REPORT

Methods used to implement Naive Bayesian Classifier:

ComputeMean:

Takes a list of numbers as input and computes and returns mean of the elements. Mean is sum of all elements/number of elements.

ComputeStandardDeviation:

Takes a list of numbers and mean of the list as input and returns standard deviation of the data. Standard deviation is square root of (sum of squares of all elements from mean /number of elements).

GuassianValue:

Takes mean,standard deviation and a value ‘x’ as input and computes and returns the value of gaussian function at x for given mean and variance.
Gaussian value = $(1/\sigma\sqrt{2\pi})e\hat{}(-((x-\mu)/\sigma)^2/2)$

ExtractColumn:

Takes a 2D list and an integer id with column count of the 2D array and returns the list of all elements in that column.

DiscreteProbabilities:

Takes a list of elements as input and returns a dictionary with distinct integer values in the list as keys and corresponding probabilities(Number of occurrences of given key/size of list) as values.

BayesianClassifier:

Takes a 2D list and an integer,denoting class 0 or , and returns a dictionary containing descriptions of likelihood values of each column of input 2D array. For continuous valued attributes, likelihood values can be found using mean and variance of a given column. For discrete valued attributes, likelihood probability can be found using probabilities of distinct values in that column.So,in the returned dictionary,key indicates column id and value indicates the attribute wise requirement values for likelihood probability computation.

PredictPosterior:

Takes a sample and type of class-0 or 1 and computes and returns posterior probability value of assigning the given sample to given class. It is the product of probability of the class with product of individual probabilities of each attribute in given sample associated with given class.(Stored in likelihood_class_0 or_1 dictionary ,computed using above defined BayesianClassifier function).(neglects probability of data in posterior formula as it is constant)
Posterior = P(C|d) = P(d|C)*P(C) = P(attr1|C)*P(attr2|C)*...P(attrn|C)*P(C)
Here C is a class and d is data with n attributes(attr1 to attrn).Also P(D) in denominator is neglected as the formula is used only for comparison here and P(D) is a constant.

PredictTargetLabels:

Takes a 2D list of samples as input and uses the above described PredictPosterior function to compute posterior probabilities of assigning data to class 0 and class 1 and assigns the class id that has higher posterior probability value as target label for each sample in the data and returns this list of target labels.

Accuracy:

Takes a list of samples containing training attributes as well as target label for each sample and computes the target labels using PredictTargetLabels function described above and returns the percentage of number of samples for which predicted target label is same as the given value in data.

PrintAccuracies:

This function is used to train a naive bayesian classifier for part 1 and 2 of the assignment. It takes the input as attribute size in global variable train_data and a string to be printed as classifier type in output and trains a naive bayesian classifier and prints the test accuracies. The algorithm first divides the training data into training and cross validation in 4:1 ratio five times by considering the 5 different 20%s as cross validation and remaining as training data.Now the algorithm trains only on the training data and cross validation is used to select the trained model with highest cross validation accuracy and then computes and prints the test set accuracy for this trained model.

PCA_Compute:

This function is used to perform part 2 of the assignment. It first normalises the training data and test data and performs pca algorithm to reduce dimensions to retain .95 variance and then calls the above defined PrintAccuracies function to train a naive bayesian classifier and prints corresponding accuracies.

PCA_Graph_Plot:

This function is used to plot the graph of components vs variance ratio using matplotlib library functions after reducing the dimensions by pca algorithm.

SequentialBackwardSelection:

This function takes an integer input denoting number of attributes in train_data and a string to be printed in output to describe this model and performs sequential backward selection method to reduce number of attributes and trains bayesian model on selected subset of attributes and prints corresponding accuracies.

RemoveOutliers:

Takes the whole dataset as input and removes all the samples which are outliers. Outlier samples are the ones with more than half the attributes having values greater than mean+3*standard deviation or less than mean-3*standard deviation values corresponding to those attributes.

Data Structure:

Python Dictionary is used to store mean and variance for continuous valued attributes and a 2nd level dictionary in this containing probability of particular value against the value for each value in discrete valued distribution.
So for example, if attr1 is continuous valued attribute with column id 1,then data_structure[1]['mean'] gives mean of the attribute in training set and data_structure[1]['sd'] gives respective standard deviation.
If attr2 is discrete valued attribute with column id 2 and values 0,1,2 as possible labels for that attribute, then data_structure[2][0] gives the probability of value 0 in the column with id as 2.

Implementation:

Given dataset is loaded and converted to a 2D list.The string valued columns are encoded to integer values. Code is executed in three parts as per assignment description.
Part -1:
Data is divided into 80% training and 20% test data set and a naive bayesian classifier is modeled using train dataset.Trainset is again considered as a combination of 5 splits each with 20% size of input train set one 20% split is considered as cross validation set in each run of a 5 length for loop.Remaining 80% is used to train model and the cross validation set is used to select a model with highest cross validation accuracy and test set output for this model is printed finally. In Bayesian Classifier, likelihoods related values are calculated for the train data and stored in a dictionary in format described above. These are again used along with class probabilities to assign a class to given sample while finding accuracies.

Part-2:
In this part,sklearn library functions are used to fit train data and then transform train and test data to normalise before using pca algorithm and then applied pca algorithm using PCA class in sklearn. Then for this new data with reduced dimensions, bayesian classifier is trained in a similar way as previous part. Corresponding accuracies are computed and printed.Graph is plotted using matplotlib library for pca component and variance ratio obtained by implementing library pca function.

