

Машинное обучение

Лекция 3

Решающее дерево (decision tree, DT)

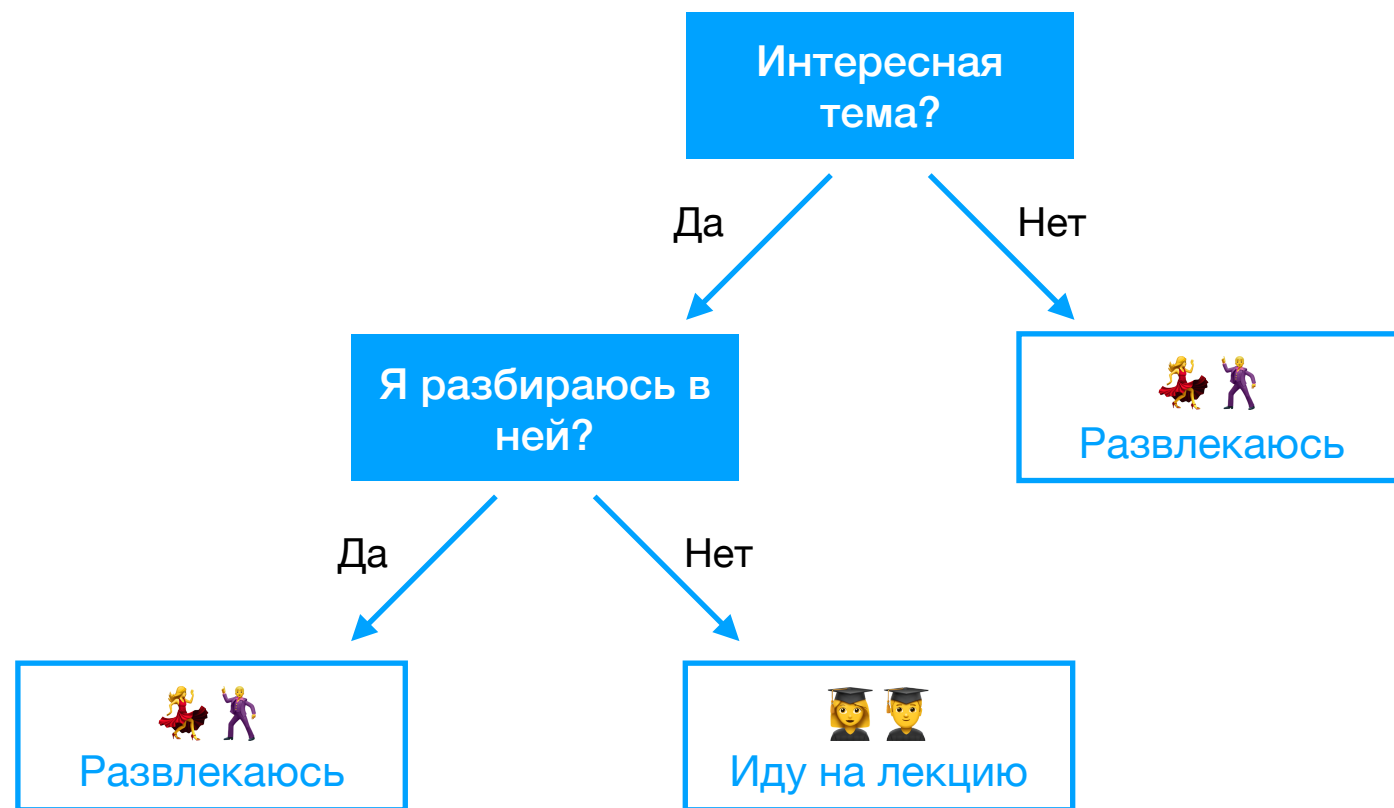
Власов Кирилл Вячеславович



Деревья принятия решений

Логический алгоритм классификации, основанный на поиске конъюнктивных закономерностей.

Пойду ли я на факультатив по МО сегодня?



Деревья принятия решений

Дерево решений служит обобщением опыта экспертов, средством передачи знаний будущим сотрудникам или моделью бизнес-процесса компании.

Решение о выдаче кредита заемщику принималось на основе некоторых интуитивно (или по опыту) выведенных правил, которые можно представить в виде дерева решений.

