Fake news detection, methods and data processing

Soham Petkar

10/15/2023

Abstract

In this project we studied several models to discern fake news articles from real news articles to find the best method. Fake news has become very prevalent in the digital age. There are thousands of articles that were written with the intent to mislead in various subjects: politics, social issues, and conspiracies. Leveraging two datasets that have true news articles and fake news articles we used 4 different machine learning models and compared them to each other to find the best method. We looked at the accuracy, the features, and the percentage of false positives the models gave. We found that the best model out of the four that we used was the custom neural network which got an accuracy of 95.7%.

1. Introduction

In the digital age the threat of fake news has become more and more prevalent. Misinformation has become very common in politics, social issues, conspiracies, and many more [1]. Fake news can lead people to have a distrust in the media and can promote the spread of false information. The goal for this project is to use machine learning to find features to identify these fake news articles.

There were two datasets used for this project [2]. The first one was filled with various fake news articles, and the other dataset was filled with various true news articles. These datasets will then be combined and stripped down to just the important words in their titles and split into training and testing data.

The problem with the traditional method of trying to verify the information from multiple sources is that the vast amount of content on the internet makes it difficult for normal users to cross reference articles themselves because it would be difficult to verify the authenticity of those articles. Machine learning can overcome these issues by already knowing what to look for. Based on the data that the model was trained on it can look at those articles and make a prediction based on that which will be just as accurate and faster.

2. Background

There were four models used for this project. The logistic regression model, then K-nearest neighbors (kNN), a multi layer perceptron (MLP), and finally a custom NN model. The logistic regression model takes in the input and turns it into a number which is the probability between 0 and 1. if it was a 0 then the output would be false and if it was 1 then it would say true.

The logistic regression model is used for classification and works by creating a decision boundary on a graph and classifies the input based on the location compared to the line.

Figure 1: This shows a visualization of what the model is looking at for a logistic regression model

The kNN model works by looking at similar words in a "bag of words" and compares it to any resemblance to nearby examples or since we are using words it would look at the nearby features. A bag of words is just the text converted into numerical representation which is useful for simplification and comparing frequencies. It then decides to put the word in the category by looking at the neighboring points which can be varied by changing K.

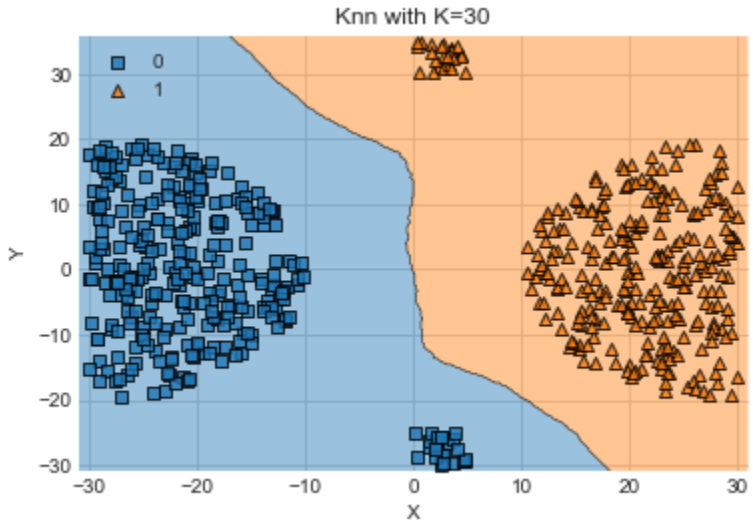

Figure 2: This shows a visualization of what the model is looking at for the kNN model

The MLP model is a network of nodes layered up all working together. The number of layers and their sizes can be adjusted to enhance accuracy according to specific requirements. These layers work together to calculate weights for the inputs which is then passed through the activation function to get the classification. The final model that was tested was the custom model. The custom NN model processes the input step by step and updates itself to minimize loss.

The optimizer used was Adam. The name is derived from adaptive moment estimation and is called Adam because it uses estimations of the first and second of the gradient to change the learning rate for each weight of the neural network. Because Adam adjusts the learning rate (a parameter that determines the step size) during training it is able to train faster. The difference

between Adam and gradient descent is that there is a lack of adaptivity with gradient descent and it can have slower convergence than Adam.

3. Dataset

There were two datasets used for this[2]. The first one was filled with 23,481 fake articles with their titles, the test, the subject, and the date it was published. The second was filled with 21,417 true articles with their titles, the text, the subject, and the date it was published(Refer to Table 1). These two datasets were then compiled into a new dataset with only their titles and a label stating if they were true news or fake news. In order to be able to use that new dataset we would have to get only the features out of it which are the individual words in this scenario. They are essential for building these models as they serve as inputs for the models to make predictions. First the dataset was tokenized which meant that each word inside of the title was split into its own separate words so it is easier for the processing that follows. Next was to get rid of the stop words and all of the punctuation marks. The reason that we do that is because stop words(and, but, this, etc) and punctuation marks carry little meaning and no significance to the message of the article so it would be best to remove them in order to cut down on the total number of features. Finally the last step was to transform the data into a bag of words. Once that is finished then the dataset is ready and we can split it into the training data and the testing data. 80% of the dataset was used for training while the remaining 20% was used for testing the accuracy.

