Machine Learning – UE22CS352A

Naive Bayes Classifier – Lab

Student Guidelines

Note: You are advised to read this document completely to understand the overview and the goal of this assignment along with the evaluation policy.

Naive Bayes is a simple and fast classification algorithm used for tasks like spam detection and text classification. It works by calculating the probability of a label (e.g., spam or not spam) given certain features (like words in a message). The algorithm assumes that all features (words) are independent of each other, which simplifies the process, though this assumption is not always true. Despite its simplicity, Naive Bayes is surprisingly effective, especially for text data, and is widely used because it is easy to implement, efficient with large datasets, and performs well even with noisy or small data.

<u>Assignment Structure:</u>

- *naive_bayes_SRN.py*: This is the boilerplate code where you are required to complete three functions in the NaiveBayesClassifier class namely *preprocess()*, *fit()* and *predict().*
  The description of each function along with their return type has been added in the boiler plate to help you understand better.
  Ensure that you replace the SRN with your actual SRN.
- *naive_bayes_test.py*: This file is responsible for conducting tests and comparing the predictions with the expected results.

<u>Input Data:</u>

- The data given to you is comprised of two things:
  1. Sentences: This contains a set of sentences related to a particular topic/domain.
  2. Categories: This list corresponds to the categories (i.e topic/domain) of the sentences.

Note: You are not required to do any additional data encoding.

<u>Instructions:</u>

- In the *naïve_bayes_SRN.py* file you are supposed to complete three functions:
  1. preprocess(sentences, categories): The dataset here may contain incorrect labels denoted by "wrong_label" or missing labels denoted by None, you are required to eliminate such records. Additionally, you are required to balance the dataset as well to ensure there is no bias.
  2. fit(X,y): This function rains the Naive Bayes Classifier using the provided training data
  3. predict(X, class_probs, word_probs, classes): This function predicts the classes for the given test data using the trained classifier.

Note: The .py file contains a more detailed explanation about the functions.

- Run the *naïve_bayes_test.py* to test your classifier implementation against the predefined test cases. Ensure that all test cases pass successfully.

Note:

- *Ensure that you change the import statement in this file as per the naming format that has your actual SRN.*
- *Do not modify anything else in this file.*

This file contains a series of test cases, each consisting of a test sentence and the correct category it should be classified into. Your goal is to ensure your NaiveBayesClassifier implementation correctly predicts these categories for each test sentence. The test cases are auto-evaluated when you run this python file. Do note that, your program will be evaluated against a set of hidden test cases as well.

Command to run:

```
python naïve_bayes_test.py
```

Evaluation Policy:

This lab will be mainly evaluated on two parameters:

1. Accurate implementation of the NaiveBayesClassifer Class
2. Passing all the test cases including the hidden test cases.