Деревья принятия решений

Деревья решений являются одним из наиболее эффективных инструментов интеллектуального анализа данных и предсказательной аналитики, которые позволяют решать задачи классификации и регрессии.

Перед тем как непосредственно перейти к решению задач с использование данного инструмента рассмотрим общее понятие "дерево" в информатике и способы задания деревьев в языке Python.

Деревья принадлежат к числу основных структур данных, используемых в программировании. Древовидная структура является одним из способов представления иерархической структуры в графическом виде. Такое название она получила потому, что граф выглядит как перевернутое дерево. Корень дерева (корневой узел) находится на самом верху, а листья (потомки) — внизу.

Деревья широко применяются в компьютерных технологиях. Примером является файловая система, представляющая собой иерархическую структуру из файлов и каталогов.

Схематично дерево и его основные элементы приведены на рисунке ниже.

tree

На рисунке изображены родительские отношения (ребра, ветви дерева) между узлами (вершинами) дерева. На верхнем уровне каждый «родитель» указывает на своих «потомков». То есть в этой иерархической структуре вершина всегда «знает» своих потомков.

Для того чтобы более точно оперировать структурой Дерево, нужно дать определение некоторым ключевым понятиям:

корневой узел — самый верхний узел дерева, он не имеет предков;

лист, листовой или терминальный узел — конечный узел, то есть не имеющий потомков;

внутренний узел — любой узел дерева, имеющий потомков, то есть не лист.

С корневого узла начинается выполнение большинства операций над деревом. Чтобы получить доступ к любому элементу структуры, необходимо, переходя по ветвям, перебирать элементы, начиная с головы — корневого узла. Корневой узел — это своеобразный вход в дерево. Большинство алгоритмов работы с деревом строятся на том, что каждый узел дерева рассматриваются как корневой узел поддерева, «растущего» из этого узла. Такой подход дает возможность зацикливать выполнение операций при прохождении по элементам дерева. Но в связи с тем, что при прохождении по дереву (в отличие от массива) неизвестно сколько шагов будет в этом цикле, используется другой инструмент — рекурсивный вызов.

Двоичное (бинарное) дерево — это древовидная структура данных, где каждый узел имеет не более двух детей. Этих детей называют левым (Л) и правым (П) потомком или «сыном». На рисунке выше дерево является двоичным.

Основы объектно-ориентированного программирования в Python

В предыдущих разделах мы рассматривали в основном традиционное программирование на Python, когда вся программа разбивается (или не разбивается) на отдельные модули, содержащие функции.

Такое программирование соответствует парадигме структурного программирования. Само структурное программирование оказалось колоссальным шагом в построении программ. Однако еще большим шагом является парадигма объектно-ориентированного программирования. В этом подходе программа состоит из отдельных классов, которые объединяют в себе как переменные, называемые полями класса, так и функции, называемые методами класса.

На самом деле мы уже сталкивались с классами, когда создавали объекты для решения задач классификации и регрессии в Scikit-learn. В данном разделе подробнее познакомимся с основами объектно-ориентированного программирования (ООП).

Объектно-ориентированное программирование состоит из трех китов:

инкапсуляция;

наследование;

полиморфизм.

Рассмотрим на примерах эти понятия. Первое - инкапсуляция - это объединение в одном объекте данных и программного кода таким образом, что для внешней работы внутренняя часть объекта может быть скрыта от пользователя. Инкапсуляция может быть реализована не только с помощью классов, но и с помощью модулей, но классы позволяют сделать инкапсуляцию естественным путем. Создадим класс в Python. Для этого необходимо определить класс (новый тип данных) и создать объект, называемый экземпляром класса. Мы рекомендуем имена классов начинать с заглавной буквы "T", подчеркивая тем самым, что речь идет о типе данных. Делается это так:

```
class TAnimal:
  name = ""
  def __init__(self, name):
      self.name = name
  def say(self):
      print(self.name)
```

Теперь создадим экземпляр этого класса. Экземпляр класса представляет собой переменную, с которой можно работать обычным образом.

```
Animal = TAnimal("Обезьяна")
Animal.say()
```

Рассмотрим синтаксис Python при создании классов. Все начинается с ключевого слова class. Далее в блоке из отступов мы определяем переменные, которые будем называть

полями и функции, которые называются методами. Методы определяются, как обычные функции и могут возвращать значения. Единственное отличие состоит в том, что у всех методов есть обязательный первый параметр, который по традиции всегда называем self в котором передается ссылка на экземпляр класса. Поэтому когда внутри класса метод хочет обратиться к своему полю, то необходимо использовать конструкцию self.name. Заметим, что при вызове методов мы первый параметр не задаем.

Далее, у каждого класса есть метод, с именем **init**, который называется конструктором класса. Этот метод вызывается в момент создания экземпляра Animal = TAnimal("Обезьяна"). Конструктор может иметь любое количество параметров. Предположим, что теперь нам нужно сделать класс для описания конкретного животного - кошки. Для это мы используем наследование классов, когда можно определять новые классы, как наследники существующих. При этом новый класс будет иметь все поля и методы наследуемого класса. Вот как это делается:

```
class TAnimal:
    name = ""
    def __init__(self, name):
        self.name = name
    def say(self):
        print(self.name)


class TCat(TAnimal):
    def may(self):
        print("Мяу!")

Cat = TCat("Кошка")
Cat.say()
Cat.may()
```

Мы видим, что у наследованного класса сохранился конструктор и метод say. В последнем примере мы выдели, что наследный класс, также как и исходный имеет конструктор, который принимает в качестве параметра - название животного тогда, что в данном случае излишне. Для решения этой проблемы мы воспользуемся объектно-ориентированным механизмом - полиморфизмом. Полиморфизм - это возможность замены методов при наследовании. Сделаем так, чтобы не нужно было передавать в конструкторе название "Кошка".

```
 class TCat(TAnimal):
    def __init__(self):
        super().__init__("Кошка")
    def may(self):
        print("Мяу!")

Cat = TCat()
Cat.say()
Cat.may()
```

Результат выполнения этой программы будет аналогичный, но теперь при использовании этого класса нам не нужно передавать в конструкторе никаких параметров. Полиморфное перекрытие методов делается простым объявлением метода (в данном случае конструктора). При этом нельзя можно менять входные параметры. Если в результате написания кода метода возникает необходимость вызвать перекрытый метод, то для этого необходимо использовать функцию super(), которая по сути просто возвращает ссылку на родительский класс. Самое удивительное в полиморфизме, что изменяя метод, он меняется даже когда на него есть ссылки родительского класса. Рассмотрим еще один пример. Пусть у нас есть класс:

```
class TDo:
    def Operation(self, x, y):
        return x + y
    def Run(self):
        x = int(input("Enter x > "))
        y = int(input("Enter y > "))
        z = self.Operation(x, y)
        print("Result = " + z.__str__())

Do = TDo()
Do.Run()
```

С помощью полиморфизма заменим функцию Operation на другую в наследном классе:

```
class TDo2(TDo):
    def Operation(self, x, y):
        return x * y
```

1.1.1 Пример

Необходимо разработать виртуальную модель процесса обучения. В программе должны быть объекты-ученики, учитель, кладезь знаний.

Потребуется три класса – "учитель", "ученик", "данные". Учитель и ученик во многом похожи, оба – люди. Значит, их классы могут принадлежать одному надклассу "человек". Однако в контексте данной задачи у учителя и ученика вряд ли найдутся общие атрибуты. Определим, что должны уметь объекты для решения задачи "увеличить знания":

Ученик должен уметь брать информацию и превращать ее в свои знания. Учитель должен уметь учить группу учеников. Данные могут представлять собой список знаний. Элементы будут извлекаться по индексу.

```
class Data:
    def __init__(self, *info):
        self.info = list(info)
    def __getitem__(self, i):
        return self.info[i]

class Teacher:
```

```
    def teach(self, info, *student):
        for i in student:
            i.take(info)

class Student:
    def __init__(self):
        self.knowledge = []
    def take(self, info):
        self.knowledge.append(info)

lesson = Data('class', 'object', 'inheritance', 'polmorphism',
'encapsulation')
marIvanna = Teacher()
vasy = Student()
pety = Student()
marIvanna.teach(lesson[2], vasy, pety)
marIvanna.teach(lesson[0], pety)
print(vasy.knowledge)
print(pety.knowledge)
```

```
['inheritance']
['inheritance', 'class']
```

1.1.2 Пример

Напишите программу по следующему описанию. Есть класс "Воин". От него создаются два экземпляра-юнита. Каждому устанавливается здоровье в 100 очков. В случайном порядке они бьют друг друга. Тот, кто бьет, здоровья не теряет. У того, кого бьют, оно уменьшается на 20 очков от одного удара. После каждого удара надо выводить сообщение, какой юнит атаковал, и сколько у противника осталось здоровья. Как только у кого-то заканчивается ресурс здоровья, программа завершается сообщением о том, кто одержал победу.

```
import random
class Warrior:
    def __init__(self, health):
        self.health = health

    def hit(self, target, target1):
        if target.health > 0:
            target.health -= 20
        if target1 == warrior1:
            target1 = 'warrior1'
        if target1 == warrior2:
            target1 = 'warrior2'
        print(target1, 'has attacked')
        print(target.health, "health left")
        if target.health == 0:
            print(target1, 'has won')
```

```
warrior1 = Warrior(100)
warrior2 = Warrior(100)
q = int(input('Enter 1 to attack. Enter 2 to stop'))

while q != 2:
    if q == 1:
        j = random.randint(1, 3)
        if j % 2 == 0:
            warrior1.hit(warrior2, warrior1)
            q = int(input('Enter 1 to let someone attack'))
        else:
            warrior2.hit(warrior1, warrior2)
            q = int(input('Enter 1 to let someone attack'))
    else:
        print('Wrong input')
        break
```

1.1.3 Пример

Создайте класс по работе с дробями. В классе должна быть реализована следующая функциональность:

сложение дробей;

вычитание дробей;

умножение дробей;

деление дробей.

```
class Rational:

    @staticmethod
    def gcd(a,b):
        while (b != 0):
            (a,b) = (b, a%b)
        return a

    @staticmethod
    def sgn(x):
        if x > 0:
            return 1
        elif x < 0:
            return -1
        else:
            return 0

    def __init__(self, n, d):
        if n == 0:
            self.num = 0
```

```
            self.den = 1
        else:
            z = self.sgn(n)*self.sgn(d)
            n = abs(n)
            d = abs(d)
            k = self.gcd(n,d)
            self.num = z*n//k
            self.den = d//k

    def __str__(self):
        if self.num == 0:
            return '0'
        else:
            return str(self.num)+'/'+str(self.den)

    def __add__(self, o):
        n1 = self.num
        d1 = self.den
        if type(o) == int:
            n2 = o
            d2 = 1
        else:
            n2 = o.num
            d2 = o.den
        n = n1*d2+n2*d1
        d = d1*d2
        return Rational(n, d)

    def __radd__(self, o):
        n1 = self.num
        d1 = self.den
        if type(o) == int:
            n2 = o
            d2 = 1
        else:
            n2 = o.num
            d2 = o.den
        n = n1*d2+n2*d1
        d = d1*d2
        return Rational(n, d)

    def __sub__(self, o):
        n1 = self.num
        d1 = self.den
        n2 = o.num
        d2 = o.den
        n = n1*d2-n2*d1
        d = d1*d2
        return Rational(n, d)
```

```
    def __mul__(self, o):
        n1 = self.num
        d1 = self.den
        n2 = o.num
        d2 = o.den
        n = n1*n2
        d = d1*d2
        return Rational(n, d)

    def __floordiv__(self, o):
        n1 = self.num
        d1 = self.den
        n2 = o.num
        d2 = o.den
        n = n1*d2
        d = d1*n2
        return Rational(n, d)

d1 = Rational(1, 2)
d2 = Rational(1, 3)
d3 = d1+d2
print(d3)
d4 = d1-d2
print(d4)
d5 = d1*d2
print(d5)
d6 = d1*d2
print(d6)
d7 = d1//d2
print(d7)
d8 = 6+d1
print(d8)
```

```
5/6
1/6
1/6
1/6
3/2
13/2
```

Задание

Создайте класс по работе с тригонометрическими функциями. В классе должны быть реализованы функции вычисления:

косинуса;

синуса;

тангенса;

арксинуса;

арккосинуса;

арктангенса;

перевода из градусов в радианы.

```
import math as m

class Trigonometry:
    def __init__(self, angle):
        self.angle = angle

    def sin(self):
        return m.sin(self.angle)

    def cos(self):
        return m.cos(self.angle)

    def tan(self):
        return m.tan(self.angle)

    def arcsin(self):
        return m.asin(self.angle)

    def arccos(self):
        return m.acos(self.angle)

    def arctan(self):
        return m.atan(self.angle)

    def to_radians(self):
        return m.radians(self.angle)

    def to_degrees(self):
        return m.degrees(self.angle)

angle = 1
trig = Trigonometry(angle)

print(trig.sin())
print(trig.cos())
print(trig.tan())
print(trig.arcsin())
print(trig.arccos())
print(trig.arctan())
print(trig.to_radians())
print(trig.to_degrees())
```

```
0.8414709848078965
0.5403023058681398
1.5574077246549023
1.5707963267948966
0.0
0.7853981633974483
0.017453292519943295
57.29577951308232
```

Реализация деревьев в Python

Любое представление графов, естественно, можно использовать для представления деревьев, потому что деревья — это особый вид графов. Однако, деревья играют свою большую роль в алгоритмах, и для них разработано много соответствующих структур и методов. Большинство алгоритмов на деревьях (например, поиск по деревьям) можно рассматривать в терминах теории графов, но специальные структуры данных делают их проще в реализации.

