Tips for Lab Sessions

- Get preparation: get familiar / try to tackle the problems **before** coming to lab.
- Do **ALL** the problems!
- Format you answer: use **Markdown** to organize your answer / conclusion.
- Watch the time: the DDL is at **XX:20 PM**, not XX:30 PM.
- Learn to Google for **usage of basic functions**.
- Do NOT mail your answer: submit your work **ONLY** via **NTU-Learn**.

Lab 5. Classification Tree

labels, records, and assigns variables to discrete classes

Classification Tree

- Goal: how to **predict categorical** variable Y given a good variable X (i.e. $\mathrm{corr}(X, Y)$ is high)?
- Rationale: **partition** data points into different groups (**leaves**) according to some rules (**conditions on a univariate**).
- How: **fit** a **classification tree** model; **evaluate** the model performance on the testing data using **confusion matrix**.

Classification Tree Workflow (Lab Mark Checkpoints)

1. Split your Dataset: **randomly split** the dataset into training v.s. test dataset.

2. Fit & Print Classification Tree: **fit** & **plot** classification trees with different variables X on the training set.

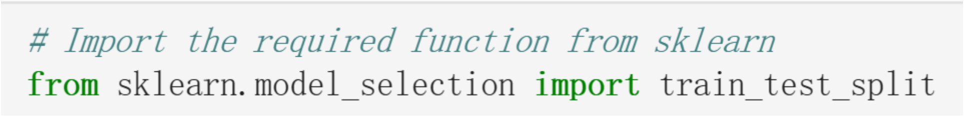
3. Calc. Confusion Matrix: calculate the **confusion matrix** on **both** train & test datasets.

4. Filter Out Misclassified Samples: identify the **leaf** with max. **false positive** samples.

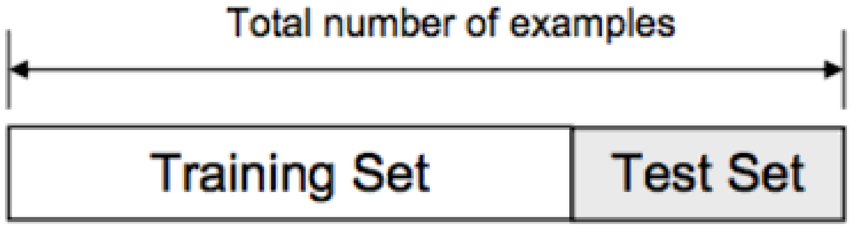
Mark Checkpoint 1: Split the dataset

- **Randomly** split the train / test dataset.

- **Visualize** the correlation between X, Y via boxplot.



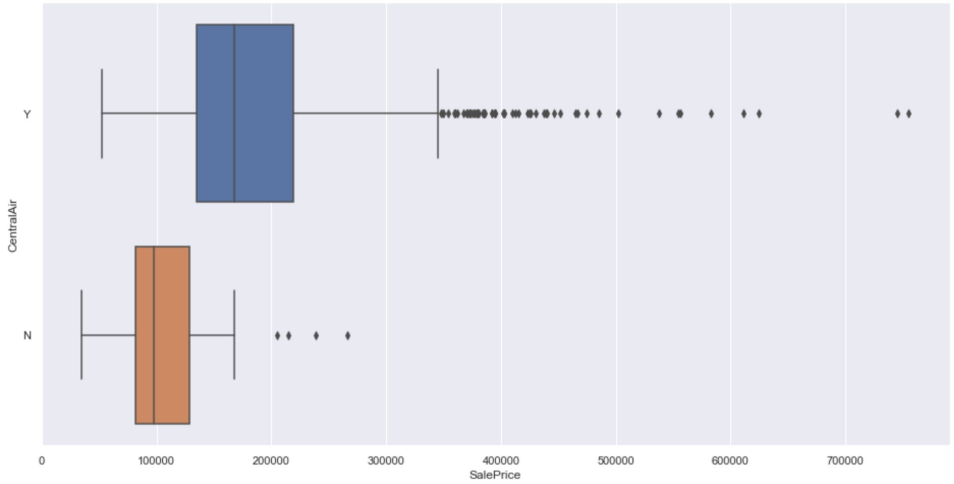
```
# Import the required function from sklearn
from sklearn.model_selection import train_test_split
```



Total number of examples

Training Set

Test Set

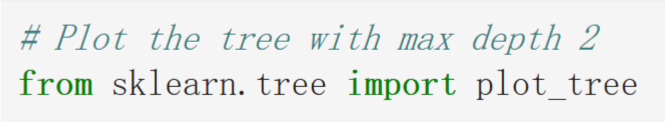


Mark Checkpoint 2: Fit & Plot Classification Tree

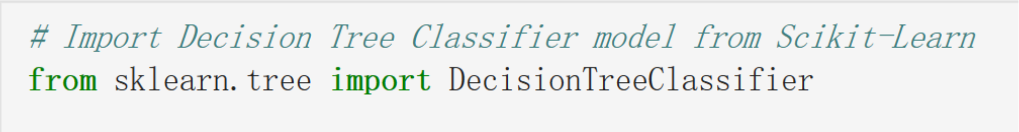
- **Fit** classification trees with **max. depth 2 & 4**.

- Google for basic attributes of DecisionTreeClassifier().