Пример: Кредитный скоринг



Деревья принятия решений



`sklearn.datasets.load_iris`

`sklearn.tree.DecisionTreeClassifier`

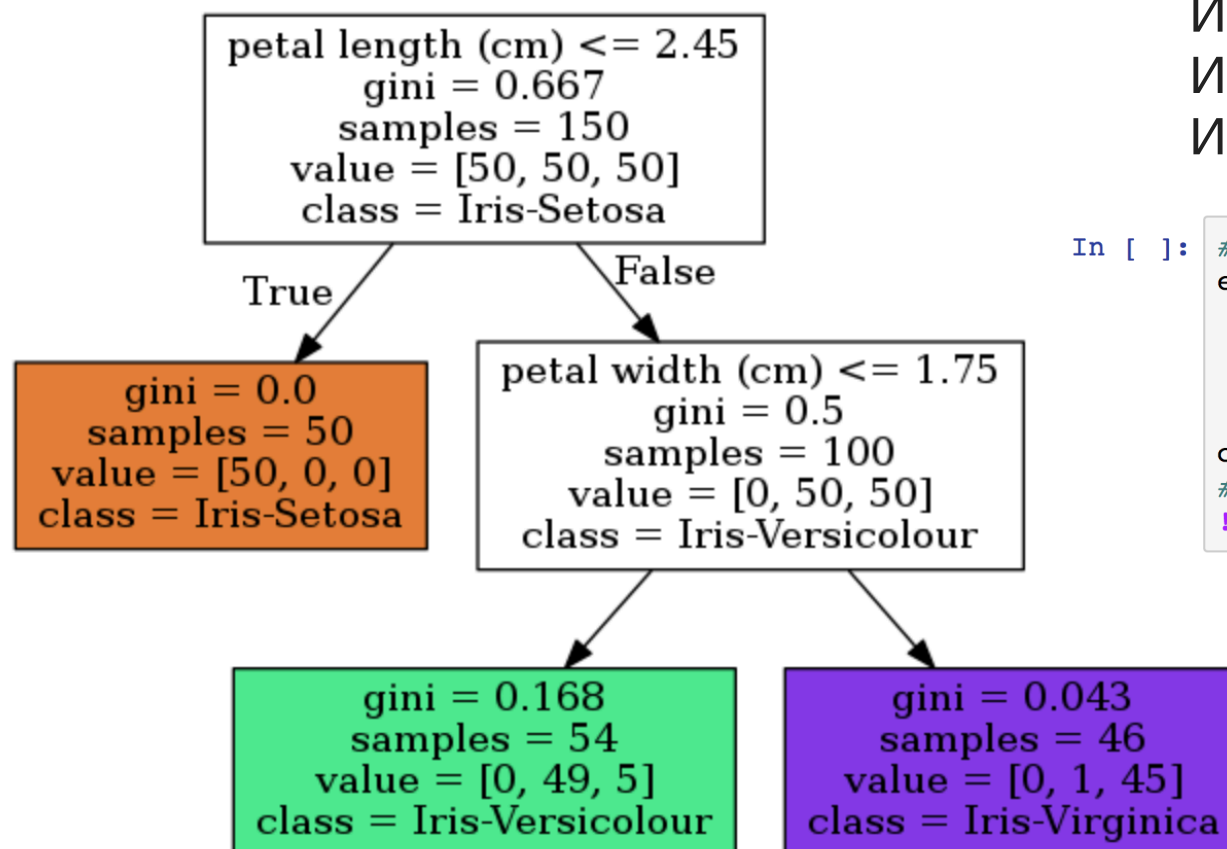
`sklearn.tree.export_graphviz`

Классы:

Ирис щетинистый (Iris setosa)

Ирис виргинский (Iris virginica)

Ирис разноцветный (Iris versicolor)



```
In [ ]: # Отрисует дерево
export_graphviz(tree, feature_names=load_iris()['feature_names'],
                class_names=['Iris-Setosa',
                             'Iris-Versicolour',
                             'Iris-Virginica'],
                out_file='iris_tree.dot', filled=True)
# для этого понадобится библиотека pydot (pip install pydot)
!dot -Tpng 'iris_tree.dot' -o 'iris_tree.png'
```

Признаки:

длина чашелистика (см)

ширина чашелистика (см)

длина лепестка (см)

ширина лепестка (см)

Деревья принятия решений



```
sklearn.datasets.load_iris
```

```
sklearn.tree.DecisionTreeClassifier
```

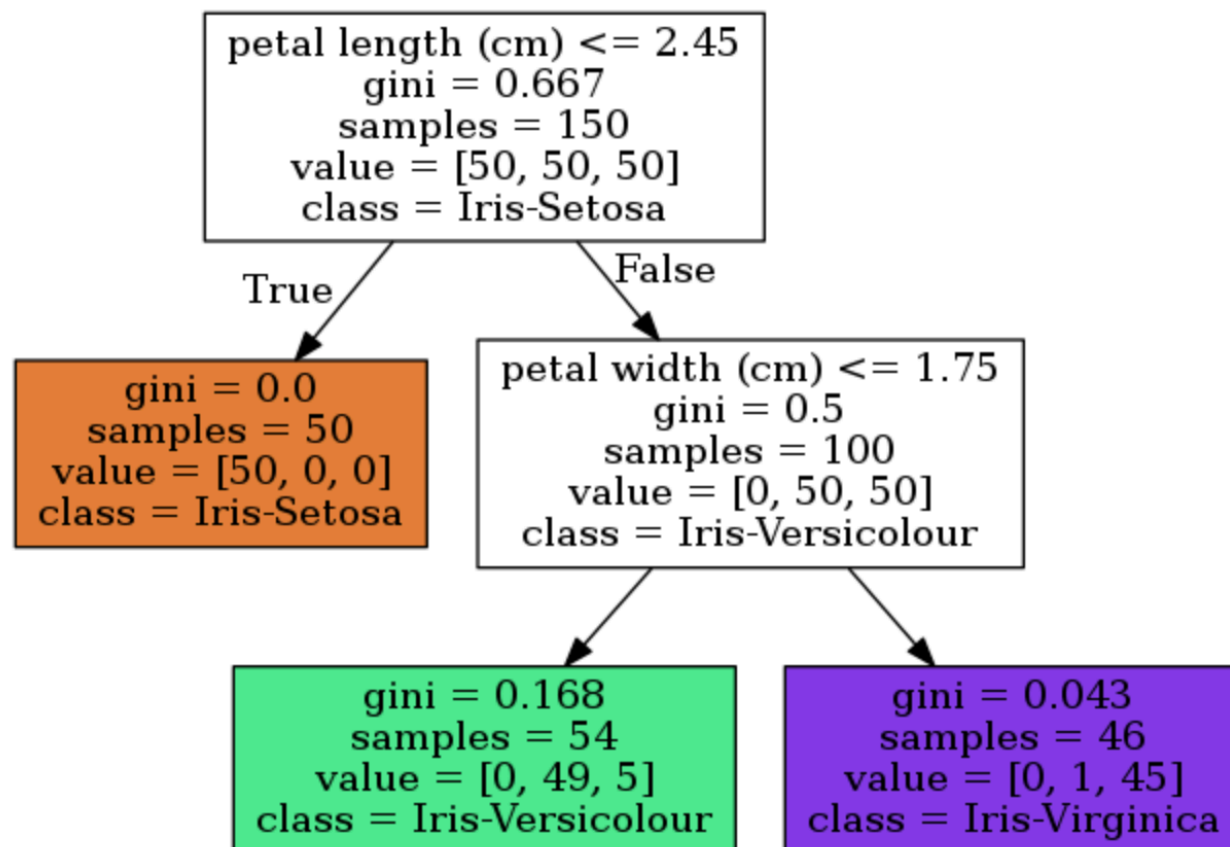
```
sklearn.tree.export_graphviz
```

Неопределенность Джини (Gini impurity):

$$G_i = 1 - \sum_{k=1}^n (p_{ik})^2$$

$$G_{split} = \frac{L}{N} \times G_L + \frac{R}{N} \times G_R \rightarrow \min$$

L - Количество элементов в левой ветке
R - Количество элементов в правой ветке
N - Количество элементов в узле