Проще всего описать представление дерева с корнем, в котором ребра спускаются вниз от корня. Такие деревья часто отображают иерархическое ветвление данных, где корень отображает все объекты (которые, возможно, хранятся в листьях), а каждый внутренний узел показывает объекты, содержащиеся в дереве, корень которого — этот узел. Это описание можно использовать, представив каждое поддерево списком, содержащим все его поддеревья-потомки. Рассмотрим простое дерево, показанное на рисунке ниже.

tree2

Мы можем представить это дерево как список списков:

```
T = [["a", "b"], ["c"], ["d", ["e", "f"]]]
print(T[0][1])
print(T[2][1][0])
```

Каждый список в сущности является списком потомков каждого из внутренних узлов. Во втором примере мы обращаемся к третьему потомку корня, затем ко второму его потомку и в конце концов — к первому потомку предыдущего узла (этот путь отмечен на рисунке). В ряде случаев возможно заранее определить максимальное число потомков каждого узла. (Например, каждый узел бинарного дерева может иметь до двух потомков). Поэтому можно использовать другие представления, скажем, объекты с отдельным атрибутом для каждого из потомков как в листинге ниже.

1.2.1 Пример

Определите класс бинарного дерева и задайте его объекты с отдельным атрибутом для каждого из потомков.

```
class Tree:
    def __init__(self, left, right):
```

```
        self.left = left
        self.right = right

t = Tree(Tree('a', 'b'), Tree('c', 'd'))
t.left.left
```

```
'a'
```

1.2.2 Пример

Для обозначения отсутствующих потомков можно использовать None (в случае если у узла только один потомок). Само собой, можно комбинировать разные методы (например, использовать списки или множества потомков для каждого узла).

Распространенный способ реализации деревьев, особенно на языках, не имеющих встроенной поддержки списков, это так называемое представление «первый потомок, следующий брат». В нем каждый узел имеет два «указателя» или атрибута, указывающих на другие узлы, как в бинарном дереве. Однако, первый из этих атрибутов ссылается на первого потомка узла, а второй — на его следующего брата (т.е. узел, имеющий того же родителя, но находящийся правее, — прим. перев). Иными словами, каждый узел дерева имеет указатель на связанный список его потомков, а каждый из этих потомков ссылается на свой собственный аналогичный список. Таким образом, небольшая модификация бинарного дерева даст нам многопутевое дерево, показанное в листинге ниже.

```
class Tree:
    def __init__(self, kids, next=None):
        self.kids = self.val = kids
        self.next = next

t = Tree(Tree('a', Tree('b', Tree('c', Tree('d')))))
t.kids.next.next.val
```

```
'c'
```

Задание

Представьте дерево показанное на рисунке с использованием списка из списков. Выведите на печать корень дерева, а также его левое и правое поддеревья.

tree3

```
class Tree:
    def __init__(self, root, left=None, right=None):
        self.left = left
        self.right = right
        self.root = root

    def print_tree(self):
        print("Корень дерева:", self.root)
```

```
        print("Левое поддерево:", self.left)
        print("Правое поддерево:", self.right)

    def __str__(self):
        return f"{self.root}"

t = Tree('a', Tree('b', Tree('d'), Tree('e')), Tree('c', None,
Tree('f')))

t.print_tree()
```

```
Корень дерева: a
Левое поддерево: b
Правое поддерево: c
```

Задание

Дан класс, описывающий бинарное дерево.

```
class Tree:
   def __init__(self, data):
      self.left = None
      self.right = None
      self.data = data
   def PrintTree(self):
      print(self.data)
```

Реализуйте в классе функцию для вставки нового элемента в дерево по следующим правилам:

Левое поддерево узла содержит только узлы со значениями меньше, чем значение в узле.

Правое поддерево узла содержит только узлы со значениями меньше, чем значение в узле.

Каждое из левого и правого поддеревьев также должно быть бинарным деревом поиска.

Не должно быть повторяющихся узлов.

Метод вставки сравнивает значение узла с родительским узлом и решает куда добавить элемент (в левое или правое поддерево). Перепишите, метод PrintTree для печати полной версии дерева.

```
class Tree:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data

    def PrintTree(self):
        if self.left:
```

```
            self.left.PrintTree()
        print(self.data, end=" ")
        if self.right:
            self.right.PrintTree()

    def add(self, value):
        if value < self.data:
            if self.left is None:
                self.left = Tree(value)
            else:
                self.left.add(value)
        else:
            if self.right is None:
                self.right = Tree(value)
            else:
                self.right.add(value)


root = Tree(10)
root.add(5)
root.add(15)
root.add(3)
root.add(7)
root.add(13)
root.add(18)

print("Дерево в порядке in-order:")
root.PrintTree()
```

```
Дерево в порядке in-order:
3 5 7 10 13 15 18
```

Деревья решений

Дерево решений – это один из наиболее часто и широко используемых алгоритмов контролируемого машинного обучения, который может выполнять как регрессионные, так и классификационные задачи.

Использование деревьев решений для прогнозного анализа имеет ряд преимуществ:

Деревья решений могут быть использованы для прогнозирования как непрерывных, так и дискретных значений, т. е. они хорошо работают как для задач регрессии, так и для задач классификации.

Они требуют относительно меньших усилий для обучения алгоритма.

Они могут быть использованы для классификации нелинейно разделимых данных.

Они очень быстры и эффективны по сравнению с KNN и другими алгоритмами классификации.

Решим модельные примеры классификации и регрессии, разобранные в предыдущих рабочих тетрадях, но с использованием деревьев принятия решений.

1.3.1 Пример

Построим дерево решений для задачи классификации, для этого, построим границу решения для каждого класса. В качестве данных будем использовать уже знакомый нам и встроенный в библиотеку sklearn набор данных ирисов Фишера. Импортируем библиотеки, набор данных и посмотрим его характеристики.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
dataset = sns.load_dataset('iris')
dataset
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

```
dataset.shape
```

```
(150, 5)
```

```
dataset.head()
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

```
dataset.iloc[:, :-1]
```

```
     sepal_length  sepal_width  petal_length  petal_width
0             5.1          3.5           1.4          0.2
1             4.9          3.0           1.4          0.2
2             4.7          3.2           1.3          0.2
```

```
3             4.6          3.1           1.5          0.2
4             5.0          3.6           1.4          0.2
..            ...          ...           ...          ...
145           6.7          3.0           5.2          2.3
146           6.3          2.5           5.0          1.9
147           6.5          3.0           5.2          2.0
148           6.2          3.4           5.4          2.3
149           5.9          3.0           5.1          1.8

[150 rows x 4 columns]
```

```
dataset.iloc[:, -1]
```

```
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
        ...
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: species, Length: 150, dtype: object
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(dataset.iloc[:, :-
1], dataset.iloc[:, -1], test_size=0.2)

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((120, 4), (30, 4), (120,), (30,))
```

```
X_train.head()
```

```
     sepal_length  sepal_width  petal_length  petal_width
137           6.4          3.1           5.5          1.8
75            6.6          3.0           4.4          1.4
59            5.2          2.7           3.9          1.4
136           6.3          3.4           5.6          2.4
81            5.5          2.4           3.7          1.0
```

```
y_train.head()
```

```
137     virginica
75     versicolor
59     versicolor
136     virginica
```

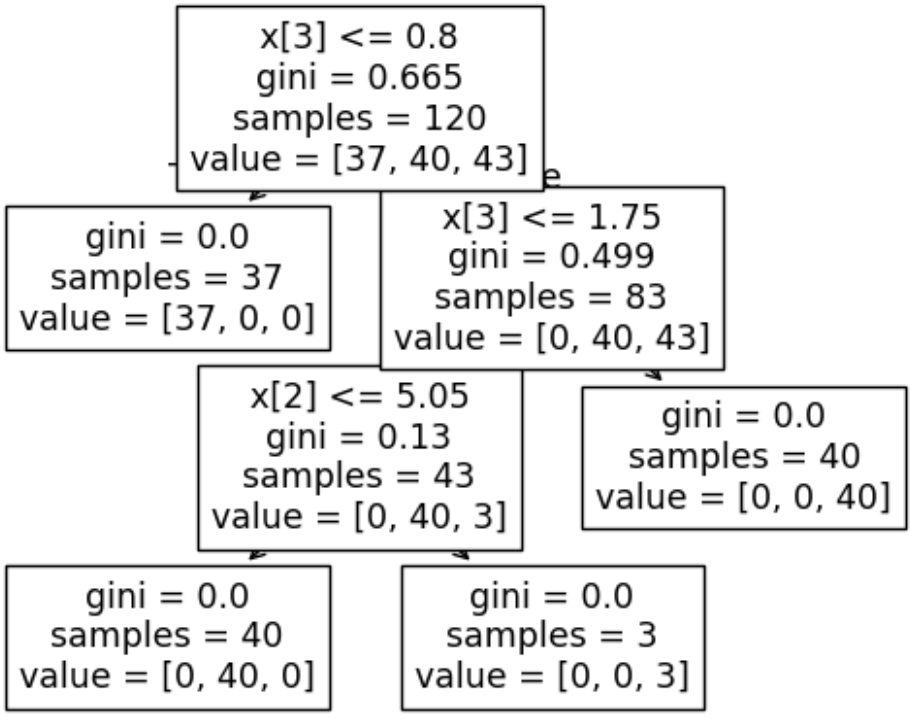
```
81     versicolor
Name: species, dtype: object
```

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier()
```

```
from sklearn import tree
tree.plot_tree(classifier)
```

```
[Text(0.4, 0.875, 'x[3] <= 0.8\ngini = 0.665\nsamples = 120\nvalue =
[37, 40, 43]'),
 Text(0.2, 0.625, 'gini = 0.0\nsamples = 37\nvalue = [37, 0, 0]'),
 Text(0.30000000000000004, 0.75, 'True  '),
 Text(0.6, 0.625, 'x[3] <= 1.75\ngini = 0.499\nsamples = 83\nvalue =
[0, 40, 43]'),
 Text(0.5, 0.75, '  False'),
 Text(0.4, 0.375, 'x[2] <= 5.05\ngini = 0.13\nsamples = 43\nvalue =
[0, 40, 3]'),
 Text(0.2, 0.125, 'gini = 0.0\nsamples = 40\nvalue = [0, 40, 0]'),
 Text(0.6, 0.125, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
 Text(0.8, 0.375, 'gini = 0.0\nsamples = 40\nvalue = [0, 0, 40]')]
```




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

```
array(['setosa', 'versicolor', 'versicolor', 'virginica', 'virginica',
       'versicolor', 'virginica', 'virginica', 'setosa', 'setosa',
       'setosa', 'versicolor', 'setosa', 'versicolor', 'setosa',
       'virginica', 'virginica', 'setosa', 'versicolor', 'versicolor',
       'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'setosa', 'virginica', 'setosa', 'setosa'],
dtype=object)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[13  0  0]
 [ 0  8  2]
 [ 0  2  5]]
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        13
  versicolor       0.80      0.80      0.80        10
   virginica       0.71      0.71      0.71         7

    accuracy                           0.87        30
   macro avg       0.84      0.84      0.84        30
weighted avg       0.87      0.87      0.87        30
```

Задание

Постройте классификатор на основе дерева принятия решений следующего датасета:

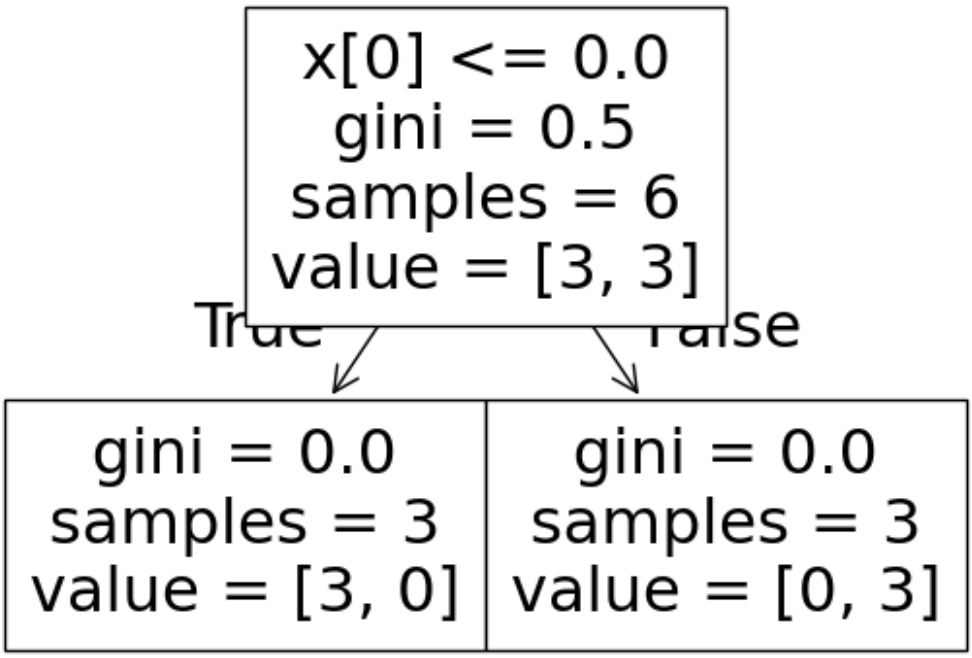
```
X = np.array([[-1,-1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
target = [0, 0, 0, 1, 1, 1]
```

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X, target)
```

```
DecisionTreeClassifier()
```

```
from sklearn import tree
tree.plot_tree(classifier)
```

```
[Text(0.5, 0.75, 'x[0] <= 0.0\ngini = 0.5\nsamples = 6\nvalue = [3,
3]'),
 Text(0.25, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.375, 0.5, 'True  '),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.625, 0.5, '  False')]
```




```
y_pred = classifier.predict(X)
y_pred
```

```
array([0, 0, 0, 1, 1, 1])
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_pred, target))
print(classification_report(y_pred, target))
```

```
[[3 0]
 [0 3]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00         3

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6
```

Дерево решений для регрессии

Дерево решений для регрессии

Процесс решения регрессионной задачи с деревом решений с помощью Scikit Learn очень похож на процесс классификации. Однако для регрессии мы используем класс DecisionTreeRegressor древовидной библиотеки.

Кроме того, оценочные показатели регрессии отличаются от показателей классификации. В остальном процесс почти такой же.

Построим регрессию с использованием дерева решений в Python и библиотеки scikit-learn. В качестве исходного набора данных будем использовать зависимость заработной платы от опыта работы из предыдущей тетради:

https://raw.githubusercontent.com/AnnaShestova/salary-years-simple-linear-regression/master/Salary_Data.csv

1.4.1 Пример

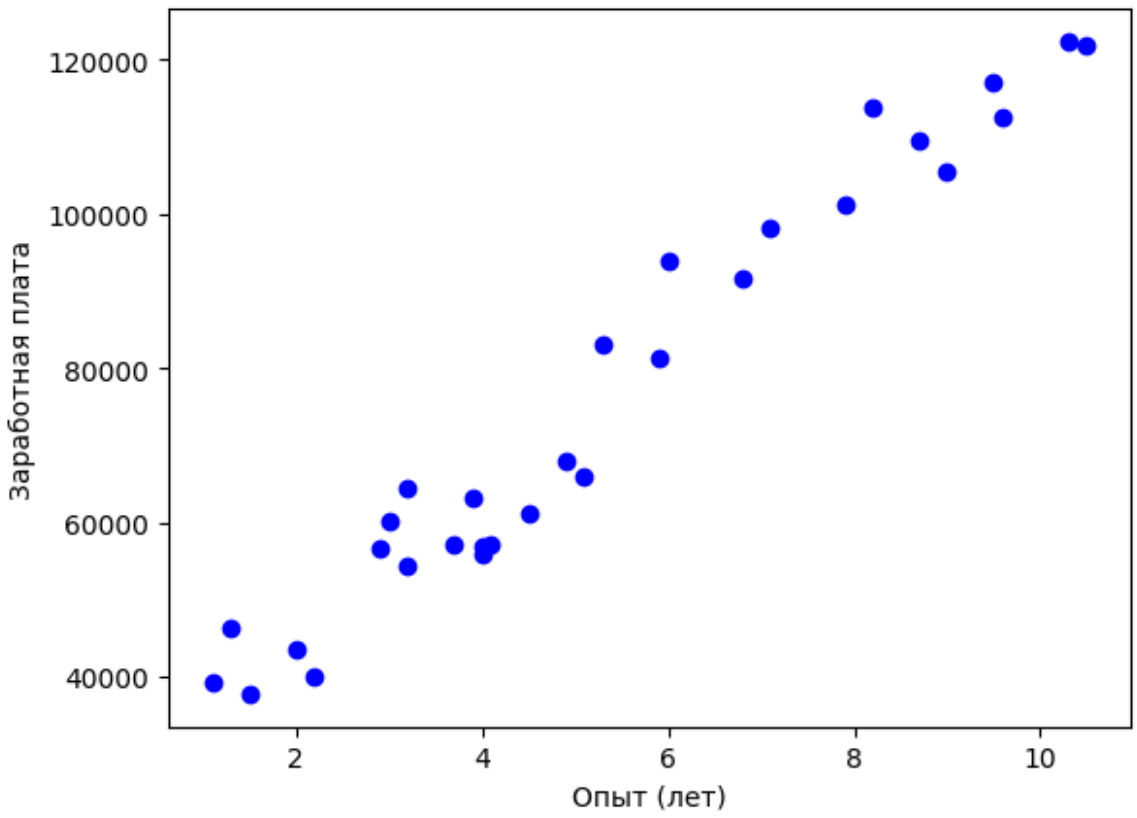
Постойте регрессию с использованием дерева решений, реализованного в Python.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

url = r'https://raw.githubusercontent.com/AnnaShestova/salary-years-simple-linear-regression/master/Salary_Data.csv'
dataset = pd.read_csv(url)

print(dataset.head())
print(dataset.shape)
dataset.describe()
```

```
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
(30, 2)
```

```
       YearsExperience         Salary
count        30.000000      30.000000
mean          5.313333   76003.000000
std           2.837888   27414.429785
min           1.100000   37731.000000
25%           3.200000   56720.750000
50%           4.700000   65237.000000
75%           7.700000  100544.750000
max          10.500000  122391.000000
```

```
plt.scatter(dataset['YearsExperience'], dataset['Salary'], color = 'b', label = 'Заработная плата')
plt.xlabel('Опыт (лет)')
plt.ylabel('Заработная плата')
plt.show()
```




```
from sklearn.tree import DecisionTreeRegressor
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
print(X)
print(y)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)

regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)
```

```
[[ 1.1]
 [ 1.3]
 [ 1.5]
 [ 2. ]
 [ 2.2]
 [ 2.9]
 [ 3. ]
 [ 3.2]
 [ 3.2]
 [ 3.7]
 [ 3.9]
 [ 4. ]
```

```
 [ 4. ]
 [ 4.1]
 [ 4.5]
 [ 4.9]
 [ 5.1]
 [ 5.3]
 [ 5.9]
 [ 6. ]
 [ 6.8]
 [ 7.1]
 [ 7.9]
 [ 8.2]
 [ 8.7]
 [ 9. ]
 [ 9.5]
 [ 9.6]
 [10.3]
 [10.5]]
[ 39343.  46205.  37731.  43525.  39891.  56642.  60150.  54445.
64445.
  57189.  63218.  55794.  56957.  57081.  61111.  67938.  66029.
83088.
  81363.  93940.  91738.  98273. 101302. 113812. 109431. 105582.
116969.
 112635. 122391. 121872.]
```

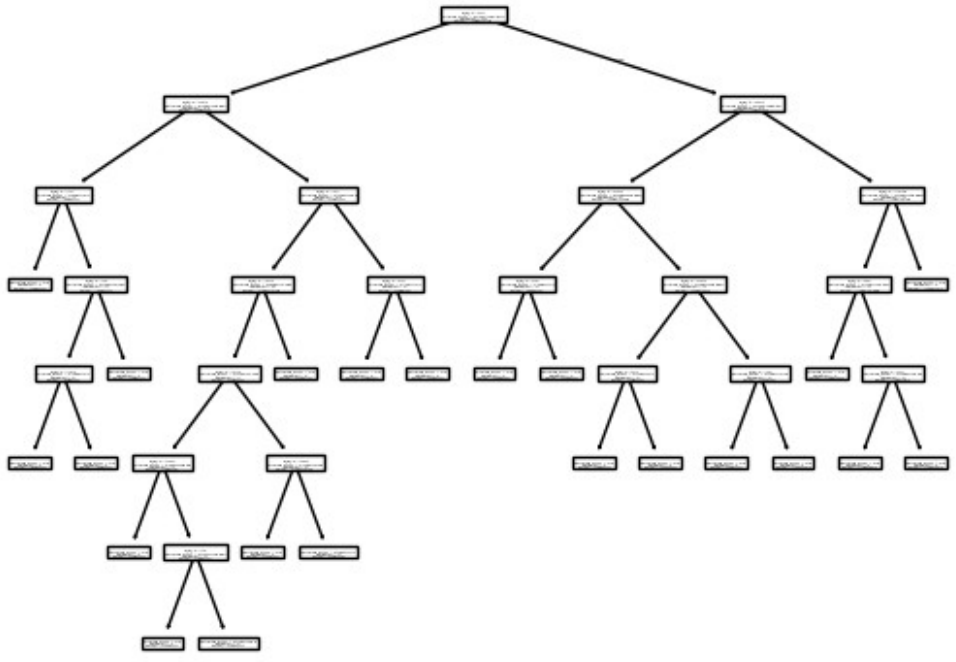
```
DecisionTreeRegressor()
```

```
from sklearn import tree
tree.plot_tree(regressor)
```

```
[Text(0.4956896551724138, 0.9375, 'x[0] <= 5.2\nsquared_error =
614737637.832\nsamples = 24\nvalue = 73886.208'),
 Text(0.20689655172413793, 0.8125, 'x[0] <= 2.55\nsquared_error =
81200345.857\nsamples = 14\nvalue = 54976.0'),
 Text(0.35129310344827586, 0.875, 'True  '),
 Text(0.06896551724137931, 0.6875, 'x[0] <= 1.2\nsquared_error =
7820714.0\nsamples = 4\nvalue = 42241.0'),
 Text(0.034482758620689655, 0.5625, 'squared_error = 0.0\nsamples = 1\
nvalue = 39343.0'),
 Text(0.10344827586206896, 0.5625, 'x[0] <= 2.1\nsquared_error =
6694994.667\nsamples = 3\nvalue = 43207.0'),
 Text(0.06896551724137931, 0.4375, 'x[0] <= 1.65\nsquared_error =
1795600.0\nsamples = 2\nvalue = 44865.0'),
 Text(0.034482758620689655, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 46205.0'),
 Text(0.10344827586206896, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 43525.0'),
 Text(0.13793103448275862, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 39891.0'),
```

```
 Text(0.3448275862068966, 0.6875, 'x[0] <= 4.7\nsquared_error =
19731272.6\nsamples = 10\nvalue = 60070.0'),
 Text(0.27586206896551724, 0.5625, 'x[0] <= 4.25\nsquared_error =
9499922.484\nsamples = 8\nvalue = 58341.625'),
 Text(0.2413793103448276, 0.4375, 'x[0] <= 3.45\nsquared_error =
9604901.143\nsamples = 7\nvalue = 57946.0'),
 Text(0.1724137931034483, 0.3125, 'x[0] <= 2.95\nsquared_error =
14313358.25\nsamples = 4\nvalue = 58920.5'),
 Text(0.13793103448275862, 0.1875, 'squared_error = 0.0\nsamples = 1\
nvalue = 56642.0'),
 Text(0.20689655172413793, 0.1875, 'x[0] <= 3.1\nsquared_error =
16777116.667\nsamples = 3\nvalue = 59680.0'),
 Text(0.1724137931034483, 0.0625, 'squared_error = 0.0\nsamples = 1\
nvalue = 60150.0'),
 Text(0.2413793103448276, 0.0625, 'squared_error = 25000000.0\nsamples
= 2\nvalue = 59445.0'),
 Text(0.3103448275862069, 0.3125, 'x[0] <= 3.85\nsquared_error =
372490.889\nsamples = 3\nvalue = 56646.667'),
 Text(0.27586206896551724, 0.1875, 'squared_error = 0.0\nsamples = 1\
nvalue = 57189.0'),
 Text(0.3448275862068966, 0.1875, 'squared_error = 338142.25\nsamples
= 2\nvalue = 56375.5'),
 Text(0.3103448275862069, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 61111.0'),
 Text(0.41379310344827586, 0.5625, 'x[0] <= 5.0\nsquared_error =
911070.25\nsamples = 2\nvalue = 66983.5'),
 Text(0.3793103448275862, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 67938.0'),
 Text(0.4482758620689655, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 66029.0'),
 Text(0.7844827586206896, 0.8125, 'x[0] <= 8.05\nsquared_error =
160167356.45\nsamples = 10\nvalue = 100360.5'),
 Text(0.6400862068965517, 0.875, '  False'),
 Text(0.6379310344827587, 0.6875, 'x[0] <= 5.95\nsquared_error =
53566814.556\nsamples = 6\nvalue = 91617.333'),
 Text(0.5517241379310345, 0.5625, 'x[0] <= 5.6\nsquared_error =
743906.25\nsamples = 2\nvalue = 82225.5'),
 Text(0.5172413793103449, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 83088.0'),
 Text(0.5862068965517241, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 81363.0'),
 Text(0.7241379310344828, 0.5625, 'x[0] <= 6.95\nsquared_error =
13823368.688\nsamples = 4\nvalue = 96313.25'),
 Text(0.6551724137931034, 0.4375, 'x[0] <= 6.4\nsquared_error =
1212201.0\nsamples = 2\nvalue = 92839.0'),
 Text(0.6206896551724138, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 93940.0'),
 Text(0.6896551724137931, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 91738.0'),
```

```
 Text(0.7931034482758621, 0.4375, 'x[0] <= 7.5\nsquared_error =
2293710.25\nsamples = 2\nvalue = 99787.5'),
 Text(0.7586206896551724, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 98273.0'),
 Text(0.8275862068965517, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 101302.0'),
 Text(0.9310344827586207, 0.6875, 'x[0] <= 10.05\nsquared_error =
33407056.688\nsamples = 4\nvalue = 113475.25'),
 Text(0.896551724137931, 0.5625, 'x[0] <= 8.6\nsquared_error =
13207004.222\nsamples = 3\nvalue = 110676.333'),
 Text(0.8620689655172413, 0.4375, 'squared_error = 0.0\nsamples = 1\
nvalue = 113812.0'),
 Text(0.9310344827586207, 0.4375, 'x[0] <= 9.3\nsquared_error =
12436202.25\nsamples = 2\nvalue = 109108.5'),
 Text(0.896551724137931, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 105582.0'),
 Text(0.9655172413793104, 0.3125, 'squared_error = 0.0\nsamples = 1\
nvalue = 112635.0'),
 Text(0.9655172413793104, 0.5625, 'squared_error = 0.0\nsamples = 1\
nvalue = 121872.0')]
```