Part-3:
In this part, outlier samples are first eliminated from input dataset. Any sample with more than half of attributes having outlier values is considered as outlier sample. For this new data set, sequential backward selection algorithm is used to reduce number of considered attributes and then bayesian classifier is trained similar to first part. In sequential backward selection algorithm,from a set initially containing all input attributes,each attribute is gradually removed and cross validation accuracy is measured.If cross validation accuracy decreases after removing an attributed,then it is added back into the set of attributes to be considered or lese it is just neglected after removing from the set.After doing the above process for all input attributes,model is trained with the final attributes that remained at the end of the algorithm. These attributes are printed along with test accuracy of the naive bayesian classifier model.

Results:

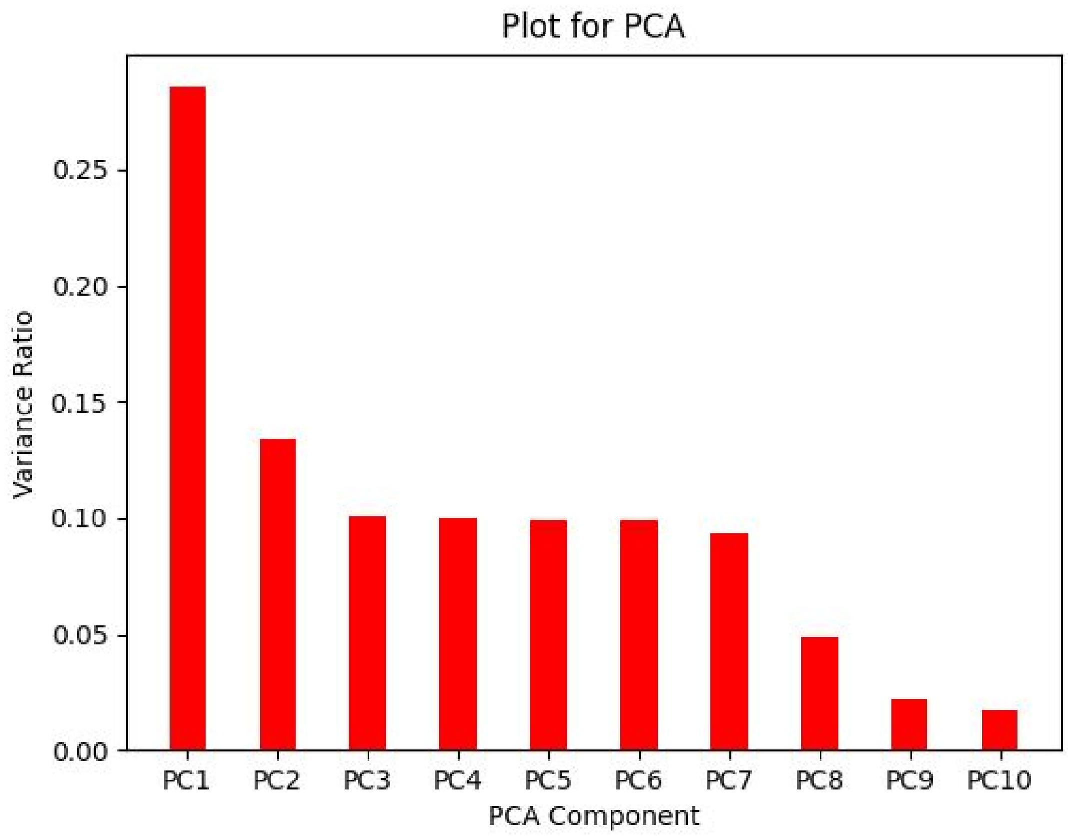
For Part-1:

After training all the 5 train data sets on Naive Bayesian Classifier, for the best cross-validation accuracy of 76.656% we got test accuracy as 76.633% . The following is the accuracy results for the 5 fold cross validation method:

Sl No.	Cross Validation Accuracy (%)
1	76.622
2	76.590
3	76.656
4	76.544
5	76.655

Final Test Accuracy = 76.633%

For Part-2:



Here all the variance ratios of Principal components that can be gotten from the table are given. Here as it is observed the Var_Ratio (PC1) > Var_Ratio (PC2) > Var_Ratio (PC3)....... So the ratios obtained here are in descending order so as to pick the top n components that give a sum of 0.95. If the bar graph is observed carefully we can see that the fall in variance ratio is nearly the same as expected i.e., as more components are chosen its particular variance reduces and effectively becomes constant in the end (observe PC9 and PC10).

After applying PCA Algorithm we got 8 principal components whose sum of variance ratios is just greater 0.95. So here we have retained 95% of the variance after 8 principal components are chosen by the algorithm. For the best cross-validation accuracy of 85.794% we got test accuracy as 85.699% . The following is the accuracy results for the 5 fold cross validation method:

Sl No.	Cross Validation Accuracy (%)
1	85.606
2	85.786
3	85.794
4	85.761
5	85.525

Final Test Accuracy = 85.699%

Part-3:

After removing the data samples with more than half outlier features, we have run the Sequential Backward Selection algorithm and got the following test accuracies for five fold cross validation method:

Sl No.	Test Accuracy (%)
1	87.625
2	87.604
3	87.592
4	87.638
5	87.628

The Best Test Accuracy is = 87.638%

The following are the 4 features that are remaining after running the Sequential Backward Selection algorithm:

-> **Driving_License**
-> **Annual_Premium**
-> **Policy_Sales_Channel**
-> **Vintage**