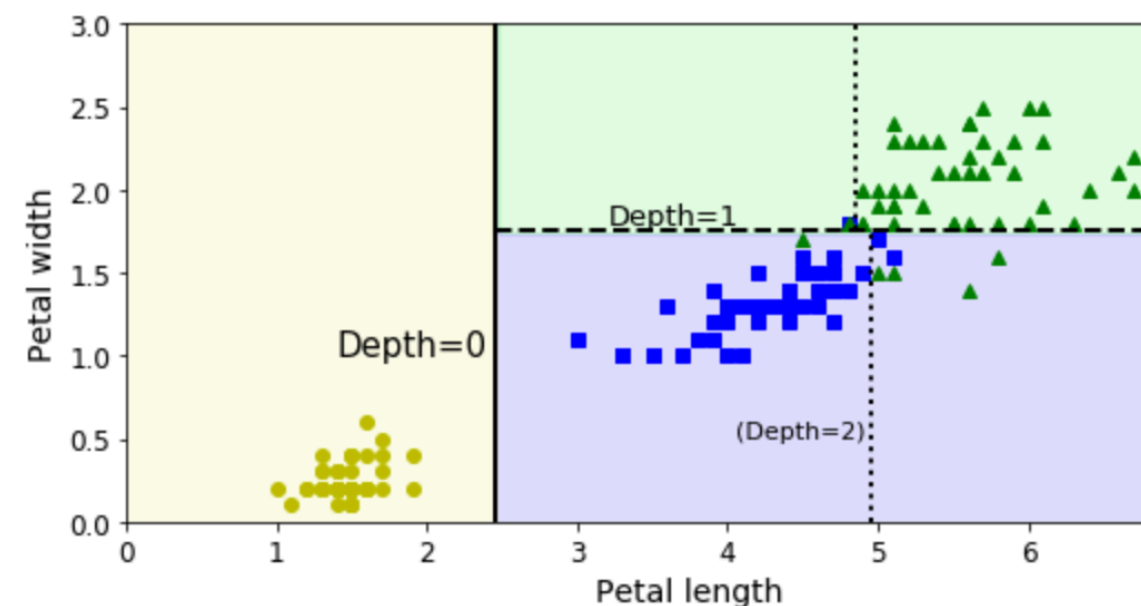
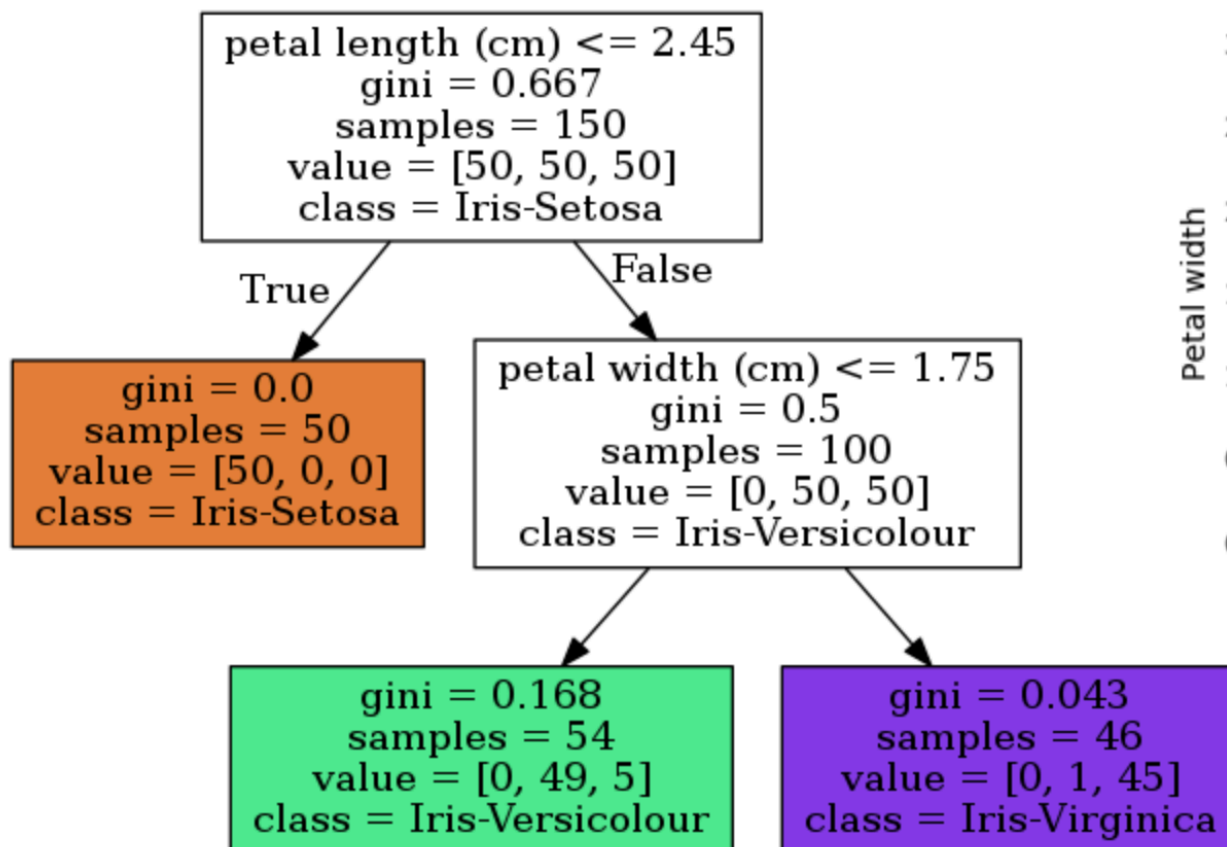
Деревья принятия решений



```
sklearn.datasets.load_iris
```

```
sklearn.tree.DecisionTreeClassifier
```

```
sklearn.tree.export_graphviz
```