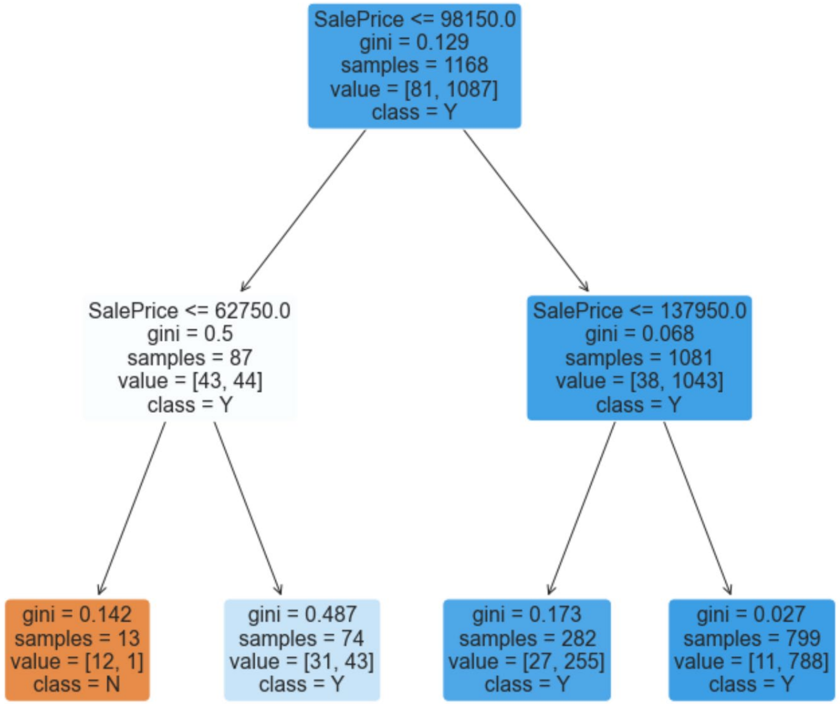
- **Plot** those regression trees.



```
# Plot the tree with max depth 2
from sklearn.tree import plot_tree
```



```
# Import Decision Tree Classifier model from Scikit-Learn
from sklearn.tree import DecisionTreeClassifier
```



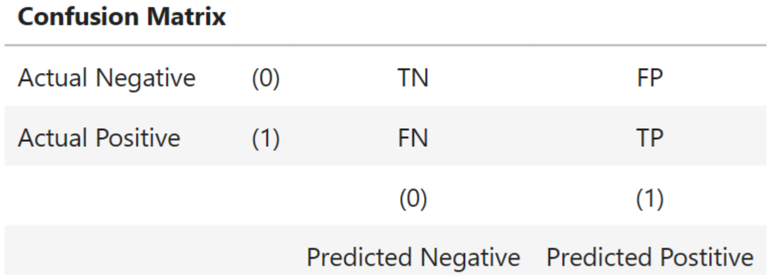
Mark Checkpoint 3: Model Selection

- ✓ **Compute& show** confusion matrix on **both** train & test datasets.

- ✓ **Print** (Markdown) for both the trees the Classification **Accuracy**, True Positive Rate, False Positive Rate (**TPR & FPR**).

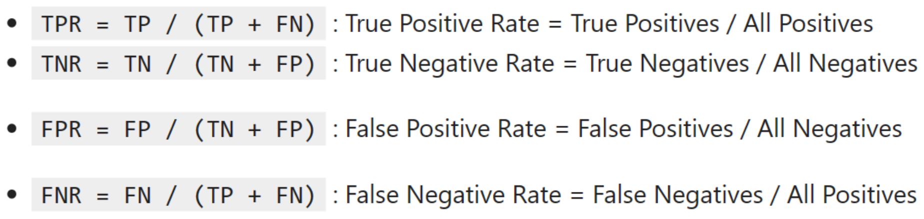
Not sure

- **Explain** in a few sentences: which tree (variates / depth) is better w.r.t Acc., TPR and FPR.



Confusion Matrix

Actual Negative	(0)	TN	FP
Actual Positive	(1)	FN	TP
		(0)	(1)
		Predicted Negative	Predicted Postitive



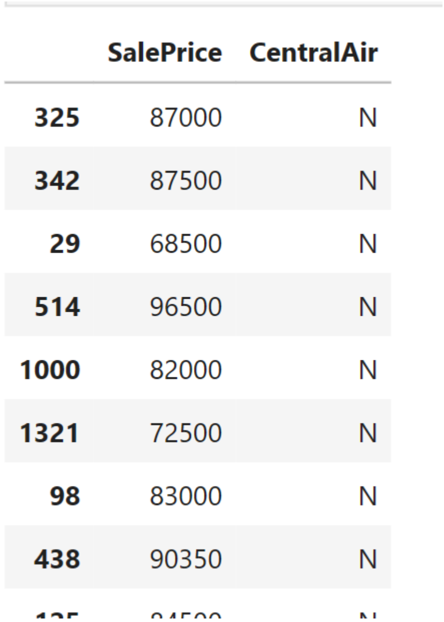
- `TPR = TP / (TP + FN)` : True Positive Rate = True Positives / All Positives
- `TNR = TN / (TN + FP)` : True Negative Rate = True Negatives / All Negatives
- `FPR = FP / (TN + FP)` : False Positive Rate = False Positives / All Negatives
- `FNR = FN / (TP + FN)` : False Negative Rate = False Negatives / All Positives

Mark Checkpoint 4: Find Misclassified Samples

- **Identify** the leaf (of the previous depth-4 tree) with maximal **False Positives**.

- Print the specific **condition** leading to this leaf.
- $x < \text{ SalePrice} < y$

- Print the **samples** assigned to this leaf.



	SalePrice	CentralAir
325	87000	N
342	87500	N
29	68500	N
514	96500	N
1000	82000	N
1321	72500	N
98	83000	N
438	90350	N
125	84500	N