Detailed Report on Spam News Detection and Handwritten Digit Recognition

Name: Manojkumar
Phone No: 6302678854
Email: nagarammanojkumar3@gmail.com
GitHub Project Link: AI-MachineLearning

Spam News Detection

Introduction

Spam news detection involves identifying whether a piece of news is genuine (true) or fabricated (fake). This task uses Natural Language Processing (NLP) and machine learning to classify news articles based on their textual content.

Steps Involved:

1. **Data Collection**:
   - News datasets containing labeled articles (True and Fake) are collected. Each article is associated with a label: 0 for true and 1 for fake.
2. **Data Preprocessing**:
   - **Cleaning**: Convert text to lowercase, remove non-alphabetical characters (numbers, symbols), and clean any unwanted tokens.
   - **Tokenization and Lemmatization**: Break text into individual words and reduce them to their base form.
   - **Stop Words Removal**: Remove common words (e.g., "the", "is", "in") that do not contribute to the meaning.
3. **Feature Extraction**:
   - **TF-IDF Vectorizer**: Converts the cleaned text data into numerical data that can be used by machine learning models. It calculates the importance of words in the context of the dataset.
4. **Model Training**:

   - A **Multinomial Naive Bayes** classifier is used, which works well for text classification tasks like this.

5. **Model Evaluation**:
   - Accuracy scores are calculated for both the training and testing datasets to evaluate the model's performance.
6. **Prediction**:
   - The model can predict whether new, unseen text is true or fake.

Step 1: import libraries.

```
import numpy as np
import pandas as pd
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer  #
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
```

Step 2: Read DataSet using pandas(pd.read) function.

```
true_news = pd.read_csv('True_News.csv')
fake_news = pd.read_csv('Fake_News.csv')
```

Step 3: Create new column for label which is used to assign True or False Value(0, 1) to our text

```
true_news['label'] = 0
fake_news['label'] = 1
dataset1 = true_news[['text', 'label']]
dataset2 = fake_news[['text', 'label']]

# This dataset will have two data points they are (Text(From all datasets),
```

Label(0, 1)).

```
dataset = pd.concat([dataset1, dataset2])
```

Step 4: Shuffling dataset.

```
dataset = dataset.sample(frac=1)
```

Step5: Data Cleaning Function.

```
ps = WordNetLemmatizer()
stopwords = stopwords.words('english')
nltk.download('wordnet')

def clean_row(row):
    row = row.lower()
    row = re.sub('[^a-zA-Z]', ' ', row)  # this line replace all the nums and
special with space(" ")
    # split
    token = row.split()

    news_txt = [ps.lemmatize(word) for word in token if word not in stopwords]

    cleanned_news = " ".join(news_txt)
    return cleanned_news

dataset['text'] = dataset['text'].apply(lambda x: clean_row(x))  # clean data
using function clean_row(x)
```

Step 7: Converting text data into numerical data

```
vectorizer = TfidfVectorizer(max_features=50000, lowercase=False,
ngram_range=(1, 2))
x = dataset.iloc[:35000, 0]
y = dataset.iloc[:35000, 1]
```

Step 8: Train-Test-Split

```
train_data, test_data, train_label, test_label = train_test_split(x, y,
test_size=0.2, random_state=0)
```

Step 9: vectorized Data and convert it into an array.

```
vec_train_data = vectorizer.fit_transform(train_data).toarray()
vec_test_data = vectorizer.transform(test_data).toarray()
training_data = pd.DataFrame(vec_train_data,
columns=vectorizer.get_feature_names_out())
testing_data = pd.DataFrame(vec_test_data,
columns=vectorizer.get_feature_names_out())
```

Step 10: Train Model

```
clf = MultinomialNB()
clf.fit(vec_train_data, train_label)
```

Step 11: Predict labels for test and train and print their Accuracy.

```
y_pred_test = clf.predict(vec_test_data)
y_pred_train = clf.predict(vec_train_data)

print(f"\n The accuracy of testing data is: \n{accuracy_score(test_label,
y_pred_test)}\n")
print(f"\n The accuracy of training data is: \n{accuracy_score(train_label,
y_pred_train)}\n")
```