Деревья принятия решений



```
sklearn.datasets.load_iris
```

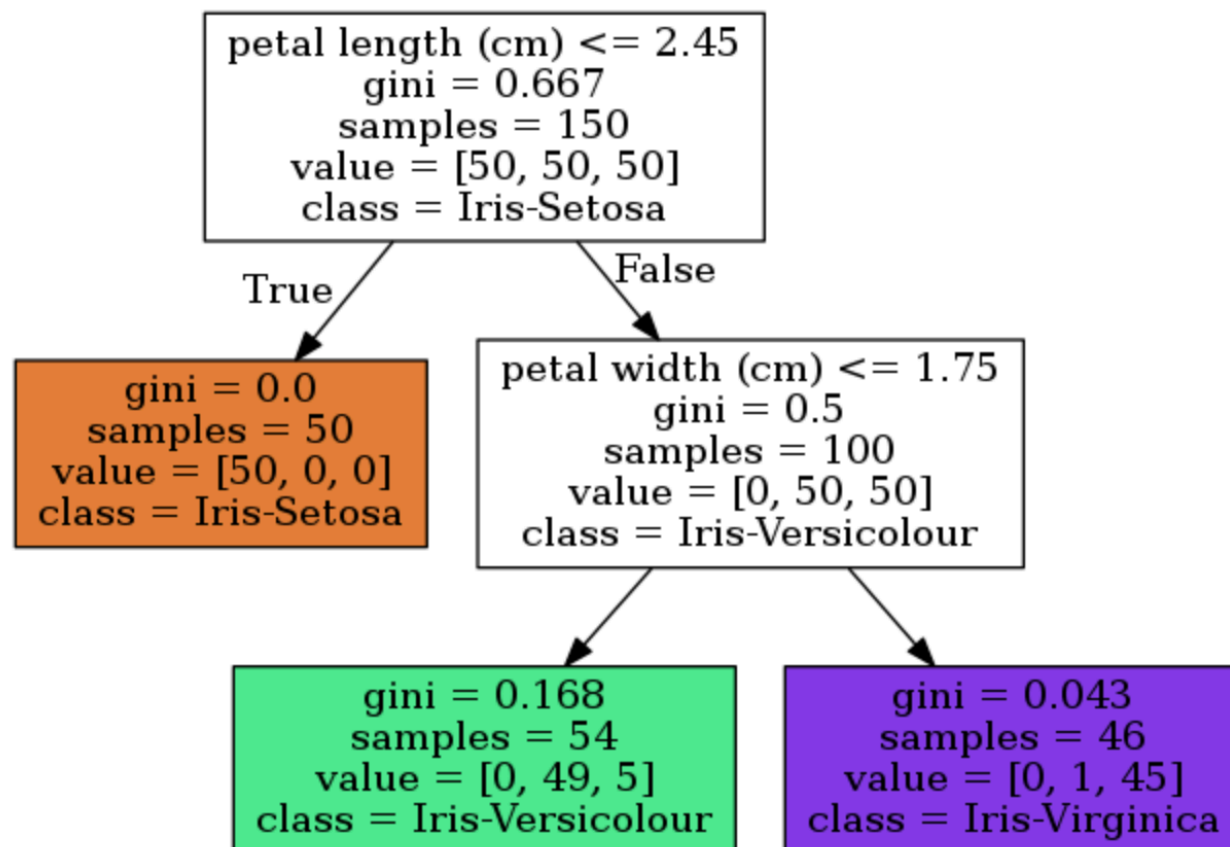
```
sklearn.tree.DecisionTreeClassifier
```

```
sklearn.tree.export_graphviz
```

Энтропия Шеннона:

$$S_i = - \sum_{k=1}^n p_{ik} \log p_{ik}$$

p_{ik} - Вероятность нахождения в состоянии k



Деревья принятия решений



```
sklearn.datasets.load_iris
```

```
sklearn.tree.DecisionTreeClassifier
```

```
sklearn.tree.export_graphviz
```

