CS6375: Machine Learning (Spring ‘23)

Programming Assignment #3

Naïve Bayes

DUE: 21 March 11:59 PM

Problem: Implement a Multinomial Naïve Bayes algorithm for classifying emails into spam and not spam. The algorithm calculates priors and class conditional likelihood of the features to fit a Naïve Bayes model. The code skeleton as well as the datasets for this assignment can be found on e-Learning

Data Set: The data set (in the folder ./data/) is obtained from a Kaggle Dataset on classification of Emails[1]. The dataset consists of 5172 samples and 3001 columns split into two files for training and testing. The feature “Prediction” is the target feature containing 2 classes, namely, “Spam” (1) and “Not Spam” (0). Refer to the link in the footer for more details.

a. (**Reporting Performance**, 30 points) From the code skeleton provided create a working Naïve Bayes algorithm. Define functions for calculating precision, recall, and f1-score from a prediction. Fit a model using the train data provided and calculate your accuracy, precision, recall and f1-score for test. From the features provided in the training data set, report the top three words that have the highest class-conditional likelihoods for both the “Spam” and “Not Spam” classes along with their log-likelihood score.

b. (**Testing On Sample Emails**, 20 points) For the two sample texts provided, predict whether they are spam or not spam using your Naïve Bayes algorithm. Modify the predict_example function to allow predictions on sentences. Report the posterior likelihood for both classes. Try running your naïve bayes model on a sentence of your choice report the prediction.

c. (**scikit-learn**, 20 points) For the email classification dataset, compare your model’s results to those of the standard sklearn Naïve Bayes Algorithms[2] . That includes sklearn’s Gaussian NB, Multinomial NB and Bernoulli NB algorithms. Report the performances of the sklearn models and discuss the differences in the algorithms and their results. Which metric do you believe should be prioritized more for the email classification problem? Explain your reasoning.
Optional: Compare these models’ performance to your earlier Logistic Regression model or Sklearn’s Logistic Regression model

d. (**Bar Plots** 20 points) For each of the models and evaluation metrics reported above, plot bar charts[3] that can visually compare the scores of the various models. (Group bar charts of the same model together)

e. (**Saving the Model**, 10 points) Once you train your model save it as a pickle file in the format ‘NETID_nb.obj’. If you are working as a team, save both the NETIDs separated with an underscore. The object file will be loaded for grading and the class functions will be individually tested.

[1] https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

[2] https://scikit-learn.org/stable/modules/naive_bayes.html

[3] https://www.geeksforgeeks.org/bar-plot-in-matplotlib/