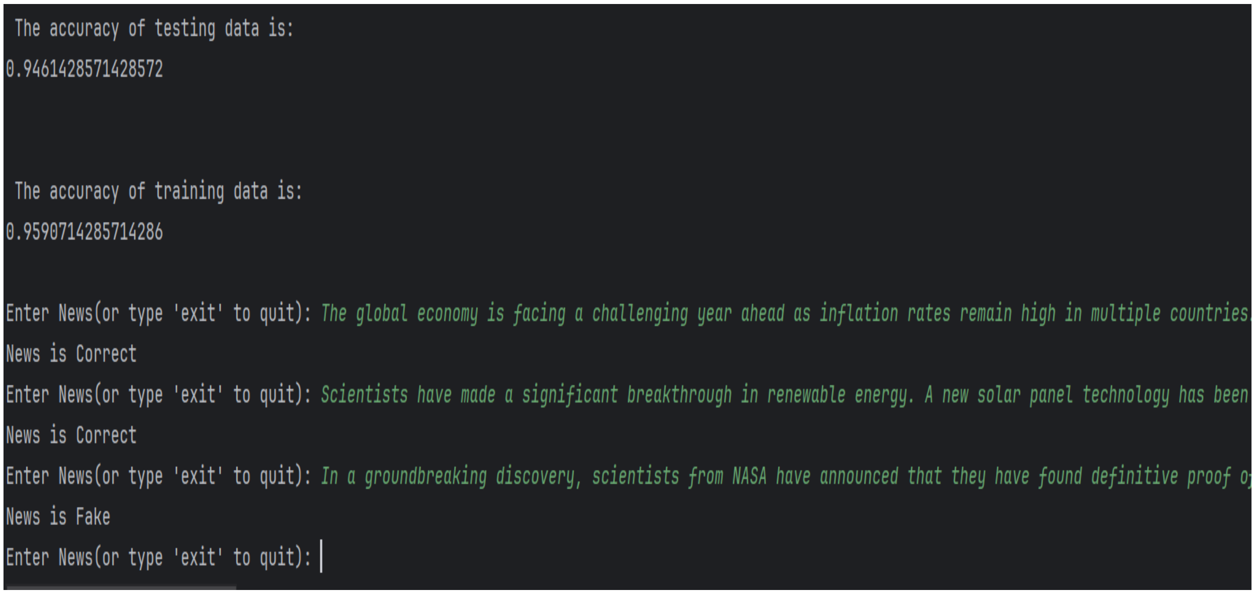
Step 12: Continuous input for predictions.

```
while True:
    txt = input("Enter News(or type 'exit' to quit): ")
    if txt.lower() == 'exit':
        print("Exiting the program.")
        break
```

```
# cleaning input text and make predictions.
news = clean_row(str(txt))
pred = clf.predict(vectorizer.transform([news]).toarray())

if pred == 0:
    print("News is Correct")
else:
    print("News is Fake")
```

OUTPUT:-



```
 The accuracy of testing data is:
0.9461428571428572



 The accuracy of training data is:
0.9590714285714286


Enter News(or type 'exit' to quit): The global economy is facing a challenging year ahead as inflation rates remain high in multiple countries
News is Correct
Enter News(or type 'exit' to quit): Scientists have made a significant breakthrough in renewable energy. A new solar panel technology has been
News is Correct
Enter News(or type 'exit' to quit): In a groundbreaking discovery, scientists from NASA have announced that they have found definitive proof o
News is Fake
Enter News(or type 'exit' to quit): |
```

Conclusion

- **Spam News Detection** is an important task for filtering out misleading information on the internet. By using machine learning algorithms like Naive Bayes, we can achieve high accuracy in classifying news articles as either true or fake.

Handwritten Digit Recognition

Introduction

Handwritten digit recognition involves identifying the digits (0-9) in images of handwritten text. Machine learning models, especially Convolutional Neural Networks (CNNs), are typically used to solve this problem.

Steps Involved:

1. **Data Collection**:
   - Datasets like MNIST contain images of handwritten digits (labeled 0-9).
2. **Data Preprocessing**:
   - Images are resized, normalized, and flattened for input to machine learning models.
3. **Model Training**:
   - A deep learning model, such as a CNN, is trained on the labeled images.
4. **Prediction**:
   - The model is used to predict the digit in a new image.

Code Explanation

The following is a detailed breakdown of the provided code for Spam News Detection.

Code Overview:

1. **Library Imports**:
   - Libraries such as pandas, nltk, sklearn are imported for data manipulation, natural language processing, and machine learning.
2. **Dataset Loading**:
   - The datasets containing true and fake news articles are loaded using pandas.read_csv.
3. **Label Assignment**:

- A new column is added to the datasets (True_News.csv and Fake_News.csv) to indicate whether the news is true (0) or fake (1).

4. **Data Preprocessing**:
   - The clean_row() function is defined to clean the text data by converting it to lowercase, removing non-alphabet characters, and lemmatizing words.
5. **TF-IDF Vectorization**:
   - The text data is converted into a numerical format using **TF-IDF Vectorizer**. This is a crucial step in transforming textual data into a format that machine learning algorithms can process.
6. **Train-Test Split**:
   - The dataset is split into training and testing sets using train_test_split from sklearn.
7. **Model Training**:
   - A **Multinomial Naive Bayes** classifier is trained on the transformed data.
8. **Evaluation**:
   - Accuracy scores are calculated for both the training and test datasets.
9. **Prediction**:
   - The model allows continuous input for predictions, where the user can enter text, and the system will predict whether it’s true or fake news.

Handwritten Digit Recognition is a classic example of image recognition using deep learning models. The MNIST dataset is a standard benchmark for testing models in this field.