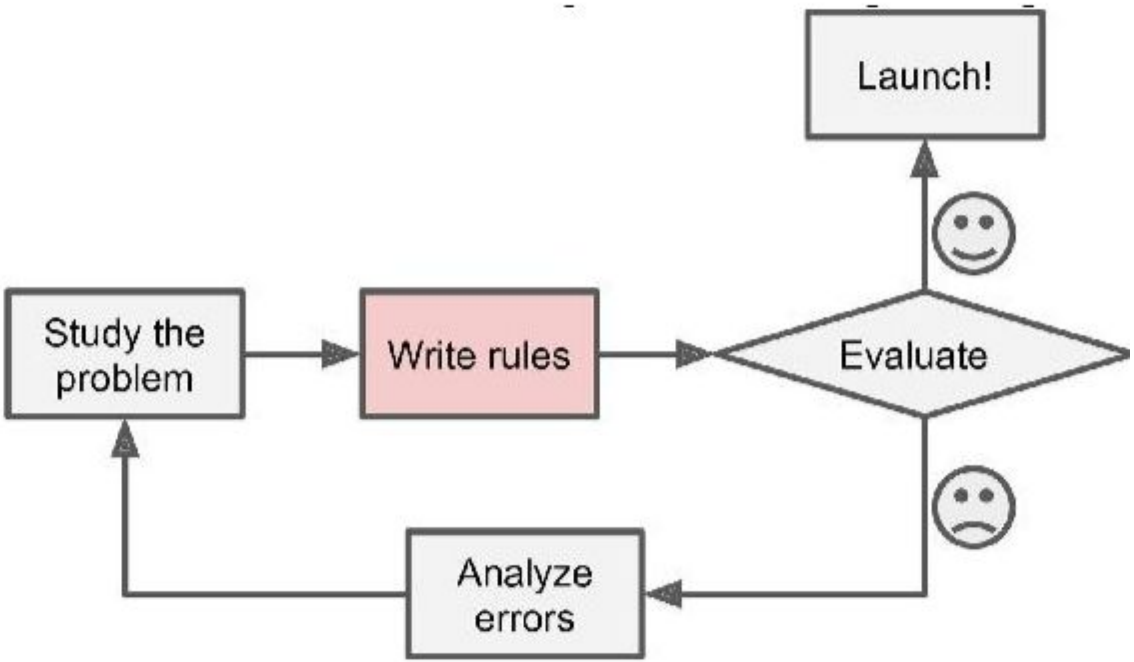
Lecture 6
Non-parametric Estimator

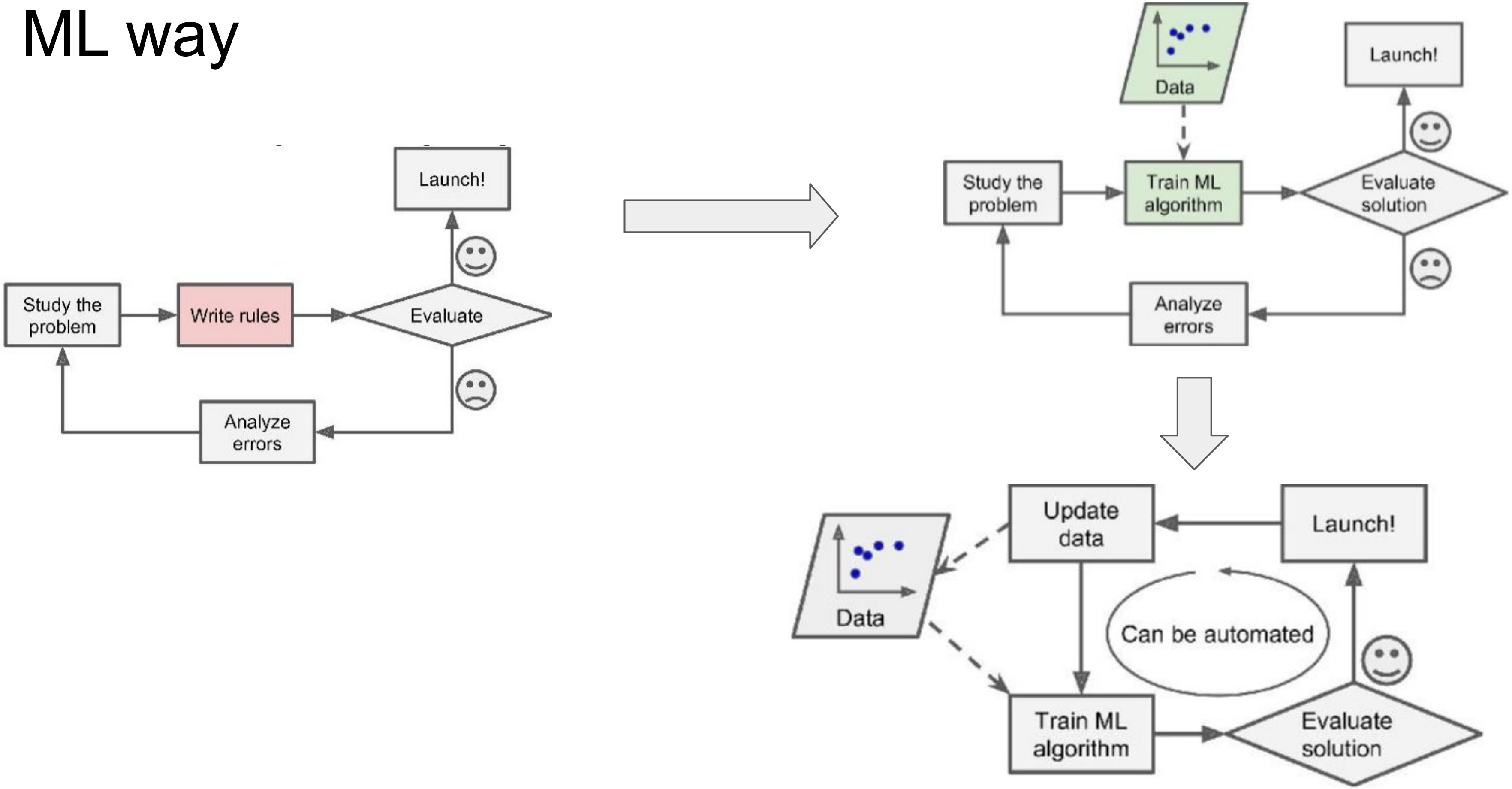
Aug 02

Why use ML?

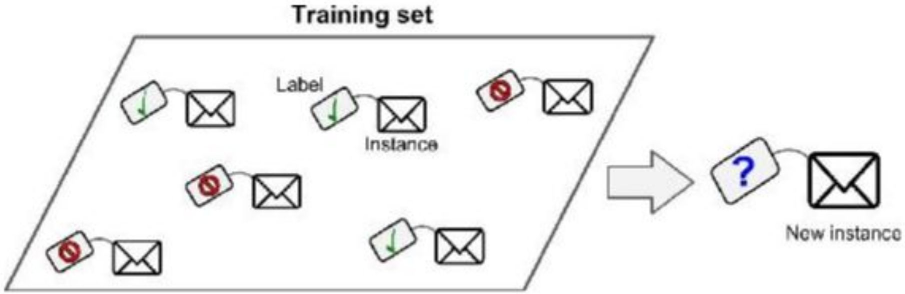


Launch!
Study the problem
Write rules
Evaluate
Analyze errors