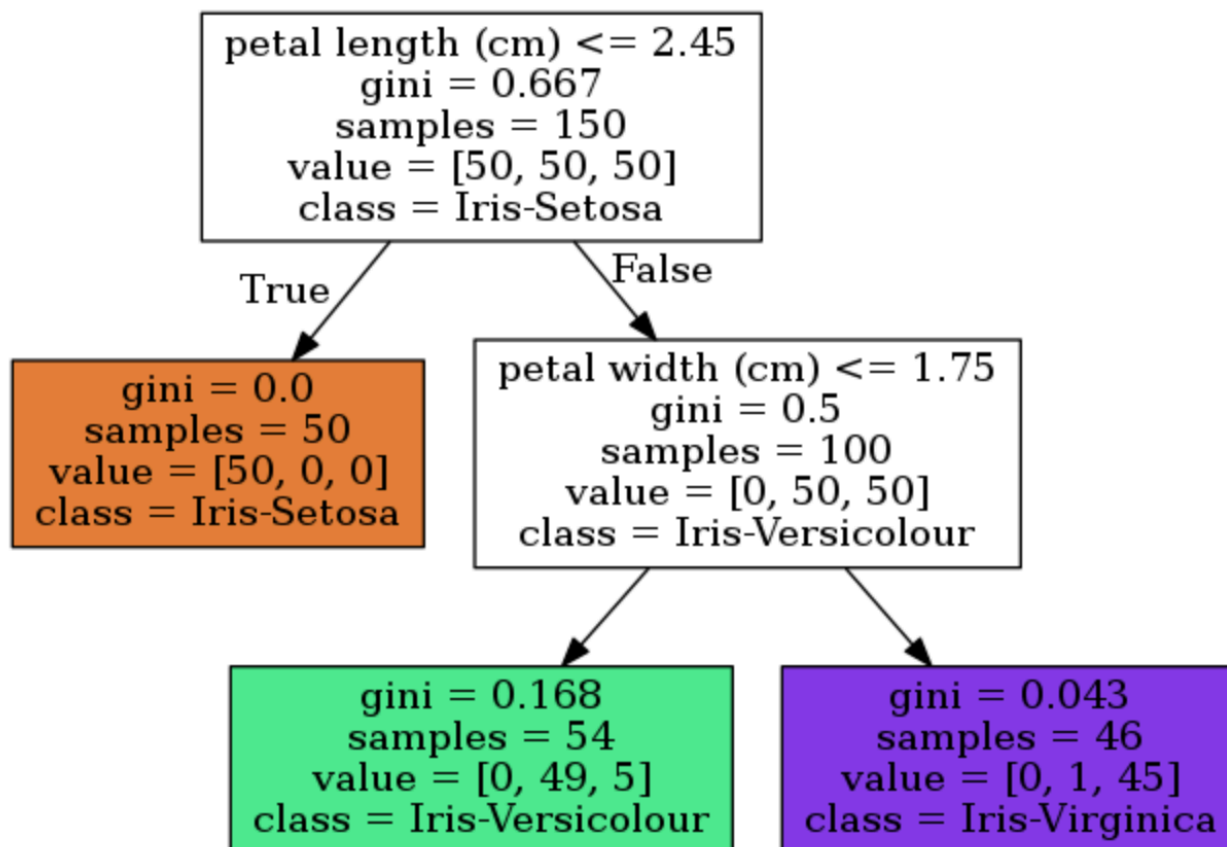
```
tree.predict_proba( [2,3,3,1] )
```

Классы:

Ирис щетинистый (Iris setosa) - 0

Ирис виргинский (Iris virginica) - 0,907

Ирис разноцветный (Iris versicolor) - 0,093



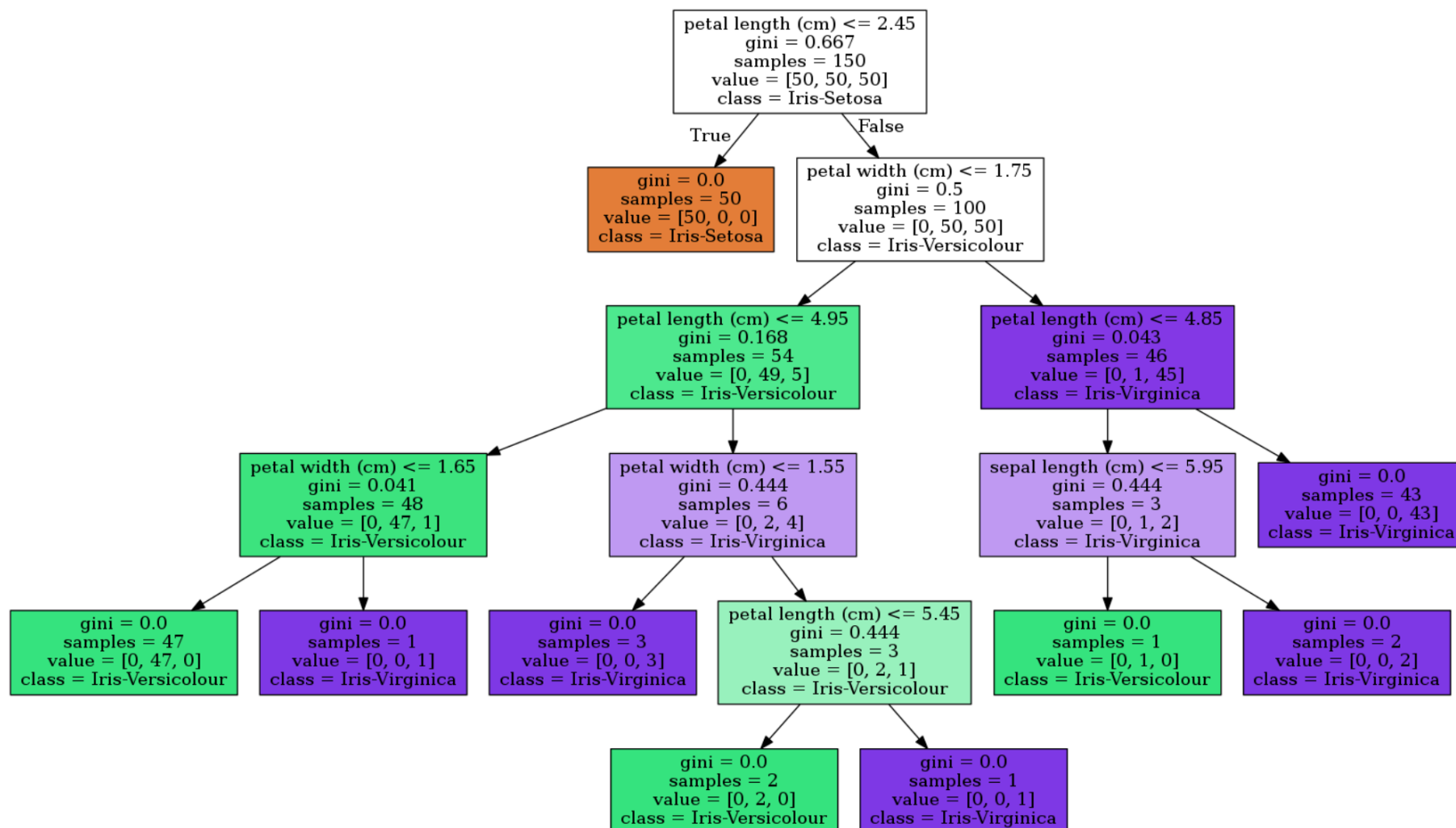
$$p_1 = \frac{0}{54} \quad p_2 = \frac{49}{54} \quad p_3 = \frac{5}{54}$$

Деревья принятия решений



`sklearn.tree`.DecisionTreeClassifier

(*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False*)

