Fake news detection using NLP

Akintya Nigam
akintnigam@gmail.com
SRM Institute of Science and Technology,
Chennai, Tamil Nadu

Aradhya Dhruv
aradhya.dhruv003@gmail.com
SRM Institute of Science and Technology,
Chennai, Tamil Nadu

ABSTRACT

All Counterfeit news is the purposeful spread of deception or promulgation by conventional news media and through online media. Such news stories can come in numerous structures, including: accidental mistakes submitted by news aggregators, inside and out bogus stories, or the tales which are created to deceive and impact per user's assessment. Likewise, since bogus data can spread so quick, not just it can hurt individuals yet additionally can be hindering to tremendous enterprises and financial exchanges. In the previous decade, we have seen dramatic development of data accessible on the web. It is getting difficult to interpret valid from the bogus. It is essential for us to know that how much of what we read on supposedly credible news site is trustworthy. False information can cause panic among citizens. Likewise lies can be utilized to control different people choices for casting a ballot bid or basically whatever else that can have enduring consequences. Bogus data spreads extremely quick, this is shown by the way that when one phony news site is brought down another immediately has its spot. Besides, it is fit for demolishing the equilibrium of the news environment. Political plans and control are one of the numerous intentions since counterfeit news is created. Today, fortunately we have advances machine learning and Language Processing (NLP) instruments offer incredible guarantee for specialists to construct frameworks which could naturally identify counterfeit news. Hence it would be beneficial to discuss the methods of detecting online deceptions. Our undertaking can be essentially utilized by any media organization to foresee if the coursing news is phony. The interaction should be possible consequently without having people physically audit a huge number of information related articles. Approaching the problem from a purely NLP perspective, it will be possible for us to predict whether the news is fake or real based only on its content. A significant piece of the objective is to think about and report the outcomes from numerous distinctive model executions, and present an investigation of the discoveries. We will use several architectures like naïve based classifier and random forests to come up with an accurate prediction.

Keywords— *Fake news, Natural Language Processing, Count Vectorizer, tf-idf, Lemmatization, Stemming, Machine Learning, Random forests, Decision Trees, Confusion Matrix.*

1. INTRODUCTION

These days' fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets. Fake news and lack of trust in the media are growing problems with huge ramifications in our society. Obviously, a purposely misleading story is "fake news "but lately blathering social media's discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.

The importance of disinformation within American political discourse was the subject of weighty attention, particularly following the American president election. The term 'fake news' became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views. In this paper, it is seemed to produce a model that can accurately predict the likelihood that a given article is fake news.

Facebook has been at the epicentre of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user see's it; they have also said publicly they are working on to to distinguish these articles in an automated way. Certainly, it is not an easy task. A given algorithm must be politically unbiased – since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what Fake News is. Later, it is needed to look into how the techniques in the fields of machine learning, natural language processing helps us to detect fake news.

There is a Kaggle competition called as the "Fake News Challenge" and Facebook is employing AI to filter fake news stories out of users' feeds. Combatting the fake news is a classic text classification project with a straight forward proposition. Is it possible for you to build a model that can differentiate between "Real "news and "Fake" news? So, a proposed work on assembling a dataset of both fake and real news and employ a Random Forestclassifier in order to create a model to classify an article into fake or real based on its words and phrases.

The main objective is to detect the fake news, which is a classic text classification problem with a straight forward proposition. It is needed to build a model that can differentiate between "Real" news and "Fake" news