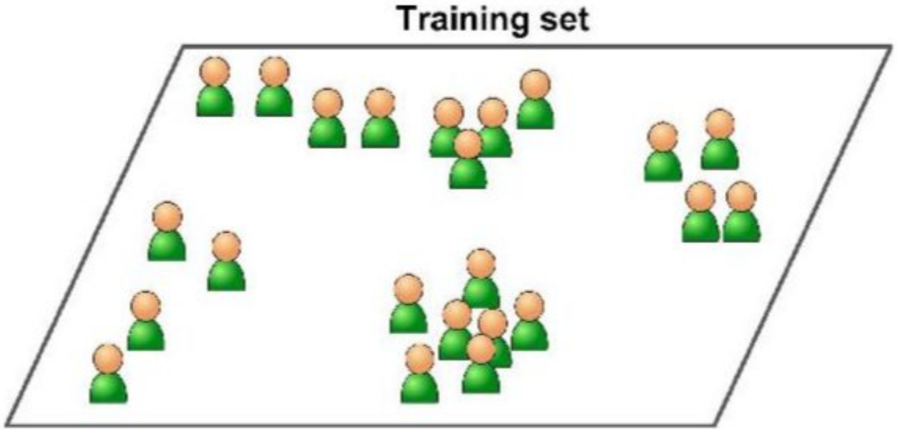
ML way


Launch!
Study the problem
Write rules
Evaluate
Analyze errors
Data
Launch!
Study the problem
Train ML algorithm
Evaluate solution
Analyze errors
Update data
Launch!
Data
Can be automated
Train ML algorithm
Evaluate solution

