PROGRESS REPORT

Fake News Detection and Sentiment Analysis

(This report contains progress of 1-5 weeks mentioned in Gantt Chart)

Submitted to

Syed Asim Ali

by

Muhammad Anas Atiq ---- B19102067

Abdul Hannan Shaikh ----- B19102003

Ameer Hamza Khan ------- B19102016

Muhammad Anas --------- B19102066

Current Progress:

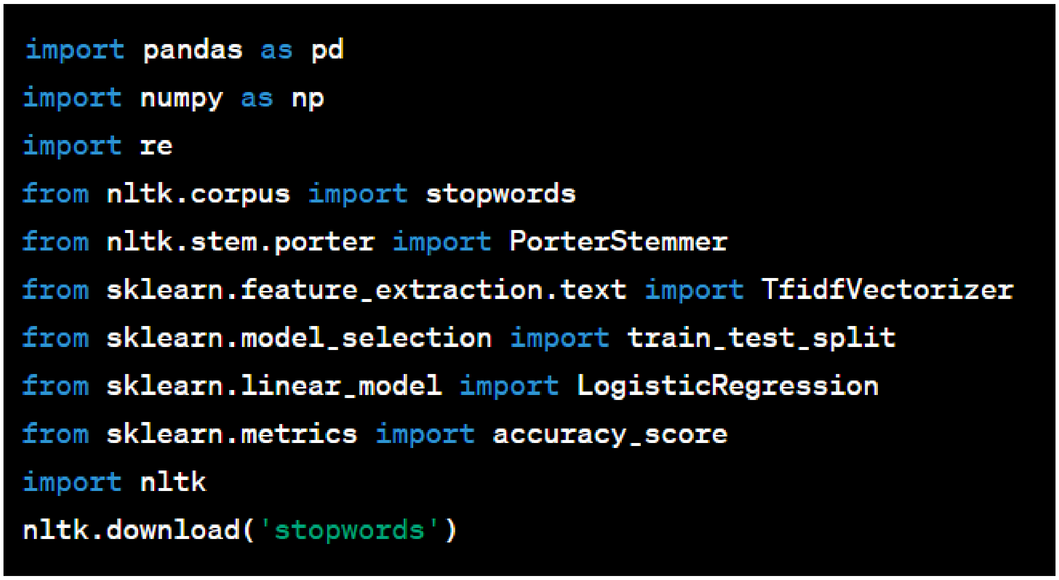
Although this project has made significant progress in detecting fake news, there are several areas that need improvement. Firstly, the dataset should be expanded to include a wider variety of news articles to enhance the model's ability to generalize. Secondly, advanced NLP techniques, such as lemmatization, sentiment analysis, and deep learning-based embeddings, should be explored to achieve more accurate classification results. Moreover, practical implications of the model's predictions should be considered, including addressing issues related to false positives and false negatives. Additionally, a more in-depth analysis of model interpretability and a broader range of evaluation metrics can provide a more comprehensive assessment of the model's performance.

Project Overview:

The purpose of this project is to develop a fake news detection system using machine learning techniques. Fake news has become a growing concern in recent years, and this project aims to build a model that can classify news articles as either real or fake based on their content.

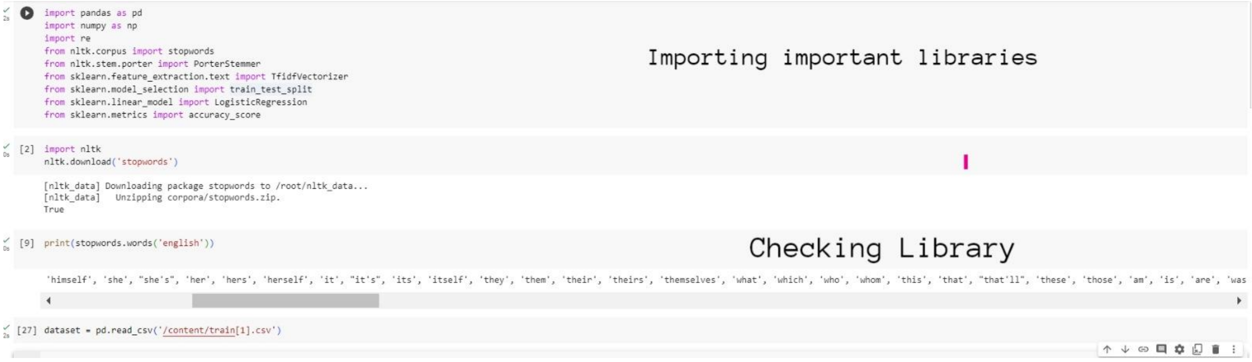
1- **Importing Libraries**:

Code:



```
import pandas as pd
import numpy as np
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import nltk
nltk.download('stopwords')
```

Output:



```
import pandas as pd
import numpy as np
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
[2] import nltk
    nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

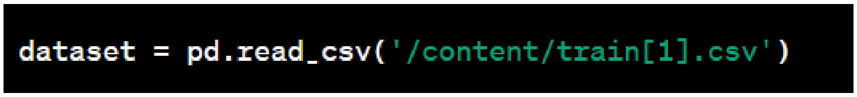
```
[9] print(stopwords.words('english'))

 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was

[27] dataset = pd.read_csv('/content/train[1].csv')
```

Explanation:

In this section, we import the necessary Python libraries for data manipulation, text preprocessing, machine learning, and natural language processing (NLP). The NLTK library is used to download English stopwords, which are common words that are often excluded from text analysis.

2- **Reading the Dataset**:

Code:



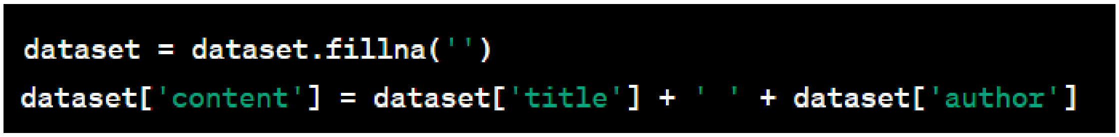
```
dataset = pd.read_csv('/content/train[1].csv')
```

Explanation:

This code reads the dataset from a CSV file named 'train[1].csv' using the Pandas library. The dataset contains information about news articles, which will be used for training and testing the fake news detection model.

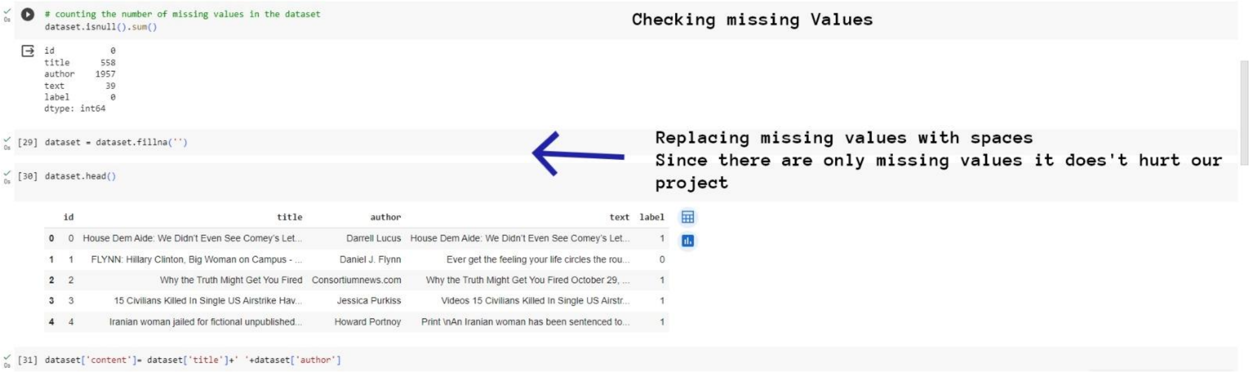
3- **Data Preprocessing**:

Code:



```
dataset = dataset.fillna('')
dataset['content'] = dataset['title'] + ' ' + dataset['author']
```

Output:



```
# counting the number of missing values in the dataset
dataset.isnull().sum()
```

Checking missing Values

```
id          0
title     558
author   1957
text       39
label       0
dtype: int64
```

