"FAKE NEWS PROJECT"

Submitted by:

Sarika Thorat

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Car Price Prediction” project also huge thanks to my academic team “Data Trained”. Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

The website that I referred are:

https://learning.datatrained.com

https://www.w3schools.com

https://medium.com/coders-camp

https://github.com

https://www.geeksforgeeks.org

https://www.javatpoint.com/nlp

https://www.educative.io/answers/preprocessing-steps-in-natural-language-processing-nlp

https://www.youtube.com/watch?v=5ctbvkAMQO4

https://www.youtube.com/watch?v=X2vAabgKiuM

INTRODUCTION

Business Problem Framing

Fake news has become one of the biggest problems of our age. It has a 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.

Fake news's 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. This is often done to further or impose certain ideas and is often achieved with political agendas.

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 main goal of the assignment is to show how you could design a Fake news filtering system from scratch.

In this project, we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression), Scikit Learn.

Natural Language Processing

Machine learning data only works with numerical features so we have to convert text data into numerical columns. So, we have to

pre-process the text and that is called natural language processing. In-text pre-process we are cleaning our text by steaming, lemmatization, removing stopwords, removing special symbols and numbers, etc. After cleaning the data, we have to feed this text data into a vectorizer which will convert this text data into numerical features.

Review of Literature

There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. I have inserted one label column zero for fake news and one for true news:

- Title: Headlines of the news.

- Text: Content of the news.

- Subject: Subject of the news.

- Date: Date of the news.

- Label: News is True (1)/False (0)

Motivation for the Problem Undertaken

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to

cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed.

The sensationalism of not-so-accurately eye-catching and intriguing headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of information broadcast. On social networking websites, the reach and effects of information spread are however significantly amplified and occur at such a fast pace, that distorted, inaccurate, or false information acquires a tremendous potential to cause real impacts, within minutes, for millions of users.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

-Information of the dataset:

Information

```
news.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 0 to 21416
Data columns (total 4 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   label    44898 non-null  int64 
dtypes: int64(1), object(3)
memory usage: 1.7+ MB
```

-Description of the dataset:

```
news.describe()
```

	label
count	44898.000000
mean	0.477015
std	0.499477
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

Data Sources and their formats

There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news.

Data Preprocessing Done

In data pre-processing, I have done the various steps to clean the dataset, as the dataset contains the comment that are in object datatype, wh ich cannot be read by the model, so before giving the features to the model I had to convert that object datatype to meaningful data and that can be understand by the model, so for this I have used the NLP (Natural Processing Language).

“Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence (AI) concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.”

Data Inputs- Logic- Output Relationships

Used TF-IDF Vectorizer to encode the comments section.

“TfidfVectorizer is the base building block of many NLP pipelines. It is a simple technique to vectorize text documents i.e., transform sentences into arrays of numbers and use them in subsequent tasks.”

Hardware and Software Requirements and Tools Used

Hardware required: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software/s required: -

1.Anaconda

Libraries required:

To run the program and to build the model we need some basic libraries as follows

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.ensemble import AdaBoostClassifier,GradientBoostingClassifier,RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import *
from sklearn.model_selection import KFold,cross_val_score

