

CSE 482 Exercise 6 (Date: December 5, 2022)

The purpose of this exercise is to help you learn how to use some of the predictive modeling methods available in Python as well as association rule mining (not available in python). Follow the instructions below to complete the exercise. Save your IPython notebook as exercise6.ipynb.

1. Download the data sets *train.csv* and *test.csv* from the class Web page. 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). Implement the following steps in an ipython notebook:
 - a. Use pandas to load the train.csv file into a DataFrame object named "data". After the file is successfully loaded, type data.head() to display the first 5 rows of the table. Remove the "date" column from the table using the drop() function since the column is not predictive of the class (unless you extract the time of day and day of week information). Use the last column, Occupancy, as your target (class) variable.
 - b. Train the following classifiers (using 5-fold cross validation) on the train.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 (maxdepth = 1, 5, 10, 50, 100)
 - ii. K-nearest neighbor (k = 1, 2, 3, 4, 5, 10,15)
 - iii. Logistic regression (C = 0.001, 0.01, 0.1, 0.5, 1)

Your code would look something like this:

```
hyperparams = [ .... ]
accuracy = []
for param in hyperparams:
    clf = CLASSIFIER_DEFINITION ( set param here )
    scores = cross_val_score(clf, X, Y, cv=5)
    acc.append(scores.mean())
```

Use the plot to choose the hyperparameter that gives 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 using the entire train.csv dataset (with the hyperparameter you had chosen from the previous step). Apply the model to the test.csv data. Draw a bar chart to compare their test accuracies using the 3 methods.
2. Download the apriori software from the following website <http://www.borgelt.net/apriori.html>. Download the data set *Titanic.csv* from the class Web page. Generate the association rules from the data with the following thresholds: minimum support =1% and minimum confidence = 90%.

Save the rules generated in a file called `titanic.rules`. Execute the program from ipython notebook.

Deliverables: Submit (via D2L) the files `exercise6.ipynb` and `titanic.rules`.