



FAKE NEWS DETECTION

Submitted by:
Rijul Kumar

ACKNOWLEDGMENT

Reference that i have used are:

- Data Trained Education online video
- Materials provided by Flip Robo
- Youtube
- Kaggle
- Geeks for Geeks
- Stackoverflow

INTRODUCTION

- Business Problem Framing

1. Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.
2. Fake news' simple meaning is to incorporate information that leads people to the wrong path. Nowadays fake news spreading like water and people share this information without verifying it.
3. For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. So it is necessary to detect fake news.

- Conceptual Background of the Domain Problem

The fake news can be detected with the help of some factors. Such as:

- Some labeled text data
- Certain recurring words
- And so on...

- Motivation for the Problem Undertaken

- For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. So it is necessary to detect fake news.
- In this project, we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression), Scikit Learn.
- This model will then be used to predict whether a news is true or fake.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
 - First of all i imported data from both csv files to dataframes using pandas.
 - After that i added label column (fake = 1 / true = 0) to the 2 dataframes and then i concatenated the 2 dataframes.
 - Then i used .dtypes to know data type of each column of dataframe.
 - After that i used .describe() to know the statistical information (such as max, min value,etc) of continuous data columns in dataframe.
 - Then i used .shape to know shape of dataframe.
- Data Sources and their formats
 - Training data has been extracted from 'Fake.csv' and 'True.csv' which has been obtained from flip robo (from kaggle)
 -
 - 'Fake_news_project.docx' from flip robo

- Data Preprocessing Done

- First of all for data preprocessing i checked whether there is a NULL value or not in dataframe using heatmap as well as .isnull()
- After that i converted mail text into lower case alphabets and removed special characters from it.
- Next i tokenized the reviews and removed the stop words from them.
- After that i applied Lemmatization method.
- Afterwards i interpreted them into Bag of words model.
- Next i used Density plots from seaborn library to plot all continuous columns for visualisation.

- Data Inputs- Logic- Output Relationships

- Data Input :

These are basically the news which are to be tested for true or fake news.

- Data Output :

Our Target variable is 'label' which tells us if a news is true = 0 or fake = 1.

- Hardware and Software Requirements and Tools Used

- Hardware used:
 - i. Laptop with intel core i5 7th gen
 - ii. Internet connection for web scraping
- Software used:
 - i. Jupyter notebook
 - ii. Required python libraries such as numpy, pandas, seaborn, matplotlib, etc
 - iii. Required libraries for model such as sklearn, etc

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
 - After preprocessing data, i separated columns into features and target based on target variable 'label'.
 - As this is a Binary classification problem of NLP field so i tried Naive Bayes Classifier.

- Run and Evaluate selected models

I defined a function model and then tried Naive Bayes Classification model using it

```
from sklearn.metrics import roc_curve

def model_selection(algorithm_instance, features_train, target_train, features_test, target_test):
    algorithm_instance.fit(features_train, target_train)
    model_1_pred_train = algorithm_instance.predict(features_train)
    model_1_pred_test = algorithm_instance.predict(features_test)
    print("Accuracy for the training model : ", accuracy_score(target_train, model_1_pred_train))
    print("Accuracy for the testing model : ", accuracy_score(target_test, model_1_pred_test))
    print("Confusion matrix for model : \n", confusion_matrix(target_test, model_1_pred_test))
    print("Classification Report for train data : \n", classification_report(target_train, model_1_pred_train))
    print("Classification Report for test data : \n", classification_report(target_test, model_1_pred_test))

    Train_accuracy = accuracy_score(target_train, model_1_pred_train)
    Test_accuracy = accuracy_score(target_test, model_1_pred_test)

    for j in range(2, 10):
        cv_score = cross_val_score(algorithm_instance, feature, target, cv=j)
        cv_mean = cv_score.mean()
        print("At cross fold " + str(j) + " the cv score is " + str(cv_mean) + " and accuracy score for training is " + str(Train_accuracy))
        print("\n")

    #Plotting auc roc curve
    plt.figure(figsize=(8, 6))

    # calculate roc curves
    lr_fpr, lr_tpr, _ = roc_curve(target_test, model_1_pred_test)

    plt.plot(lr_fpr, lr_tpr, marker='.', label=algorithm_instance, color='r')
    plt.plot((0, 1), (0, 1), marker='*', label='No skill', color='b')

    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')

    plt.legend()
    plt.show()
```

