Fake News Detection

Andrew Istfan
Caleb Huck
Abishek Ajai Satnur
Virginia Polytechnic Institute and State University
Blacksburg, Virginia, United States

1 INTRODUCTION

1.1 Problem Description

Fake news has become much more common and prevalent with the rise of social media and digital communication. Inaccurate information, political divisiveness, and public distrust of media organizations are just a few of the negative effects that fake news may have.

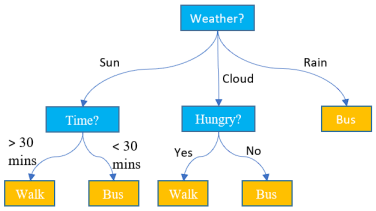
Fake news is much more common in today's world than it was twenty years ago. This is because of the rise of social media and digital communication as a whole. Inaccurate information, political divisiveness, and public distrust of media organizations are just a few of the negative effects that fake news may have. Whole communities can be influenced by fake news, resulting in mass misinformation.

Letting this problem build-up leads to the internet being a dangerous, untrustworthy form of communication. To combat this we have created a reliable system for identifying fake news is essential for preserving the authority of news sources and guaranteeing that the general public receives accurate news.

2 ALGORITHM DESCRIPTIONS

2.1 Decision Tree

Decision Tree Classifiers are a supervised machine learning that rely on an intuitive rule based approach for making decisions, very similar to how humans make decisions each day. They are built on the concepts of Decision Trees, which are a decision support hierarchical model that uses a tree-like model of decisions. Trees are constructed with nodes where a decision is made based upon a pre-defined rule. A decision tree can be followed until it reaches a leaf where its value is a classification. An example decision tree that a human may implement themselves for determining whether or not to walk or take the bus is depicted below:



The decision tree classification model is trained by creating a decision tree to parsed for new values. To build the decision tree, we begin with the root node and determine which feature is strongest for creating the split. While many split criterion can be used, we worked entropy to calculate the information gained before and after splits. As such, we begin at the node and recursively split the tree by the feature with the highest information gain. This continues until a stopping criteria is reached or no more classes exist. In our case, we have one hyperparamter, max_depth, which determines the maximum number of features that can be included the given decision tree. When a new row is being predicted, based off its feature values, it will traverse down this tree and reach a leaf node where it will ultimately make a prediction of whether or not it believes the article is real or fake.

2.2 Naïve Bayes

A popular approach for text classification applications is the Multinomial Naive Bayes classifier. It is based on the Bayes theorem and presumes that each feature is independent. The classifier determines if a news story is fake or authentic based on its linguistic characteristics in the context of fake news detection.

The Naive Bayes algorithm is based on Bayes' theorem, a fundamental principle in probability theory and statistics. Bayes' theorem relates the conditional probability of an event given another event. Mathematically, it is expressed as:

$$P(A|B) = (P(B|A) * P(A))/P(B)$$

Here, P(A|B) is the conditional probability of event A occurring given that event B has occurred, P(B|A) is the conditional probability of event B occurring given that event A has occurred, P(A) is the prior probability of event A, and P(B) is the prior probability of event B.

The steps taken to implement the model in practice are as follows:

(1) Data preparation: In this section, the code imports required libraries, reads in the training and testing datasets, and preprocesses the text data by cleaning, tokenizing, and removing stop words. This is done to prepare the data for feature extraction using the bag-of-words representation.
(2) Feature extraction: CountVectorizer from the sklearn.feature_extraction.text module is used to turn the text data into a representation of a bag of words for feature extraction. Each row in the bag-of-words representation corresponds to a document, and each column to a distinct word in the corpus. The word count in each column and row of the document is represented by the value in each cell.

Virginia Tech '23, ML-4824, Blacksburg, VA,

(3) The data is split into training and validation sets using the train_test_split function from sklearn.model_selection. This is done to evaluate the performance of the classifier on unseen data.
(4) Naive Bayes classifier implementation: The MultinomialNaiveBayes class is implemented to perform the Naive Bayes classification. The class has two main methods - fit and predict. In the fit method, the classifier estimates the prior probabilities and conditional probabilities for each class ci and feature fj based on the training data. This is done by counting occurrences of each class and feature in the training data. In the predict method, the classifier calculates the posterior probabilities for each class ci using the Bayes' theorem and the independence assumption. Then, it assigns the class with the highest probability to the input document.
(5) Model training and evaluation: The Naive Bayes classifier is trained on the training set using the fit method, and its performance is evaluated on the validation and test sets using the predict method. The accuracy_score from sklearn.metrics is used to calculate the accuracy of the classifier.