import pickle
import warnings
warnings.filterwarnings('ignore')
```

MODEL/S DEVELOPMENT AND EVALUATION

Identification of possible problem-solving approaches (methods)

- **EDA**
- **Description**
- **Visualization**
- **Data cleaning**
- **Data Pre-processing (NLP)**
- **Word Cloud**
- **Encoding**
- **Model Building**
- **Select the best model**
- **Cross-Validation**

Testing of Identified Approaches (Algorithms)

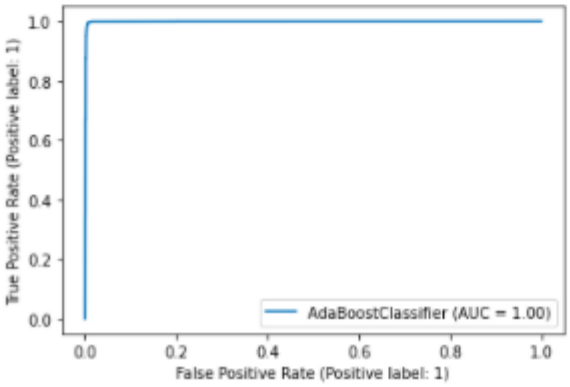
Algorithms used for the training and testing:

- AdaBoost Classifier
- GradientBoosting Classifier
- KNeighbors Classifier
- RandomForest Classifier
- Logistic Regression
- Decision Tree

Run and Evaluate selected models

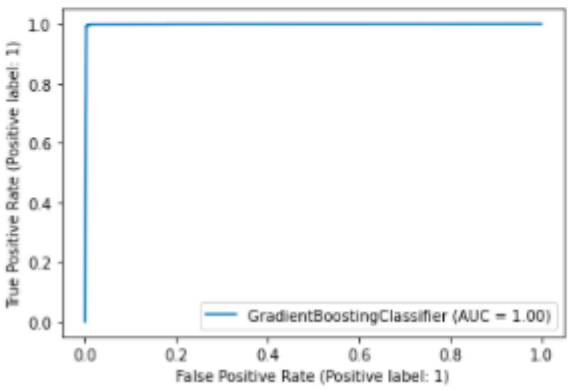
- AdaBoost Classifier

```
 ----- Train Result -----

Accuracy Score: 0.9958423662875301

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     17634
           1       0.99      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17537    97]
 [   43 15996]]

 ----- Test Result -----

Accuracy Score: 0.9941202672605791

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      5847
           1       0.99      1.00      0.99      5378

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225


 ----- Confusion matrix -----
 [[5804   43]
 [  23 5355]]

 ----- Roc Curve -----
```



- GradientBoosting Classifier

```
 ----- Train Result -----

Accuracy Score: 0.9973569328542156

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17571    63]
 [   26 16013]]

 ----- Test Result -----

Accuracy Score: 0.9942984409799555

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      5847
           1       0.99      1.00      0.99      5378

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225


 ----- Confusion matrix -----
 [[5809   38]
 [  26 5352]]

 ----- Roc Curve -----
```

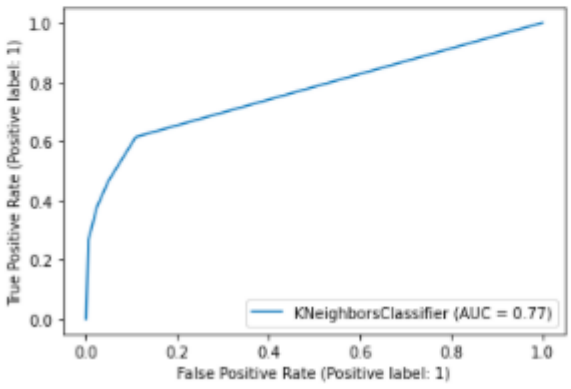


- KNeighbors Classifier

```
 ----- Train Result -----

Accuracy Score: 0.7449588691236302

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       0.68      0.98      0.80     17634
           1       0.96      0.48      0.64     16039

    accuracy                           0.74     33673
   macro avg       0.82      0.73      0.72     33673
weighted avg       0.81      0.74      0.73     33673


 ----- Confusion matrix -----
 [[17333   301]
 [ 8287  7752]]

 ----- Test Result -----

Accuracy Score: 0.6896213808463252

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       0.63      0.98      0.77      5847
           1       0.94      0.38      0.54      5378

    accuracy                           0.69     11225
   macro avg       0.78      0.68      0.65     11225
weighted avg       0.78      0.69      0.66     11225
```

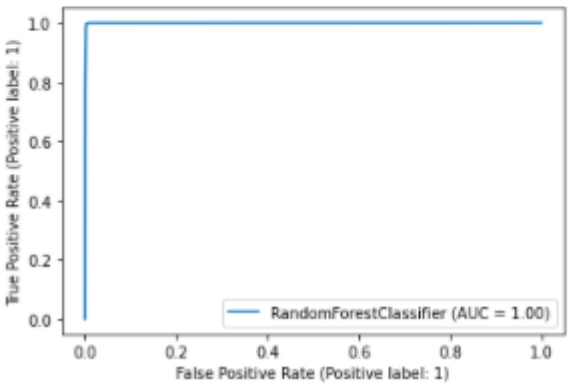
click to scroll output; double click to hide