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

array([ 46205. , 121872. ,  56375.5,  56375.5, 112635. , 105582. ])
```

```
df = pd.DataFrame({'Actual:':y_test, 'Predicted:':y_pred})
df
```

```
    Actual:  Predicted:
0   37731.0     46205.0
1  122391.0    121872.0
2   57081.0     56375.5
3   63218.0     56375.5
4  116969.0    112635.0
5  109431.0    105582.0
```

```
from sklearn import metrics
print('Root of Mean Squared Error: ',
metrics.root_mean_squared_error(y_test, y_pred))
print('Mean Absolute Error: ', metrics.mean_absolute_error(y_test,
y_pred))
```

```
Root of Mean Squared Error:  5049.652306512466
Mean Absolute Error:  4120.666666666667
```

```
metrics.mean_absolute_error(y_test, y_pred) / np.average(y) * 100
```

```
np.float64(5.421715809463662)
```

Задание

Задание. Постройте модель регрессии для данных из предыдущей рабочей тетради.

Для примера можно взять потребления газа (в миллионах галлонов) в 48 штатах США или набор данных о качестве красного вина:

https://raw.githubusercontent.com/likarajo/petrol_consumption/master/data/petrol_consumption.csv

https://raw.githubusercontent.com/aniruddhachoudhury/Red-Wine-Quality/master/winequality-red.csv

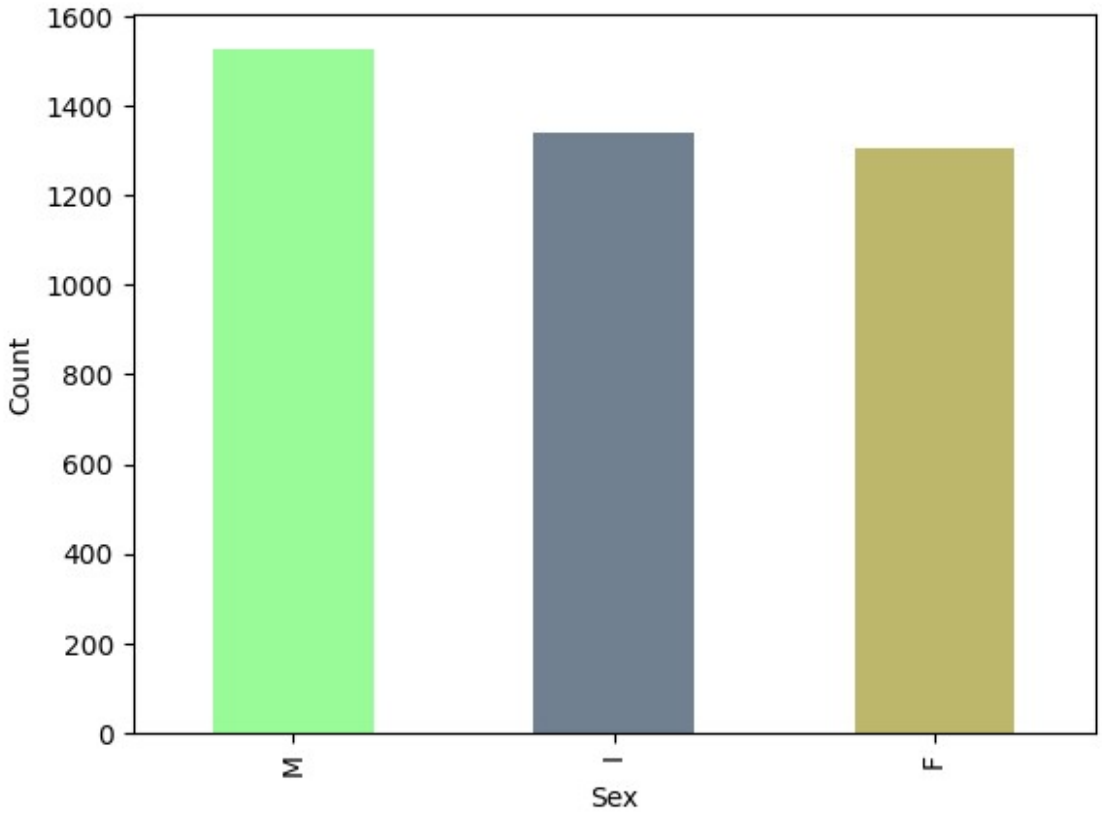
Постройте прогноз. Оцените точность модели.

```
df = pd.read_csv('abalone.csv')
print(df.head())
print(df.describe())
print(df.shape)
```

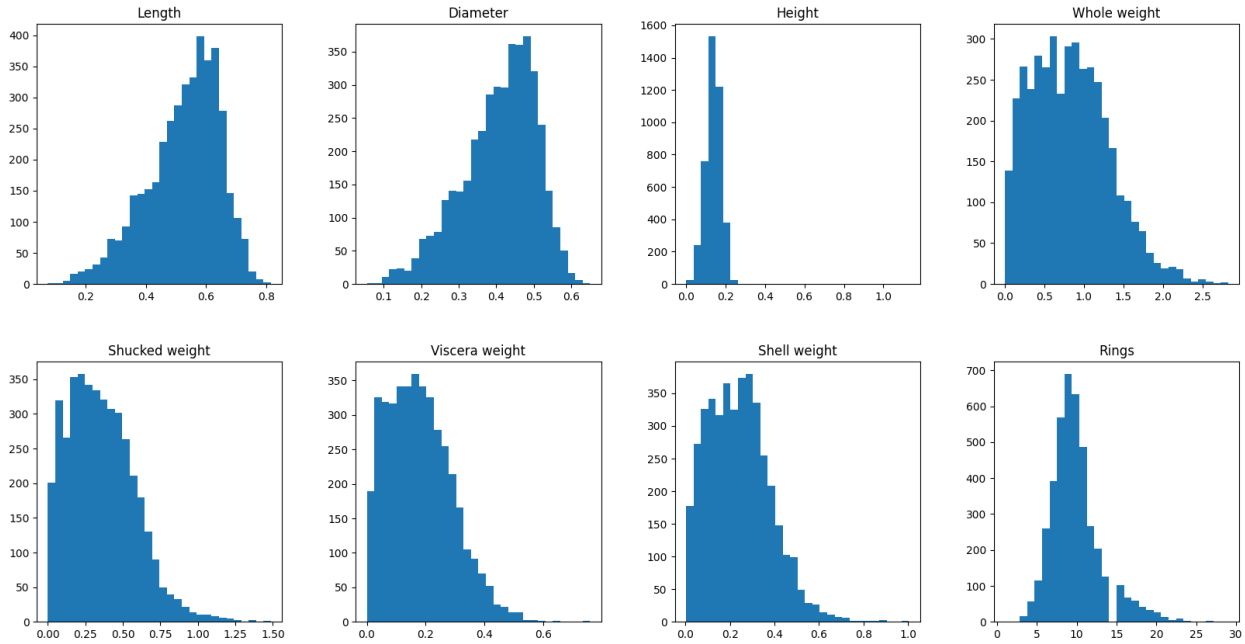
```
  Sex  Length  Diameter  Height  Whole weight  Shucked weight  Viscera
weight  \
0   M   0.455     0.365   0.095        0.5140          0.2245
0.1010
1   M   0.350     0.265   0.090        0.2255          0.0995
0.0485
2   F   0.530     0.420   0.135        0.6770          0.2565
```

```
0.1415
3   M   0.440     0.365   0.125         0.5160          0.2155
0.1140
4   I   0.330     0.255   0.080         0.2050          0.0895
0.0395

   Shell weight  Rings
0         0.150     15
1         0.070      7
2         0.210      9
3         0.155     10
4         0.055      7
            Length     Diameter       Height  Whole weight  Shucked
weight  \
count  4177.000000  4177.000000  4177.000000   4177.000000
4177.000000
mean      0.523992     0.407881     0.139516      0.828742
0.359367
std       0.120093     0.099240     0.041827      0.490389
0.221963
min       0.075000     0.055000     0.000000      0.002000
0.001000
25%       0.450000     0.350000     0.115000      0.441500
0.186000
50%       0.545000     0.425000     0.140000      0.799500
0.336000
75%       0.615000     0.480000     0.165000      1.153000
0.502000
max       0.815000     0.650000     1.130000      2.825500
1.488000

