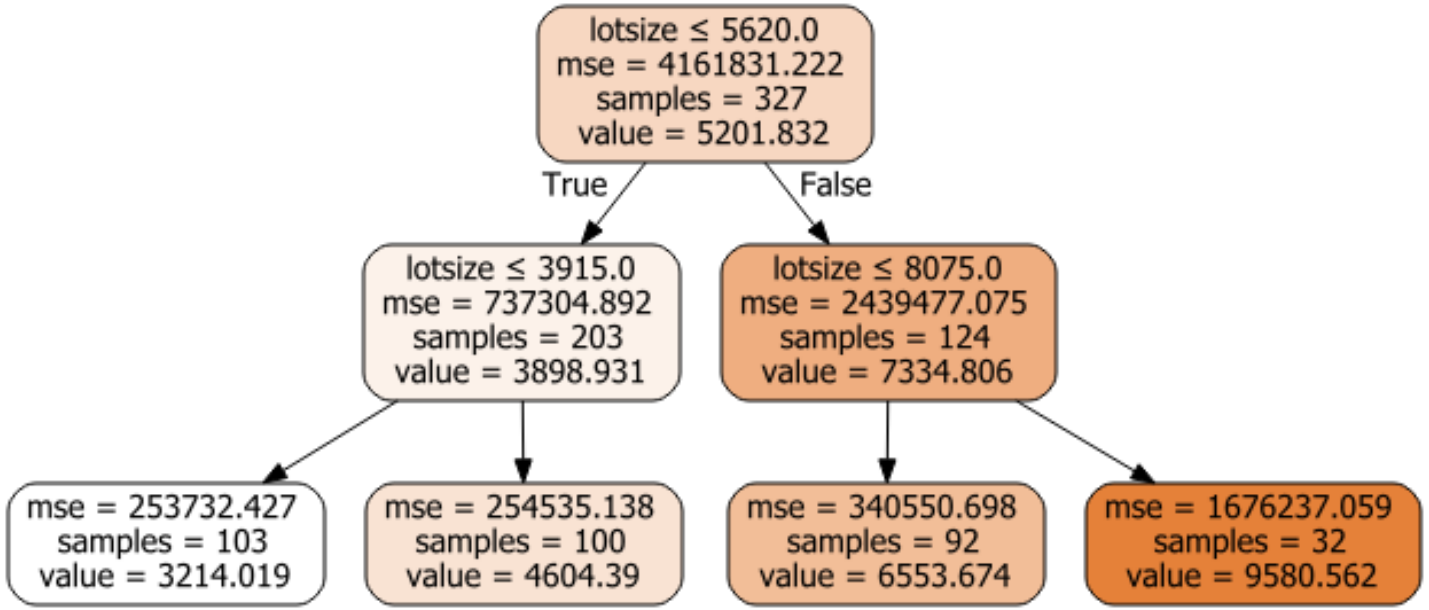
Sane's STATS Academy of Statistics

Regression Tree

Typical Regression Tree Output

- In case of regression trees, the difference is that on the leaf nodes we have the means of the response variable values.



Regression Tree

- The data gets divided into two parts in the interest of decreasing the variation of response variable
- The child nodes have lesser variation than their respective parent nodes for response variable

Comparison in types of trees

	Classification	Regression
Response Variable Type	Categorical	Numerical
Measuring Homogeneity	Gini, Entropy	MSE
Prediction	Majority Class in the leaf node	Mean of response variable in the data in leaf node
Evaluation	Confusion Matrix metrics, ROC(only for 2 categories)	MSE, MAE, R^2

Regression Tree in Python

- From scikit-learn, we import package tree
- We instantiate the class of tree.DecisionTreeRegressor
- Call the method fit() on it
- On the built model, we call predict()

```
clf = tree.DecisionTreeRegressor(max_depth=2)
clf2 = clf.fit(X_train, y_train)
```

```
y_pred = clf2.predict(X_test)
```

Example : Sales Prices of Houses in the City of Windsor

- Description
  - a cross-section from 1987
  - number of observations : 546
  - country : Canada
- A dataframe containing :
  - price : sale price of a house
  - lotsize : the lot size of a property in square feet
  - bedrooms : number of bedrooms
  - bathrms : number of full bathrooms
  - stories : number of stories excluding basement
  - driveway : does the house has a driveway ?
  - recroom : does the house has a recreational room ?
  - fullbase : does the house has a full finished basement ?
  - gashw : does the house uses gas for hot water heating ?
  - airco : does the house has central air conditioning ?
  - garagepl : number of garage places
  - prefarea : is the house located in the preferred neighbourhood of the city ?

Program and Output

```
Housing = pd.read_csv("F:/Python Material/ML with Python/Cases/Real Estate/Housing.csv")
dum_Housing = pd.get_dummies(Housing.iloc[:,1:11], drop_first=True)

from sklearn.model_selection import train_test_split
from sklearn import tree
X = dum_Housing
y = Housing.iloc[:,1]

# Create training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
                                                    random_state=42)

clf = tree.DecisionTreeRegressor(max_depth=2)
clf2 = clf.fit(X_train, y_train)
```

```
y_pred = clf2.predict(X_test)

from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

```
In [138]: mean_squared_error(y_test, y_pred)
Out[138]: 909968.9687529949

In [139]: mean_absolute_error(y_test, y_pred)
Out[139]: 601.9324009754998

In [140]: r2_score(y_test, y_pred)
Out[140]: 0.8337797533789042
```

Sane's STATS Academy of Statistics

Sane's STATS Academy of Statistics

Questions?