16 179]					[[15 179]				
[2 293]]					[18 277]]				
precision	recall	f1-score	support		precision	recall	f1-score	support	
fake	0.89	0.08	0.15	195	fake	0.45	0.08	0.13	194
real	0.62	0.99	0.76	295	real	0.61	0.94	0.74	295
accuracy			0.63	490	accuracy			0.60	489
macro avg	0.75	0.54	0.46	490	macro avg	0.53	0.51	0.43	489
weighted avg	0.73	0.63	0.52	490	weighted avg	0.55	0.60	0.50	489

Figure 3.1: Resulting accuracies and confusion matrix for decision tree

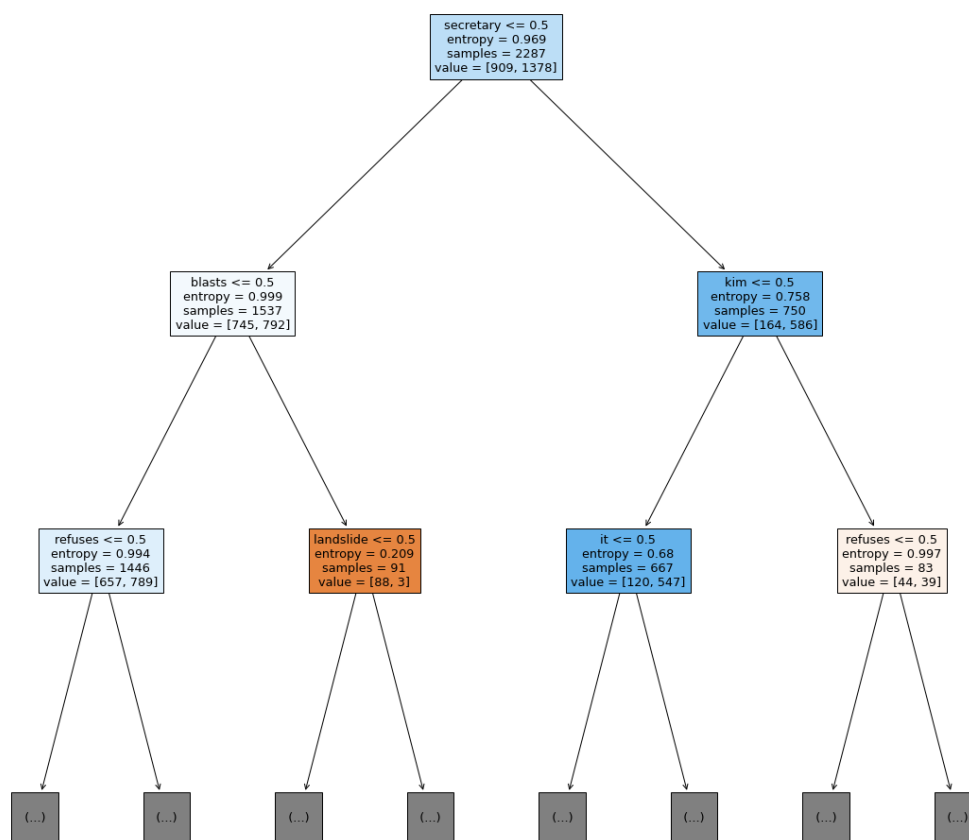


Figure 3.2: Visualization of the 2 layers of the decision tree.

3.2 Logistic Regression

Secondly, for the logistic regression I wrote `select_modelLR(train, train_label, test, test_label, val, val_label, features)` function, I trained the model using 5 different values of solver parameter which specifies the algorithm to use in the optimization problem. The model performs the best when solver algorithm was liblinear. After testing, the accuracy became 62.2%. Here are the results of all models, including the testing:

---Logistic Regression Training---					4th Model Accuracy: 65.306122 %				
1st Model Accuracy: 66.326531 %					[[40 155]				
[[50 145]					[15 280]]				
	precision	recall	f1-score	support		precision	recall	f1-score	support
fake	0.71	0.26	0.38	195	fake	0.73	0.21	0.32	195
real	0.65	0.93	0.77	295	real	0.64	0.95	0.77	295
accuracy			0.66	490	accuracy			0.65	490
macro avg	0.68	0.59	0.57	490	macro avg	0.69	0.58	0.54	490
weighted avg	0.68	0.66	0.61	490	weighted avg	0.68	0.65	0.59	490
2nd Model Accuracy: 62.040816 %					5th Model Accuracy: 64.693878 %				
[[93 102]					[[82 113]				
[84 211]]					[60 235]]				
	precision	recall	f1-score	support		precision	recall	f1-score	support
fake	0.53	0.48	0.50	195	fake	0.58	0.42	0.49	195
real	0.67	0.72	0.69	295	real	0.68	0.80	0.73	295
accuracy			0.62	490	accuracy			0.65	490
macro avg	0.60	0.60	0.60	490	macro avg	0.63	0.61	0.61	490
weighted avg	0.61	0.62	0.62	490	weighted avg	0.64	0.65	0.63	490
3rd Model Accuracy: 65.306122 %					Testing Model Accuracy: 62.167689 %				
[[40 155]					[[42 152]				
[15 280]]					[33 262]]				
	precision	recall	f1-score	support		precision	recall	f1-score	support
fake	0.73	0.21	0.32	195	fake	0.56	0.22	0.31	194
real	0.64	0.95	0.77	295	real	0.63	0.89	0.74	295
accuracy			0.65	490	accuracy			0.62	489
macro avg	0.69	0.58	0.54	490	macro avg	0.60	0.55	0.53	489
weighted avg	0.68	0.65	0.59	490	weighted avg	0.60	0.62	0.57	489

Figure 3.3: Resulting accuracies and confusion matrix for logistic regression

According to the results, logistic regression performed best. Decision tree gave the worst results when the splitting criteria was gini with max depth of 20. Also, decision tree overfit the most.

4. *Conclusion*

In the end, fake news detection is an important aspect of the uses of machine learning algorithms as spreading it has become a serious issue in the current social media world. In this project, I briefly analyzed a set of fake and real news headlines and trained a model using decision tree and logistic regression to distinguish between them which will facilitate the detection of fake news.

Bibliography

- [1] Geron, A., 2020. *Hands-On Machine Learning With Scikit-Learn And Tensorflow*. 1st ed. Sebastopol: O'Reilly Media Inc.
- [2] Kaggle.com. 2020. *Bag Of Words Meets Bags Of Popcorn* . [online]. Available at: <https://www.kaggle.com/c/word2vec-nlp-tutorial/overview/part-1-for-beginners-bag-of-words>. [Accessed 30 October 2020].
- [3] McKinney, W., 2017. *Python For Data Analysis*. 2nd ed. Sepastopol: O'Reily Media Inc.
- [4] Monkeylearn.com. 2020. *Text Analysis*. [online] Available at: <https://monkeylearn.com/text-analysis/>. [Accessed 31 October 2020].
- [5] Mithrakumar, M., 2020. *How To Tune A Decision Tree?*. [online] Medium. Available at: <https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680> [Accessed 31 October 2020].

[6] Pietro, M., 2020. *Text Analysis and Feature Engineering With NLP*. [online] Medium. Available at: <https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d> [Accessed 31 October 2020].

[7] Pandas.pydata.org. 2020. *Pandas Documentation* [online] Available at: <https://pandas.pydata.org/pandas-docs/stable/index.html> [Accessed 31 October 2020].

[8] Scikit-learn.org. 2020. *ScikitLearn: Machine Learning In Python*. [online] Available at: <https://scikit-learn.org/stable/index.html> [Accessed 31 October 2020].