ASSIGNMENT 3

In this assignment, you will implement various classification techniques on the Pima dataset. More description about this dataset can be obtained here:

https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

The data is here:
https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data

The description of the data is here:
https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.names

There are various parts to this assignment. You need to submit a separate folder for each part. You are encouraged to do this assignment in R programming language.

I. Exploratory Data Analysis:

In the first phase of the project, you will analyze the independent variables or attributes and examine their properties such as range, variation, and distribution.

Note that the independent variables are also referred to as attributes and refer to the variables other than the class variable.

Perform the following analysis. Submit code and plots as applicable.

1. Create the following plots: histogram, and barplot.

Write a short note on the distribution of the variables that you observe from the plots. Are they normally distributed?

2. Find the correlation between each of the attributes and the class variable.
Note: You will need to convert the factor variable to numeric to perform this calculation. You can use R's cor function to compute correlations.

Which attributes seem to have a strong correlation with the output (class) variable?

3. Compute the correlation between all pairs of the 8 attributes. Which two attributes have the highest mutual correlation?
You should do this question by writing a loop that automatically computes the correlation between all the attributes and outputs the pair having the highest value.

II. Naïve Bayesian Classifier

In this section, you will build a naïve Bayesian classifier for classifying the Pima dataset.

1. Find a good R package that does naïve Bayesian classification or write your own Java/Python code. Clearly mention the approach that you take.

2. Using a random number generator, split the Pima dataset into a ratio of 90:10 for training and testing. Use the training data to build a naïve Bayesian model and then use the model to find the prediction on the test data.

Repeat this experiment 10 times using different samples each time.

You can create training/test data using R's **sample** function.

3. For each experiment, compute the accuracy. Also, report the average accuracy of the 10 experiments.

As always, accuracy is defined as = Number of correct predictions/ Total instances in the test data.

Report the output in a table:

Experiment	Accuracy
1	90%
2	95%
..	..

Overall accuracy = 80%

III. SVM Classifier

1. Repeat the steps in part II, but this time with a SVM classifier using default kernel. Report the accuracy on 10 different training samples and also the average accuracy (similar to part II)

2. Change the value of the kernel from the default to the following values:

- linear
- polynomial

- radial basis
- sigmoid

Carry out 10 experiments using different samples and each of the kernels above.
Report only the average accuracy in each case. That is, report the output as below:

Kernel	Average accuracy of 10 experiments
Linear	90%
Polynomial	95%
Radial	60%
Sigmoid	…

IV. kNN

Next carry out the experiments using a k-Nearest Neighbor classifier.
Choose 5 different values of k (3, 5, 7, 9, and 11) and carry out 10 experiments each using different samples of training and test data.

Report results as below:

k	Average accuracy of 10 experiments
3	90%
5	95%
7	60%
9	…
11	…

Write a short paragraph (in the README file) on which method works the best for this dataset.

Turn in a zip file containing a folder for each part that contains code, answers to questions (if any), and plots (if any).