|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«Классификация с помощью деревьев решений»**

**ДИСЦИПЛИНА: «Технологии анализа данных»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-82Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Ерохин И.И. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2024

**Цель:** формирование практических навыков работы с деревьями принятия решений.

**Задачи:** ознакомиться с работой деревьев принятия решений.

**Вариант 5**

Считать данные из файла iris\_df.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Sepal width».

**Листинг:**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeRegressor, plot\_tree

from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot as plt

iris\_df = pd.read\_csv('iris\_df.csv')

label\_encoder = LabelEncoder()

iris\_df['Species'] = label\_encoder.fit\_transform(iris\_df['Species'])

X = iris\_df.drop(columns=['Sepal width'])

y = iris\_df['Sepal width']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

depths = [2, 3]

models = []

for depth in depths:

    model = DecisionTreeRegressor(max\_depth=depth)

    model.fit(X\_train, y\_train)

    models.append(model)

for i, model in enumerate(models, 1):

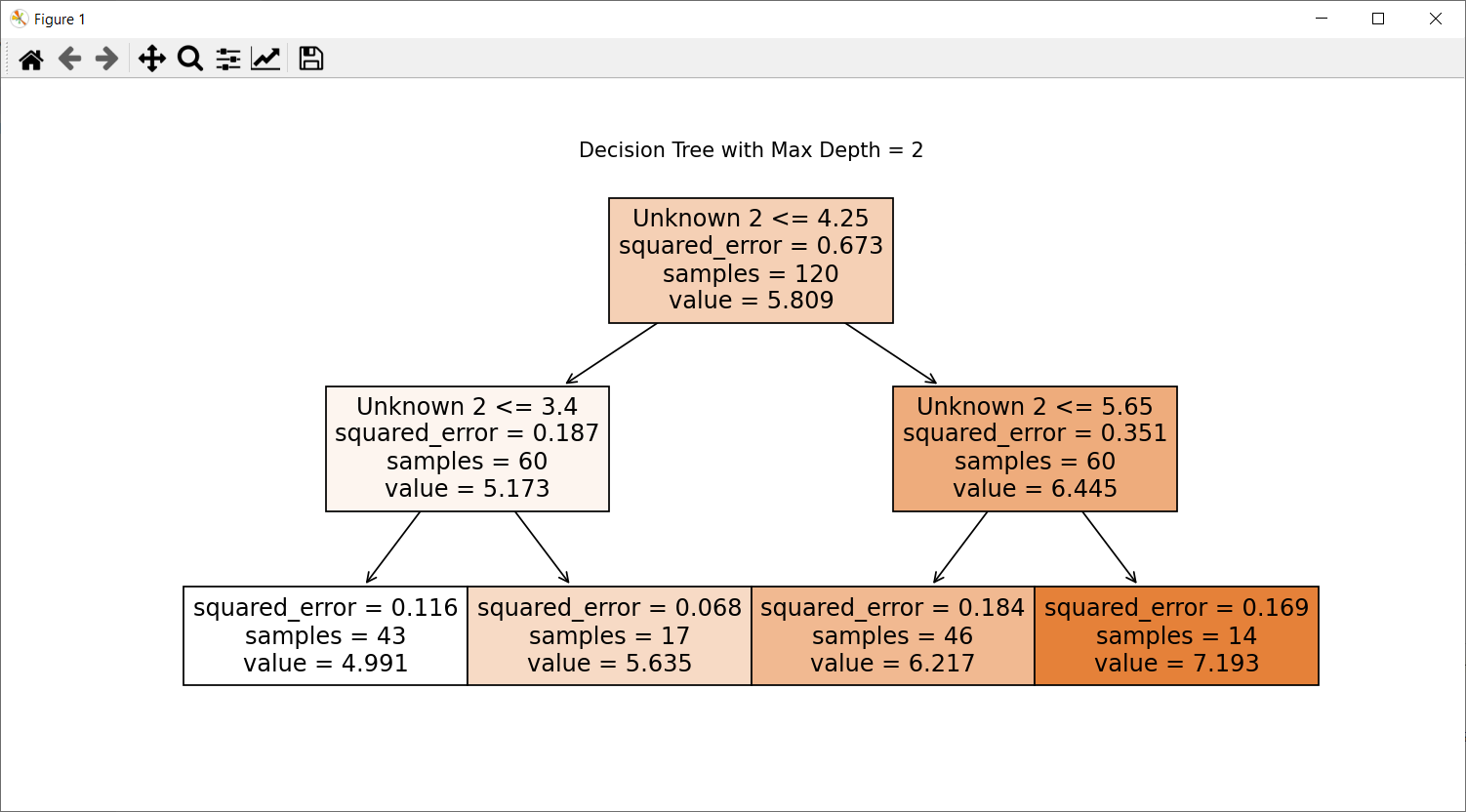
    plt.figure(figsize=(12, 6))

    plot\_tree(model, filled=True, feature\_names=X.columns)

