CSPI-P556 - Applied Machine Learning - Fall 2021

Fake News Detection

Anuhya Sankranti, Krithika Shivaram, Suyash Chaudary

December 15, 2021

Abstract

Fake news is a widely disseminated source of great worry due to its potential to create significant social harm. Our goal is to create a model that uses recurrent neural networks and machine learning approaches to detect fake news. Two different RNN approaches and five separate machine classification algorithms were explored and compared.

Keywords
Fake news, RNN, SVM, Machine learning

1 Introduction

Fake news is becoming a rising threat in society. It is a difficult effort to detect it. Fake news is a serious threat to not only the credibility of some media sources, but also to the government and society. More than 62 percent of adults in the United States acquire their news from social media, according to reports. Furthermore, the sheer volume of material released and the speed with which it spreads on social media sites makes determining its reliability in a timely manner extremely challenging. The detection of fraudulent content in online and offline sources is a significant research issue.

The objective of the paper is to compare performances of five machine learning (ML) algorithms on a standard dataset using a novel set of features and statistically validate the results using accuracies.The five machine learning techniques that we chose are, Support Vector Machine, Logestic regression algorithm, Ensembling modeling, Random forest, Naive Bayes algorithms. These are the different classification algorithms that help us classify the text as fake or real.

1. Single Classifier based Prediction

Classifiers are algorithm which maps the input data to specific type of category .

i) Support Vector Machine

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

ii) Naive Bayes Classifier

The Naive Bayes algorithm assumes that all features are independent of one another and contribute equally to the output. Another assumption is that all attributes are of equal value. It has various uses in today's world, including as spam filtering and document classification. Naive Bayes takes only a little amount of training data to estimate the needed parameters. Furthermore, a Naive Bayes classifier is much quicker than more sophisticated and advanced classifiers. The Naive Bayes classifier, on the other hand, is known for being poor at estimating because it assumes all characteristics are of equal value, which is not true in most real-world cases. This is based on Bayes'theorem . A, B: events, P: Probability, P(A|B): how often A happens given that B happens, P(B|A): how often B happens given that A happens, P(A), P(B): The independent probabilities of A and B. P(A|B) = P(A) P(B|A) P(B)

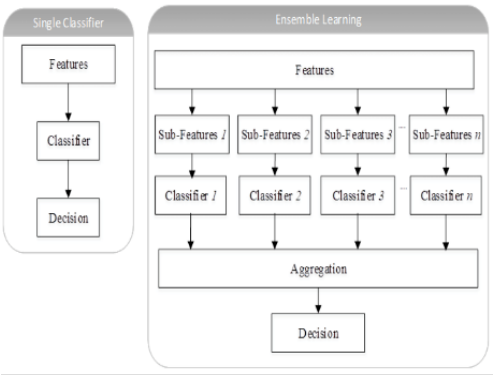
iii) Logestic Regression Classifier

Binomial logestic regression is used when the target variable is categorical. In our case the target variable is predicting if the news if fake or real. In many ways, binomial logistic regression is similar to linear regression, with the exception of the measurement type of the dependent variable (i.e., linear regression uses a continuous dependent variable rather than a dichotomous one). However, unlike linear regression, you are not attempting to determine the predicted value of the dependent variable, but the probability of being in a particular category of the depen-

dent variable given the independent variables. An observation is assigned to whichever category is predicted as most likely. As with other types of regression, binomial logistic regression can also use interactions between independent variables to predict the dependent variable.

2. Ensemble Approach based Classifiers

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.



i) Random Forest Classifier

The random forest classifier fits multiple decision trees on different dataset sub-samples. It uses the average to enhance its predictive accuracy and manage overfitting. The size of the sub-sample is always identical to the size of the input sample; however, the samples are generated using replacement.

The random forest classifier has a unique benefit in that it reduces overfitting. Furthermore, our classifier outperforms decision trees in terms of accuracy. However, it is a far slower approach for real-time prediction and a highly sophisticated algorithm, making it extremely difficult to execute successfully.

ii) Voting Classifier

The voting classifier functions similarly to an electoral system, making predictions on new data points based on a voting system among members of a group of machine learning models. The hard and soft voting types are available.