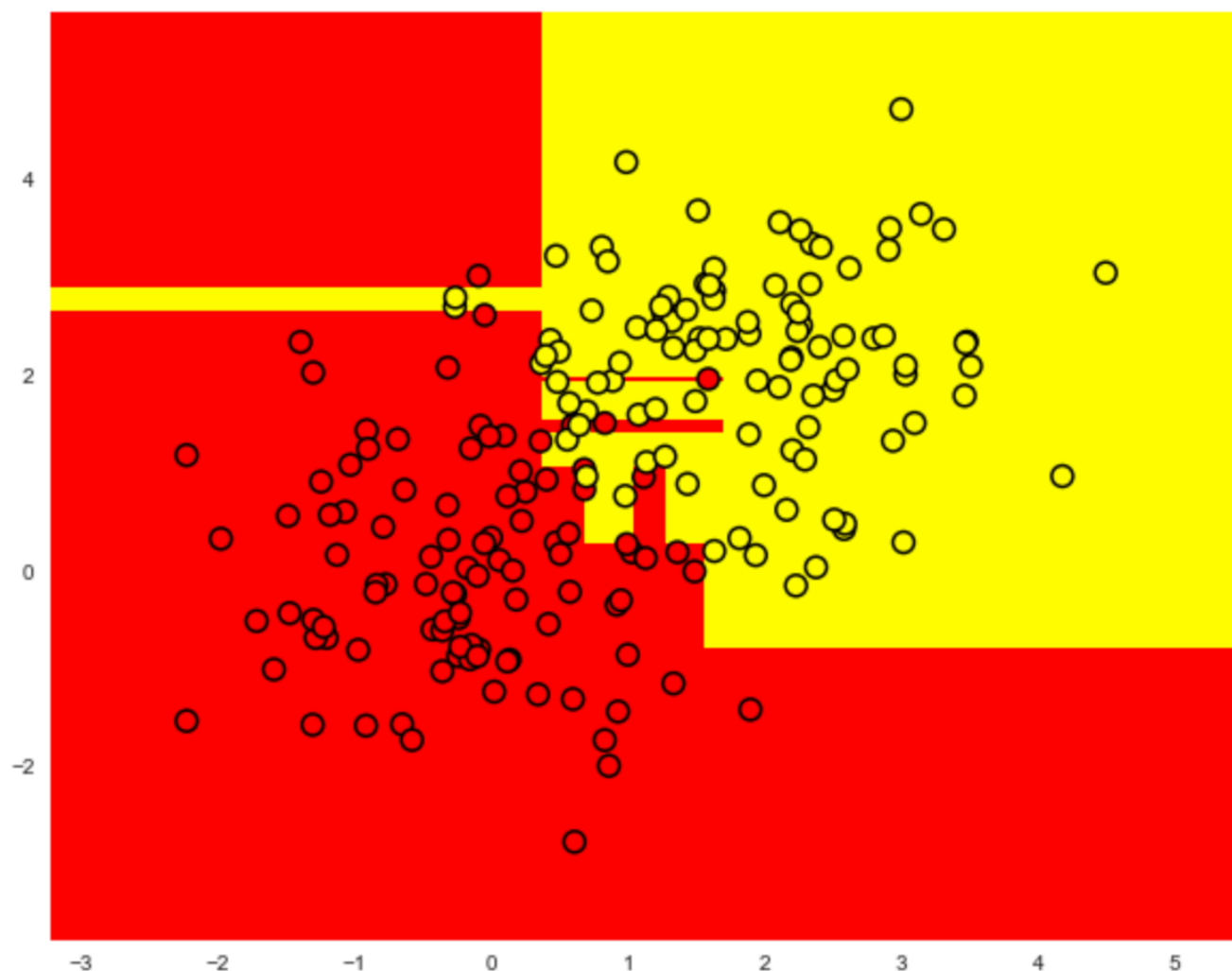


Регуляризация деревьев



`sklearn.tree.DecisionTreeClassifier`

(*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False*)



max_depth - глубина дерева

min_samples_split - Минимальное количество объектов, прежде чем можно сделать разделение

