ENGG2112 Coding Assignment

Due on 23 September 2022, 11.59pm

9 SEPTEMBER 2022

Instructions

- This is an individual assignment and the submitted work must be your original work. You are allowed to discuss the method of solution with others, however the submitted code must be entirely written by you.
- Submit your work as a Python notebook in the template provided.
- Submissions must be made through Canvas only, and not by e-mail. The deadline will be strictly enforced: 11:59pm on 23 September 2022. (Students with disability adjustments will be contacted separately.)
- Please plan your time according to your own ability and schedule, seek help from the teaching team and peers in a timely fashion, and try not to ask for deadline extensions.

Problem 1

Download the file "Alternative Fuel Vehicles US.csv" from the Canvas website under Assignments » Coding Assignment. (As this data is from the US, fuel economy is in miles per gallon (mpg) and distances are in miles.)

1. Plot the following histograms in Python:

    a) (10% of total) "Conventional Fuel Economy Combined" of hybrid electric vehicles[1] of European makes and Japanese makes in two plots (Google which makes are European and Japanese if necessary). Is it clear from the histograms whether European or Japanese makes are more fuel efficient?

    b) (10% of total) "All Electric Range" of all electric[2] vehicles. Write Python code to save the top ten percent of these vehicles in terms of "All Electric Range" in a file named TopTenElectric.csv. This file will be a part of your assignment submission.

[1] Fuel = "Hybrid Electric" only, not "Plug-in Hybrid", etc.

[2] Fuel = "Electric" only.

2. Now consider only those vehicles with non-empty "Conventional Fuel Economy Combined" entries (there are 444 of them). Using columns A, D, K, L, M and Q as features and column E (Fuel) as the target, write a Python program using logistic regression as the classification method. For column D (Manufacturer), simplify by grouping according to European, American and Asian manufacturers.

    a) (10% of total) Explain any pre-processing that you had to perform.

    b) (10% of total) Find the accuracy of your method by averaging over 20 random train/test splits at each of 5 train-test ratios of your choice. Display the best accuracy.

    c) (10% of total) Repeat the above with a multi-layer perceptron having one hidden layer with any number of neurons you wish. (Again, experiment with various numbers and display the best result.)

Problem 2

For this problem, use the file heart.csv. There are 918 records with 11 features and 1 target (the HeartDisease column). The target is 0 if the patient has no heart disease and 1 otherwise.

1. (10% of total) Explain how you would transform the categorical variables into numerical variables. Should we use one-hot encoding on all of them? Examine the data carefully and comment on whether further pre-processing is necessary. If it is, execute the pre-processing method(s) you suggested.

2. (15% of total) Use all the available features to train a Gaussian Naive Bayes classifier with a random 75/25 train/test split (approximately). Find the area under the curve of the receiver operating characteristic (ROC) and the accuracy of the classifier for 10 independent instances of the train/test split, and present a table of your results.

3. (10% of total) Suppose a patient came in with incomplete data, e.g. cholesterol and resting ECG are missing. Suggest a way to still use the classifier developed above to predict the probability that the patient has heart disease. Write Python code to implement your suggestion.

4. (15% of total) Next, use K Nearest Neighbours (KNN) to tackle the same classification problem. Experiment with different values of K, and also compare the weighted and unweighted versions of KNN in terms of accuracy and AUC. Present a table with columns comprising K, Accuracy (KNN), Accuracy (WNN), AUC (KNN), and AUC (WNN).