Exercise 6 : Classification Tree

Workflow

1. Create a folder on your Desktop and name it CZ1016_[LabGroup], where [LabGroup] is the name of your Group
2. Download the .ipynb files and data files posted corresponding to this exercise and store in the aforesaid folder
3. Open Jupyter Notebook (already installed on the Lab computer) and navigate to the aforesaid folder on Desktop
4. Open and explore the .ipynb files (notebooks) that you downloaded, and go through "Preparation", as follows
5. The walk-through videos posted on NTU Learn (under Course Content) may help you with this "Preparation" too
6. Create a new Jupyter Notebook, name it Exercise6_solution.ipynb, and save it in the same folder on the Desktop
7. Solve the "Problems" posted below by writing code, and corresponding comments, in Exercise6_solution.ipynb

Try to solve the problems on your own. Take help and hints from the "Preparation" codes and the walk-through videos. **If you are still stuck, talk to your friends in the Lab to get help/hints.** If that fails too, approach the Lab Instructor.

Note : Don't forget to import the Essential Python Libraries required for solving the Exercise. Write code in the usual "Code" cells, and notes/comments in "Markdown" cells of the Notebook. Check the preparation notebooks for guidance.

Preparation

M4 ClassificationTree.ipynb Check how to perform basic Classification on the Pokemon data (pokemonData.csv)

Objective

Note that our Housing Data has a Binary (two-level) Categorical Variable named "CentralAir", with values "Y" and "N". In a former Example Class, we have identified and analyzed some of the most relevant numeric variables in this dataset. In this Example Class, we will try to predict if a House has Central Air Conditioning or not using those Numeric Variables.

Problems

Download the dataset **train.csv** and the associated text file **data_description.txt** posted with this Exercise.

Problem 1 : Predicting CentralAir using SalePrice

Import the complete dataset "train.csv" in Jupyter, as `houseData = pd.read_csv('train.csv')`

Note : In this exercise, we will not extract the variables from the dataset, as we did the last time.

a) Plot the binary distribution of `houseData['CentralAir']` using `catplot` to check the ratio of Y against N. Note that the classes Y and N are quite unbalanced; do you think this will create any problem in our Classification?

b) Plot `houseData['CentralAir']` vs `houseData['SalePrice']` using `boxplot`, and note the strong relationship. Also check the mutual relationship by plotting the two variables using a `swarmplot`, and note the difference.

c) Import Classification Tree model from Scikit-Learn : `from sklearn.tree import DecisionTreeClassifier`

d) Partition the complete dataset `houseData` into `houseData_train` (1100 rows) and `houseData_test` (360 rows).

e) Training : Fit a Decision Tree model for classification of CentralAir using SalePrice using the following variables.

```
y_train = pd.DataFrame(houseData_train['CentralAir'])
X_train = pd.DataFrame(houseData_train['SalePrice'])
```

f) Visualize the Decision Tree model using graphviz (needs the packages to be installed; check if they are installed).

g) Predict CentralAir for the train dataset using the Decision Tree model, and plot the Two-Way Confusion Matrix. Predict CentralAir for the test dataset using the Decision Tree model, and plot the Two-Way Confusion Matrix.

h) Print all the accuracy parameters of the decision tree model, including its Classification Accuracy, True Positive Rate, True Negative Rate, False Positive Rate and False Negative Rate, based on the aforesaid confusion matrix.

Problem 2 : Predicting CentralAir using Other Variables

Perform all the above steps on `'CentralAir'` against each of the variables `'GrLivArea', 'LotArea', 'TotalBsmtSF'` one-by-one to obtain individual Decision Trees. Discuss with your Friends about the models, compare the Classification Accuracy, check the True Positives and False Positives, and determine which model is the best to predict `'CentralAir'`.

Extra Resources

You may read more about the `DecisionTreeClassifier` model you use in this exercise in the following references.

`DecisionTree` : https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Other Tree Models (Scikit Learn) : https://scikit-learn.org/stable/modules/classes.html#module-sklearn.tree

Bonus Problems

1. Note that `DecisionTreeClassifier` model can take more than one Predictor to model the Response variable. Try using this feature to fit a Decision Tree model to predict `'CentralAir'` using all the four variables `'SalePrice', 'GrLivArea', 'LotArea', 'TotalBsmtSF'`. Find the accuracy of this multi-variate model.

2. Fit a Decision Tree model to predict `'CentralAir'` using all the numeric variables in the given dataset. You may use all the numeric variables from Exercise 2. Find the accuracy parameters of this multi-variate model.

Are the False Positive Rates of the various Decision Tree models significantly higher/lower than the False Negative Rates? Does this have anything to do with the unbalanced classes Y and N of `'CentralAir'`? Experiment, and think about it.