‘features”	“description”
“titles”	The title of the article
“text”	The text of the article
“subject”	The subject of the article
“date”	The date at which the article was posted

Table 1: This shows all of the columns that were included in the data set and a brief description of the purpose for each one.

4. Methodology / Models

The first method used was a logistic regression model. That model worked well with 95.6% accuracy. This method uses weights on the words that are adjusted during the training process. Then the model makes a linear combination which is the weighted sum from the bag of words vector element and the bias term. This then makes a line and if the output is determined based if it falls above or below the drawn line.

The next method tried was K nearest neighbors. Varying the amount of neighbors used we found that with 1 neighbor gave an accuracy of 88.5% which means that the model was able to correctly predict if an article was real or fake 88.5% of the time.

num neighbors	accuracy
1	0.8849665924
2	0.84922049
3	0.8819599109
4	0.8646993318
5	0.884298441
6	0.8667037862
7	0.8815144766
8	0.8648106904
9	0.8775055679
10	0.8582405345

Table 2: This shows the number of neighbors and the accuracy that they achieved

In order to get this accuracy the model uses the bag of words, which is all of the features in the dataset transformed into vectors, and calculates the similarity between each word and the words that are near it. The main hyperparameter in the K nearest neighbors model in K which represents how many neighbors the model considers when calculating the distance. If your K is larger then the model is more robust to local variations or less likely to be affected by individual data points. The opposite is true for a smaller K. The smaller it is then there is more of a bias towards local patterns and it is more sensitive to local changes in the data.

Next we tried a MLP classifier which got an accuracy of 95.6% which was the same accuracy as the logistic regression model. Through testing multiple amounts of layers and the sizes of the layers we found that the hidden layer sizes that had the highest accuracy were 3, 4, and 6. The features are inputted into the first layer and given a weight that corresponds with the importance of each feature. The weights are then adjusted during the training of the model.

The final one that was tried was a custom NN model. The model had the same structure as the MLP classifier with 3 hidden layers with sizes of 3, 4, and 6. The activation function used was sigmoid which is primarily used for predicting probabilities. With 3 epochs this got an accuracy of 95.7%. The input layer had a shape of 800 because there were 800 features and the sizes of the layers was the same as the MLP. We only did 3 epochs because after that the model began to overfit to the training data.

5. Results and Discussion

From the four different models that were tested, logistic regression, kNN, MLP classifier, and a custom NN model, the one that had the best performance was the custom NN model with 96.2% accuracy.

From the logistic regression model we extracted the features that had the highest weight for fake and true news articles. The top 5 features that had the most weight for fake news articles were, VIDEO, obama, TRUMP, ATTACK, and TWEETS. The top 5 features that had the most weight for true news articles were, factbox, presidency, minister, bid, trade, and budget. Fake

news articles try to get an emotional reaction from the reader by using words like “ATTACK” in all caps and controversial political figures like Obama and Trump. However, 68% of the articles that were fake news related to news or politics, this may have an influence on the political names relating to fake news.

We varied the numbers of neighbors for kNN and the one with the best performance was 1 neighbor[Figure 1]. The number of neighbors tested was 1-15,20 and 30. The even number of neighbors always did worse than their neighboring numbers and as the number of neighbors the accuracy decreased with a slope of -0.002. After we got the accuracy that worked the best we plotted it onto a confusion matrix to see if there were an unusual amount of false positives or negatives. We found that the majority of the mistakes were from mislabeling fake news as true. 15.7% of the fake news was predicted as true from the kNN model compared to the 8.1% true news that was labeled as fake news.

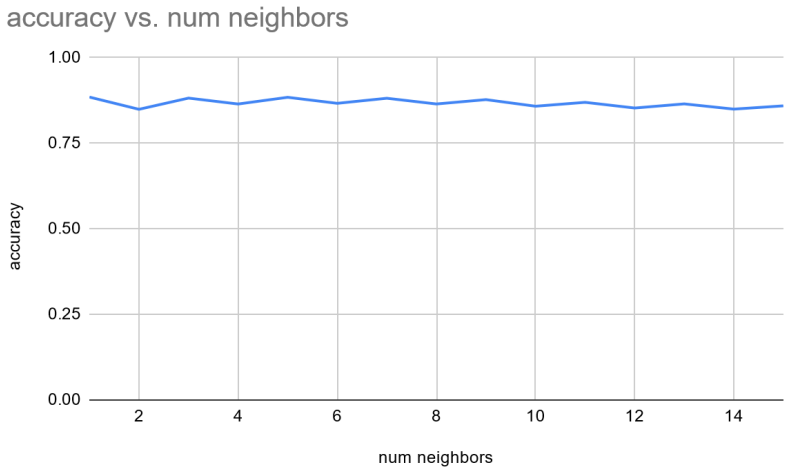

Figure 3: This shows the relationship between the accuracy and the number of neighbors that was used to train the model

To find the best accuracy for the mlp classifier I iteratively went through each hidden layer size to find the one with the highest accuracy. Almost every combination was +-2% within 95% besides the few combinations in which the model was guessing. The combination that had the highest accuracy was 3, 4 and 6 which got an accuracy of 95.6%.