2. PROPOSED FAKE NEWS DETECTION SYSTEM

The proposed system will consist of 2 modules:
- A pre-processor which provides the following functionalities:
- Removing punctuations
- Removing stop words
- Removing unnecessary numeric data
- Tokenization of text
- Lemmatization of text
- Count Vectorization
- Tfidf Vectorization (Term Frequency Inverse Document Frequency)
- A Machine Learning module which implements SVC (Support Vector Classifier) to carry out he predictions.
- The proposed classifier tampers and boosts the accuracy of the given module by changing the values of the
- Regularization Parameter which is denoted by C.
- The value of C has to be positive.
- C has a default value of 1.0 which gives it the standard accuracy.
- When the value of C is increased to 2, there is a significant increase in the accuracy and when it is increased to 3 the accuracy is boosted further.
- After 3 the value of accuracy doesn't increase much therefore the optimal value of the Regularization parameter for our given module is 3.
- This gives us an optimal accuracy of: 97.48%

3. PROTOTYPE IMPLEMENTATION

Solving this problem from a purely NLP perspective, we first followed the basic preliminary data pre-processing steps. All stop words such as 'I', 'he', 'she', 'it', 'the' 'that' etc were removed. We had done so, because stop words have very low significance when it comes to predicting whether an article is fake or not. Once the stop words had been removed, we carried out lemmatization on our dataset and created a "word cloud" to see what words are the most frequently-occurring ones within our dataset.

```
In [37]:  from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

In [38]:  svc_cm= confusion_matrix(y_test, predictions)
          svc_clf= classification_report(y_test, predictions)
          svc_acc= accuracy_score(y_test, predictions)

In [39]:  print(svc_acc)
          print(svc_clf)
          print(svc_cm)

          0.9666942833471417
                        precision    recall  f1-score   support

                     0       0.97      0.97      0.97      3450
                     1       0.96      0.96      0.96      2585

              accuracy                           0.97      6035
             macro avg       0.97      0.97      0.97      6035
          weighted avg       0.97      0.97      0.97      6035

          [[3357   93]
           [ 108 2477]]

In [41]:  text_clf.predict(['Videos 15 Civilians Killed'])

          0-Fake news
```
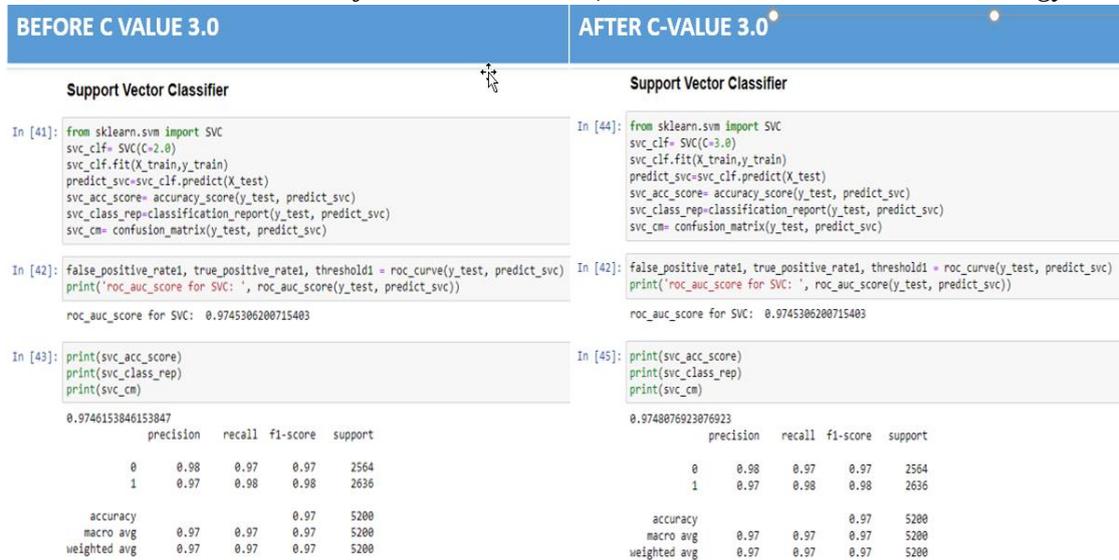
Figure 1: Accuracy of SVM Classifier

After getting a clean-data we trained it over multiple machine learning models such as logistic regression, Naïve bayes, random forests, Support vector machine etc. and drew a comparative study to find out which algorithm was the most accurate one. In our case it was SVC. Once the most efficient algorithm had been identified by us, we fine-tuned its parameters to get even more promising results. The table below illustrates the accuracy before the and after the C-value was modified.

