CS504 Spring 2022 Project 3
Due: 5/6/22 by 11:59PM

You can do this project individually or in a group that can have up to 3 students. You need to sign up accordingly for project 4 individuals or groups on Blackboard. If you are in a group, the late days for your group will be the average late days rounding down.

Description:

In this project, you will work on the wine.csv file uploaded on Blackboard and write Python code to classify three different classes of wine.
More information about the dataset is contained in wine-names.txt file uploaded on Blackboard.

Step 1: Data preprocessing

Explore the dataset to identify the features and the class attribute. In general, scikit-learn doesn't deal with categorical data well. More specifically, KNN works best with scaled data while MultinomialNB doesn't accept negative values. Consider if there are any missing values, outliers, and attributes that have no predict power. You also need to convert pandas DataFrames into numpy arrays that can be used by scikit-learn. Show your data after being preprocessed. If none of the techniques described below is able to achieve close to or above 90% accuracy, examine your data again to see if you can preprocess the data in a different way.

Step 2: Applying techniques

Apply the following techniques to your preprocessed data set, and see which one yields the highest accuracy as measured with 10-fold cross validation.

Decision tree

- Create a single train/test split of your data. Set aside 75% for training, and 25% for testing. Use **tree.DecisionTreeClassifier** to create a model and fit it to your training data. Measure the accuracy of the resulting decision tree model using your test data. (Hint: you don't have to visualize the tree and you can use **score** method to get the accuracy.)
- Instead of a single train/test split, use 10-fold cross validation to get a measure of your model's accuracy. (Hint: use **model_selection.cross_val_score** and use **mean** method to find the average)

Random forest

- Use **ensemble.RandomForestClassifier** with n_estimators=10 and use 10-fold cross validation to get a measure of the accuracy. Does it perform better than decision tree?

KNN

- Use **neighbors.KNeighborsClassifier** with n_neighbors=10 and use 10-fold cross validation to get a measure of the accuracy.
- Try different values of K. Write a for loop to run KNN with K values ranging from 1 to 50 and see if the value of K makes a substantial difference. Make a note of the best performance you could get out of KNN by 10-fold cross validation.

Naive Bayes

- Use **naive_bayes.MultinomailNB** and use 10-fold cross validation to get a measure of the accuracy.
- Use **naive_bayes.GaussianNB** and use 10-fold cross validation to get a measure of the accuracy. Does it perform better than MultinomialNB?

Extra credit: (10 points)

SVM

- Use **svm.SVC** and use 10-fold cross validation to get a measure of the accuracy.
- Try different kernels. Write a for loop to run svm with linear, sigmoid, and poly kernels respectively and make a note of what kernel performs best using 10-fold cross validation.

Deliverable:

Include the following files:

1. Python source code in .py
2. The output of your source code including both steps
3. Your answer to each question in step 2 and a conclusion regarding your results.

Note that you can also include everything in a single .ipynb file.