For majority rule voting, the hard voting type is applied to projected class labels. This is based on the principle of "majority carries the vote," which states that whoever receives more than half of the vote wins. We used hard type in our model

iii) Gradient Boosting Classifier

Gradient boosting classifier is a set of machine learning algorithms that include several weaker models to combine them into a strong big one with highly predictive output. Models of a kind are popular due to their ability to classify datasets effectively. Gradient boosting classifier usually uses decision trees in model building.

The weak learners are fit in such a way that each new learner fits into the residuals of the previous step so as the model improves. The final model aggregates the result of each step and thus a strong learner is achieved. A loss function is used to detect the residuals. For instance, mean squared error (MSE) can be used for a regression task and logarithmic loss (log loss) can be used for classification tasks. It is worth noting that existing trees in the model do not change when a new tree is added. The added decision tree fits the residuals from the current model.

3. Catboost Classifier

Catboost is a boosted decision tree machine learning algorithm developed by Yandex. It works in the same way as other gradient boosted algorithms such as XGBoost but provides support out of the box for categorical variables, has a higher level of accuracy without tuning parameters and also offers GPU support to speed up training.

4. Recurrent Neural Network

LSTM stands for Long-Short Term Memory. LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Having a good hold over memorizing certain patterns LSTMs perform fairly better. As with every other NN, LSTM can have multiple hidden layers and as it passes through every layer, the relevant information is kept and all the irrelevant information gets discarded in every single cell. How does it do the keeping and discarding you ask?

One good reason to use LSTM is that it is effective in memorizing important information. In LSTM we can use a multiple word string to find out the class to which it belongs. This is very helpful while working with Natural language processing. If we use appropriate layers of embedding and encoding in LSTM, the model will be able to find out the actual meaning in input string and will give the most accurate output class.

2 Process

The fake news detection dataset was obtained from Kaggle. The data set consisted of five columns, the title of the news, the actual news, the subject to which news belongs to, date of news published and the last column is our target column that says if the particular news is fake or real. All the columns contain categorical data except the date column. You can get a bet-

ter picture of data referring to the figure below, that shows the summary of the data.

```
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   title    44898 non-null  object
 1   text     44898 non-null  object
 2   subject  44898 non-null  object
 3   date     44898 non-null  object
 4   label    44898 non-null  object
```

Text pre-processing is the process of preparing text data so that machines can use the same to perform tasks like analysis, predictions, etc. Preprocessing the data is required as the data is categorical. The preprocessing included Removing Stop Words, Removing Unnecessary Characters, Tokenization, Encoding and Feature Scaling.

Removing Stop words: Stop words are the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text. Examples of a few stop words in English are “the”, “a”, “an”, “so”, “what”. Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text in order to give more focus to the important information. In order words, we can say that the removal of such words does not show any negative consequences on the model we train for our task. Removal of stop words definitely reduces the dataset size and thus reduces the training time due to the fewer number of tokens involved in the training.

Removing Unnecessary Characters: This process include removing the special characters like @, , etc., that don’t give any meaning to the word, removing the punctuation like ’,’ , ’.’, ’:’,’!’, etc., as we consider each word as separate entity and we do not care about the punctuation of sentence anymore. This process also include removing the white spaces and duplicate letters.

Tokenization: Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. For example, we can divide a chunk of text into words, or we can divide it into sentences. Depending on the task at hand, we can define our own conditions to divide the input text into meaningful tokens.

