NAIVE BAYES CLASSIFIER FOR RATING PREDICTION FROM REVIEWS

-*DEEPANSH NAGARIA*

PROBLEM STATEMENT AND DATASET:

Building and training a naive Bayes algorithm for text classification. The dataset is a subset of Amazon product data. First entry of each row is the rating followed by the review. Given a review, task is to predict the rating given by the reviewer. A review comes from one of the five categories (class label). Here, class label represents rating given by the user along with the review. The solution needed to be presented in the form of 3 scripts, as I did, namely: train.py, test.py and accuracy_check.py. The role of each is explained in the below sections.

SUMMARY OF SOLUTION:

I have implemented the Naïve Bayes classifier for multi class text classification from scratch without using any available framework as specified. The given classification problem is implemented on the “Amazon product dataset”. We are supplied with two separate files for training and testing purposes. The format was such that each line contained the class (one of the 5) followed by the review. The code is includes all the specifications given along with the problem statement.

I have commented in a very better manner that even a novice will be able to understand the code and used the variable name accordingly so that there is no problem

understating the role of that variable. The process that I expect the tester to follow while evaluation is:

First run the file train.py which will generate some files which will be useful to the test.py file and further test files will generate some files which will be used by accuracy.py to check the accuracy. Also, the paths of the train and test files should be changed in the code at the highlighted spots.

Generation of the files has been done because reading and extracting data again in the test file will again take time.

I have counted the number of occurrences a word is having in a line while classification.

ALGORITHM:

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability. The code can be broken into fragments for better understanding as follows:

- **Handling data:** This involves loading raw data from a CSV file. It also involves the conversion of data received after pre-processing into floating point vectors and achieving the data in the required format.

- **Pre-processing:** As mentioned above we received raw data which needs to be processed before any further use. Some words, formally called stop words are quite common

and populate large portions of datasets and transfer no particular meaning in the context of classifications. Removal of these words improve the accuracy to some extend mostly. In the pre-processing involved care is taken to remove these stop words along with punctuation marks. Stemming is also done so as to treat different forms of same word as one.

- **Summarize Data:** The naive bayes model is comprised of a summary of the data in the training dataset. This summary is then used when making predictions. The summary of the training data collected involves the mean and the standard deviation for each attribute, by class value. We can break the preparation of this summary data down into the following sub-tasks:
    - Separate Data By Class
    - Calculate Mean
    - Calculate Standard Deviation
    - Summarize Dataset
    - Summarize Attributes By Class

- **Making Predictions:** We shall now be ready to make predictions using the summaries prepared from our training data. Making predictions involves calculating the probability that a given data instance belongs to each class, then selecting the class with the largest probability as the prediction. We can divide this part into the following tasks:
    - Calculate Gaussian Probability Density Function
    - Calculate Class Probabilities
    - Make a Prediction
    - Estimate Accuracy

- **Getting accuracy:** The predictions can be compared to the class values in the test dataset and a classification accuracy can be calculated as an accuracy ratio between 0& and 100%.

Why me? :

I am very enthusiastic towards working with your company as an intern as it would definitely be a great learning experience for me. Moreover my interest in the topic shall be my motivation to work and shall bring out the best in me. I on my side am very much looking forward to work with you and share a long bond with your company.