Supervised, semi-supervised & Unsupervised



Supervised

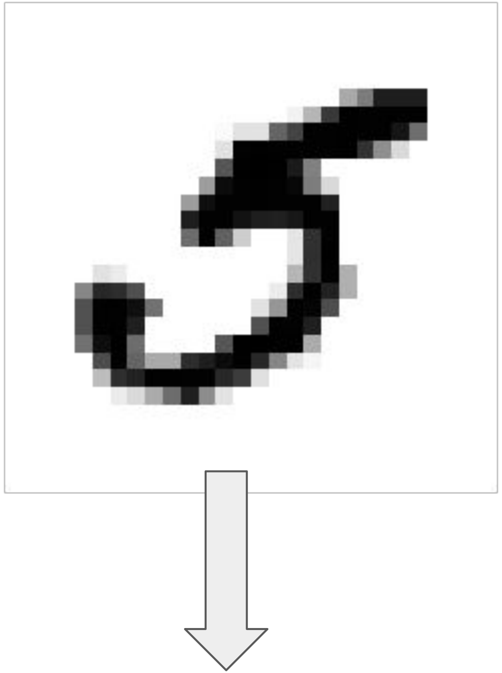


Unsupervised

Classification & Regression

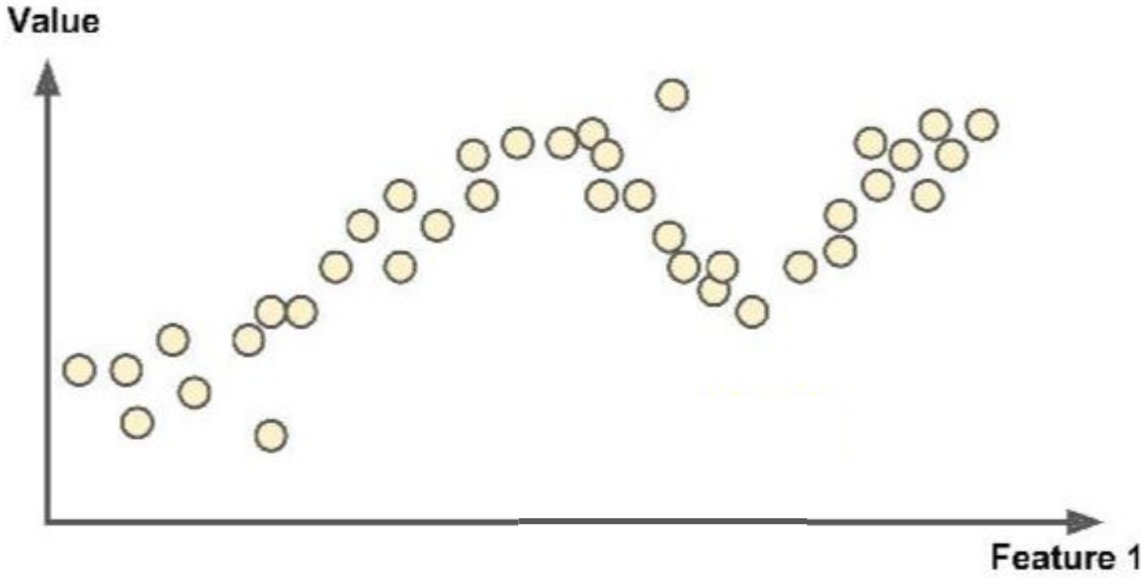
- Classification
  - Image recognition
  - Spam Filtering
- Regression
  - Numeric estimate
  - Predictors
  - House Prices