```
 ----- Confusion matrix -----
 [[5708  139]
 [3345 2033]]

 ----- Roc Curve -----
```



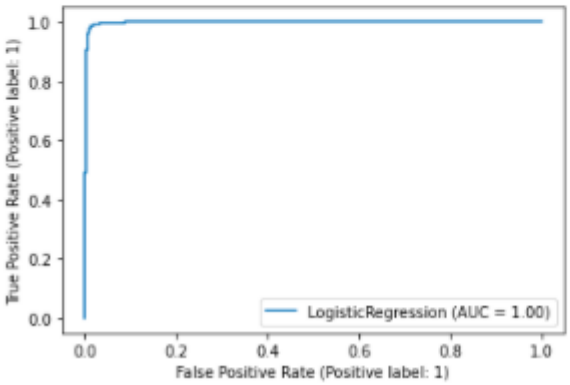
- # RandomForest Classifier

```
 ----- Train Result -----

Accuracy Score: 0.9999703026163395

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


 ----- Confusion matrix -----
 [[17634     0]
 [    1 16038]]

 ----- Test Result -----

Accuracy Score: 0.9970601336302896

 ----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5847
           1       1.00      1.00      1.00      5378

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225


 ----- Confusion matrix -----
 [[5826   21]
 [  12 5366]]

 ----- Roc Curve -----
```



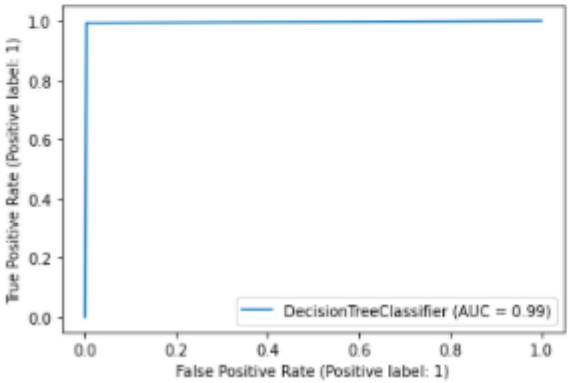
- # Logistic Regression

```
----- Train Result -----

Accuracy Score: 0.9914471535057762

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     17634
           1       0.99      0.99      0.99     16039

    accuracy                           0.99     33673
   macro avg       0.99      0.99      0.99     33673
weighted avg       0.99      0.99      0.99     33673


----- Confusion matrix -----
[[17454   180]
 [  108 15931]]

----- Test Result -----

Accuracy Score: 0.9851224944320712

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      5847
           1       0.98      0.99      0.98      5378

    accuracy                           0.99     11225
   macro avg       0.98      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225


----- Confusion matrix -----
[[5747  100]
 [  67 5311]]

----- Roc Curve -----
```



- # Decision Tree

```
----- Train Result -----

Accuracy Score: 0.9999703026163395

----- Classification Report -----
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17634
           1       1.00      1.00      1.00     16039

    accuracy                           1.00     33673
   macro avg       1.00      1.00      1.00     33673
weighted avg       1.00      1.00      1.00     33673


----- Confusion matrix -----
[[17634     0]
[    1 16038]]

----- Test Result -----

Accuracy Score: 0.9948329621380846

----- Classification Report -----
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      5847
           1       1.00      0.99      0.99      5378

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225


----- Confusion matrix -----
[[5830   17]
[  41 5337]]

----- Roc Curve -----
```



Interpretation Of the Results

RandomForest Classifier is giving the best result as compared to others.

CONCLUSION

Key Findings and Conclusions of the Study

Apply computing theory, languages, and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analyses. Formulate and use appropriate models of data analysis to solve hidden solutions to business-related challenges. Perform well in a group.