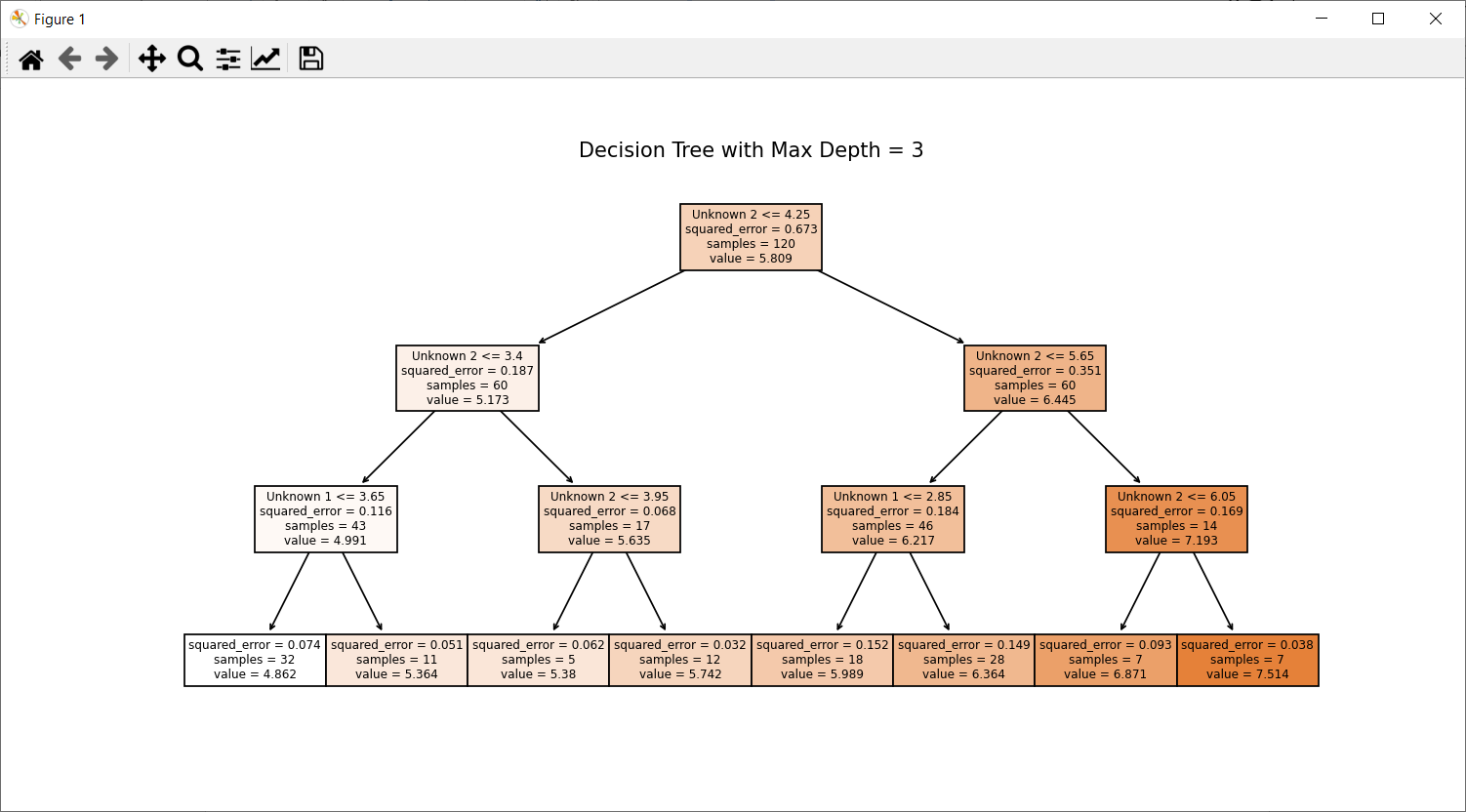
    plt.title(f'Decision Tree with Max Depth = {depths[i-1]}')

    plt.show()

**Результат:**



**Рис. 1.** Дерево принятия решений глубиной 2



**Рис. 2.** Дерево принятия решений глубиной 3

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки работы с деревьями принятия решений.