Classification



5

Regression



Popular supervised learning algorithms

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Decision Trees

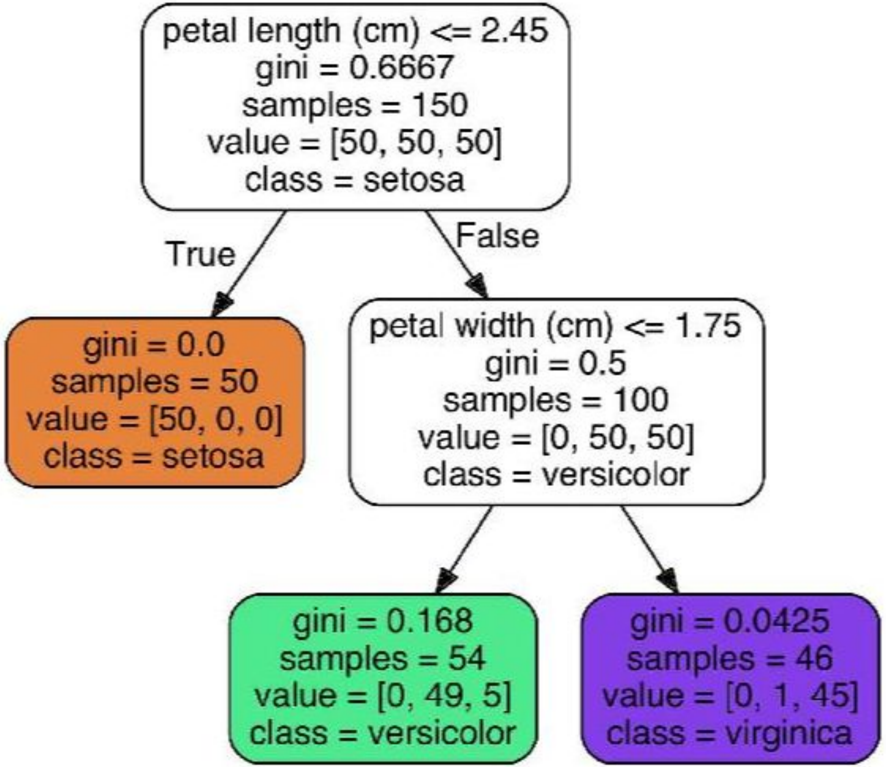
- Can perform both classification and regression tasks.
- Can handle non-linearity in the data
- Capable of fitting complex datasets.
- Fundamental components of Random Forests

Train a tree

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # petal length and width
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(X, y)
```

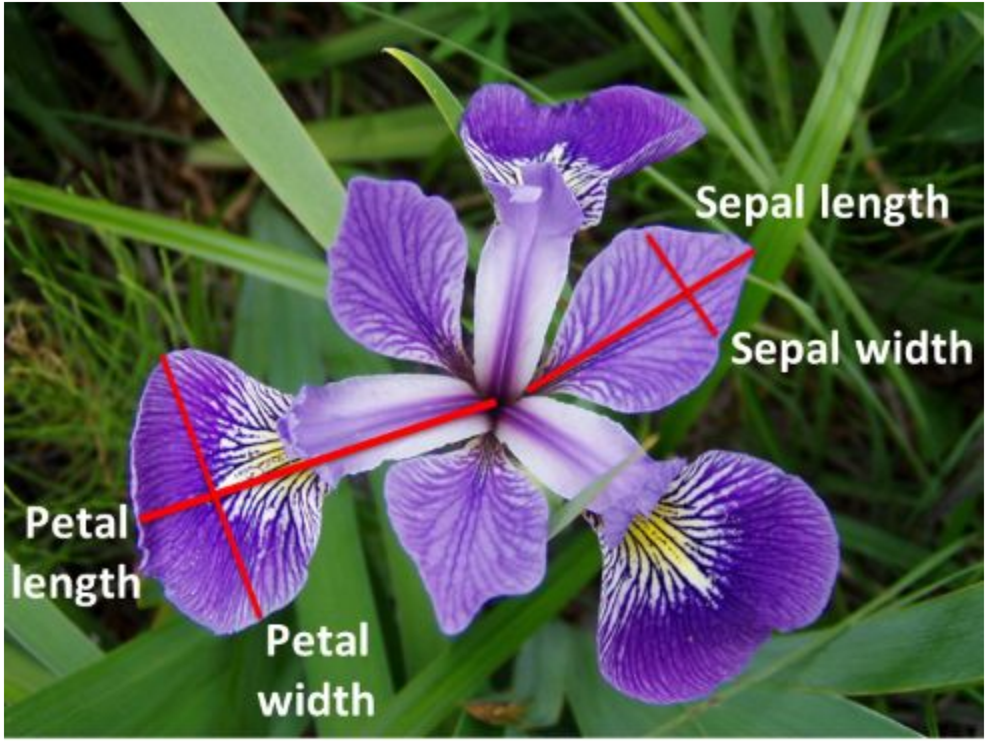
Visualize the tree