       Viscera weight  Shell weight        Rings
count     4177.000000   4177.000000  4177.000000
mean         0.180594      0.238831     9.933684
std          0.109614      0.139203     3.224169
min          0.000500      0.001500     1.000000
25%          0.093500      0.130000     8.000000
50%          0.171000      0.234000     9.000000
75%          0.253000      0.329000    11.000000
max          0.760000      1.005000    29.000000
(4177, 9)
```

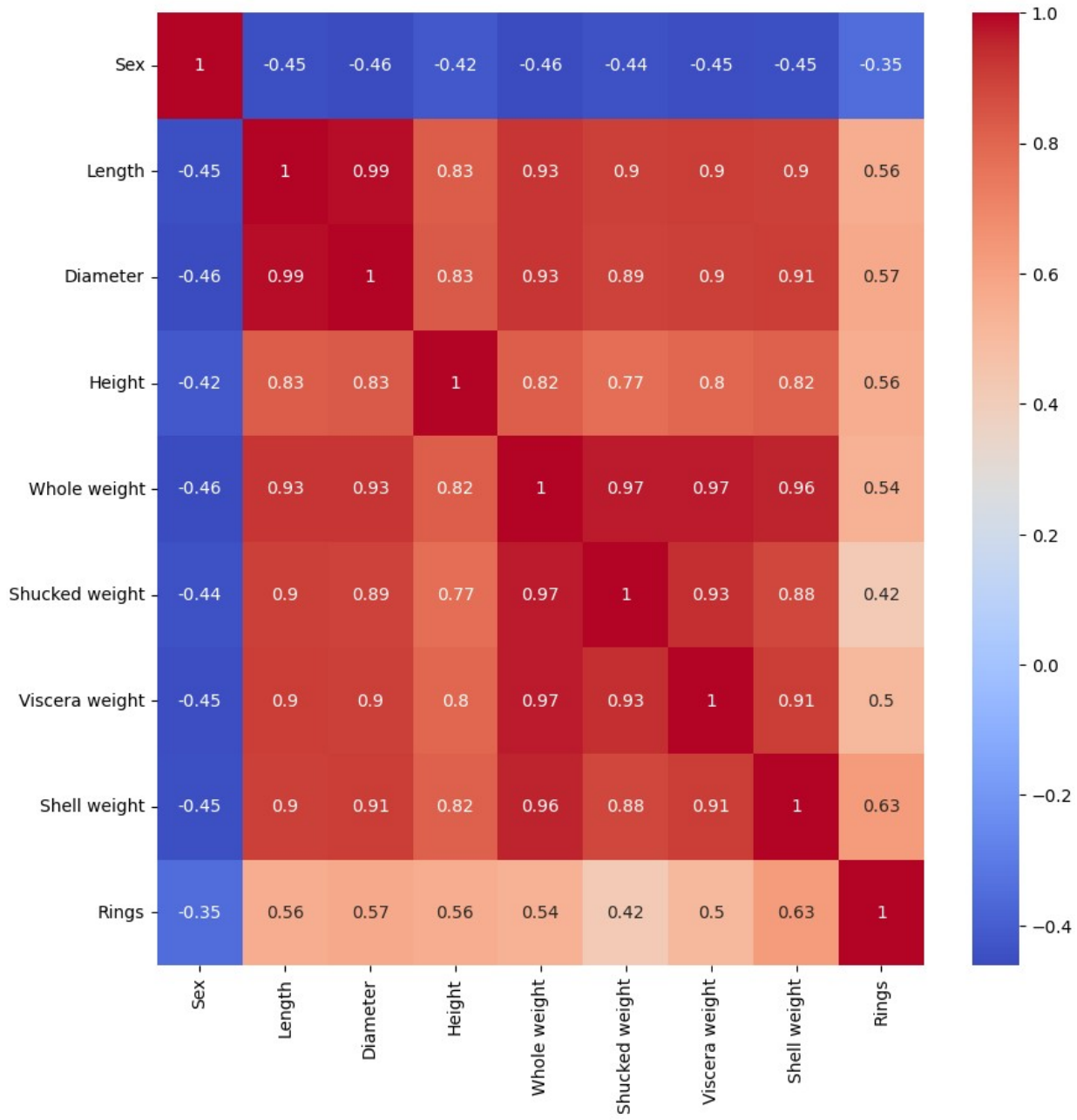
```
import seaborn as sns
ax = df.value_counts('Sex').plot(kind='bar', color=["palegreen",
'slategray', 'darkkhaki'])
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```




```
df.hist(figsize=(20,10), grid=False, layout=(2, 4), bins = 30)

array([[<Axes: title={'center': 'Length'}>,
        <Axes: title={'center': 'Diameter'}>,
        <Axes: title={'center': 'Height'}>,
        <Axes: title={'center': 'Whole weight'}>],
       [<Axes: title={'center': 'Shucked weight'}>,
        <Axes: title={'center': 'Viscera weight'}>,
        <Axes: title={'center': 'Shell weight'}>,
        <Axes: title={'center': 'Rings'}>]], dtype=object)
```




```
# Корреляция между признаками
df['Sex'] = df['Sex'].map({'M': 0, 'F': 1, 'I': 2})
plt.figure(figsize=(10, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()
```




```
from sklearn.tree import DecisionTreeRegressor
X = df[['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked
weight', 'Viscera weight', 'Shell weight']]
y = df['Rings']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.1, random_state=0)
regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(df)
```

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test,
y_pred))
print('Root Mean Squared Error:',
metrics.root_mean_squared_error(y_test, y_pred))
```

```
      Actual  Predicted
668       13       10.0
1580       8        7.0
3784      11       11.0
463        5        6.0
2615      12       14.0
...      ...        ...
3326      12       23.0
3877      15       19.0
2769      11       17.0
3359      20       16.0
104       12       11.0

[418 rows x 2 columns]
Mean Absolute Error: 2.0598086124401913
Root Mean Squared Error: 2.8816362975219776
```

```
from sklearn import tree
tree.plot_tree(regressor)
```

```
[Text(0.4162257622150095, 0.984375, 'x[6] <= 0.168\nsquared_error =
10.343\nsamples = 3759\nvalue = 9.937'),
 Text(0.1166373637241437, 0.953125, 'x[6] <= 0.059\nsquared_error =
4.655\nsamples = 1275\nvalue = 7.575'),
 Text(0.2664315629695766, 0.96875, 'True  '),
 Text(0.02362039146437616, 0.921875, 'x[6] <= 0.027\nsquared_error =
2.353\nsamples = 321\nvalue = 5.701'),
 Text(0.00869113409853101, 0.890625, 'x[5] <= 0.01\nsquared_error =
1.096\nsamples = 103\nvalue = 4.408'),
 Text(0.0036043499271499837, 0.859375, 'x[4] <= 0.005\nsquared_error =
0.8\nsamples = 50\nvalue = 3.86'),
 Text(0.00139523222986451, 0.828125, 'x[2] <= 0.027\nsquared_error =
0.688\nsamples = 4\nvalue = 2.25'),
 Text(0.0009301548199096732, 0.796875, 'x[1] <= 0.078\nsquared_error =
0.25\nsamples = 2\nvalue = 1.5'),
 Text(0.0004650774099548366, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 1.0'),
 Text(0.00139523222986451, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 2.0'),
 Text(0.0018603096398193465, 0.796875, 'squared_error = 0.0\nsamples =
2\nvalue = 3.0'),
 Text(0.0058134676244354574, 0.828125, 'x[6] <= 0.015\nsquared_error =
0.565\nsamples = 46\nvalue = 4.0'),
 Text(0.00418569668959353, 0.796875, 'x[3] <= 0.024\nsquared_error =
0.39\nsamples = 43\nvalue = 3.93'),
```

```
 Text(0.002325387049774183, 0.765625, 'x[6] <= 0.008\nsquared_error =
0.391\nsamples = 13\nvalue = 3.615'),
 Text(0.0018603096398193465, 0.734375, 'x[4] <= 0.007\nsquared_error =
0.38\nsamples = 11\nvalue = 3.727'),
 Text(0.0009301548199096732, 0.703125, 'x[4] <= 0.005\nsquared_error =
0.247\nsamples = 9\nvalue = 3.556'),
 Text(0.0004650774099548366, 0.671875, 'squared_error = 0.0\nsamples =
3\nvalue = 4.0'),
 Text(0.00139523222986451, 0.671875, 'x[2] <= 0.022\nsquared_error =
0.222\nsamples = 6\nvalue = 3.333'),
 Text(0.0009301548199096732, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.0018603096398193465, 0.640625, 'squared_error = 0.0\nsamples =
4\nvalue = 3.0'),
 Text(0.00279046445972902, 0.703125, 'x[5] <= 0.005\nsquared_error =
0.25\nsamples = 2\nvalue = 4.5'),
 Text(0.002325387049774183, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.0032555418696838564, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 4.0'),
 Text(0.00279046445972902, 0.734375, 'squared_error = 0.0\nsamples =
2\nvalue = 3.0'),
 Text(0.006046006329412876, 0.765625, 'x[6] <= 0.01\nsquared_error =
0.329\nsamples = 30\nvalue = 4.067'),
 Text(0.005115851509503203, 0.734375, 'x[5] <= 0.009\nsquared_error =
0.299\nsamples = 19\nvalue = 4.263'),
 Text(0.004650774099548366, 0.703125, 'x[6] <= 0.009\nsquared_error =
0.222\nsamples = 18\nvalue = 4.333'),
 Text(0.00418569668959353, 0.671875, 'squared_error = 0.0\nsamples =
7\nvalue = 4.0'),
 Text(0.005115851509503203, 0.671875, 'x[3] <= 0.033\nsquared_error =
0.248\nsamples = 11\nvalue = 4.545'),
 Text(0.004650774099548366, 0.640625, 'x[3] <= 0.028\nsquared_error =
0.234\nsamples = 8\nvalue = 4.375'),
 Text(0.00418569668959353, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.005115851509503203, 0.609375, 'x[2] <= 0.043\nsquared_error =
0.139\nsamples = 6\nvalue = 4.167'),
 Text(0.004650774099548366, 0.578125, 'x[6] <= 0.01\nsquared_error =
0.222\nsamples = 3\nvalue = 4.333'),
 Text(0.00418569668959353, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.005115851509503203, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.00558092891945804, 0.578125, 'squared_error = 0.0\nsamples =
3\nvalue = 4.0'),
 Text(0.00558092891945804, 0.640625, 'squared_error = 0.0\nsamples =
3\nvalue = 5.0'),
 Text(0.00558092891945804, 0.703125, 'squared_error = 0.0\nsamples =
```

```
1\nvalue = 3.0'),
 Text(0.006976161149322549, 0.734375, 'x[5] <= 0.006\nsquared_error =
0.198\nsamples = 11\nvalue = 3.727'),
 Text(0.006511083739367713, 0.703125, 'squared_error = 0.0\nsamples =
2\nvalue = 3.0'),
 Text(0.007441238559277386, 0.703125, 'x[0] <= 0.212\nsquared_error =
0.099\nsamples = 9\nvalue = 3.889'),
 Text(0.006976161149322549, 0.671875, 'squared_error = 0.0\nsamples =
6\nvalue = 4.0'),
 Text(0.007906315969232223, 0.671875, 'x[1] <= 0.155\nsquared_error =
0.222\nsamples = 3\nvalue = 3.667'),
 Text(0.007441238559277386, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 3.0'),
 Text(0.00837139337918706, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.007441238559277386, 0.796875, 'x[2] <= 0.053\nsquared_error =
2.0\nsamples = 3\nvalue = 5.0'),
 Text(0.006976161149322549, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.007906315969232223, 0.765625, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.013777918269912036, 0.859375, 'x[6] <= 0.018\nsquared_error =
0.824\nsamples = 53\nvalue = 4.925'),
 Text(0.011626935248870915, 0.828125, 'x[1] <= 0.145\nsquared_error =
0.72\nsamples = 17\nvalue = 4.529'),
 Text(0.01116185783891608, 0.796875, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.012092012658825752, 0.796875, 'x[3] <= 0.063\nsquared_error =
0.489\nsamples = 15\nvalue = 4.333'),
 Text(0.011626935248870915, 0.765625, 'x[5] <= 0.016\nsquared_error =
0.388\nsamples = 14\nvalue = 4.429'),
 Text(0.01116185783891608, 0.734375, 'x[3] <= 0.048\nsquared_error =
0.38\nsamples = 11\nvalue = 4.273'),
 Text(0.010231703019006406, 0.703125, 'x[1] <= 0.165\nsquared_error =
0.25\nsamples = 6\nvalue = 4.5'),
 Text(0.009766625609051569, 0.671875, 'x[2] <= 0.053\nsquared_error =
0.24\nsamples = 5\nvalue = 4.4'),
 Text(0.009301548199096732, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.010231703019006406, 0.640625, 'x[5] <= 0.012\nsquared_error =
0.222\nsamples = 3\nvalue = 4.667'),
 Text(0.009766625609051569, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 4.0'),
 Text(0.010696780428961243, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.010696780428961243, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.012092012658825752, 0.703125, 'x[2] <= 0.048\nsquared_error =
0.4\nsamples = 5\nvalue = 4.0'),
```