2.3 Convolutional Neural Network

Another popular method for Fake News Detection is Convolutional Neural Networks. Using 1D convolution layers, since we are analyzing only text and not an image that would require 2D layers and the pre-processed text, CNNs are useful due to their filter-down nature in terms of binary classifiers.
Our CNN uses the same data pre-processing stated earlier but works in a few different key ways. Our CNN was built so that the input fits what each entry would be coming in, which would be each individual entry's bag of words and the label associated with it. The model has a few Convolutional 1D layers to break down the information as it goes down the chain until it gets down its logic into the binary classifier of the file. It then does this with every file, learning and adapting the Neural Network as it learns and validates the data.

3 EXPERIMENTAL RESULTS

3.1 Decision Tree

3.1.1 Results.

Decision Tree Performance			
Accuracy	Precision	Recall	F-Score
0.485	0.0	0.0	0.0

3.1.2 Analysis of Results. The results of this classification are extremely poor and are likely due to the fact that the model was extremely computationally heavy. As such, the PC running the model was only able to work with a max depth of 50 with a reduced dataset to see the model complete in reasonable time. Given the dataset, had over 100,000 attributes when fully vectorized, it is not shocking to see the shockingly poor results. While the model does work for other smaller datasets, it needs to be significantly improved in efficiency so that it can be run on larger datasets such as these. For comparison, however, a Decision Tree from sklearn was utilized as well to assess its performance. It was far more efficient in training and was able to generate strong accuracies; however, appeared to suffer from a brief amount of overfitting which is typical from a Decision Tree Classification.

3.1.3 Remaining Errors. The model is currently extremely computationally expensive. As it is currently recursively navigating through each feature on a massive vectorized dataset, it takes an extreme amount of time to train. As such, in the future optimizations could be made to attempt to improve the efficiency of the model. For example, when using the scklearn model without a limit on the number of features, the model would run in just a handful of minutes.

3.1.4 Advantages and Disadvantages. The model is extremely trivial to interpret as decision trees are a very intuitive concept. They closely follow the way we process and make decisions as humans each day. As such, it makes implementation and interpretation simple. However, the model is extremely prone to overfitting with just one tree and as such it could be improved upon with the implementation of multiple trees, bagging and random feature selection (Random Forest).

3.2 Naïve Bayes

3.2.1 Results.

Naïve Bayes Performance			
Accuracy	Precision	Recall	F-Score
0.9646	0.9919	0.9368	0.9636

3.2.2 Analysis of Results. The Naive Bayes machine learning model ended with an accuracy of 0.9646. That is a good level of accuracy. Precision was very high, and recall and F-score metrics were also good. The Naive Bayes model was overall successful in detecting fake news and has a relatively less training time of around 10 minutes.

3.2.3 Remaining Errors. The model does make a substantial amount of assumptions that do not necessarily hold true. These include assumptions that all the classes are independent and weighed equal. This is not necessarily true in the case of fake news.

3.2.4 Advantages and Disadvantages. One of the main advantages is the simplicity of the model and its low running time. Naive Bayes does not require a large dataset for it to train. Naive Bayes works well with text data, and it handles missing values well.

3.3 Convolutional Neural Network

3.3.1 Results.

Convolutional Neural Network Performance			
Accuracy	Precision	Recall	F-Score
0.9976	0.9996	0.9956	0.9976

3.3.2 Analysis of Results. The CNN did exceptionally well getting very high percentages in all categories as well as having a very low loss score of 0.0052. This shows a stength in the model's ability to assess a dataset, even when it may take a long time to do so.

3.3.3 Remaining Errors. Not necessarily an error but the model still takes a significant amount of time to run at about an hour per epoch, making it very hard to fine-tune quickly and debug, as well as generally taking a lot of processing power to run. In a future attempt, I would try to see if I could potentially optimize the layers of the CNN or run on a computer that isn't a laptop to see if I can get a more significant processing power to run faster and be able to fine-tune more.

3.3.4 Advantages and Disadvantages. One of the main advantages of this model was its ability to have a very strong accuracy of prediction but with too many epochs was prone to overfitting. Hyperparameter tuning was a key part of making this model functional. The original base epochs of 10 got the accuracy closer and closer to 1.00 but when running at the test data only managed to do slightly better than a random guess, meaning it probably overfit the data.

3.4 Consolidated Results

3.4.1 Results.

Consolidated Comparison of Models				
Model	Accuracy	Precision	Recall	F-Score
Decision Tree	0.485	0.0	0.0	0.0
Naïve Bayes	0.9646	0.9919	0.9368	0.9636
CNN	0.9976	0.9996	0.9956	0.9976

4 TEAN MEMBER ROLES

4.1 Andrew Istfan

Main contributor to Decision Trees. Pre-processing of data, research, report and presentation.

4.2 Caleb Huck

Main contributor to Convolutional Neural Network. Pre-processing of data, research, organizing of main group and files.

4.3 Abishek Ajai Satnur

Main contributor to Naive Bayes. Worked on parts of the report and presentation.