Exercise 5 : Classification Tree

Workflow

1. Download the .ipynb files and data files posted with this exercise and store them all in a folder on your Desktop.
2. Open Jupyter Notebook (already installed on the Lab computers) and navigate to the aforesaid folder on Desktop.
3. Open and explore the .ipynb files (notebooks) that you downloaded, and go through "Preparation", as follows.
4. The walk-through videos posted on NTU Learn (under Course Content) may help you with this "Preparation" too.
5. Create a new Jupyter Notebook, name it Exercise5_solution.ipynb, and save it in the same folder on the Desktop.
6. Solve the "Problems" posted below by writing code, and corresponding comments, in Exercise5_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 your 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 the previous sessions, we have seen some numeric variables in this dataset that are important to predict "SalePrice". In this lab session, we will try to predict if a house has Central Air Conditioning or not using some other variables.

Typical steps to follow while building a supervised machine learning model on a given dataset:

- **Partition** the labeled dataset into two random portions – 75% to Train the model and 25% to Test the model.
- **Fit** the desired supervised machine learning model on the Train set to predict response using the predictors.
- **Predict** response using the predictors on the Test set using the machine learning model fit on the Train data.
- **Check** the Prediction Accuracy of the model on both the Train and the Test sets using the "Confusion Matrix".

Disclaimer: There may be several ways to solve these problems and there is no single correct answer. Try to explore on your own, talk to your friends and the Lab Instructor, and make sure you are happy with your own justifications. You will get marks for your solutions as long as your justifications make sense, and you can explain those clearly.

Marks distribution

4 points for Problem 1 2 points for train-test set and classification in (a) + 2 points for metrics in (b)

3 points for Problem 2 2 points for the two classifications + 1 point for comparison and justifications

3 points for Problem 3 2 points for printing samples in the leaf in (a) + 1 point for isolating FPs in (b)

Problems

Problem 1 : Predicting CentralAir using SalePrice

In this problem, you will build a Classification Tree model to predict `CentralAir` using `SalePrice` and judge its accuracy.

a) Create appropriate datasets for Train and Test in an 75:25 ratio and fit two Classification Tree models (of max depth 3 and max depth 4) on the Train set to predict `CentralAir` using `SalePrice`. Print the tree in each case.

b) Print the confusion matrix on Train set and Test set for both the aforesaid models. Compute and print for both the trees the Classification Accuracy, True Positive Rate, False Positive Rate. Which of the two models is better?

Hints and Pointers

- If you take just the first 75% of the data as train and the next 25% as test, it may not be the best train test split.
- If you want to change the max depth of the tree, check the input parameters needed to instantiate the model.
- Think about the accuracy metrics for a tree carefully to determine which model is better and in which metrics.

Problem 2 : Predicting CentralAir using Other Variables

Following the steps from the previous problem, build two new uni-variate Classification Tree models (of max depth 3) to predict `CentralAir` using the variables `OverallQual` and `YearBuilt`. Justify which of the variables is the best predictor.

Hints and Pointers

- Same as Problem 1, just on other variables. You can compare models using the metrics you are printing anyway.
- Optional: You may think of writing a simple Python function to do classification on variable(s) in a given dataset.

Problem 3 : Understanding the Misclassified Samples

In this problem, you will consider finer details of the tree model and try to isolate the misclassified samples for a model.

(a) Print the Classification Tree of max depth 3 that you fit in Problem 1(a). Note that each leaf node of your tree represents a specific partition of your training data, based on certain conditions given by the splits in the tree. Find out which leaf node of this tree has the highest number of False Positives (FP) in the training dataset and print all samples in the Train set (rows from the dataframe) that end up in this leaf node (partition) during fit.

(b) Print ONLY the False Positive cases (rows from the dataframe) from the samples (rows) extracted in part (a).

Hints and Pointers

- In Part (a), you may print the model in Problem 1(a), or fit the max depth 3 classification tree model once again.
- You may simply "observe" which leaf node has the maximum number of False Positives; no need to code for it.
- Think about how data points (samples) end up in a leaf node based on tree splits; it will help you isolate samples.
- Part (b) is a simple continuation of part (a), where you need to "filter" out the False Positives from the samples.

Hints are not meant to tell you exactly what to do for the problems; use these as pointers to search online. Take a close look at **Pandas DataFrame documentation** and check **Classification Tree LAMS** carefully to solve most of these problems.

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