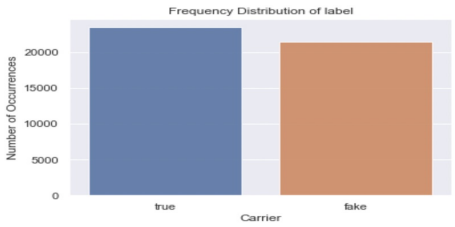
Natural Language Processing

['Natural', 'Language', 'Processing']

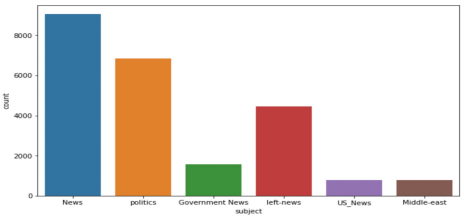
Encoding: Encoding is the process of converting categorical data to numeric data. Encoding is necessary for our data as our data is categorical. We used word vectorizer in the process. Word Embeddings or Word vectorization is a methodology to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics. The process of converting words into numbers are called Vectorization.

Feature Scaling: Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. For feature extraction we have selected ’text’ and ’label’ columns from our dataset.

Once preprocessing the data is done, we are left with clean and standard dataset. To analyse the features of the dataset we did the following analysis.

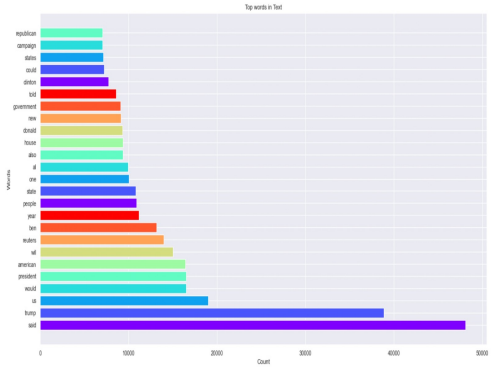


This plot shows the frequency of fake and real news in the dataset. We can observe that the distribution is almost equal with a very minor difference in the frequency.



This plot shows the distribution of subject column in our dataset. We can observe that news and politics have the highest frequency of news in the dataset.

The frequency of the top words used in the data can be visualized from the plot below. This plot shows the top words and their respective count in the dataset.



After all this preprocessing we have applied different machine learning classifiers to find if the news is fake or real. We also implemented recurring neural networks. We used LSTM in this process. The Long Short Term-memory model(LSTM model) is a type of recurrent neural network that is better than other recurrent neural networks in terms of memory. It has a good hold over memorizing certain patterns in the text. LSTM retains the useful information out of the text and it discards useless information in every cell. While traditional neural networks suffer from problems like short term memory and vanishing gradient, LSTM overcomes these challenges by efficiently memorizing useful information that is important to find patterns within the text/corpus. To fit our data on an LSTM RNN model, we had to make it compatible with it first. After preprocessing, we created the vectors we had to tokenize our data so that our model can interpret and analyze our input. After we tokenize our data, we create word embeddings for our input data and model using the w2vec model and our vector vocabulary. We then finally implement an RNN model with an LSTM layer consisting of 128 units and a Dense layer acting as an output layer. Our model compiles using an 'Adam' optimizer and binary cross-entropy for loss type. Through training it for almost 8 epochs, we get a presentable result.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 1000, 125)         29059625