BEFORE C VALUE 3.0	AFTER C-VALUE 3.0

Support Vector Classifier (Before C Value 3.0)

```
In [41]: from sklearn.svm import SVC
         svc_clf= SVC(C=2.0)
         svc_clf.fit(X_train,y_train)
         predict_svc=svc_clf.predict(X_test)
         svc_acc_score= accuracy_score(y_test, predict_svc)
         svc_class_rep=classification_report(y_test, predict_svc)
         svc_cm= confusion_matrix(y_test, predict_svc)
```

```
In [42]: false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(y_test, predict_svc)
         print('roc_auc_score for SVC: ', roc_auc_score(y_test, predict_svc))

         roc_auc_score for SVC:  0.9745306200715403
```

```
In [43]: print(svc_acc_score)
         print(svc_class_rep)
         print(svc_cm)

         0.9746153846153847
                       precision    recall  f1-score   support

                    0       0.98      0.97      0.97      2564
                    1       0.97      0.98      0.98      2636

             accuracy                           0.97      5200
            macro avg       0.97      0.97      0.97      5200
         weighted avg       0.97      0.97      0.97      5200
```

Support Vector Classifier (After C-Value 3.0)

```
In [44]: from sklearn.svm import SVC
         svc_clf= SVC(C=3.0)
         svc_clf.fit(X_train,y_train)
         predict_svc=svc_clf.predict(X_test)
         svc_acc_score= accuracy_score(y_test, predict_svc)
         svc_class_rep=classification_report(y_test, predict_svc)
         svc_cm= confusion_matrix(y_test, predict_svc)
```

```
In [42]: false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(y_test, predict_svc)
         print('roc_auc_score for SVC: ', roc_auc_score(y_test, predict_svc))

         roc_auc_score for SVC:  0.9745306200715403
```

```
In [45]: print(svc_acc_score)
         print(svc_class_rep)
         print(svc_cm)

         0.9748076923076923
                       precision    recall  f1-score   support

                    0       0.98      0.97      0.97      2564
                    1       0.97      0.98      0.98      2636

             accuracy                           0.97      5200
            macro avg       0.97      0.97      0.97      5200
         weighted avg       0.97      0.97      0.97      5200
```

Figure 2: Before and after modifying C value

4. CONCLUSIONS

The model provides proof that Support vector classifier provides higher accuracy points as compared to Random Forest, Naive Bayes and Logistic Regression. We get this result as our model can change the C value, thereby identifying better-fit patterns with about the same computational capacity. With these findings, we can safely infer that Fake news can be predicted with highest accuracy using the given classifier and the value of C at 3 gives the highest accuracy. The accuracy can be altered using C which is also known as the regularization parameter. NLP played an important role in the pre-processing of the data and therefore helped the classifier to understand the data set better. For future work findings and scope, we would like to develop ways to detect sarcasm in news articles. Sarcasm is one of the biggest challenges in NLP and every tech giant is trying their best to develop ways as to detect sarcasm in text messages.

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] IEEE Conference 2019: - Fake News Detection in Social Networks Using Machine Learning and Deep Learning: Performance Evaluation.
[2] IEEE Conference, 2019 1st International Conference on Advances in Information Technology: - Fake News Detection Using Deep Learning Techniques.
[3] IEEE Conference 2019: - Fake News Detection Using Machine Learning approaches: A systematic Review.
[4] 2018 4th International Conference on Computing Communication and Automation (ICCCA): - Fake News Detection Using A Deep Neural Network.

BIOGRAPHIES

Akintya Nigam
Student,
SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

Aradhya Dhruv
Student,
SRM Institute of Science and Technology, Chennai, Tamil Nadu, India