Result that i got for model :

- Naive Bayes Classifier :

Accuracy for the training model :

0.9853272823454075

Accuracy for the testing model :

0.9858574610244989

At cross fold 2 the cv score is 0.892845546126935

and accuracy score for training is

0.9853272823454075 and accuracy score for testing

is 0.9858574610244989

Confusion matrix for model :

```
[[4165 102]
```

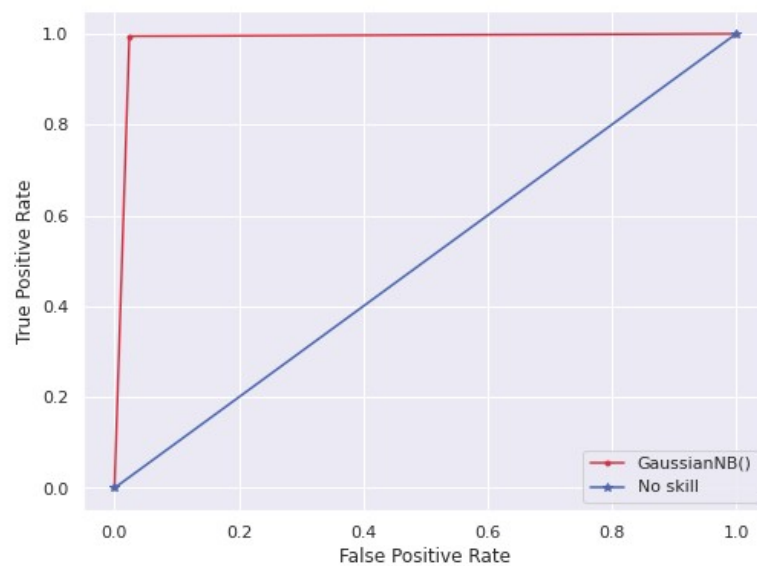
```
[ 25 4688]]
```

Classification Report for train data :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.99 | 0.98 | 0.98 | 17149 |
| 1 | 0.98 | 0.99 | 0.99 | 18768 |
| accuracy | | | 0.99 | 35917 |
| macro avg | 0.99 | 0.98 | 0.99 | 35917 |
| weighted avg | 0.99 | 0.99 | 0.99 | 35917 |

Classification Report for test data :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.99 | 0.98 | 0.98 | 4267 |
| 1 | 0.98 | 0.99 | 0.99 | 4713 |
| accuracy | | | 0.99 | 8980 |
| macro avg | 0.99 | 0.99 | 0.99 | 8980 |
| weighted avg | 0.99 | 0.99 | 0.99 | 8980 |



- Interpretation of the Results

So the results which i got were:

- We got Training Accuracy as 98.53% and Testing Accuracy as 98.59%.
- Since we have approximately balanced data (equal number of true and fake news), the precision is high for the data.

CONCLUSION

- Key Findings and Conclusions of the Study

- From this study i learnt that many news editor use certain words a lot while writing news which can be used to check if news is fake or not.
- More number of frequent words we take, higher the accuracy will be in the model till a threshold point.

- Learning Outcomes of the Study in respect of Data Science

Some problems faced and their solution (using visualisation and algorithm) used were:

- When i used 500 most frequent words in bag of words the testing accuracy was higher than when i used 100 most frequent words in bag of words but the time taken for code execution was much longer in case of 500 most frequent words. For 1000 most frequent words code execution was taking much longer hence i chose the case with 500 most frequent words for my model.

- Limitations of this work and Scope for Future Work

Some limitations are :

- There are many other factors which are not in the data which may play major role in fake news such as human error, etc.
- With evolving technology, there may be increase in intelligently hidden fake news which are also required to be taken care of.

Scope for future work :

- This can be made further accurate by taking more and more factors as such as taking human error in consideration .