```
 Text(0.011626935248870915, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 3.0'),
 Text(0.012557090068780589, 0.671875, 'x[0] <= 0.227\nsquared_error =
0.188\nsamples = 4\nvalue = 4.25'),
 Text(0.012092012658825752, 0.640625, 'squared_error = 0.0\nsamples =
3\nvalue = 4.0'),
 Text(0.013022167478735426, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.012092012658825752, 0.734375, 'squared_error = 0.0\nsamples =
3\nvalue = 5.0'),
 Text(0.012557090068780589, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 3.0'),
 Text(0.015928901290953153, 0.828125, 'x[3] <= 0.063\nsquared_error =
0.765\nsamples = 36\nvalue = 5.111'),
 Text(0.013952322298645098, 0.796875, 'x[4] <= 0.019\nsquared_error =
0.4\nsamples = 5\nvalue = 6.0'),
 Text(0.013487244888690263, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.014417399708599935, 0.765625, 'x[4] <= 0.025\nsquared_error =
0.188\nsamples = 4\nvalue = 5.75'),
 Text(0.013952322298645098, 0.734375, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.014882477118554772, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.01790548028326121, 0.796875, 'x[6] <= 0.025\nsquared_error =
0.676\nsamples = 31\nvalue = 4.968'),
 Text(0.017440402873306372, 0.765625, 'x[6] <= 0.024\nsquared_error =
0.638\nsamples = 28\nvalue = 5.071'),
 Text(0.015812631938464446, 0.734375, 'x[1] <= 0.178\nsquared_error =
0.325\nsamples = 17\nvalue = 4.706'),
 Text(0.014882477118554772, 0.703125, 'x[6] <= 0.019\nsquared_error =
0.139\nsamples = 6\nvalue = 5.167'),
 Text(0.014417399708599935, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.015347554528509609, 0.671875, 'squared_error = 0.0\nsamples =
5\nvalue = 5.0'),
 Text(0.01674278675837412, 0.703125, 'x[6] <= 0.022\nsquared_error =
0.248\nsamples = 11\nvalue = 4.455'),
 Text(0.01627770934841928, 0.671875, 'x[4] <= 0.023\nsquared_error =
0.122\nsamples = 7\nvalue = 4.143'),
 Text(0.015812631938464446, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.01674278675837412, 0.640625, 'squared_error = 0.0\nsamples =
6\nvalue = 4.0'),
 Text(0.017207864168328955, 0.671875, 'squared_error = 0.0\nsamples =
4\nvalue = 5.0'),
 Text(0.019068173808148303, 0.734375, 'x[4] <= 0.041\nsquared_error =
0.595\nsamples = 11\nvalue = 5.636'),
 Text(0.018603096398193464, 0.703125, 'x[2] <= 0.057\nsquared_error =
```

```
0.36\nsamples = 10\nvalue = 5.8'),
 Text(0.01813801898823863, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.019068173808148303, 0.671875, 'x[3] <= 0.078\nsquared_error =
0.222\nsamples = 9\nvalue = 5.667'),
 Text(0.018603096398193464, 0.640625, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.019533251218103138, 0.640625, 'x[5] <= 0.021\nsquared_error =
0.24\nsamples = 5\nvalue = 5.4'),
 Text(0.019068173808148303, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 5.0'),
 Text(0.019998328628057976, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.019533251218103138, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 4.0'),
 Text(0.018370557693216046, 0.765625, 'squared_error = 0.0\nsamples =
3\nvalue = 4.0'),
 Text(0.038549648830221314, 0.890625, 'x[6] <= 0.039\nsquared_error =
1.783\nsamples = 218\nvalue = 6.312'),
 Text(0.026816436126341187, 0.859375, 'x[1] <= 0.203\nsquared_error =
0.803\nsamples = 81\nvalue = 5.753'),
 Text(0.023079466469008766, 0.828125, 'x[5] <= 0.023\nsquared_error =
0.906\nsamples = 25\nvalue = 6.12'),
 Text(0.021742368915388614, 0.796875, 'x[1] <= 0.192\nsquared_error =
0.864\nsamples = 19\nvalue = 6.368'),
 Text(0.02046340603801281, 0.765625, 'x[5] <= 0.019\nsquared_error =
0.25\nsamples = 4\nvalue = 5.5'),
 Text(0.019998328628057976, 0.734375, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.020928483447967647, 0.734375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.023021331792764412, 0.765625, 'x[3] <= 0.106\nsquared_error =
0.773\nsamples = 15\nvalue = 6.6'),
 Text(0.02185863826787732, 0.734375, 'x[6] <= 0.031\nsquared_error =
0.472\nsamples = 12\nvalue = 6.833'),
 Text(0.020928483447967647, 0.703125, 'x[4] <= 0.036\nsquared_error =
0.247\nsamples = 9\nvalue = 6.556'),
 Text(0.02046340603801281, 0.671875, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.021393560857922485, 0.671875, 'x[4] <= 0.039\nsquared_error =
0.222\nsamples = 6\nvalue = 6.333'),
 Text(0.020928483447967647, 0.640625, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.02185863826787732, 0.640625, 'x[4] <= 0.051\nsquared_error =
0.222\nsamples = 3\nvalue = 6.667'),
 Text(0.021393560857922485, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.02232371567783216, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
```

```
 Text(0.022788793087786995, 0.703125, 'x[6] <= 0.035\nsquared_error =
0.222\nsamples = 3\nvalue = 7.667'),
 Text(0.02232371567783216, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.02325387049774183, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.024184025317651504, 0.734375, 'x[4] <= 0.055\nsquared_error =
0.889\nsamples = 3\nvalue = 5.667'),
 Text(0.02371894790769667, 0.703125, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.024649102727606342, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.02441656402262892, 0.796875, 'x[5] <= 0.026\nsquared_error =
0.222\nsamples = 6\nvalue = 5.333'),
 Text(0.023951486612674086, 0.765625, 'squared_error = 0.0\nsamples =
4\nvalue = 5.0'),
 Text(0.02488164143258376, 0.765625, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.030553405783673604, 0.828125, 'x[1] <= 0.222\nsquared_error =
0.671\nsamples = 56\nvalue = 5.589'),
 Text(0.0277084400649655, 0.796875, 'x[0] <= 0.273\nsquared_error =
0.552\nsamples = 46\nvalue = 5.457'),
 Text(0.025811796252493434, 0.765625, 'x[4] <= 0.035\nsquared_error =
0.56\nsamples = 5\nvalue = 4.8'),
 Text(0.025346718842538595, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.02627687366244827, 0.734375, 'x[5] <= 0.036\nsquared_error =
0.25\nsamples = 4\nvalue = 4.5'),
 Text(0.025811796252493434, 0.703125, 'x[1] <= 0.207\nsquared_error =
0.222\nsamples = 3\nvalue = 4.333'),
 Text(0.025346718842538595, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.02627687366244827, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.026741951072403104, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.029605083877437568, 0.765625, 'x[5] <= 0.021\nsquared_error =
0.493\nsamples = 41\nvalue = 5.537'),
 Text(0.028137183302267617, 0.734375, 'x[2] <= 0.075\nsquared_error =
0.09\nsamples = 10\nvalue = 5.1'),
 Text(0.027672105892312778, 0.703125, 'squared_error = 0.0\nsamples =
9\nvalue = 5.0'),
 Text(0.028602260712222452, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.031072984452607522, 0.734375, 'x[2] <= 0.082\nsquared_error =
0.541\nsamples = 31\nvalue = 5.677'),
 Text(0.029532415532132126, 0.703125, 'x[4] <= 0.058\nsquared_error =
0.454\nsamples = 28\nvalue = 5.786'),
 Text(0.027846509921045842, 0.671875, 'x[3] <= 0.107\nsquared_error =
```

```
0.426\nsamples = 21\nvalue = 5.952'),
 Text(0.026567547043670044, 0.640625, 'x[6] <= 0.03\nsquared_error =
0.25\nsamples = 6\nvalue = 5.5'),
 Text(0.026102469633715205, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.02703262445362488, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 5.0'),
 Text(0.029125472798421644, 0.640625, 'x[0] <= 0.282\nsquared_error =
0.382\nsamples = 15\nvalue = 6.133'),
 Text(0.027962779273534553, 0.609375, 'x[4] <= 0.051\nsquared_error =
0.24\nsamples = 5\nvalue = 6.6'),
 Text(0.027497701863579714, 0.578125, 'x[1] <= 0.207\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.02703262445362488, 0.546875, 'x[5] <= 0.028\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.026567547043670044, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.027497701863579714, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.027962779273534553, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.028427856683489388, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.030288166323308736, 0.609375, 'x[4] <= 0.049\nsquared_error =
0.29\nsamples = 10\nvalue = 5.9'),
 Text(0.029358011503399062, 0.578125, 'x[0] <= 0.315\nsquared_error =
0.222\nsamples = 3\nvalue = 5.333'),
 Text(0.028892934093444227, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.029823088913353897, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.03121832114321841, 0.578125, 'x[0] <= 0.305\nsquared_error =
0.122\nsamples = 7\nvalue = 6.143'),
 Text(0.03075324373326357, 0.546875, 'squared_error = 0.0\nsamples =
6\nvalue = 6.0'),
 Text(0.031683398553173245, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.03121832114321841, 0.671875, 'x[5] <= 0.022\nsquared_error =
0.204\nsamples = 7\nvalue = 5.286'),
 Text(0.03075324373326357, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.031683398553173245, 0.640625, 'x[4] <= 0.092\nsquared_error =
0.139\nsamples = 6\nvalue = 5.167'),
 Text(0.03121832114321841, 0.609375, 'squared_error = 0.0\nsamples =
5\nvalue = 5.0'),
 Text(0.03214847596312808, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.032613553373082915, 0.703125, 'x[3] <= 0.111\nsquared_error =
0.222\nsamples = 3\nvalue = 4.667'),
```

```
 Text(0.03214847596312808, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 4.0'),
 Text(0.03307863078303776, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.0333983715023817, 0.796875, 'x[3] <= 0.113\nsquared_error =
0.76\nsamples = 10\nvalue = 6.2'),
 Text(0.03246821668247203, 0.765625, 'x[4] <= 0.038\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.032003139272517196, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.03293329409242687, 0.734375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.03432852632229138, 0.765625, 'x[6] <= 0.038\nsquared_error =
0.204\nsamples = 7\nvalue = 5.714'),
 Text(0.033863448912336544, 0.734375, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.034793603732246214, 0.734375, 'x[6] <= 0.039\nsquared_error =
0.222\nsamples = 3\nvalue = 5.333'),
 Text(0.03432852632229138, 0.703125, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.03525868114220105, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.050282861534101435, 0.859375, 'x[4] <= 0.064\nsquared_error =
2.069\nsamples = 137\nvalue = 6.642'),
 Text(0.04525784545623004, 0.828125, 'x[3] <= 0.154\nsquared_error =
3.583\nsamples = 54\nvalue = 7.167'),
 Text(0.042002303586546184, 0.796875, 'x[2] <= 0.062\nsquared_error =
2.307\nsamples = 47\nvalue = 6.766'),
 Text(0.04153722617659135, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.04246738099650102, 0.765625, 'x[4] <= 0.057\nsquared_error =
1.959\nsamples = 46\nvalue = 6.674'),
 Text(0.03851422301188491, 0.734375, 'x[5] <= 0.028\nsquared_error =
2.16\nsamples = 31\nvalue = 7.032'),
 Text(0.03618883596211073, 0.703125, 'x[4] <= 0.053\nsquared_error =
2.876\nsamples = 11\nvalue = 6.182'),
 Text(0.035026142437223635, 0.671875, 'x[3] <= 0.128\nsquared_error =
1.556\nsamples = 9\nvalue = 5.667'),
 Text(0.03409598761731396, 0.640625, 'x[3] <= 0.124\nsquared_error =
0.889\nsamples = 6\nvalue = 6.333'),
 Text(0.03363091020735912, 0.609375, 'x[4] <= 0.04\nsquared_error =
0.4\nsamples = 5\nvalue = 6.0'),
 Text(0.03316583279740429, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.03409598761731396, 0.578125, 'x[3] <= 0.117\nsquared_error =
0.188\nsamples = 4\nvalue = 6.25'),
 Text(0.03363091020735912, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.03456106502726879, 0.546875, 'squared_error = 0.0\nsamples =
```

```
3\nvalue = 6.0'),
 Text(0.03456106502726879, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.035956297257133306, 0.640625, 'x[5] <= 0.025\nsquared_error =
0.222\nsamples = 3\nvalue = 4.333'),
 Text(0.03549121984717847, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.03642137466708814, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 4.0'),
 Text(0.03735152948699782, 0.671875, 'x[6] <= 0.048\nsquared_error =
2.25\nsamples = 2\nvalue = 8.5'),
 Text(0.036886452077042976, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.037816606896952654, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.04083961006165909, 0.703125, 'x[5] <= 0.038\nsquared_error =
1.15\nsamples = 20\nvalue = 7.5'),
 Text(0.039676916536772, 0.671875, 'x[4] <= 0.047\nsquared_error =
1.09\nsamples = 16\nvalue = 7.312'),
 Text(0.038746761716862324, 0.640625, 'x[4] <= 0.044\nsquared_error =
0.222\nsamples = 3\nvalue = 8.333'),
 Text(0.03828168430690749, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.039211839126817166, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.04060707135668167, 0.640625, 'x[2] <= 0.073\nsquared_error =
0.994\nsamples = 13\nvalue = 7.077'),
 Text(0.04014199394672684, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.04107214876663651, 0.609375, 'x[3] <= 0.126\nsquared_error =
0.96\nsamples = 10\nvalue = 6.8'),
 Text(0.04014199394672684, 0.578125, 'x[0] <= 0.29\nsquared_error =
1.0\nsamples = 2\nvalue = 8.0'),
 Text(0.039676916536772, 0.546875, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.04060707135668167, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.042002303586546184, 0.578125, 'x[3] <= 0.139\nsquared_error =
0.5\nsamples = 8\nvalue = 6.5'),
 Text(0.04153722617659135, 0.546875, 'x[5] <= 0.03\nsquared_error =
0.5\nsamples = 4\nvalue = 6.0'),
 Text(0.04107214876663651, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.042002303586546184, 0.515625, 'x[0] <= 0.312\nsquared_error =
0.222\nsamples = 3\nvalue = 5.667'),
 Text(0.04153722617659135, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.04246738099650102, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
```

