Report on Fake News Detection

1. Data Exploration and Model Building Steps :

The dataset comprises news articles with columns labeled `title`, `text`, and `label`, where the `label` indicates whether a news article is `FAKE` or `REAL`. Our primary goal is to develop a machine-learning model that can accurately classify these articles based on their content.

To begin with, the dataset is loaded using pandas to inspect its structure. The next step involves splitting the data into training and testing sets, ensuring the model's performance can be evaluated on unseen data. This is achieved using the `train\_test\_split` function from scikit-learn, which divides the data into a training set (80%) and a testing set (20%).

For the text data to be usable in machine learning algorithms, it is vectorized using `TfidfVectorizer` from scikit-learn. This converts the text into numerical features while also normalizing the importance of terms based on their frequency across the documents.

The machine learning model chosen for this task is the `PassiveAggressiveClassifier`, known for its effectiveness in online learning and text classification tasks. The classifier is initialized and trained on the vectorized training data. After training, the model's predictions are made on the testing set.

To evaluate the model's performance, the accuracy is calculated and a confusion matrix is constructed. The accuracy provides a straightforward measure of the proportion of correctly classified instances, while the confusion matrix offers deeper insights into the model's performance by detailing the true positives, true negatives, false positives, and false negatives.

1. Code Snippets :

* Loading the dataset is straightforward with pandas:

**import pandas as pd**

**file\_path = ‘news.csv'**

**news\_df = pd.read\_csv(file\_path)**

**print(news\_df.head())**

* Splitting the data into training and testing sets is accomplished with the following code:

**from sklearn.model\_selection import train\_test\_split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(news\_df['text'], news\_df['label'], test\_size=0.2, random\_state=7)**

* Text vectorization using `TfidfVectorizer` is implemented as:

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**tfidf\_vectorizer = TfidfVectorizer(stop\_words='english', max\_df=0.7)**

**tfidf\_train = tfidf\_vectorizer.fit\_transform(X\_train)**

**tfidf\_test = tfidf\_vectorizer.transform(X\_test)**

* Building and training the `PassiveAggressiveClassifier` involves:

**from sklearn.linear\_model import PassiveAggressiveClassifier**

**pac = PassiveAggressiveClassifier(max\_iter=50)**

**pac.fit(tfidf\_train, y\_train)**

**y\_pred = pac.predict(tfidf\_test)**

* Finally, evaluating the model with accuracy and confusion matrix is done as follows:

**from sklearn.metrics import accuracy\_score, confusion\_matrix**

**score = accuracy\_score(y\_test, y\_pred)**

**print(f'Accuracy: {round(score\*100, 2)}%')**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

**print('Confusion Matrix:')**

**print(conf\_matrix)**

1. Model’s Performance Evaluation :

The model's performance is noteworthy, achieving an accuracy of 92.82%. This high accuracy indicates the model's effectiveness in distinguishing between fake and real news articles. The confusion matrix further breaks down the model's performance, correctly classifying 591 fake news articles and 585 real news articles. However, it misclassified 47 fake news articles as real and 44 real news articles as fake. Despite the inherent challenges in text classification, these results underscore the robustness of the `PassiveAggressiveClassifier` for this task.

Overall, the model demonstrates strong performance, with a high accuracy and well-distributed confusion matrix, making it a reliable tool for fake news detection.

