Comparing Logistic Regression and Decision Trees for Binary Classification of News Article

Introduction

With the rise of social media came an increase in fake news. Validating and classifying this fake news can be tedious and require large amount of background research. However, with the help of machine learning, fake news articles can be flagged for further verification. This can help distinguish conspiracy from fact and help in keeping people informed.

This report compares two different machine learning methods for binary classification of new articles from true (1) and fake (0). The Two different methods are discussed in more detail in methods section and the two models are compared in results section. The code used for pre-precession and training the methods can be found in appendix.

Problem Formulation

The data points are news articles. The dataset can be found on Kaggle. The dataset includes 17903 unique datapoints labelled fake and 20826 unique datapoints labelled genuine(true). The features are the title and sentence or phrase from the news. The label is either 1 or 0 depending if the news article is either true (1) or false (0). The textual features need to be cleaned, stemmed and vectorised so that it is easier to apply machine learning methods.

Methods

The dataset used in this project is a popular dataset frequently used to train machine learning to detect fake news. The dataset can be found as two CSV files, one for fake news and one for genuine news. The first pre-processing step was to label these files with 0 for fake and 1 for true, and merge them into one. Pandas library is used for this and by using “concat” method the two files were merged.

The next step was to get the X (features) and y (label) from the data set. The title and text both contain information whether the article is false or real. Hence, the column’s title and text were both merged with a space between them. Other unrelated columns such as date and subject were removed from the data-frame. After these steps there were 44898 unique data points.

The next step is to Tokenize the string and Vectorize. During Tokenization texts are broken down into smaller pieces. For example, “This is an ml project. Python is being used…” is converted into “ ’This is an ml project’, ‘Python is being used…’ ”. There are several Vectorizers, such as Count Vectorizer, Hash Vectorizer and TF-IDF Vectorizer. TF-IDF Vectorizer was chosen as it can more

accurately vectorize the text by using the frequency of a token in the text and its reoccurrence in the whole corpora. TF stands for term-frequency and IDF stands for inverse document frequency. This method can be found from “sklearn.feature_extraction.text” library. Once all the pre-processing is applied the result is a sparce matrix with a shape (44898, 122513). Then the data was randomly divided into test and training/validation sets with an 80/20 split. Then the training and validation set was further split into training set and validation set with a 70/30 split. The method “train_test_split” from “sklean.model_selsection” library was used to split the initial data into two sets first, then the later set was again split into two set which was used for training and validation of the model.

Since the main goal of this project is to classify if the given text belongs to fake (0 category) or true (1 category). Two binary classifiers were chosen. The first is Logistic Regression and second Decision Tree with depth size varying between one and twenty to find the optimum depth size.

Logistic Regression is a classification algorithm which is based on a Sigmoidal function. Logistic Regression was chosen because of large datasets and it is easy and efficient to train. To calculate the training and validation errors for Logistic Regression model, logistic loss was used. It was imported from “sklearn.metrics” library. Logistic loss function was used instead of 0/1 loss as, Logistic loss is more sensitive to and leads to a better estimation. Since Logistic Regression is a simple model, having a complex loss function leads to better and more realistic model. Logistic loss is defined as

$$L_{\log}(y, p) = -(y \log(p) + (1 - y)\log(1 - p)$$

Where, for some data with a label *y* is equal to 0 or 1.

The second model chosen was Decision Tree with varying depth length from “sklearn.trees” library. Decision Tree is a tree like model where the next node can be reached from the previous node with a true or false paths. After iterating through the tree, when a leaf node is reached, it is then associated with the label. Decision Tree was chosen as a contrast to the first model (Logistic Regression). Logistic Regression is a simple and easy model to train, while Decision Tree is a bit more complex and takes considerably more computational power to train. Hence, after both the models have been trained, we can compare if Logistic Regression is sufficient for the sake of classifying true and fake article or more complex binary classifiers such as Decision Tree is required. To calculate training, validation and testing errors for decision tree, 0/1 (hinge loss) was used from “sklear.metrics” library. 0/1 loss was used as we want to penalise misclassifications. 0/1 loss is defined as

$$L_{0-1}(y_i, \hat{y_i}) = 1(\hat{y_i} \neq y_i)$$

where the function returns 1 if the predicted value is the same and the original value, and 0 if the predicted value is not the same as the original value.

