FAKE NEWS CLASSIFIER

Laura Soto

University of Miami
lms338@miami.edu

ABSTRACT

This project aims to test different machine learning approaches on a news dataset containing both fake and real news. The objective of the classifiers is to be able to correctly classify a news as fake or real. In order to perform these tasks, a dataset was created by combining two datasets extracted from Kaggle. All the features of the dataset were analyzed and the best features to train and test the models were selected. After a process of combining, extracting, transforming, and cleaning the data four models were trained, and their accuracy was compared.

1. INTRODUCTION

The term fake news stands for a piece of false or misleading information that is presented in the form of news. A fake news is most likely created under the objective of damaging someone's reputation as well as an advertising tool to make money.

The thread that fake news bring is the fact that most people cannot identify the difference between fake and real news, and this can create biases, as well as misleading thoughts and actions. In fact, conciseness about fake news started in 2016 around the United States elections when afterward analysis has shown the huge influence that the spread of fake news on social media had on the results. What this example proves is how fake news in the hands of something powerful can influence major decisions.

Furthermore, ever since then, fake news about politics, climate change, economics, and other topics became more frequent on social platforms. At the same time, the need of creating models to identify them became a necessity. Indeed, identifying fake news became an important task for all major social media platforms, as well as for the users. Furthermore, big technology companies like Facebook, Google, or Twitter, have tried to filter fake news or add a fake news label to it.

There are many ways to identify a fake news. From a user point of view, the recommended way is to search the source to establish how reliable it is. On the other hand, this solution can be tedious and not very effective. Therefore, this searching process can be automated by creating a successful model to classify a news as fake or real based on collected data. Indeed, involving machine learning models to identify similarities between each class and being able to successfully label them.

2. THE DATASET

In order to be able to create a successful algorithm to classify fake and real news, two datasets were retrieved from Kaggle. The first dataset only contained real news, and the second one only contained fake news. The real news dataset contained a total of 20826 instances, while the fake news dataset contained a total of 17903 instances. Each data set had the same number of features, which are title, text, subject, and date.

The first step in this project consisted of merging both datasets and adding an additional feature, which corresponded to the class label. As a result of this merging, the final dataset contained a total of 38729 unique instances.

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

Fig. 1. Datasets extracted from Kaggle

	title	text	subject	Class
0	budget fight looms u republicans flip fiscal s...	government shutdown defense " discretionary " ...	politicsNews	Real
1	pentagon monday u military accept transgender ...	federal law banning gender protect transgender...	politicsNews	Real
2	senior u job republican senator let mr mueller	seemed election trump administration allies ja...	politicsNews	Real
3	fbi russia probe helped nyt australian diploma...	conversation election special counsel led papa...	politicsNews	Real
4	charge much trump wants postal service amazon ...	46 percent stock prices n), postal service de...	politicsNews	Real
...	...	...	...	...
44893	john mccain furious iran treated us sailors we...	iran quickly defuse far appreciation world dar...	Middle-east	Fake
44894	users action mail privacy class 0 justice yaho...	advertising purposes billion dollar company fa...	Middle-east	Fake
44895	us take territorial booty northern syria sunni...	seemed ambitions ambassador ask remain islamic...	Middle-east	Fake
44896	blow al jazeera america finally calls 700 mill...	qatar owned al jazeera purchased current tv te...	Middle-east	Fake
44897	10 u iranian military – signs neocon political...	iran neocon stuntalready far election official...	Middle-east	Fake

44898 rows × 4 columns

Fig. 2. Final Dataset after performing some text transformations

3. DATASET ANALYSIS

Once the dataset for this project was created, the first step before starting to apply the machine learning models consisted of analyzing the data.

The importance of analyzing the data, before applying any model, is that it helps to understand better what components to consider, as well as how to avoid overfitting and common errors.

The first step is to see if there any null values and to make sure we didn't have any duplicates. By using Python and the Pandas library, performing these two actions was very simple. The dataset did not have any null values, and a few duplicates were removed. In order to understand the dataset, each feature was analyzed.

3.1. Date

The date feature represents the date that the news was published. Based on domain knowledge, since the project aims to perform natural language processing to classify news, this feature was not considered to bring any needed and influential data. Therefore, the date feature was not considered and was removed from the dataset.

3.2. Class Label

Since our data was divided into two classes, analyzing the distribution of each class was important. This distribution was observed by creating a plot using the Matplot library in Python. Moreover, the final dataset was formed by 52% of fake news, and 48% of real news, which is considered to be a good class distribution.

Fig. 3. Pie chart representing the class distribution

3.3. Subject

The subject feature represented the type of news. That means if it is a piece of political news, economic news, or other categories. On an initial belief, some types of news are more likely to be fake than real. Indeed, the initial theory was that political news had a higher probability of being fake.