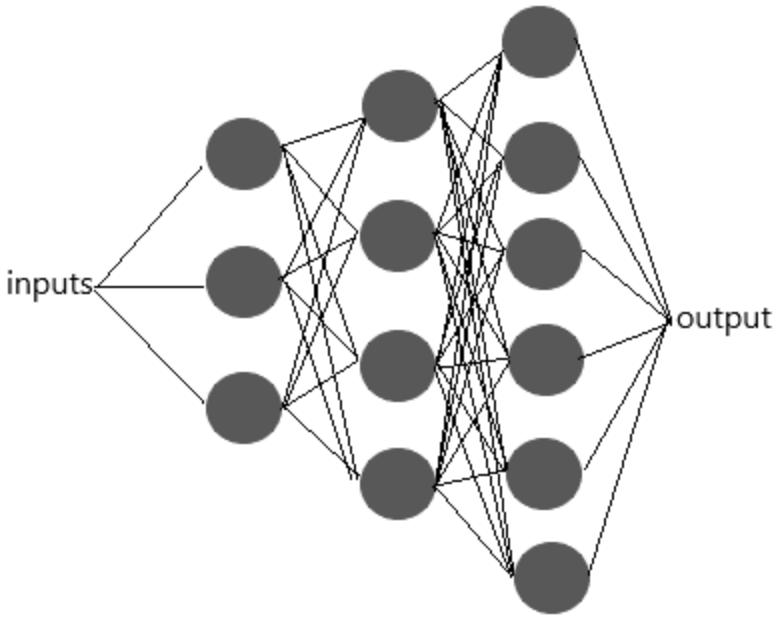

Figure 4: This is a representation of what the neural network would look like. It shows the 3,4,6 neurons in their different layers.

We plotted a confusion matrix for this as well and it performed better than the kNN model in every category. This had 5.3% mislabeling the fake news as true and 4.0% true news labeled as fake news.

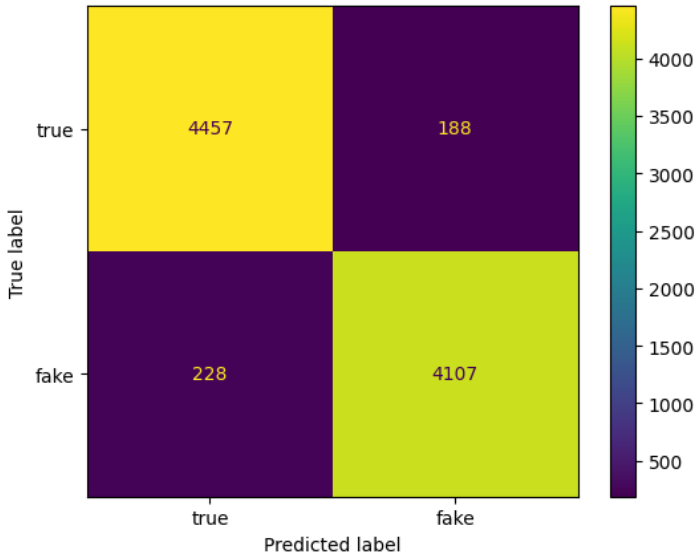

Figure 5: This shows the frequency that the model was making an incorrect prediction and the frequency of the model making a correct prediction

The best model out of all of them was the custom NN model: 95.7%. The dense layer sizes used were the same used for the MLP because that was the one that worked the best. The activation function used was sigmoid and the optimizer was Adam. After 2 epochs the model

began to overfit to the training data which means that model learns the training data too well to the point that it cannot perform well on something that it has not seen before.

6. Conclusions

The goal for the project was to find the best model to flag fake news articles. Based on out experiments, we found that the best model for identifying fake news articles is the custom NN model. We consider this to be the best because it got the highest accuracy out of all the models that we tested. There were 4 models used and ordering them from the best performance to worst performance goes: the custom NN model, MLP classifier, logistic regression, and the kNN model.

Looking at the top features that showed an association with fake news articles we learned that fake news articles usually center around controversial topics and figures. They try to get emotional reactions by using bait words like “ATTACK” and “TRUMP”.

References

[1] Soroush Vosoughi and Deb Roy and Sinan Aral, 2018, “The spread of true and false news online”

[2] Clément Bisaillon, 2019, ”Fake and real news dataset”