In addition to training, validation and testing errors, F1 scores have been used to evaluate the model’s performance. F1 scores have been calculated using the confusion matrix from “sklearn.metrics” library. F1 score has been defined using the precision and recall of the model,

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

where precision is defined as,

$$\mathrm{Precision} = \frac{\mathrm{True\ Positive}}{\mathrm{True\ Positive} + \mathrm{False\ Positive}}$$

and Recall is defined as,

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Results

Logistic Regression:

The trained Logistic model was used to calculate the predicted labels using features from training and validation. Using the logistics loss function the error in predicted labels have been tabulated in table 1.

Table 1: Training error, Validation error, and F1 score of Logistic Regression model

Training error	Validation error	F1 score
0.6895	0.4743	0.9871

From table 1, it can be observed that the training and validation error seems to be very high. This might hint towards underfitting. One of the reasons might be because the words in training set might be harder to learn compared to validation set. For example, “Everyone will receive 1 million euros tomorrow” compared to “The US president said….”. Even without domain knowledge one can understand or suspect that the first phrase might be somehow fake. While the second phrase needs domain knowledge and can be difficult to verify. The high training error and validation error can be explained due to the fact that Logistic Regression is a simple model and was not able to account for all the variation in the data. Another reason could be because the texts are only a sample from the whole article, most identifiable phrase could be left out, which could make predicting if the article is true of fake harder. One way to improve is to collect more data and include more text from the article rather than samples. Although the training and validation errors are quite high, the F1 score error seem to be quite low. In the case of fake/true news calcification, misclassification is not the worst as article readers can give input if the article, they read is true or fake. Which then can be corrected.

Decision Tree:

The next model used was Decision Tree with different depth values ranging from 2 to 20. Using the 0/1 loss the predicted labels for training and validation were tested against the true labels. Training, validation errors along with F1 score for the Decision Tree for 2-5 depth values has been tabulated in table 2. The rest of the errors for depth values from 6-20 can be found in appendix.

Table 2: Training error, Validation error, and F1 score of Decision tree of different depth

Depth value	2	3	4	5
Training error	0.0062	0.0051	0.0051	0.0043
Validation error	0.0058	0.0052	0.0051	0.0046
F1 score	0.9936	0.9940	0.9940	0.9947

According to table 2 the depth value of 5 has the lowest validation error. Higher depth values also have lower validation error. However, training error reaches 0, which suggests overfitting. Hence, decision tree of depth 5 has been chooses as it does not seem too overfit. Although the validation error for depth value 5 is slightly higher than training error, it can be ignored as the increase in minutes.

If we compare Logistic Regression and Decision Tree, we can observe that there is a higher error in logistic regression. This is because Decision Tree works well with text as each word or words can be linked the whole article being fake. We can also see that the F1 score error for Decision Tree is significantly lower compared to F1 score for Logistic Regression. This also suggests that Decision Tree of depth 5 is a better model for classifying true and fake news compared to logistic loss. Therefore, after considering Logistic Regression and Decision Tree of depths 2-20 the chosen model is Decision Tree model with a maximum depth of 5.

Conclusion

Based on training error, validation error and F1 score, we can observe that Logistic Regression was unable to accurately classify news articles. In contrast, Decision Tree of maximum depth of 5 seems to be able to predict accurately and its test error is 0.0051. These low values might be due to the fact that 0/1 loss is a very basic loss function.

There are many ways to improve that could result in a model which classifies true and fake news articles. First, one can choose a more robust loss function, which could penalise the model better than 0/1. Another improvement can be done during the splitting stage, where instead of just splitting the whole data set randomly, K-fold splitting could be used. Different k values could be used and the one which results in lowest validation error could be chosen.

As discussed earlier the data selection could contain more raw data and more content in the raw data. This could be beneficial as sometimes the core part of summary of the article might be left out if only a sample is taken. While selecting the raw data more variety of news articles from different countries could be chosen. This would generalise the data leading into a more robust model which could handle a variety of different news articles. However, this data set is very beginner-friendly and easy to work with as it is in a csv format.

In conclusion, the Decision Tree of maximum depth of 5 was chosen for its low validation error and high F1 score while not leading to underfitting or overfitting. While there are many improvements discussed, it was a successful project as the model was able to accurately and reliably classify true and fake news article using its title and some text from the article.

References

Course Book, A. Jung, "Machine Learning: The Basics", http://mlbook.cs.aalto.fi

Bisaillon, C. Fake and real news dataset. Retrieved 10 February 2022, from https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset?select=Fake.csv

Appendix