In order to prove or contradict these initial beliefs, the subject types were analyzed though plots. First, the distribution of the subject on the overall data can be observed as follows:

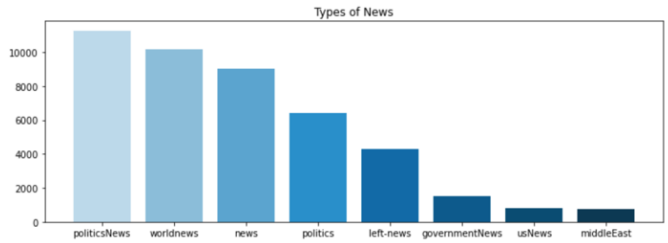

Fig. 4. Subject type distribution

The impression from this initial distribution is that there is a good distribution among the type of news. Moreover, this dataset contains eight different types of news which are political news, world news, news, politics, left news, government news, U.S news, and middle east news.

On the other hand, when analyzing the distribution of these eight subject types in each class we observed the following:

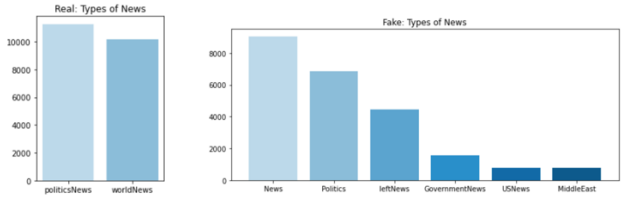

Fig. 5. Subject type distribution on the real and the fake news

As it can be observed in the above plots, the subject is a biasing feature. Indeed, both classes do not share subjects, and therefore if only this feature is used when training and testing the model, 100% accuracy will be achieved, but these models will be overfitted. Therefore, eliminating the subject feature will avoid having an overfitted algorithm.

3.4. Title and Text

Since both the title and the text feature were very similar, they were analyzed together. First of all, since this project wants to perform NLP both features presented the most promising results. Since both the title and the text are a combination of words, cleaning these words was needed. Indeed, transforming the title and text into only keywords had to be the first step before analyzing them.

Since this dataset contained almost 40 thousand instances, the computational time of cleaning and transforming this data if apply a simple loop function, was very high. In order to reduce the computational time,

PySpark was used. By using PySpark, the dataset was transformed into an RDD, which allowed the application of the transformation function using parallel computing. As a result of using PySpark to perform this transformation, only a few seconds were needed to have a clean dataset.

The transformation function applied to both features had different steps. The first step was to remove all the strings that are not in the alphabet, so symbols and numbers. The second step consisted of transforming all the characters to a lower case. After these, all the non-English words were removed, using the nlk library. Finally, using the Rake nlk library all the non-keywords were filtered out. As a result of this transformation function, only English keywords were left on each of these features. The transform RDD was saved as a CSV file.

Furthermore, once this final CSV file was created, and contained only the desired information, it was ready to be used to perform some data analysis on the title and text features.

In order to analyze both features, which contained only keywords, the WordCloud library was used. By creating a WordCloud graph, the most frequent words can be easily identified. Indeed, the size of a word in the WordCloud represents the frequency, therefore, the biggest the font size the most frequent. Four WordCloud graphs were created and were divided into fake, real, and the feature.

