Multiclass Text Classification using Naive Bayes in Python
Project Overview

Business Overview

Natural Language Processing, often abbreviated as NLP, gives the ability to machines to understand, read, and get meaningful insights from human language. In the last project of this series, we understood how to perform text preprocessing and classification using Logistic regression which can be found here. While we dealt with binary classification, many of the fields are concerned about multiclass classification. This project aims to give you a brief overview of text classification where there are more than two classes available and build a classification model on processed data using the Naive Bayes algorithm. This project also explains the working of the Naive Bayes algorithm and related terminology.

Aim

To understand the Naive Bayes algorithm and build a multiclass classification model.

Data Description

The dataset contains more than two million customer complaints about consumer financial products. Amongst the various available columns, we have a column that contains the actual text of the complaint and one column containing the product for which the customer is raising the complaint.

Tech Stack

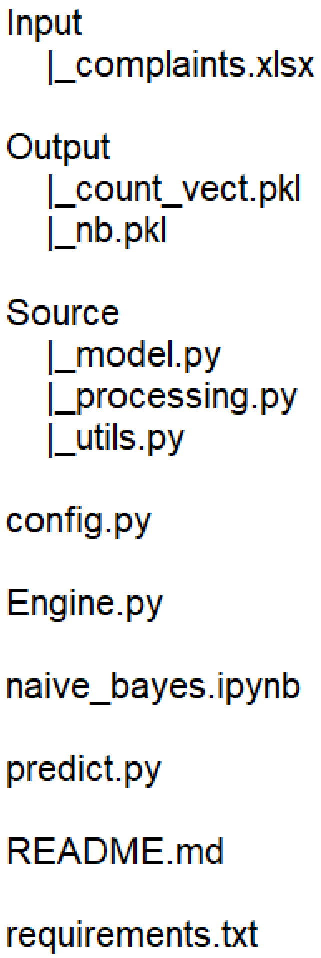
- ➔ Language: Python
- ➔ Libraries: pandas, seaborn, matplotlib, sklearn, nltk

Approach

1. Introduction to Naive Bayes algorithm
2. Data Description and visualization
3. Data Preprocessing
    a. Conversion to lower case
    b. Tokenization
    c. Stopwords removal
    d. Punctuation removal
4. Model Building and Accuracy

5. Predictions on new reviews

Modular Code Overview



```
Input
   |_complaints.xlsx

Output
   |_count_vect.pkl
   |_nb.pkl

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

config.py

Engine.py

naive_bayes.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 complaints.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 naive_bayes.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. What is Multiclass Classification?
3. Understanding Conditional Probability using an example
4. Derivation of Baye Theorem
5. What is the Naive Bayes Algorithm?
6. Data Exploration and visualization
7. Performing tokenization using word_tokenization from nltk library
8. How to remove stopwords?
9. Removing the punctuations
10. Word Count and count plots
11. Class mapping

12. Building Naive Bayes model
13. Evaluation Performance
14. Predictions on new reviews