min_samples_leaf - Минимальное кол-во объектов в листовом узле

max_leaf_nodes - Максимальное количество листовых узлов

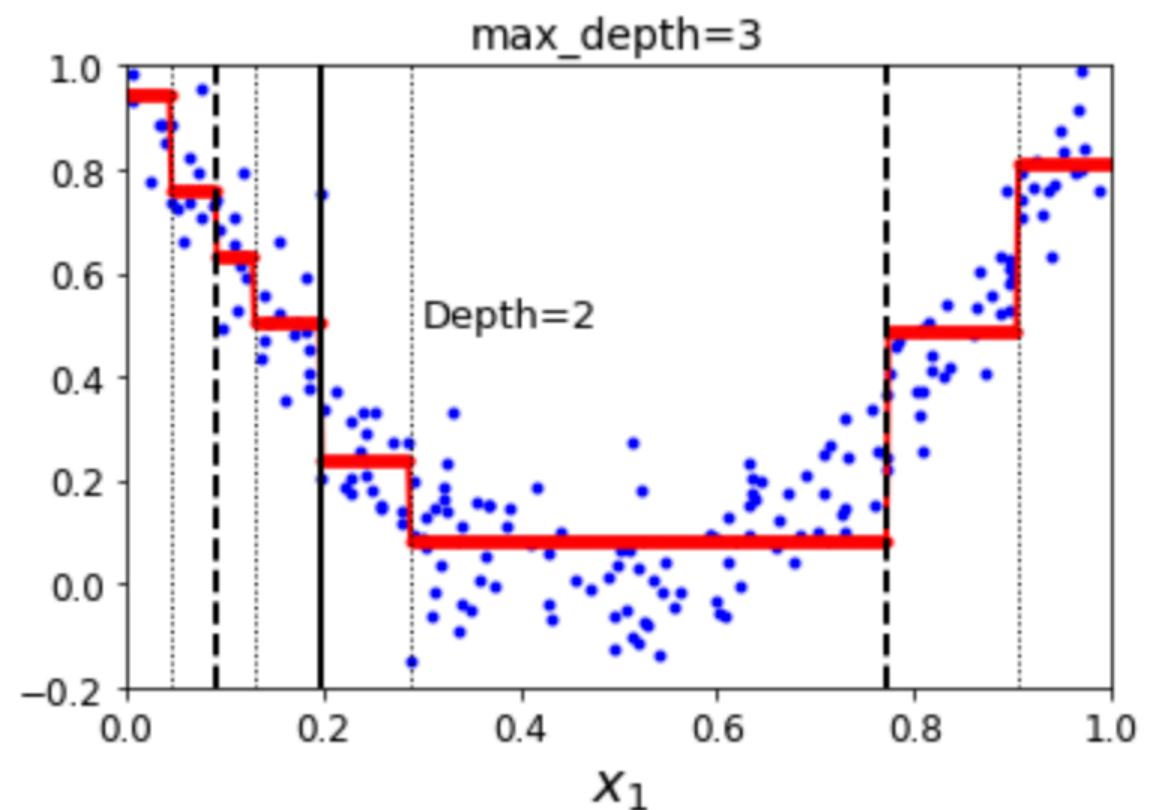
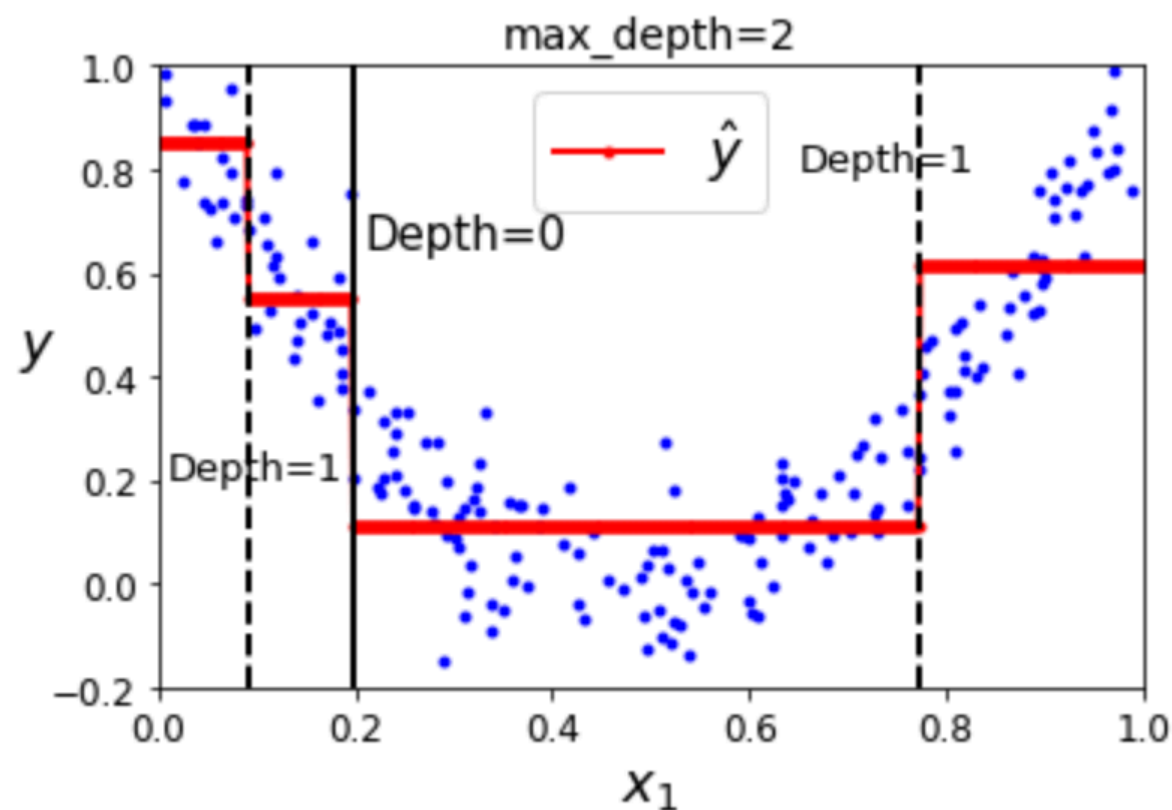
Семинар Евгения Соколова

Деревья решений для задачи регрессии



`sklearn.tree.DecisionTreeRegressor`

(*criterion='mse', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort=False*)

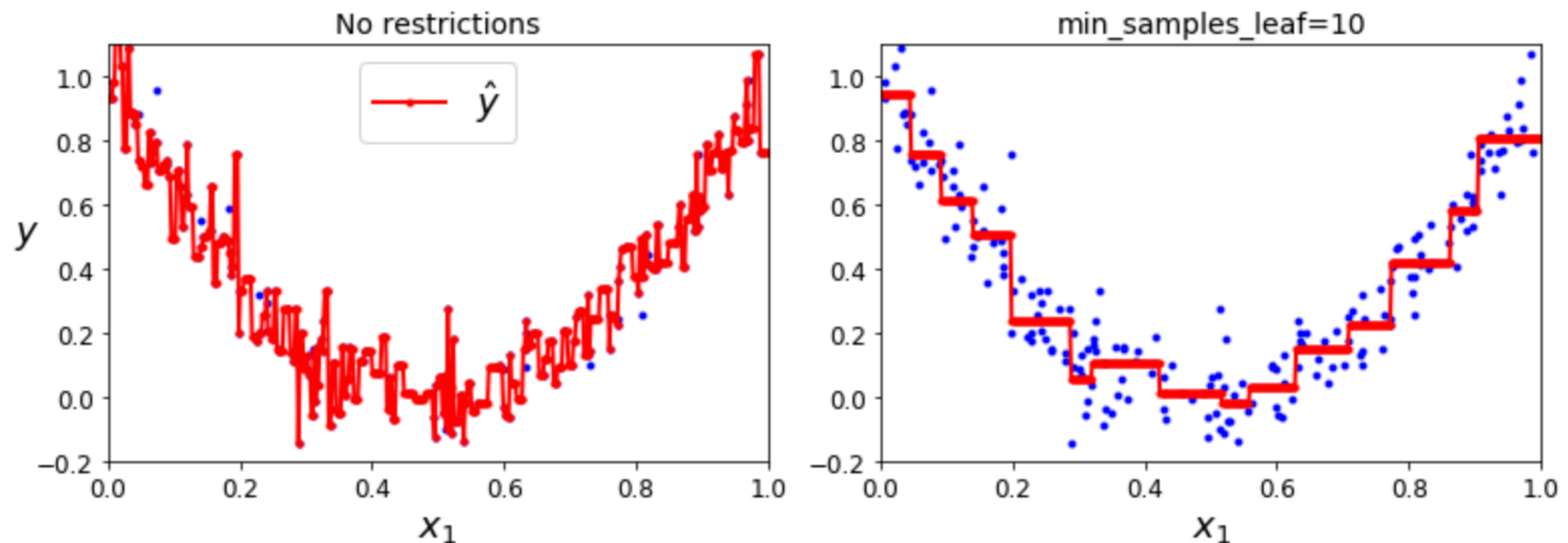


Деревья решений для задачи регрессии



`sklearn.tree.DecisionTreeRegressor`

(*criterion='mse', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort=False*)



Какой ответ деревьев в регрессии?