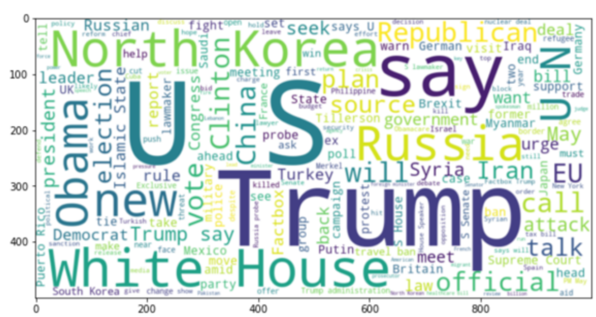

Fig. 6. WordCloud for title feature on real news

Fig. 7. WordCloud for title feature on fake news

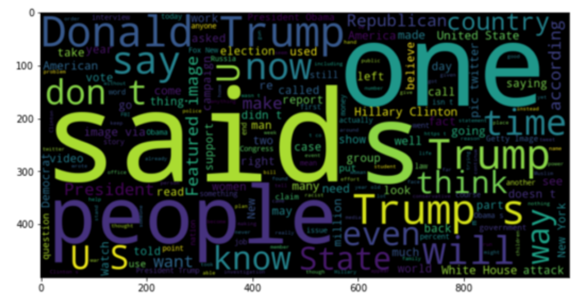

Fig. 8. WordCloud for text feature on real news

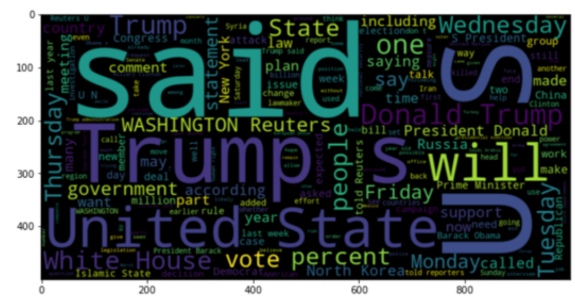

Fig. 9. WordCloud for text feature on fake news

There are different conclusions that can be made by observing the generated WordClouds. First, when observing and comparing the WordClouds generated using the feature text for real and fake classes, we can realize that there are many similarities in the most used words. Indeed, it is hard to separate them. Moreover, each text contains an average of 500 keywords, which can negatively impact the computational time of the models. Therefore, using this feature may not be the best option.

On the other hand, when observing and comparing the WordClouds generated by the feature title on fake and real classes, we can quickly observe that the most frequent words differ. Indeed, the words “Video” or “Tweet” are some of the most frequent words used on a fake news title, while they are barely used on a real news title. What we can conclude with this simple and initial analysis, is that the feature title is the most promising when finding differences between the two classes. Moreover, the average number of keywords that a title contains is eight.

3.5. Results

The result of this initial data and feature analysis revealed that the most promising feature to use to train and test the models is the title. Indeed, it is the only feature that clearly suggested a difference between the classes and does not seems to have the potential to create overfitting models.

4. THE ALGORITHMS

There are many different potential machine learning models that can be used to create a successful real and fake news classifier. Indeed, four different models with different approaches and characteristics were created based on the following machine learning techniques Naïve Bayes, Linear Support Vector Machine, K-Nearest Neighbors, and Decision Tree. All four models were trained and tested on the same data, with a split of 70 - 30, and their accuracy was compared.

Before training and testing the algorithms, the data had to be encoded. Indeed, the title feature was transformed into a float matrix, where each keyword had a unique value. This had to be done since most models in python can not be trained on string values. After encoding the data, the different machine learning techniques were performed.

4.1. Naïve Bayes

Naïve Bayes Classifier is a supervised learning algorithm that applies Baye's theorem. It consists of creating a probabilistic classifier which is created by the features having the assumption that there is conditional independence between each feature. This model was selected since it has been proved to work quite well in real-world situations like spam filtering, which is similar to fake news filtering. Furthermore, other advantages of the Naïve Bayes algorithm is that it can perform pretty fast when compared to other approaches, it is most likely to not present problems from the curse of dimensionality, and it does not require a huge amount of training data.

For the purpose of the project, a Multinomial Naïve Bayes algorithm was used. Multinomial Naïve Bayes is usually used in text classification, having the data represented as a tf-idf vector. This algorithm was used by importing the Sklearn Naïve Bayes library.

The model was trained on 70% of the total dataset, and it was tested on 30% of the total dataset. The results of this model are the following:

Naive Bayes

	precision	recall	f1-score	support
Fake	0.93	0.96	0.94	6741
Real	0.95	0.93	0.94	6539
accuracy			0.94	13280
macro avg	0.94	0.94	0.94	13280
weighted avg	0.94	0.94	0.94	13280

Accuracy: 0.94

Fig. 10. Results obtain from Naïve Bayes Model

As it can be observed from the results above, a Multinomial Naïve Bayes model showed to be useful and successful when classifying the given dataset. Indeed, an accuracy of 94% was obtained.

4.2. Linear Support Vector Machine

Support Vector Machine is a type of supervised learning model. This model is associated with learning algorithms where data is analyzed from classification and regression analysis. Indeed, the SVM training algorithm builds a model making a probabilistic binary linear classifier. This means that it maps each training example to a point in space, and it assigns a region to each class. Furthermore, when testing and classifying new instances, it does it considering where they are mapped within the established regions.

The model was created by importing the Sklearn SVM library and the results are the following:

Linear Suport Vector Machine

	precision	recall	f1-score	support
Fake	0.95	0.94	0.95	6741
Real	0.94	0.95	0.95	6539
accuracy			0.95	13280
macro avg	0.95	0.95	0.95	13280
weighted avg	0.95	0.95	0.95	13280

Accuracy: 0.95

Fig. 11. Results obtain from Linear Support Vector Machine Model

This model has an accuracy of 95% when tested on the testing set. This is considered to be a very good accuracy, and therefore this model is successful to classify fake and real news.