lstm (LSTM)                  (None, 128)               130048

dense (Dense)                (None, 1)                 129

=================================================================
Total params: 29,189,802
Trainable params: 130,177
Non-trainable params: 29,059,625
_________________________________________________________________
```

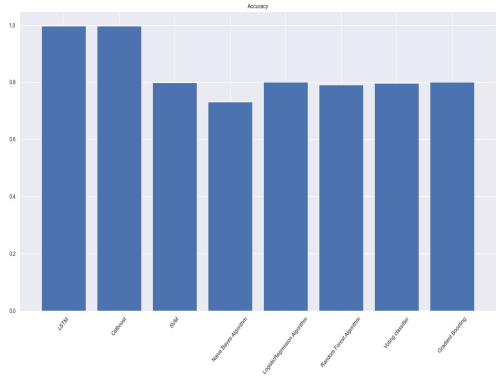
To train our catBoostClassifier model, we'll have to perform manual splitting of our data as we need to specify in our pool what the text feature of data is, we use the following hyperparameters for our model, for tokenization we'll use the sense feature as it instructs our model to detect tokens intuitively. We also specified our token types as a list of word, numbers and sentenceBreak and we interpret our sub-tokens as several/multiple tokens. Our dictionaries will have a maximum of 50000 tokens and lastly we use Bag of words as well.

3 Results & insights

Once the classification is done using different machine learning models, we had the following results: Catboost has the highest accuracy of 99.7 percent followed by LSTM with 99.6 percent. Among the Single classifier based predictions, Logestic regression and Support vector machine classifiers showed the highest accuracies of 79 percent each, while Naive bayes showed the accuracy of around 73 percent. Among the Ensemble approach based classifiers, Gradient Boosting classifier has highest accuracy of 80 percent followed by Voting classifier and Random forest classifier with 79 percent each. We can understand better from the following table

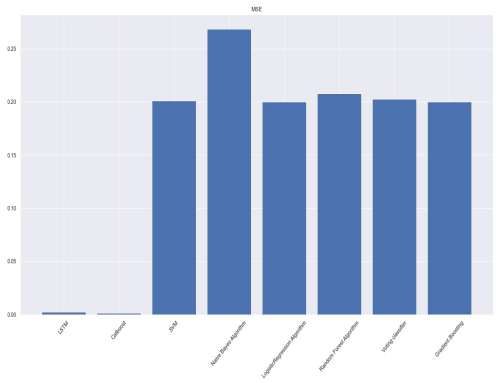
Classifier	Accuracy	MSE
LSTM	0.996800	0.002600
Catboost	0.997200	0.001300
SVM	0.799035	0.200965
Naive Bayes Algorithm	0.731552	0.268448
LogisticRegression Algorithm	0.799777	0.200223
Random Forest Algorithm	0.791982	0.208018
Voting classifier	0.797253	0.202747
Gradient Boosting	0.800223	0.199777

For analysing the performances of different algorithms we made a comparison plot with all the different classifiers we used on the x-axis and their accuracies on the y-axis.



