Data Inspection and Preparation Report: Fake News Detection Notebook

1. Introduction

In this report, I describe the steps I observed while inspecting the Kaggle notebook provided by my teacher. The notebook is titled 'Fake News Detection' and focuses on preparing textual data for further analysis. My goal was to review the dataset, understand how the data was cleaned, and identify the methods used before training. The output of the notebook was a CSV file named 'manual_testing.csv' which contained the processed data ready for testing.

2. Importing Libraries

The first section of the notebook imports several Python libraries required for data handling, visualization, and evaluation. These include pandas, numpy, seaborn, and matplotlib for data manipulation and plotting, along with modules from sklearn for data splitting and evaluation. Regular expression tools (re) and string manipulation are also imported for cleaning text data.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
import re
import string
```

This step ensures that all necessary tools are loaded so that the data can be inspected, analyzed, and prepared properly.

3. Importing the Dataset

In the next section, the notebook imports two datasets: one containing fake news articles and another containing true news articles. Both datasets are stored in CSV format and are read into pandas DataFrames.

```
df_fake = pd.read_csv('../input/fake-news-detection/Fake.csv')
df_true = pd.read_csv('../input/fake-news-detection/True.csv')
```

After importing, the notebook displays the first few rows using head() to check the structure of each DataFrame. It also uses shape to confirm the total number of rows and columns in each dataset.

4. Adding the Class Column

To help differentiate between fake and true news, a new column named 'class' is added. Fake articles are labeled as 0, and true articles are labeled as 1. This prepares the data for supervised learning later on.

```
df_fake['class'] = 0
df_true['class'] = 1
```

5. Combining Both Datasets

Once both DataFrames have their class labels, they are combined into a single dataset using the concat() function. This allows for unified inspection and cleaning.

```
df = pd.concat([df_fake, df_true], axis=0)
```

I confirmed that after merging, the index was reset and the columns remained consistent across both datasets.

6. Cleaning the Text Data

The notebook includes a custom function for text cleaning. This function removes URLs, punctuation, and non-alphabetical characters. It also converts all text to lowercase and removes extra spaces. This step is important to make sure the model receives clean and consistent input.

```
def wordopt(text):
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    text = text.lower()
    text = text.split()
    return ' '.join(text)

df['text'] = df['text'].apply(wordopt)
```

After applying this cleaning function, the text data becomes much more readable and standardized. At this stage, I could clearly see that unwanted characters and noise were removed.

7. Splitting Data for Manual Testing

Towards the end of the notebook, part of the dataset is saved into a new CSV file called 'manual_testing.csv'. This file contains a small portion of the data which can later be used to manually check predictions made by the model.

```
df.to_csv('manual_testing.csv', index=False)
```

This step completes the data preparation stage. The resulting file includes cleaned and labeled data ready for testing.

8. Conclusion

In summary, I inspected how the notebook handled data loading, cleaning, and preparation. The dataset was properly labeled, merged, and cleaned before being exported as 'manual_testing.csv'. These steps are crucial for ensuring that the data is accurate, consistent, and suitable for later analysis.

References

1. Kaggle: Fake News Detection Notebook by tdquang (https://www.kaggle.com/code/tdquang/fake-news-detection)

2. Scikit-learn Documentation (https://scikit-learn.org/stable/documentation.html)

3. Template and reformatting by ChatGPT (OpenAI, GPT-5).