4.3. K-Nearest Neighbors

The K-Nearest Neighbors algorithm is a non-parametric classification method. It is used for supervised learning classification problems. The main concept behind this method is that given an unlabeled sample, the k-closest training examples from the training set need to be identified, and the unlabeled example will be classified depending on the class label of the k-nearest neighbors. Therefore, a given sample is classified depending on the most common class among its k nearest neighbors.

The first decision when creating a K-NN model is to determine the value for k, therefore, to determined how many neighbors to considered when classifying.

In order to determine the best value for k, a function was created. The goal of this function was to try different k values from 1 to 500, store the results and finally display the k value for which the accuracy was the highest. After performing this function, the best value for k was established to be equal to 291.

The highest accuracy is obtain when using K = 291

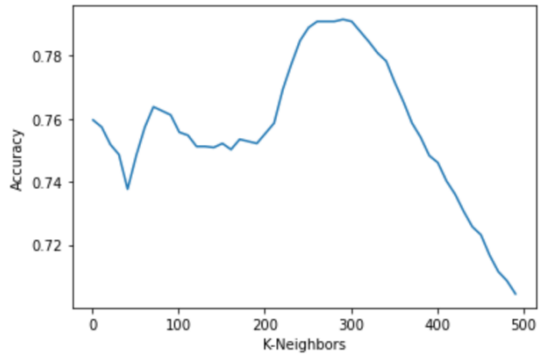

Fig. 12. Graph representing the accuracy values on different k values

After establishing the best value for k, the model was created, trained, and tested. In order to do so, the Sklearn KNN library was imported.

K-Nearest Neighbors

	precision	recall	f1-score	support
Fake	0.85	0.73	0.79	6741
Real	0.76	0.87	0.81	6539
accuracy			0.80	13280
macro avg	0.81	0.80	0.80	13280
weighted avg	0.81	0.80	0.80	13280

Accuracy: 0.8

Fig. 13. Results obtain from K-Nearest Neighbors Model

The accuracy obtained with this model is equal to 80%. This percentage is still high but as we observed with the other models it could be better. Therefore, we could conclude that a K-NN model might not be the best approach when classifying fake and real news.

4.4. Decision Tree

A Decision tree learning algorithm is a predictive model used in data mining. This approach uses a decision tree to go from the observations/conditions to the conclusions. A decision tree classifier has the goal of predicting the class label, which is represented in the leaves, and the branches represent the conjunctions of conditions that lead to that conclusion. Moreover, a decision tree can be used to visually represent the decision-making process.

The model was created using the Sklearn Decision Tree Classifier library and the results can be observed as following.

Decision Tree

	precision	recall	f1-score	support
Fake	0.99	0.48	0.64	6741
Real	0.65	1.00	0.79	6539
accuracy			0.73	13280
macro avg	0.82	0.74	0.71	13280
weighted avg	0.82	0.73	0.71	13280

Accuracy: 0.73

Fig. 14. Results obtain from Decision Tree Classifier Model

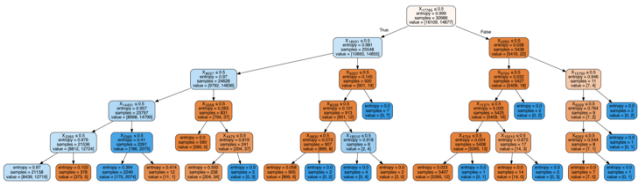

Fig. 15. Decision Tree created from the model

As observed, this model created a visual decision tree that represents the decision-making process. This model was able to reach an accuracy of 73% which is not too bad, but it is definitely not ideal. Therefore, we could conclude that it is not the best model to classify this dataset.

5. CONCLUSION

In conclusion, the project was successfully able to test different machine learning algorithms on the same dataset and was able to compare their accuracy. Overall, all models presented an accuracy greater than 70%, which can be considered very positive. On the other hand, two models performed better than the others achieving an accuracy equal to 94% and 95%. From these results, we can conclude that using a Support Vector Machine algorithm, or a Multinomial Naïve Bayes algorithm, will provide the best accuracy and the most successful results.

6. REFERENCES

[1] Bisaillon C. Fake and real news dataset. Accessed May 11,2021. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[2] 1.9. Naive Bayes — scikit-learn 0.24.2 documentation. Scikit-learn.org. Accessed May 11, 2021. https://scikit-learn.org/stable/modules/naive_bayes.html

[3] sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.24.2 documentation. Scikit-learn.org. Accessed May 11, 2021. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[4] 1.4. Support Vector Machines — scikit-learn 0.24.2 documentation. Scikit-learn.org. Accessed May 11, 2021. https://scikit-learn.org/stable/modules/svm.html

[5] sklearn.tree.DecisionTreeClassifier — scikit-learn 0.24.2 documentation. Scikit-learn.org. Accessed May 11, 2021. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[6] https://github.com/Laurasoto98/FakeNewsClassifier