NLP Project for Beginners on Text Processing and Classification
Project Overview

Business Overview

Have you ever wondered how the machine deals with the text we write, the sentence we speak? How the machine takes essential decisions based on just the text? Natural Language Processing, often abbreviated as NLP, gives the ability to machines to understand, read, and get meaningful insights from human language. Basically, NLP is the automatic handling of human languages. Nowadays, NLP is prospering due to the large availability of data and computational power. NLP has dug down its routes from healthcare, media, finance to human resources. It is growing with each coming day. In this series of projects, we will introduce NLP and associated techniques in a very lucid manner. This project aims to give you a brief overview of text preprocessing and building a binary classification model on processed data.

Aim

To understand the basic text preprocessing and build a classification model.

Data Description

The dataset contains more than a thousand reviews about an application openly available to the public. The data includes reviews and sentiment, i.e., is the review positive or negative with various other variables.

Tech Stack

- ➔ Language: Python
- ➔ Libraries: pandas, seaborn, matplotlib, sklearn, nltk

Approach

1. Data Description and visualization
2. Introduction to NLTK library
3. Data Preprocessing
    a. Conversion to lower case
    b. Tokenization
    c. Stopwords removal
    d. Punctuation removal
    e. Stemming

4. Bag of Words
    a. Binary
    b. Non-binary
    c. N-grams
5. TF-IDF
6. Model Building and Accuracy
7. Predictions on new reviews

Modular Code Overview

```
Input
   |_Canva_review.xlsx

Output
   |_model.pkl
   |_model_lr.pkl

Source
   |_model.py
   |_processing.py
   |_utils.py

config.py

Engine.py

linear_model.ipynb

predict.py

README.md

requirements.txt
```

Once you unzip the modular_code.zip file, you can find the following folders within it.

1. Input
2. Output
3. Source

1. The input folder contains the data that we have for analysis. In our case, it contains Canva_reviews.xlsx.

2. The source folder contains all the modularized code for all the above steps in a modularized manner. It includes the following.
    a. model.py
    b. processing.py
    c. utils.py
       These all python files contain helpful functions which are being used in the Engine.py file.

3. The output folder contains all the pre-trained models and vectorizers. These models can be quickly loaded and used for future use, and the user need not have to train all the models from the beginning.

4. The config.py file contains all the configurations required for this project.

5. The Engine.py file is the main file that needs to be called to run the entire code in one go. It trains the model and saves it in the output folder.
   Note: Please check the README.md file for more information.

6. The linear_model.ipynb is the original notebook we saw in the videos.

7. The predict.py file is used to predict the probability of new reviews.
   Note: Please check the README.md file for more information.

8. The README.md file contains all the information on how to run particular files and more instructions to follow.

9. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command **pip install -r requirements.txt**

Project Takeaways

1. Understanding problem statement and the approach
2. Data Exploration and visualization
3. What is Tokenization?
4. Performing tokenization using word_tokenization from nltk library
5. What are stopwords?
6. How to remove stopwords?
7. Removing the punctuations

8. What is stemming?
9. Stemming using Porter Stemmer
10. Stemming using Lancaster Stemmer
11. Distributions of words in each document
12. Creating bag of words with binary
13. Creating bag of words without binary
14. Creating bag of words with N-grams
15. What is TF-IDF?
16. How to find TF-IDF score?
17. Building Logistic regression model
18. Predictions on new reviews