```python
from sklearn.tree import export_graphviz

export_graphviz(tree_clf,
                out_file = "tree.dot",
                feature_names = iris.feature_names[2:],
                class_names=iris.target_names)

with open("tree.dot") as f:
    dot_graph = f.read()
graphviz.Source(dot_graph)
```



Iris flower


petal
sepal

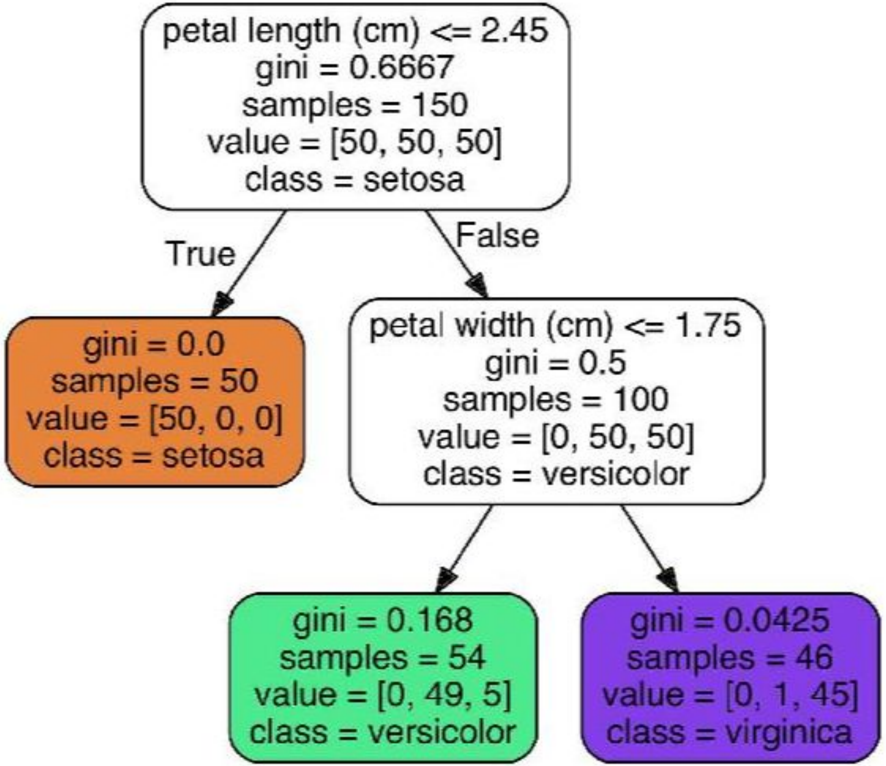

Sepal length
Sepal width
Petal
length
Petal
width

Making Prediction

Explaining Tree

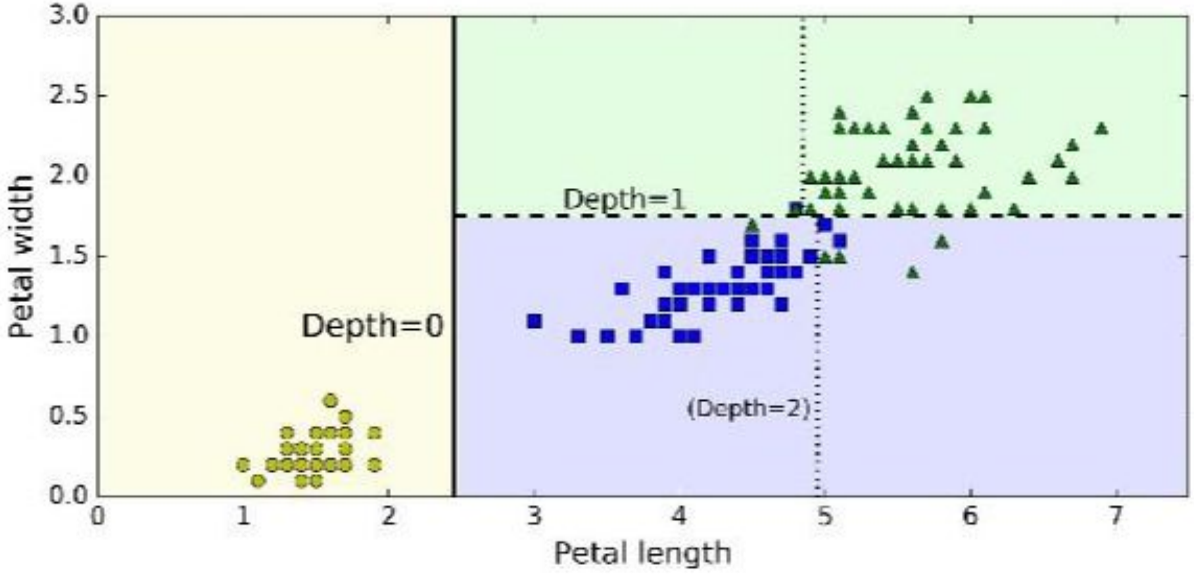
- Node 'Sample' attribute.
- Node 'Value' attribute.
- Gini impurity

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$



Decision Tree boundaries

- Max depth = 2.
- What happens when
  - max depth = 3



Class Probabilities

