Data Mining
IIIT Delhi
Sumit Bhatia

November 5, 2018
CSE 506, Monsoon 2018
Assignment

Assignment 3
Due Sunday, 11:59PM, November 11, 2018

Policy

- This assignment is to be done individually. You will be required to demo your code to the TAs.
- Any help taken from any outside source *must* be reported in your report.
- Institute's policy on plagiarism will apply. *We will use plagiarism detection tool to check for plagiarism in code.* Violation will result in 0 points for this assignment.
- Last date for doubt clearance from instructor: Friday, November 9, 4 PM.
- Last date for doubt clearance from TAs: Friday, November 9, 4 PM.
- Late submissions penalized by 10% every three hours.
- Programming can be done either in Java or Python.

1. (100 points) **Naïve Bayes Classification**: In this assignment, you will implement the Naïve Bayes classifier discussed in class.
   (a) (30 points) Implement a method called `trainNB(double[][] featureMatrix, String[] labels)` that takes as input the n-dimensional feature matrix and an array of class labels. The data is passed such that i^{th} element in the labels array is the class label for i^{th} row in the feature matrix. Assume that all feature values are *numeric*. That means, you are required to learn the probability distribution functions for each feature. Assume all features follow the *gaussian distribution*.
   (b) (20 points) Implement a method called `classifyNB(double[] testPoint)` that takes as input the feature vector of a test point and outputs the class label by using the model learned in train stage above.
   (c) (20 points) Use the file `train.txt` as input to train your classifier. Each line in the file corresponds to one training data point with its features separated by commas. The last column is the class label. Perform 10 folds cross validation and report classification accuracy.
   (d) (30 points) Use the `test.txt` file to predict the class labels for data points in `test.txt` file. The format of the test file is same as the train file except that the class label is omitted. You should submit the predicted class labels in a file called `labels.txt` along with your code. The submitted file should only contain the predicted class label for each line in test file. For example, 5^{th} line of labels.txt should contain the predicted class label of 5^{th} line in test file. You marks for this part will be based on the performance of your classifier that will be evaluated during demos to the TA. The accuracy will be computed at the evaluation time.