QUESTION 3 (5pts)

1. Define Holdout and Repeated Holdout (Random Sampling)

2. Define k-fold cross-validation (N- N/k ; N/k) holdout and repeated k-fold cross-validation holdout

3. Define Leave-one-out (N-1 ; 1) holdout

4. Bootstrap Holdout

5 .632 boostrap Holdout

PART-1

Question-3

(1) Define Holdout & Repeated Holdout ?

(A) These are two types of methods to evaluate the classifiers. these methods help to bring out the predictive accuracy of the classifier.

Holdout: In this method, the provided data is seperated into two independent disjoint sets. Typically, $\frac{2}{3}^{rd}$ of data is allocated for training & the rest $1/3^{rd}$ data for testing. So, the training data is used to train & build the classifier where as the test data is used to estimate the accuracy known as predictive accuracy. This estimate is pessimistic since the entire data is not used for the testing phase.

Repeated Holdout: This method is similar to the Holdout method but just that it's repeated for K times. The partioning of the data is done randomly ie the probability of any data point to be picked is exactly the same. Since this method is performed randomly, there might be cases where few data points might not be entirely present in the training phase. Finally when the Holdout testing/validation is performed for K-times (generally given), the overall accuracy is obtained as the average of all the accuracies obtained after each iteration.

(2) Define K-fold cross validation holdout & repeated K-fold cross validation holdout.

K-fold cross validation: This is also a method which is commonly used to evaluate a classifier. Over here, the initial data is divided randomly into K (given) disjoint sets (mutually exclusive). So, each datapoint is observed only once across all the subsets generated. In this method, there are K iterations & for each iteration, one of the K subsets is considered without repetitions as the test data. Once the test data is selected, rest of the subsets together form the training data. Since no. of subsets is K. for any iteration training data = N - N/K & test data = N/K where N is the initial number of data tuples. The accuracy estimate calculated is the total number of correct classifications across all the iterations divided by the

initial number of data points. Unlike as in "Holdout" method, each data point or data tuple is used for training & testing phase of the classifier. To be exact, each data tuple is in training phase for K-1 iterations & in testing phase just once. Most suggested number of folds is 10.

Repeated K-fold cross validation holdout:

It's the same procedure as K-fold but repeated multiple times (generally given). As, we seperate the datasets initially for K-fold; the predictive accuracy is totally dependent on this initial subsampling. This may turn out as bias & noisy estimate of the model performance. Different subsets may bring out different estimates. So Repeated K-fold repeats the whole procedure multiple times & estimates the accuracy as the mean estimate of each K-fold accuracy.

If the k-fold algorithm is repeated for n times. The total number of classifiers built are n * k. So this method is generally used for small or medium sized datasets.

(3) Leave One out holdout :

This is a special case of K-fold cross validation where K is taken as the initial number of data tuples. This means for each iteration, the testing data contains only 1 data tuple & the rest N-1 data tuples are considered as training data. This in general provides an unbiased accuracy estimate of the classifier but computationally expensive. So it is generally used for smaller sized datasets.

(4) Boot strap Holdout:

Unlike the methods mentioned above, this method subsamples the training data tuples with replacement. This means if a data point is selected randomly for training data, there's again same probability of getting selected for the data tuple similar to the first one. If we fix on number of data tuples to sample N. Then we need to select N data tuples (might not be unique) randomly from the data with repetition. The data tuples which finally do not belong to the training data turn to be the test data which can be used for predicting the accuracy. This method can be repeated multiple times till we get a meaningful statistics of accuracies. The final predictive accuracy would be the mean estimate of all the iterations.

(5) .632 Bootstrap Holdout :

This is a bootstrap method where we sample exactly the initial number of data tuples as training set. So, if the initial number of data tuples are d. Then we randomly select d data tuples from the data with repitition. Finally the data tuples which are not part of the training data will be the test dataset.

So, in this method the probability of a data point being selected is $1/d$ & the probability for not being selected is $1 - 1/d$. Since we need to do this d times, the probability of a data point not being picked in the whole phase is $(1 - 1/d)^d$. If d is large, this approaches to $e^{-1} = 0.368$. So there are 36.8% of data tuples which never come up in training data & be the test set & the remaining 63.2% will form the training set.

If we repeat this algorithm for k times. The final predictive accuracy would be.

Accuracy of Data D

$$= \sum_{i=1}^{k} (0.632 \times \text{Accuracy of } D_i \text{ bootstrap test set} + 0.368 \times \text{Accuracy of } D_i \text{ bootstrap training set}).$$

So because of the probability 0.632 of being picked in training dataset, this method is called as .632 bootstrap holdout. This works well for smaller datasets