```
>>> tree_clf.predict_proba([[5, 1.5]])
array([[ 0. ,  0.90740741,  0.09259259]])
>>> tree_clf.predict([[5, 1.5]])
array([1])
```

Algorithm CART

Classification And Regression Tree

- single feature ‘k’
- threshold t_k

- (e.g., “petal length ≤ 2.45 cm”).

CART cost function for classification

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

$$\text{where } \begin{cases} G_{\text{left/right}} \text{ measures the impurity of the left/right subset,} \\ m_{\text{left/right}} \text{ is the number of instances in the left/right subset.} \end{cases}$$

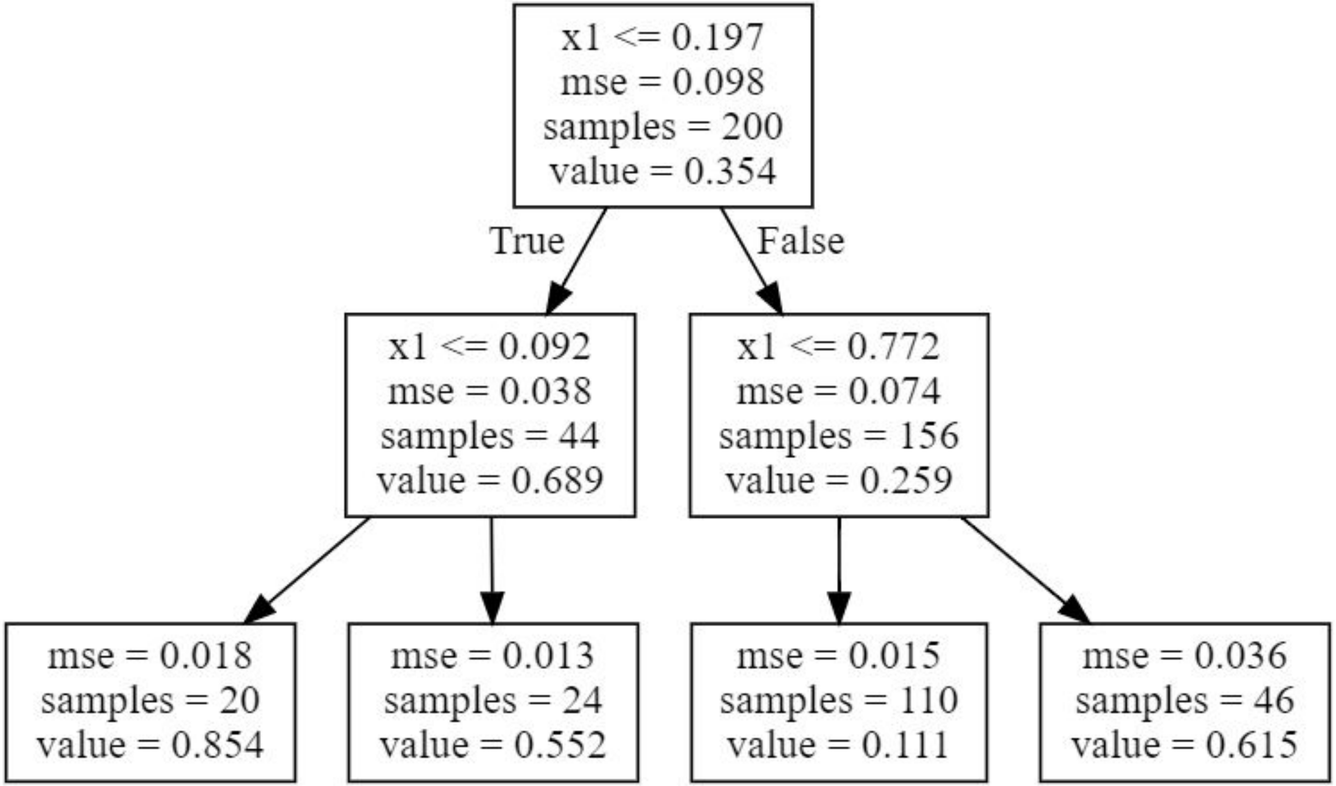
Regression Trees

- Scikit-Learn's *DecisionTreeRegressor* class,

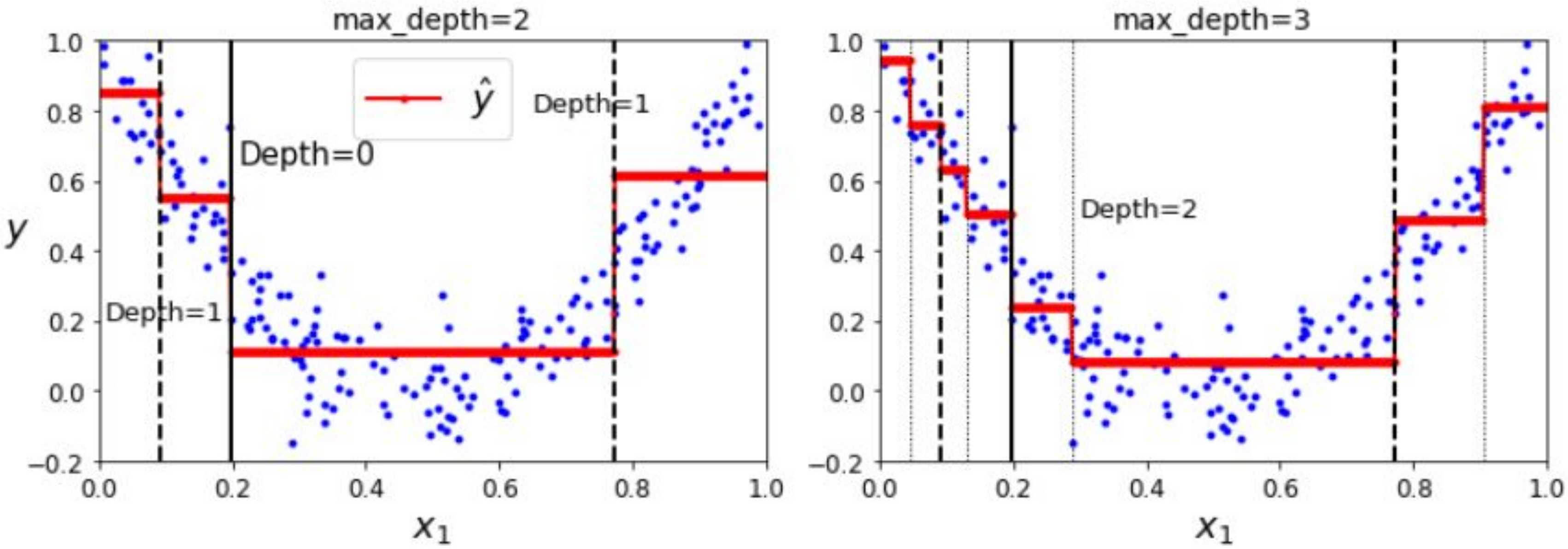
```python
# Quadratic training set + noise
np.random.seed(42)
m = 200
X = np.random.rand(m, 1)
y = 4 * (X - 0.5) ** 2
y = y + np.random.randn(m, 1) / 10
```