From this plot we can see that LSTM and Catboost classifier has highest accuracies. While all the classifiers showed decent accuracies Naive Bayes has the least accuracy when compared

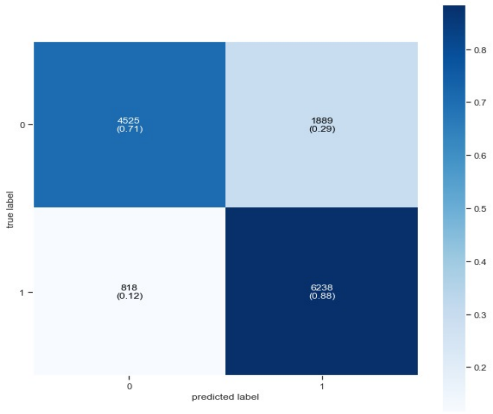
with the others. Similarly we made a comparison plot of Mean square errors of each algorithm.



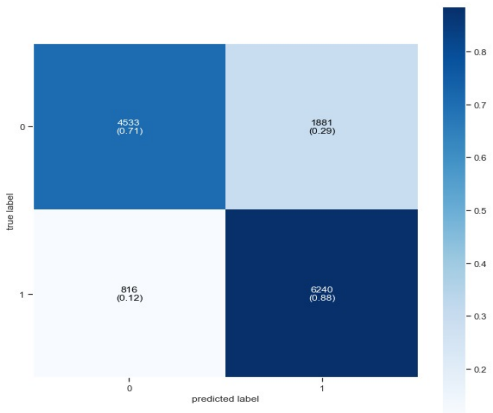
Here, we can see that the LSTM and Catboost approaches have the least mean square error compared to the other classifiers. Naive Bayes has the highest mean square error. The algorithms that has highest accuracies has least mean square error and vice versa and this does make sense.

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. Following are the confusion matrices of each approach we used.

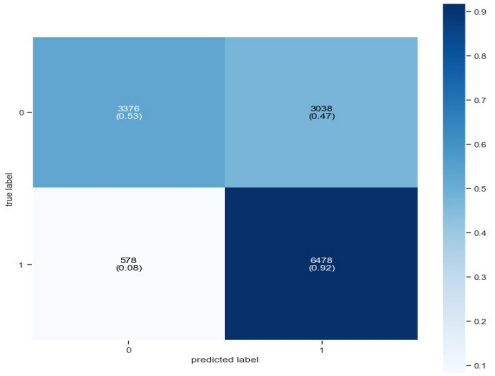
Confusion Matrix of SVM classifier



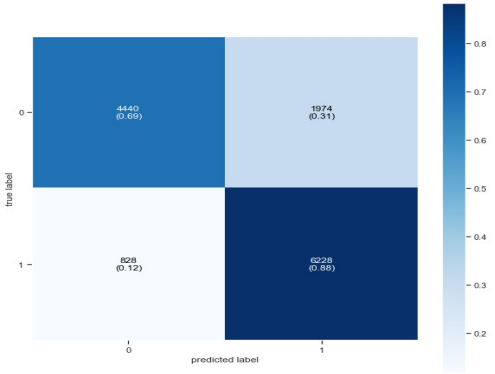
Confusion matrix of Logestic Regression classifier



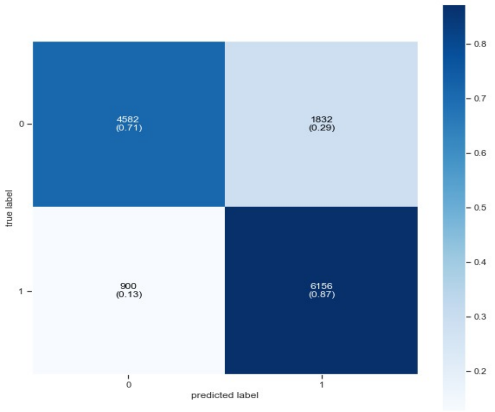
Confusion matrix of Naive Bayes approach



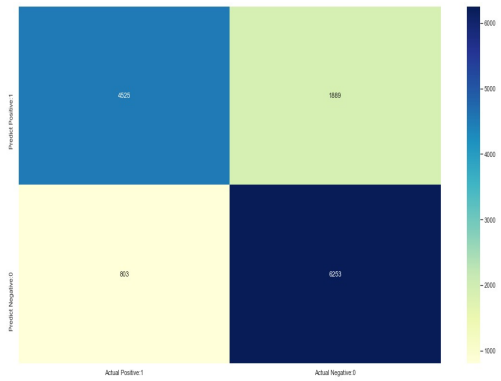
Confusion matrix of Random forest classifier



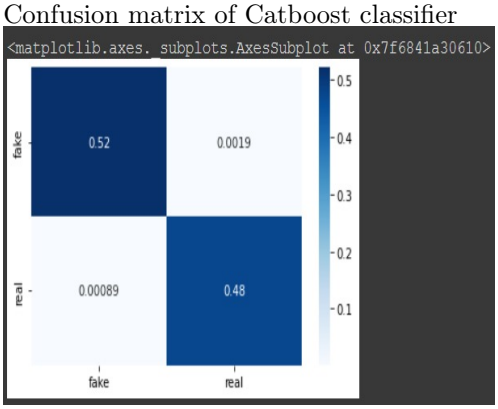
Confusion matrix of Voting classifier



Confusion matrix of Gradient Boosting algorithm



Confusion matrix of Catboost classifier



Conclusions

Predicting if a news is fake or real can help people know whether the news they are following is correct or misleading. We have seen the different approaches to classify the news as fake or real and have learnt that LSTM and Catboost approaches predict the news to be fake or real more efficiently. Hence we can further work on this approach and develop the model on both image and text data.

References

https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/
https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset T. Joachims, [U+2015] Text categorization with support vector machines: Learning with many relevant features, [U+2016] in European conference on machine learning, 1998, pp. 137–142. D. E. Walters, [U+2015] Bayes's Theorem and the Analysis of Binomial Random Variables, [U+2016] Biometrical J., vol. 30, no. 7, pp. 817–825, 1988, doi: 10.1002/bimj.4710300710.