```
[29] dataset = dataset.fillna('')
```

```
[30] dataset.head()
```

Replacing missing values with spaces
Since there are only missing values it does't hurt our project

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

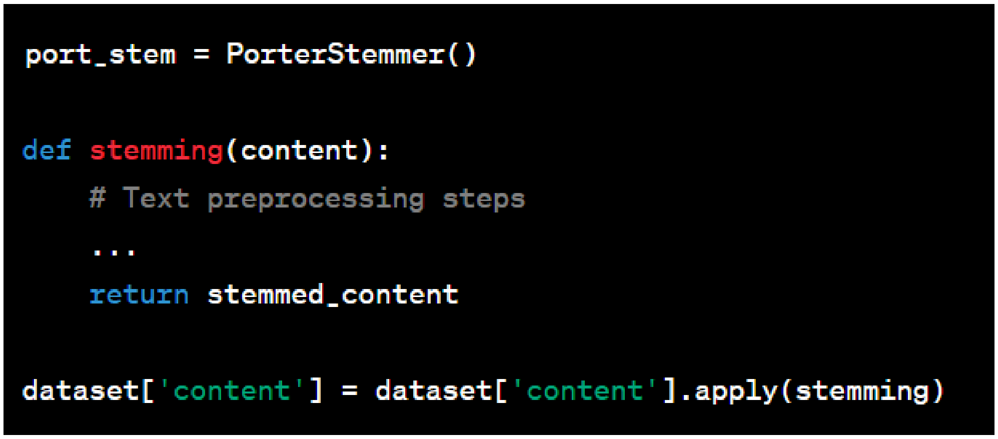
```
[31] dataset['content']= dataset['title']+' '+dataset['author']
```

Explanation:

In this section, the code fills any missing values in the dataset with empty strings. It then combines the 'title' and 'author' columns to create a new column named 'content'. This 'content' column will be the basis for feature extraction and analysis.

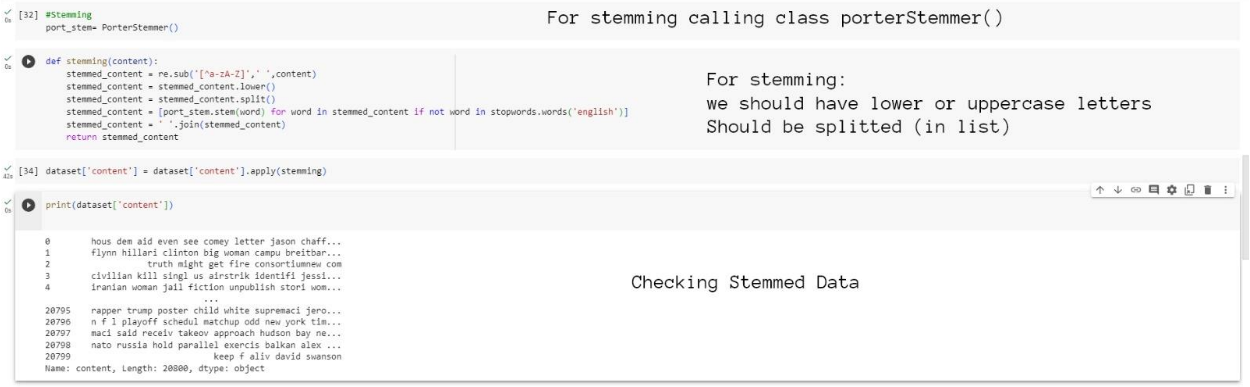
4- **Stemming**:

Code:



```
port_stem = PorterStemmer()

def stemming(content):
    # Text preprocessing steps
    ...
    return stemmed_content

dataset['content'] = dataset['content'].apply(stemming)
```

Output:



```
[32] #Stemming
     port_stem= PorterStemmer()
```

For stemming calling class porterStemmer()