```python
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(max_depth=2, random_state=42)
tree_reg.fit(X, y)
```

Regression Tree



Regression Trees 2



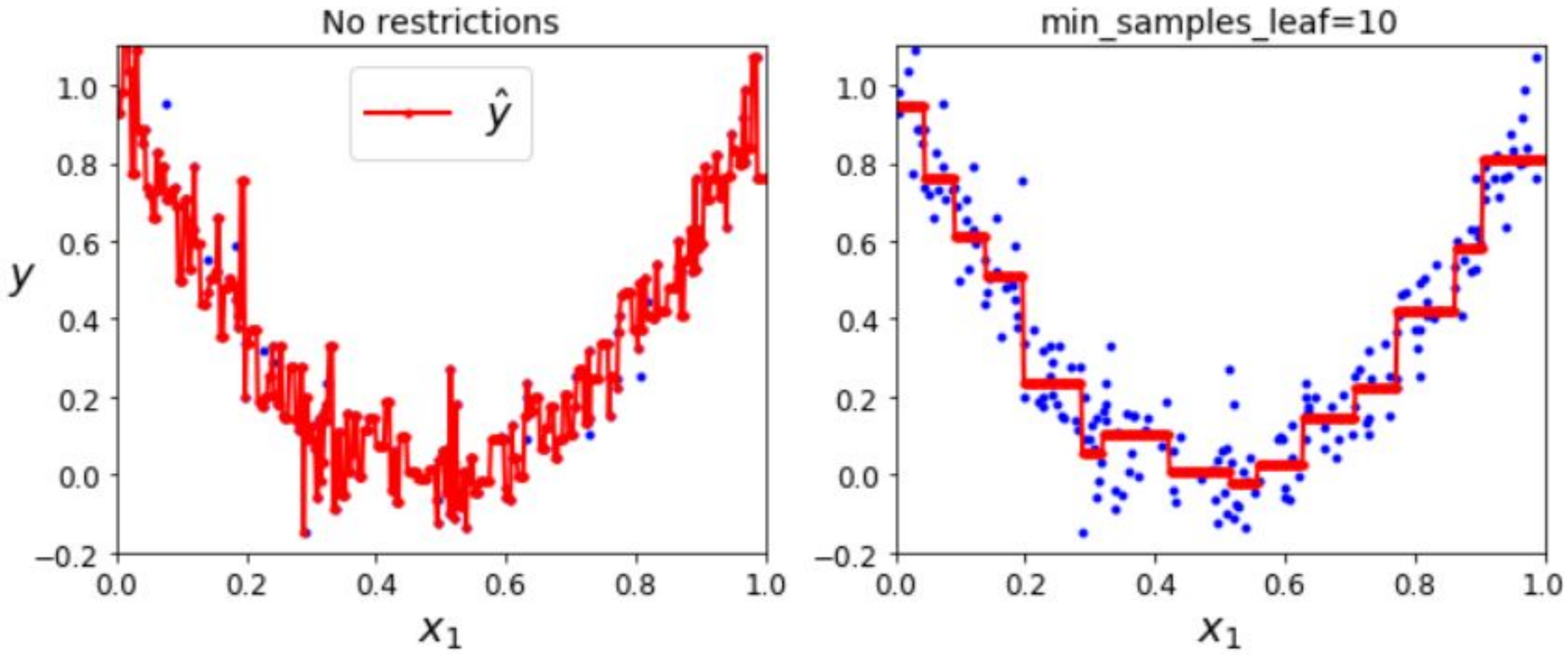
Working

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Overfitting

- What is overfitting?
  - Learning the noise rather than the feature.
- When does this problem occur?
  - Very complex model.
- How does pruning help?

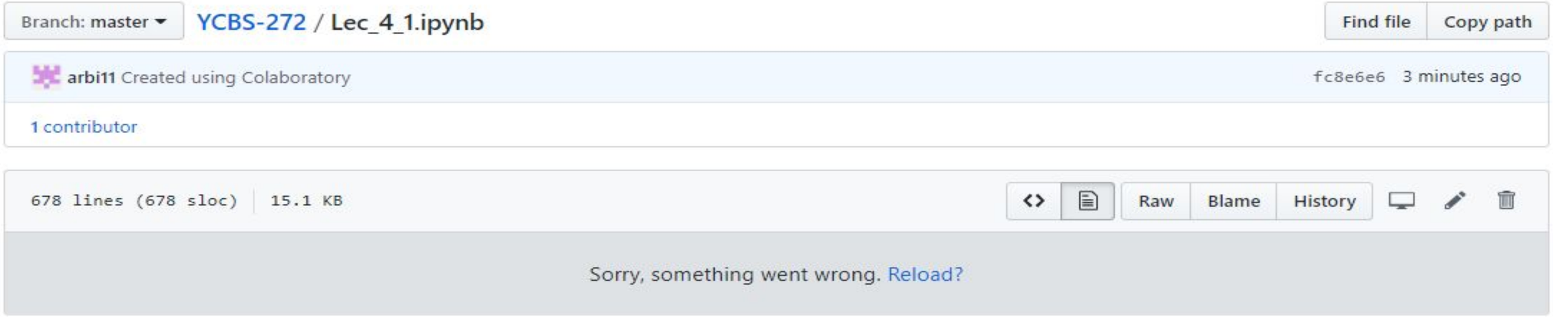
Overfitting - Regression Trees



Link for the notebook

https://github.com/arbi11/YCBS-272/blob/master/Lec_6_decision_trees.ipynb

If you see this error on github



Copy the link (of github page) and paste here:

https://nbviewer.jupyter.org/