```
 Text(0.04246738099650102, 0.546875, 'squared_error = 0.0\nsamples =
4\nvalue = 7.0'),
 Text(0.042002303586546184, 0.671875, 'x[0] <= 0.305\nsquared_error =
0.688\nsamples = 4\nvalue = 8.25'),
 Text(0.04153722617659135, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.04246738099650102, 0.640625, 'x[6] <= 0.045\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.042002303586546184, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.042932458406455855, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.04642053898111713, 0.734375, 'x[5] <= 0.034\nsquared_error =
0.729\nsamples = 15\nvalue = 5.933'),
 Text(0.0447927680462752, 0.703125, 'x[0] <= 0.317\nsquared_error =
0.617\nsamples = 9\nvalue = 6.222'),
 Text(0.04386261322636553, 0.671875, 'x[6] <= 0.048\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.04339753581641069, 0.640625, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.04432769063632037, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.04572292286618487, 0.671875, 'x[1] <= 0.273\nsquared_error =
0.56\nsamples = 5\nvalue = 5.8'),
 Text(0.04525784545623004, 0.640625, 'x[6] <= 0.043\nsquared_error =
0.25\nsamples = 4\nvalue = 5.5'),
 Text(0.0447927680462752, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.0'),
 Text(0.04572292286618487, 0.609375, 'x[0] <= 0.328\nsquared_error =
0.222\nsamples = 3\nvalue = 5.333'),
 Text(0.04525784545623004, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.046188000276139715, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.046188000276139715, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.048048309915959056, 0.703125, 'x[0] <= 0.32\nsquared_error =
0.583\nsamples = 6\nvalue = 5.5'),
 Text(0.04758323250600422, 0.671875, 'x[0] <= 0.303\nsquared_error =
0.16\nsamples = 5\nvalue = 5.8'),
 Text(0.047118155096049386, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.048048309915959056, 0.640625, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.0485133873259139, 0.671875, 'squared_error = 0.0\nsamples = 1\
nvalue = 4.0'),
 Text(0.0485133873259139, 0.796875, 'x[0] <= 0.317\nsquared_error =
3.837\nsamples = 7\nvalue = 9.857'),
 Text(0.048048309915959056, 0.765625, 'squared_error = 0.0\nsamples =
```

```
1\nvalue = 6.0'),
 Text(0.04897846473586873, 0.765625, 'x[5] <= 0.033\nsquared_error =
1.583\nsamples = 6\nvalue = 10.5'),
 Text(0.0485133873259139, 0.734375, 'squared_error = 0.0\nsamples = 1\
nvalue = 13.0'),
 Text(0.04944354214582357, 0.734375, 'x[2] <= 0.078\nsquared_error =
0.4\nsamples = 5\nvalue = 10.0'),
 Text(0.04897846473586873, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.049908619555778404, 0.703125, 'x[3] <= 0.17\nsquared_error =
0.188\nsamples = 4\nvalue = 9.75'),
 Text(0.04944354214582357, 0.671875, 'squared_error = 0.0\nsamples =
3\nvalue = 10.0'),
 Text(0.05037369696573324, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.05530787761197284, 0.828125, 'x[1] <= 0.212\nsquared_error =
0.789\nsamples = 83\nvalue = 6.301'),
 Text(0.054842800202018, 0.796875, 'squared_error = 0.0\nsamples = 3\
nvalue = 5.0'),
 Text(0.055772955021927674, 0.796875, 'x[0] <= 0.295\nsquared_error =
0.753\nsamples = 80\nvalue = 6.35'),
 Text(0.05530787761197284, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.05623803243188251, 0.765625, 'x[3] <= 0.17\nsquared_error =
0.672\nsamples = 79\nvalue = 6.316'),
 Text(0.05269908401550742, 0.734375, 'x[4] <= 0.076\nsquared_error =
0.199\nsamples = 19\nvalue = 5.895'),
 Text(0.05176892919559775, 0.703125, 'x[0] <= 0.308\nsquared_error =
0.066\nsamples = 14\nvalue = 6.071'),
 Text(0.051303851785642916, 0.671875, 'x[3] <= 0.159\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.05083877437568808, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.05176892919559775, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.05223400660555259, 0.671875, 'squared_error = 0.0\nsamples =
12\nvalue = 6.0'),
 Text(0.0536292388354171, 0.703125, 'x[3] <= 0.169\nsquared_error =
0.24\nsamples = 5\nvalue = 5.4'),
 Text(0.053164161425462264, 0.671875, 'squared_error = 0.0\nsamples =
3\nvalue = 5.0'),
 Text(0.054094316245371935, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.059776980848257597, 0.734375, 'x[0] <= 0.338\nsquared_error =
0.748\nsamples = 60\nvalue = 6.45'),
 Text(0.05595462588519128, 0.703125, 'x[4] <= 0.073\nsquared_error =
0.614\nsamples = 25\nvalue = 6.84'),
 Text(0.055024471065281605, 0.671875, 'x[1] <= 0.252\nsquared_error =
0.64\nsamples = 5\nvalue = 7.6'),
 Text(0.05455939365532677, 0.640625, 'squared_error = 0.0\nsamples =
```

```
3\nvalue = 7.0'),
 Text(0.05548954847523645, 0.640625, 'x[5] <= 0.037\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.055024471065281605, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.05595462588519128, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.05688478070510095, 0.671875, 'x[0] <= 0.317\nsquared_error =
0.427\nsamples = 20\nvalue = 6.65'),
 Text(0.05641970329514612, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.05734985811505579, 0.640625, 'x[5] <= 0.047\nsquared_error =
0.299\nsamples = 19\nvalue = 6.737'),
 Text(0.05688478070510095, 0.609375, 'x[4] <= 0.085\nsquared_error =
0.222\nsamples = 18\nvalue = 6.667'),
 Text(0.05595462588519128, 0.578125, 'x[3] <= 0.177\nsquared_error =
0.234\nsamples = 8\nvalue = 6.375'),
 Text(0.05548954847523645, 0.546875, 'x[3] <= 0.171\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.055024471065281605, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.05595462588519128, 0.515625, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.05641970329514612, 0.546875, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.05781493552501063, 0.578125, 'x[5] <= 0.032\nsquared_error =
0.09\nsamples = 10\nvalue = 6.9'),
 Text(0.05734985811505579, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.058280012934965465, 0.546875, 'squared_error = 0.0\nsamples =
9\nvalue = 7.0'),
 Text(0.05781493552501063, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.0635993358113239, 0.703125, 'x[4] <= 0.117\nsquared_error =
0.656\nsamples = 35\nvalue = 6.171'),
 Text(0.06313425840136908, 0.671875, 'x[3] <= 0.177\nsquared_error =
0.574\nsamples = 34\nvalue = 6.118'),
 Text(0.06141928545216061, 0.640625, 'x[1] <= 0.262\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.060954208042205776, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.06188436286211545, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.06484923135057753, 0.640625, 'x[3] <= 0.205\nsquared_error =
0.566\nsamples = 30\nvalue = 6.033'),
 Text(0.06281451768202512, 0.609375, 'x[5] <= 0.037\nsquared_error =
0.554\nsamples = 25\nvalue = 5.92'),
 Text(0.06014032257478481, 0.578125, 'x[5] <= 0.027\nsquared_error =
0.734\nsamples = 8\nvalue = 6.375'),
```

```
 Text(0.059210167754875136, 0.546875, 'x[3] <= 0.192\nsquared_error =
0.25\nsamples = 2\nvalue = 5.5'),
 Text(0.0587450903449203, 0.515625, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.05967524516482997, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.06107047739469448, 0.546875, 'x[4] <= 0.081\nsquared_error =
0.556\nsamples = 6\nvalue = 6.667'),
 Text(0.06060539998473965, 0.515625, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.06153555480464932, 0.515625, 'x[6] <= 0.053\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.06107047739469448, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.06200063221460416, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.06548871278926544, 0.578125, 'x[5] <= 0.048\nsquared_error =
0.325\nsamples = 17\nvalue = 5.706'),
 Text(0.06432601926437834, 0.546875, 'x[3] <= 0.195\nsquared_error =
0.248\nsamples = 11\nvalue = 5.455'),
 Text(0.06339586444446867, 0.515625, 'x[3] <= 0.179\nsquared_error =
0.222\nsamples = 6\nvalue = 5.667'),
 Text(0.06293078703451382, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.06386094185442351, 0.484375, 'x[5] <= 0.039\nsquared_error =
0.16\nsamples = 5\nvalue = 5.8'),
 Text(0.06339586444446867, 0.453125, 'x[4] <= 0.089\nsquared_error =
0.25\nsamples = 2\nvalue = 5.5'),
 Text(0.06293078703451382, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.06386094185442351, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.06432601926437834, 0.453125, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.06525617408428801, 0.515625, 'x[1] <= 0.25\nsquared_error =
0.16\nsamples = 5\nvalue = 5.2'),
 Text(0.06479109667433318, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.06572125149424285, 0.484375, 'squared_error = 0.0\nsamples =
4\nvalue = 5.0'),
 Text(0.06665140631415252, 0.546875, 'x[6] <= 0.056\nsquared_error =
0.139\nsamples = 6\nvalue = 6.167'),
 Text(0.06618632890419769, 0.515625, 'squared_error = 0.0\nsamples =
5\nvalue = 6.0'),
 Text(0.06711648372410736, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.06688394501912995, 0.609375, 'x[1] <= 0.278\nsquared_error =
0.24\nsamples = 5\nvalue = 6.6'),
 Text(0.0664188676091751, 0.578125, 'squared_error = 0.0\nsamples = 3\
```

```
nvalue = 7.0'),
 Text(0.06734902242908478, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.06406441322127875, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.20965433598391123, 0.921875, 'x[6] <= 0.112\nsquared_error =
3.851\nsamples = 954\nvalue = 8.205'),
 Text(0.1526741496895245, 0.890625, 'x[2] <= 0.107\nsquared_error =
2.686\nsamples = 459\nvalue = 7.562'),
 Text(0.12429148363327193, 0.859375, 'x[4] <= 0.122\nsquared_error =
2.467\nsamples = 353\nvalue = 7.402'),
 Text(0.10431450133164742, 0.828125, 'x[6] <= 0.086\nsquared_error =
3.09\nsamples = 208\nvalue = 7.659'),
 Text(0.08747633737005991, 0.796875, 'x[4] <= 0.08\nsquared_error =
2.504\nsamples = 180\nvalue = 7.461'),
 Text(0.07583668516076054, 0.765625, 'x[6] <= 0.065\nsquared_error =
2.622\nsamples = 47\nvalue = 8.128'),
 Text(0.07339502875849765, 0.734375, 'x[2] <= 0.093\nsquared_error =
2.381\nsamples = 28\nvalue = 7.607'),
 Text(0.0701394868888138, 0.703125, 'x[5] <= 0.028\nsquared_error =
2.286\nsamples = 21\nvalue = 8.0'),
 Text(0.06967440947885896, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.07060456429876863, 0.671875, 'x[5] <= 0.033\nsquared_error =
1.928\nsamples = 20\nvalue = 7.85'),
 Text(0.0701394868888138, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.07106964170872347, 0.640625, 'x[0] <= 0.34\nsquared_error =
1.579\nsamples = 19\nvalue = 8.0'),
 Text(0.06920933206890413, 0.609375, 'x[3] <= 0.166\nsquared_error =
1.234\nsamples = 8\nvalue = 7.375'),
 Text(0.06827917724899445, 0.578125, 'x[1] <= 0.252\nsquared_error =
0.222\nsamples = 3\nvalue = 8.667'),
 Text(0.06781409983903962, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.06874425465894929, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.0701394868888138, 0.578125, 'x[6] <= 0.061\nsquared_error =
0.24\nsamples = 5\nvalue = 6.6'),
 Text(0.06967440947885896, 0.546875, 'x[3] <= 0.181\nsquared_error =
0.222\nsamples = 3\nvalue = 6.333'),
 Text(0.06920933206890413, 0.515625, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.0701394868888138, 0.515625, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.07060456429876863, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.07292995134854281, 0.609375, 'x[1] <= 0.262\nsquared_error =
1.339\nsamples = 11\nvalue = 8.455'),
```

