An important note: you must work individually on the assignments. Violation of this is against the USU's Academic Integrity (outlined in the Syllabus) and can have serious academic consequences.

This homework aims to help you learn how to apply and evaluate predictive models using Python.

Due Date: April 03, 2022 (11:59 PM)

Total points: 100

Files needed:

- Download the data sets *train_office.csv* and *test_office.csv* from the Canvas. The data contains information about the date, temperature, humidity, and other environmental factors for the office. The goal is to predict whether the office room is occupied (1) or not (0). Use the last column, Occupancy, as your target (class) variable.
- *msft.csv*: the data contains the stock prices for Microsoft from January 2007 to December 2016.

1. [35 points] Different predictive models

   (a) Train the following classifiers (using 5-fold cross-validation) on the *train_office.csv* file and calculate the average accuracy of the cross-validation for each method given below. Vary the hyperparameters of the classifier and draw a plot that shows the average cross-validation accuracy versus hyperparameter values for each classification method shown below:

      i. Decision tree (DT) (maxdepth = 1, 5, 10, 50, 100)

      ii. K-nearest neighbor (KNN) (k = 1, 2, 3, 4, 5, 10,15)

      iii. Logistic regression (LR) (C = 0.001, 0.01, 0.1, 0.5, 1)

   (b) Use the plot to choose the hyperparameter with the highest average accuracy.

   Note: the hyperparameter chosen may not be the same with different runs due to the randomization used by the cross-validation procedure.

   (c) For each method above, train a model on the entire *train_office.csv* set with the hyperparameter you had chosen from the previous step.

   (d) Predict labels of test samples in the *test_office.csv* using the best classifiers you built in the previous step. Then, perform the followings:

      i. Print out the confusion matrices (for DT, KNN, and LR).

      ii. Draw a bar plot showing precision, recall, F1-score of the three classifiers. For the F1-score, you can use 'weighted'.

      iii. Draw ROC curves of the three classifiers in one graph. In the legend of the graph, include AUCs (area under the curve), e.g., DT (AUC=X), KNN (AUC=X), LR (AUC=X).

(e) Visualize (plot) the resulting decision tree You need to use *tree* class from sklearn–See this. Based on this tree, what do you think is the most important feature?

2. [20 points] Sklearn grid search

Sickit-learn provides a function to perform grid search i.e., evaluating all possible values of a given set of hyperparameters.

For this question, use the office dataset.

(a) Perform a grid search for Random Forest Classifier using a 5-fold cross-validation (cv=5). The hyperparameters for your grid search are {n_estimators, criterion, max_depth, min_samples_split, min_samples_leaf}. Please read the signature of Random Forest to know more about these arguments and their appropriate values.

(b) Predict the class of test samples using the best classifier. Note that, the *best classifier* is built similar to 1(c) i.e., training on the entire training data with the best hyperparameters. Then, predict labels of test samples and print accuracy, precision, recall, F1-score, and AUC. Did you obtain a better performing classifier than what you found in question 1?

3. [45 points] Regression

For this question, use the Microsoft stock dataset. Use 80% of the data for training and 20% for the test. Consider *Adj Close* as the label.

(a) Use grid search (5-fold cross-validation) and find the best decision tree regressor. The hyperparameters for your grid search include {criterion, splitter, max_depth, min_samples_split, min_samples_leaf}. Again, consider a set of reasonable values for hyperparamters.

(b) Find the hyperparameter values yielding the best performance in the previous step. Then, similar to the previous questions, train a decision tree regressor with these hyperparameters on the entire training set. Afterwards, predict labels of samples in the test set using this best regressor. What is the performance against the test set based on RMSE?

(c) Draw the learning curve plot, i.e., the size of the training set vs. the performance. For each experiment, train a classifier on k% of the training set and get the performance on the training and test sets and draw them in one plot. Consider k=$\{10\%, 20\%, 30\% \cdots 100\%\}$. What are your observations?

(d) Traditional models (e.g., decision trees) are usually sensitive to the input features' scale. Hence, it may be better to change raw feature vectors into a more suitable representation for the downstream task. In this part, try to apply different transformations to the input features aiming to have a better predictive model. To this end, use preprocessing package (study this page and sample codes carefully).

i. Transform the input features using *StandardScaler* function and repeat 3(a) and 3(b).

i. Transform the input features using *MinMaxScaler* function and repeat 3(a) and 3(b).

iii. Transform the input features using *normalize* function (section 6.3.3) and repeat 3(a) and 3(b).

Did any of the processing techniques help?

Submit the following:

- **homework6.ipnyb**: Your python notebook containing the code for questions that need programming.
- **homework6.pdf**: Your answers to questions that need explanation. You can write your explanations in homework6.ipnyb (e.g., use heading or markdown). But please write them clearly.