Какая стратегия поведения в листьях регрессионного дерева приводит к меньшему матожиданию ошибки по MSE: отвечать средним значением таргета на объектах обучающей выборки, попавших в лист, или отвечать таргетом для случайного объекта из листа (считая все объекты равновероятными)?

Какой ответ деревьев в регрессии?

Какая стратегия поведения в листьях регрессионного дерева приводит к меньшему матожиданию ошибки по MSE: отвечать средним значением таргета на объектах обучающей выборки, попавших в лист, или отвечать таргетом для случайного объекта из листа (считая все объекты равновероятными)?

- $\hat{y} = \frac{1}{n} \sum_{i=1}^n c_i$

$$E(y - \frac{1}{n} \sum_{i=1}^n c_i)^2 = Ey^2 + \left(\frac{1}{n} \sum_{i=1}^n c_i \right)^2 - 2 \left(\frac{1}{n} \sum_{i=1}^n c_i \right) Ey$$

- $\hat{y} = X$, где $X \sim U(c)$

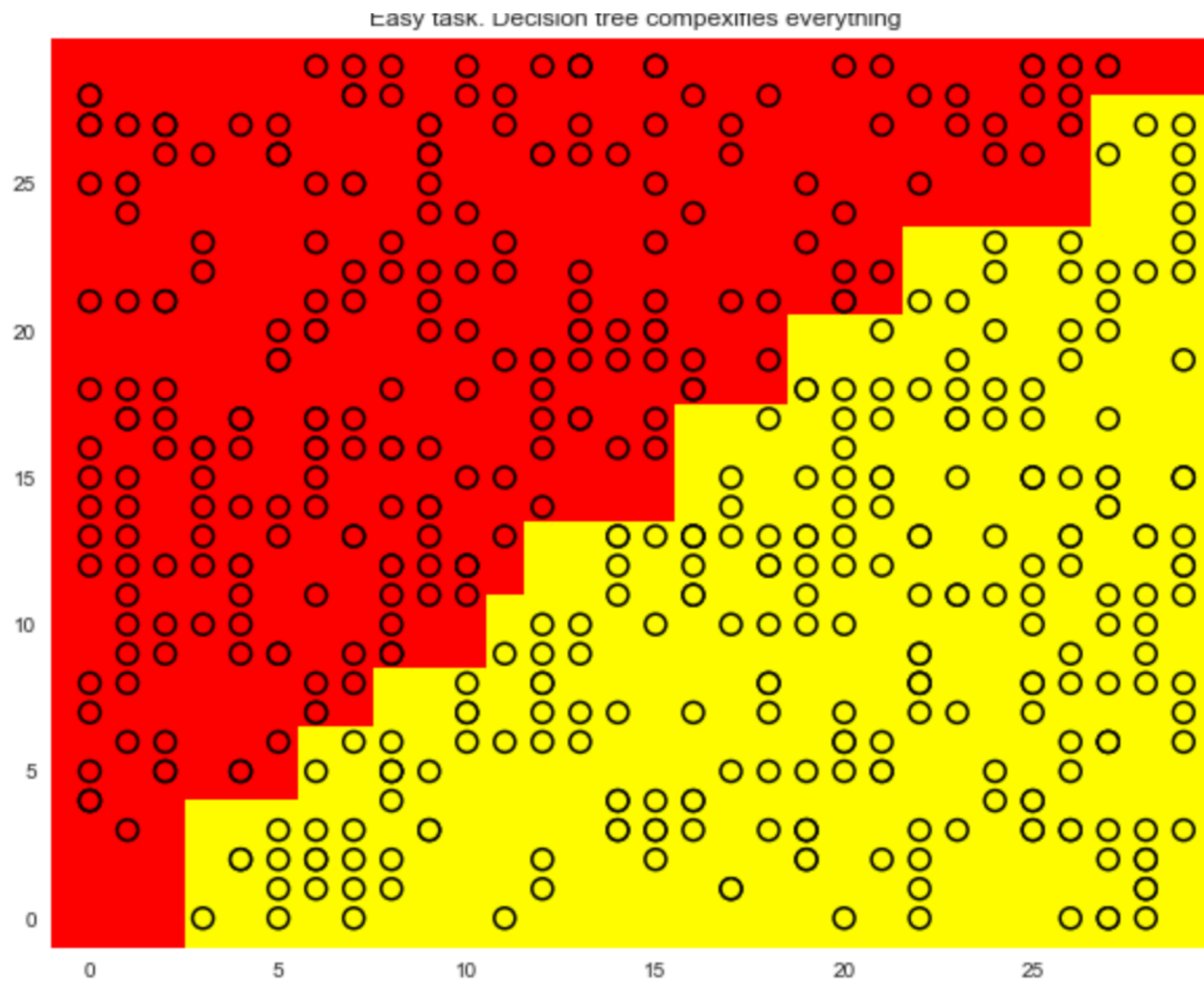
$$E \frac{1}{n} \sum_{i=1}^n (y - c_i)^2 = \frac{1}{n} \sum_{i=1}^n E(y - c_i)^2 = Ey^2 + \frac{1}{n} \sum_{i=1}^n c_i^2 - \frac{2}{n} Ey \sum_{i=1}^n c_i$$

Тогда выпишем их разность:

$$E \frac{1}{n} \sum_{i=1}^n (y - c_i)^2 - E(y - \bar{c})^2 = \frac{1}{n} \sum_{i=1}^n c_i^2 - \left(\frac{1}{n} \sum_{i=1}^n c_i \right)^2 \geq 0 \text{ (По неравенству Коши-Буняковского)}$$

Получили, что мат. ожидание ошибки для первого поведения меньше, чем для второго.

Сложные случаи для деревьев



Ссылки на использованные материалы

Открытый курс машинного обучения: Тема 3

Семинар Евгения Соколова