```
def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

For stemming:
we should have lower or uppercase letters
Should be splitted (in list)

```
[34] dataset['content'] = dataset['content'].apply(stemming)
```

```
print(dataset['content'])
```

```
0        hous dem aid even see comey letter jason chaff...
1        flynn hillari clinton big woman campu breitbar...
2                   truth might get fire consortiumnew com
3        civilian kill singl us airstrik identifi jessi...
4        iranian woman jail fiction unpublish stori wom...
                               ...
20795    rapper trump poster child white supremaci jero...
20796    n f l playoff schedul matchup odd new york tim...
20797    maci said receiv takeov approach hudson bay ne...
20798    nato russia hold parallel exercis balkan alex ...
20799                            keep f aliv david swanson
Name: content, Length: 20800, dtype: object
```

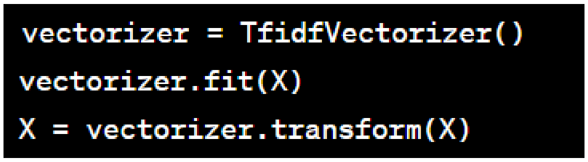
Checking Stemmed Data

Explanation:

The code initializes the Porter Stemmer for stemming text data. It defines a custom stemming function, which preprocesses the text by converting it to lowercase, removing non-alphabetic characters, tokenizing it, applying stemming, and removing English stopwords. This function is then applied to the 'content' column of the dataset to preprocess the text.

5- **TF-IDF Vectorization**:

Code:



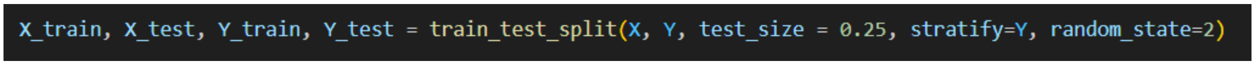
```
vectorizer = TfidfVectorizer()
vectorizer.fit(X)
X = vectorizer.transform(X)
```

Explanation:

in this snippet, the code initializes a TF-IDF vectorizer and fits it to the preprocessed text data. The TF-IDF vectorizer converts the text data into numerical vectors, which can be used for machine learning. The resulting TF-IDF features are stored in the variable X.

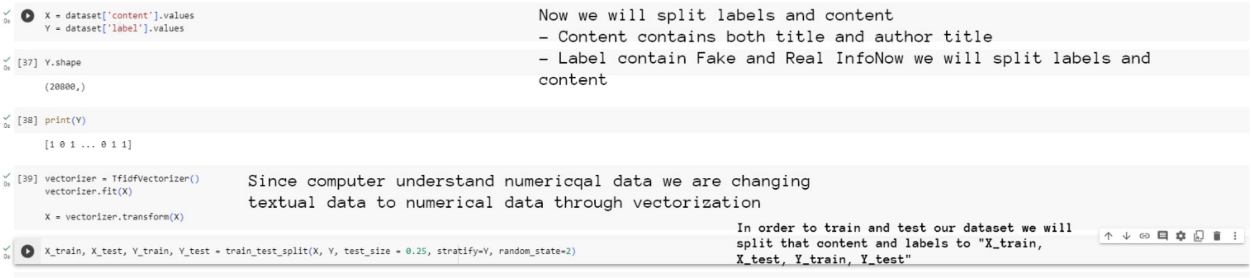
6- **Splitting the data**:

Code:



```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, stratify=Y, random_state=2)
```

Output:



```
X = dataset['content'].values
Y = dataset['label'].values
```

Now we will split labels and content
– Content contains both title and author title
– Label contain Fake and Real InfoNow we will split labels and content

```
[37] Y.shape

(20800,)
```

```
[38] print(Y)

[1 0 1 ... 0 1 1]
```

```
[39] vectorizer = TfidfVectorizer()
     vectorizer.fit(X)

     X = vectorizer.transform(X)
```

Since computer understand numericqal data we are changing textual data to numerical data through vectorization

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, stratify=Y, random_state=2)
```

In order to train and test our dataset we will split that content and labels to "X_train, X_test, Y_train, Y_test"

Explanation:

This code splits the dataset into training and testing sets using the train_test_split function from Scikit-Learn. The data is divided into 75% training and 25% testing sets, with stratification to maintain the class distribution and a fixed random seed for reproducibility.

7- **Training a Logistic regression model**:

```
model = LogisticRegression()
model.fit(X_train, Y_train)
```

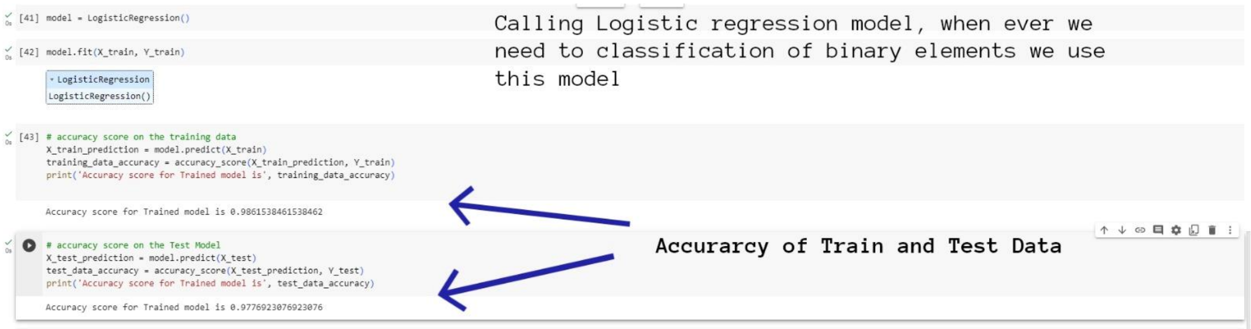
Here, a Logistic Regression model is initialized and trained on the training data. Logistic Regression is a simple and effective algorithm for binary classification tasks like fake news detection.

8- **Model Evaluation**:

Code:

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

Output:



```
[41] model = LogisticRegression()

[42] model.fit(X_train, Y_train)

     ▾ LogisticRegression
     LogisticRegression()

[43] # accuracy score on the training data
     X_train_prediction = model.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
     print('Accuracy score for Trained model is', training_data_accuracy)

     Accuracy score for Trained model is 0.9861538461538462

     # accuracy score on the Test Model
     X_test_prediction = model.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
     print('Accuracy score for Trained model is', test_data_accuracy)

     Accuracy score for Trained model is 0.9776923076923076
```

Calling Logistic regression model, when ever we need to classification of binary elements we use this model

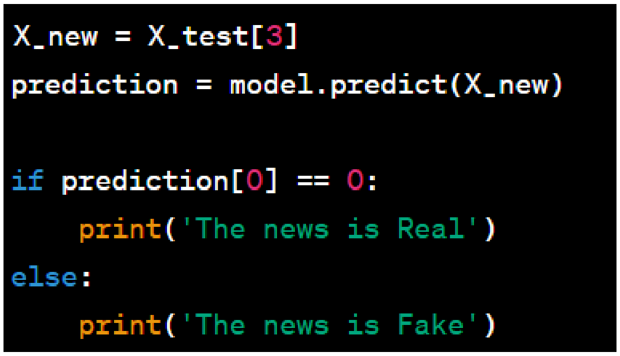
Accurarcy of Train and Test Data

Explanation:

These lines of code assess the model's performance. It calculates the accuracy of the model on both the training and testing data using the accuracy_score function.

9- **Making Predictions**:

Code:



```
X_new = X_test[3]
prediction = model.predict(X_new)

if prediction[0] == 0:
    print('The news is Real')
else:
    print('The news is Fake')
```

Explanation:

This snippet demonstrates how to use the trained model to make predictions. It selects a specific news article from the test data, predicts its authenticity (real or fake), and prints the result.

Future Progressions:

In the course of this project, significant strides have been made towards the development of a fake news detection system. Notably, the project has successfully preprocessed the dataset, implemented a TF-IDF vectorization approach, and trained a logistic regression model, demonstrating the feasibility of automated news classification. However, there are several key progressions yet to be made. First, the project is yet to initiate the analysis of sentiment in news articles, a critical aspect that can offer a richer context for classification. Additionally, the integration of a user-friendly interface, model refinement, prototype development, rigorous testing, and optimization remain important milestones for the project to evolve from a concept to a practical, reliable, and scalable system.