Assignment 1: Fake News Detection via Text Classification

Before start working on this assignment, please investigate Text Analytics for Beginners using NLTK and A Gentle Introduction to Natural Language Processing carefully. These are Python NTLK (natural language toolkit) sentiment analysis tutorials. You will learn how to create and develop sentiment analysis using Python and NLP methodologies.

For this assignment, we will propose and implement different algorithms for fake news detection. You can access the fake new + real news dataset from the link.

If you don't have a Python platform on your local machine, you are welcome to use Google Colab with a configured environment.

Task 1 – Explore Essential Information from Text Data and Preprocessing

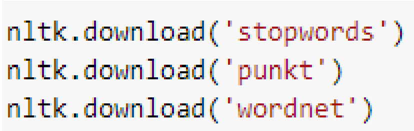
In this dataset, you will have access to both real news and fake news. Before machine learning, please explore the essential information from the textual data, e.g., the most commonly used words in the collection/real news/fake news. Note that you need to call "stopword removal" and "lemmatization" functions before calculating the word frequency.

Then, please answer the following questions:

1. What are the most commonly used words (top 100) in the collection, the most commonly used words (top 100) in the real news and most commonly used words (top 100) in the fake news?
2. By reading the preprocessed textual data, can you easily tell the difference between the real news and fake news? What does the strongest feature set (for machine learning) look like?

Hint:

1. The NLTK' functions usually take some time to process the data. It is recommended to use sample data to test the code before applying it to the whole dataset.
2. You might need to download the corresponding resource after import nltk if you have not used it before. Such as:



```
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

3. You can report the common words and their frequency in a table. In addition, common words can also be visualized by WordClouds.

Task 2 – Build Machine Learning Model

In this task, please build machine learning text classification models to classify all the news into "real" and "fake" categories. You will need to split the data into training and testing collections for this task (e.g., 70% and 30%). Note that you can choose different kinds of algorithms, e.g., regression and MultinomialNB, and you can feed machine learning algorithms different kinds of feature sets, e.g., "term-frequency" and "TFIDF" as the feature set.

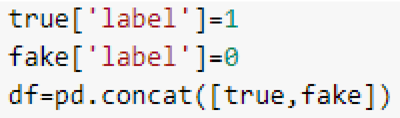
Please report the performance of different algorithms in the following table:

ML Model	Feature	Precision	Recall	Accuracy
Model 1	Feature 1			
Model 1	Feature 2			
Model 2	Feature 1			
…	…			

Please provide error analysis for the best performed results (top 2) by using "Test Confusion Matrix". Please explain this outcome.

Hint:

1. You can merge the true and fake datasets by using pd.concat function. eg:



```
true['label']=1
fake['label']=0
df=pd.concat([true,fake])
```

Task 3 – Enhanced NLP Features

After applying "POS Tagging", you can locate specific kinds of words in the collection, e.g., nouns and verbs. Please build additional classifier(s) to classify the news by leveraging POS information, e.g., only use the "nouns" or "adj" + "noun" as features. Did you witness the performance improvement (compared with the result from task 2)? why?

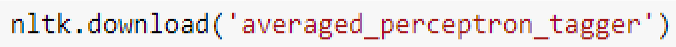
Please report the performance of different algorithms in the following table:

ML Model	Feature	Filter	Precision	Recall	Accuracy
Model 1	Feature 1	Filter 1			
Model 1	Feature 2	Filter 1			
Model 2	Feature 1	Filter 2			
…	…				

e.g., "stopword removed", "noun" "noun+adj"and as the filter set.

Hint:

1. You might need to download the POS Tagging resource via NLTK if you have not used it before.



```
nltk.download('averaged_perceptron_tagger')
```

2. POS tagset can be found on page18, NLP - Introduction.pdf.
3. Similar to stopword removal, you can filter the tokens by giving a selected POS tagset.

Task 4 – Future Work

Please tell me what you plan to do in the future if you want to further enhance the performance of the machine learning models, e.g., enhancing learning models? Or investigating novel features? Before you propose your own idea, you may want to read the following papers:

1. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques
2. Supervised Learning for Fake News Detection

Please design a simple experiment with GPT (or GPT API), and tell me if we can potentially use GPT to address this problem.

Submission

You need to submit **two separate files** (DO NOT compress your files to .zip) to Canvas: 1) your code; 2) your results doc.

1. Your code can be .py file or .ipynb file. For those who use Colab to complete this assignment, we would appreciate it if you could provide a shared link (General access: Anyone with the link can view it) to make it easy for us to review your code.
2. Your results doc can be a .pdf file or .docx file. The file needs to contain the results (e.g., figures, tables) of each task and your analysis.