Objective: The goals of this project are: (i) learn to preprocess available data, (ii) design several different classifiers based on the type of features (both categorical and numerical), and (iii) see how the selection of features affect the performance of classifiers.

Submission guideline: You should submit (i) a report that contains the information request below and (ii) all codes that can be executed by the T.A. for grading purposes.

1 Decision Tree and Random Forest Classifiers

For the first part of the project, use the dataset in `'heart-disease-classification.csv'`. The file contains the information about 918 patients with and without a heart disease (last column in the file: 0 - no heart disease, 1 - heart disease). There are 11 attributes in the first 11 columns, and the labels are contained in the last column `HeartDisease`.

Task 1: Some attributes are categorical and the others are numerical. Before we train a decision tree or random forest, you are asked to convert the numerical variables `Age`, `RestingBP`, `Cholesterol`, and `MaxHR` to categorical variables `Age_cat`, `RestingBP_cat`, `Cholesterol_cat`, and `MaxHR_cat` using different thresholds shown below.

$$\texttt{Age\_cat} = \begin{cases} \texttt{Older}, & \texttt{Age} > 55 \\ \texttt{Mid}, & 40 < \texttt{Age} \leq 55 \\ \texttt{Young}, & \texttt{Age} \leq 40 \end{cases}$$

$$\texttt{RestingBP\_cat} = \begin{cases} \texttt{High}, & \texttt{RestingBP} > 140 \\ \texttt{Normal}, & 120 < \texttt{RestingBP} \leq 140 \\ \texttt{Low}, & \texttt{RestingBP} \leq 120 \end{cases}$$

$$\texttt{Cholesterol\_cat} = \begin{cases} \texttt{High}, & \texttt{Cholesterol} > 200 \\ \texttt{Normal}, & \texttt{Cholesterol} \leq 200 \end{cases}$$

$$\texttt{MaxHR\_cat} = \begin{cases} \texttt{High}, & \texttt{MaxHR} > 165 \\ \texttt{Normal}, & 140 < \texttt{MaxHR} \leq 165 \\ \texttt{Low}, & \texttt{MaxHR} \leq 140 \end{cases}$$

After creating the new categorical variables, replace the four numerical variables with the four categorical variables, and save the new dataset in a file `'HeartDiseaseData.csv'`.

Task 2: Use the dataset in `'HeartDiseaseData.csv'` to train a decision tree and a random forest. For the random forest, create 250 decision trees, and for both the decision tree and the random forest, each decision point should have at least 200 samples for splitting.

As mentioned earlier, the labels for the samples are contained in the column `HeartDisease`. Partition the dataset in `'HeartDiseaseData.csv'` into a training dataset and a testing dataset. Use 25 percent of the samples in the dataset for testing and 75 percent for training. Train a decision tree and a random forest using the training dataset, and evaluate the test accuracy of the decision tree and the random forest using the testing dataset. Repeat this 20 times, each time using randomly selected training and testing datasets, and compute the average of the test accuracy for each for each classifier.

Repeat the above exercise after reducing the minimum number of samples for splitting to 10. Discuss whether or not the test accuracy changed significantly for either of the two classifiers. Finally, compare the performance of the two classifiers.

2 Naive Bayes, k-Nearest Neighbors, Logistic Regression, and Support Vector Machine

For the second part of the project, use the dataset stored in a file `'DiabetesData.csv'`. The file contains the information from 99,805 patients. Each patient is labeled 0 (does not have diabetes) or 1 (has diabetes). There are 16 attributes in the first 16 columns, and the labels are in the last column. All attributes are numerical, and you can use them as given in the file. Split the dataset into a training dataset and a testing dataset with 25 percent of the samples reserved for testing. For evaluating the test accuracy, take the average of 20 models by randomizing the training and testing datasets.

Task 3: Design the following classifiers – (i) k-nearest neighbors (kNN), (ii) Naive Bayes, (iii) logistic regression, and (iv) support vector machine (SVM).

- For the kNN classifier, vary the value of k from 3 to 20 with an increment of one and plot the average test accuracy.
- For the naive Bayes classifier, try both multinomial and Gaussian distributions for approximating the feature distributions. Report the average test accuracy separately.
- For the SVM classifier, you can free to choose some of the parameters, including the loss function used for maximizing the margin or the kernel function if you choose to. However, you should clearly describe the parameters and the model created and trained in your code.

Task 4: For Task 3, you were asked to use all attributes. For this task, you will use only m ($m < 16$) attributes instead. For each model you train, randomly select m out of the 16 attributes and use only the chosen attributes to perform the classification using the classifiers in Task 3. Consider $m = 3$ and 7. Generate 100 models for each classifier by selecting a new subset of m attributes each time. For the kNN algorithm, fix $k = 11$. For this part, you do not need to repeat it 20 times with randomly chosen training and testing datasets, i.e., use one partition of the dataset into training and testing datesets.

Identify the maximum test accuracy achieved among the 100 models generated for each classifier and the subset of the attributes that led to the highest test accuracy. Plot the histogram of the test accurcy of the

100 test models (one histogram for each $m \in \{3, 7\}$) with 10 bins. Discuss your observations regarding the test accuracies.