```
 Text(0.07199979652863314, 0.578125, 'x[4] <= 0.075\nsquared_error =
0.64\nsamples = 5\nvalue = 9.4'),
 Text(0.07153471911867831, 0.546875, 'squared_error = 0.0\nsamples =
3\nvalue = 10.0'),
 Text(0.07246487393858798, 0.546875, 'x[5] <= 0.038\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.07199979652863314, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.07292995134854281, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.0738601061684525, 0.578125, 'x[6] <= 0.059\nsquared_error =
0.556\nsamples = 6\nvalue = 7.667'),
 Text(0.07339502875849765, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.07432518357840733, 0.546875, 'x[3] <= 0.177\nsquared_error =
0.24\nsamples = 5\nvalue = 7.4'),
 Text(0.0738601061684525, 0.515625, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.07479026098836217, 0.515625, 'x[4] <= 0.076\nsquared_error =
0.188\nsamples = 4\nvalue = 7.25'),
 Text(0.07432518357840733, 0.484375, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.075255338398317, 0.484375, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.07665057062818151, 0.703125, 'x[0] <= 0.355\nsquared_error =
0.816\nsamples = 7\nvalue = 6.429'),
 Text(0.07618549321822668, 0.671875, 'x[5] <= 0.047\nsquared_error =
0.472\nsamples = 6\nvalue = 6.167'),
 Text(0.07572041580827184, 0.640625, 'x[4] <= 0.074\nsquared_error =
0.24\nsamples = 5\nvalue = 6.4'),
 Text(0.075255338398317, 0.609375, 'x[5] <= 0.03\nsquared_error =
0.222\nsamples = 3\nvalue = 6.667'),
 Text(0.07479026098836217, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.07572041580827184, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.07618549321822668, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.07665057062818151, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.07711564803813635, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.07827834156302343, 0.734375, 'x[3] <= 0.178\nsquared_error =
1.989\nsamples = 19\nvalue = 8.895'),
 Text(0.0778132641530686, 0.703125, 'squared_error = 0.0\nsamples = 1\
nvalue = 12.0'),
 Text(0.07874341897297828, 0.703125, 'x[3] <= 0.206\nsquared_error =
1.534\nsamples = 18\nvalue = 8.722'),
 Text(0.07827834156302343, 0.671875, 'x[3] <= 0.189\nsquared_error =
```

```
1.254\nsamples = 13\nvalue = 8.231'),
 Text(0.0778132641530686, 0.640625, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.07874341897297828, 0.640625, 'x[4] <= 0.075\nsquared_error =
1.157\nsamples = 11\nvalue = 8.455'),
 Text(0.07711564803813635, 0.609375, 'x[2] <= 0.072\nsquared_error =
0.286\nsamples = 7\nvalue = 9.0'),
 Text(0.07665057062818151, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.07758072544809118, 0.578125, 'x[3] <= 0.201\nsquared_error =
0.139\nsamples = 6\nvalue = 8.833'),
 Text(0.07711564803813635, 0.546875, 'squared_error = 0.0\nsamples =
5\nvalue = 9.0'),
 Text(0.07804580285804602, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.0803711899078202, 0.609375, 'x[1] <= 0.267\nsquared_error =
1.25\nsamples = 4\nvalue = 7.5'),
 Text(0.07944103508791053, 0.578125, 'x[3] <= 0.195\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.07897595767795569, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.07990611249786536, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.08130134472772987, 0.578125, 'x[0] <= 0.355\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.08083626731777505, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.08176642213768472, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.07920849638293312, 0.671875, 'squared_error = 0.0\nsamples =
5\nvalue = 10.0'),
 Text(0.09911598957935928, 0.765625, 'x[1] <= 0.278\nsquared_error =
2.25\nsamples = 133\nvalue = 7.226'),
 Text(0.08644626357535526, 0.734375, 'x[1] <= 0.257\nsquared_error =
2.015\nsamples = 55\nvalue = 6.8'),
 Text(0.08316165436754923, 0.703125, 'x[5] <= 0.05\nsquared_error =
7.222\nsamples = 6\nvalue = 8.333'),
 Text(0.08223149954763954, 0.671875, 'x[0] <= 0.347\nsquared_error =
2.889\nsamples = 3\nvalue = 10.667'),
 Text(0.08176642213768472, 0.640625, 'x[6] <= 0.062\nsquared_error =
0.25\nsamples = 2\nvalue = 9.5'),
 Text(0.08130134472772987, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.08223149954763954, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.08269657695759439, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.0840918091874589, 0.671875, 'x[2] <= 0.095\nsquared_error =
0.667\nsamples = 3\nvalue = 6.0'),
```

```
 Text(0.08362673177750406, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.08455688659741373, 0.640625, 'x[4] <= 0.112\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.0840918091874589, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.08502196400736857, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.08973087278316129, 0.703125, 'x[6] <= 0.066\nsquared_error =
1.054\nsamples = 49\nvalue = 6.612'),
 Text(0.08734735105714275, 0.671875, 'x[5] <= 0.035\nsquared_error =
0.508\nsamples = 21\nvalue = 6.333'),
 Text(0.08641719623723308, 0.640625, 'x[4] <= 0.084\nsquared_error =
0.889\nsamples = 3\nvalue = 5.333'),
 Text(0.08595211882727824, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.08688227364718791, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 4.0'),
 Text(0.08827750587705242, 0.640625, 'x[6] <= 0.061\nsquared_error =
0.25\nsamples = 18\nvalue = 6.5'),
 Text(0.0878124284670976, 0.609375, 'squared_error = 0.0\nsamples = 5\
nvalue = 7.0'),
 Text(0.08874258328700727, 0.609375, 'x[3] <= 0.234\nsquared_error =
0.213\nsamples = 13\nvalue = 6.308'),
 Text(0.08827750587705242, 0.578125, 'x[3] <= 0.199\nsquared_error =
0.149\nsamples = 11\nvalue = 6.182'),
 Text(0.0878124284670976, 0.546875, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.08874258328700727, 0.546875, 'x[4] <= 0.108\nsquared_error =
0.09\nsamples = 10\nvalue = 6.1'),
 Text(0.08827750587705242, 0.515625, 'squared_error = 0.0\nsamples =
8\nvalue = 6.0'),
 Text(0.0892076606969621, 0.515625, 'x[4] <= 0.113\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.08874258328700727, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.08967273810691694, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.0892076606969621, 0.578125, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.09211439450917983, 0.671875, 'x[5] <= 0.038\nsquared_error =
1.361\nsamples = 28\nvalue = 6.821'),
 Text(0.0906028929268266, 0.640625, 'x[3] <= 0.219\nsquared_error =
0.889\nsamples = 3\nvalue = 9.333'),
 Text(0.09013781551687178, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 10.0'),
 Text(0.09106797033678145, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.09362589609153305, 0.640625, 'x[3] <= 0.232\nsquared_error =
```

```
0.57\nsamples = 25\nvalue = 6.52'),
 Text(0.09199812515669112, 0.609375, 'x[3] <= 0.22\nsquared_error =
0.534\nsamples = 18\nvalue = 6.722'),
 Text(0.09106797033678145, 0.578125, 'x[3] <= 0.212\nsquared_error =
0.444\nsamples = 9\nvalue = 6.333'),
 Text(0.0906028929268266, 0.546875, 'x[3] <= 0.209\nsquared_error =
0.688\nsamples = 4\nvalue = 6.75'),
 Text(0.09013781551687178, 0.515625, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.09106797033678145, 0.515625, 'x[5] <= 0.049\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.0906028929268266, 0.484375, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.09153304774673628, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.09153304774673628, 0.546875, 'squared_error = 0.0\nsamples =
5\nvalue = 6.0'),
 Text(0.09292827997660079, 0.578125, 'x[4] <= 0.086\nsquared_error =
0.321\nsamples = 9\nvalue = 7.111'),
 Text(0.09246320256664596, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.09339335738655563, 0.546875, 'x[1] <= 0.262\nsquared_error =
0.188\nsamples = 8\nvalue = 7.25'),
 Text(0.09292827997660079, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.09385843479651046, 0.515625, 'x[5] <= 0.059\nsquared_error =
0.122\nsamples = 7\nvalue = 7.143'),
 Text(0.09339335738655563, 0.484375, 'squared_error = 0.0\nsamples =
6\nvalue = 7.0'),
 Text(0.0943235122064653, 0.484375, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.09525366702637497, 0.609375, 'x[0] <= 0.362\nsquared_error =
0.286\nsamples = 7\nvalue = 6.0'),
 Text(0.09478858961642014, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.09571874443632981, 0.578125, 'x[2] <= 0.088\nsquared_error =
0.139\nsamples = 6\nvalue = 5.833'),
 Text(0.09525366702637497, 0.546875, 'x[3] <= 0.237\nsquared_error =
0.25\nsamples = 2\nvalue = 5.5'),
 Text(0.09478858961642014, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.09571874443632981, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.09618382184628464, 0.546875, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.11178571558336331, 0.734375, 'x[0] <= 0.392\nsquared_error =
2.198\nsamples = 78\nvalue = 7.526'),
 Text(0.10581964443378641, 0.703125, 'x[6] <= 0.072\nsquared_error =
2.343\nsamples = 66\nvalue = 7.667'),
```

```
 Text(0.10025324918338947, 0.671875, 'x[4] <= 0.092\nsquared_error =
1.943\nsamples = 24\nvalue = 7.125'),
 Text(0.098044131486104, 0.640625, 'x[5] <= 0.051\nsquared_error =
2.286\nsamples = 7\nvalue = 8.0'),
 Text(0.09711397666619433, 0.609375, 'x[4] <= 0.082\nsquared_error =
0.56\nsamples = 5\nvalue = 7.2'),
 Text(0.09664889925623948, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.09757905407614915, 0.578125, 'x[4] <= 0.084\nsquared_error =
0.25\nsamples = 4\nvalue = 7.5'),
 Text(0.09711397666619433, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.098044131486104, 0.546875, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.09897428630601367, 0.609375, 'x[5] <= 0.053\nsquared_error =
1.0\nsamples = 2\nvalue = 10.0'),
 Text(0.09850920889605884, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.09943936371596851, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.10246236688067495, 0.640625, 'x[4] <= 0.11\nsquared_error =
1.356\nsamples = 17\nvalue = 6.765'),
 Text(0.10129967335578785, 0.609375, 'x[1] <= 0.282\nsquared_error =
0.61\nsamples = 10\nvalue = 6.3'),
 Text(0.10036951853587818, 0.578125, 'x[4] <= 0.102\nsquared_error =
0.56\nsamples = 5\nvalue = 6.8'),
 Text(0.09990444112592334, 0.546875, 'x[0] <= 0.38\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.09943936371596851, 0.515625, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.10036951853587818, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.10083459594583302, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.10222982817569752, 0.578125, 'x[5] <= 0.044\nsquared_error =
0.16\nsamples = 5\nvalue = 5.8'),
 Text(0.10176475076574269, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.10269490558565236, 0.546875, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.10362506040556203, 0.609375, 'x[4] <= 0.112\nsquared_error =
1.673\nsamples = 7\nvalue = 7.429'),
 Text(0.1031599829956072, 0.578125, 'squared_error = 0.0\nsamples = 1\
nvalue = 10.0'),
 Text(0.10409013781551688, 0.578125, 'x[3] <= 0.231\nsquared_error =
0.667\nsamples = 6\nvalue = 7.0'),
 Text(0.10362506040556203, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.1045552152254717, 0.546875, 'x[4] <= 0.12\nsquared_error =
```

```
0.56\nsamples = 5\nvalue = 6.8'),
 Text(0.10409013781551688, 0.515625, 'x[4] <= 0.117\nsquared_error =
0.25\nsamples = 4\nvalue = 6.5'),
 Text(0.10362506040556203, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.1045552152254717, 0.484375, 'x[6] <= 0.069\nsquared_error =
0.222\nsamples = 3\nvalue = 6.667'),
 Text(0.10409013781551688, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.10502029263542655, 0.453125, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.10502029263542655, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.11138603968418337, 0.671875, 'x[2] <= 0.093\nsquared_error =
2.309\nsamples = 42\nvalue = 7.976'),
 Text(0.10734567968520073, 0.640625, 'x[3] <= 0.272\nsquared_error =
0.465\nsamples = 16\nvalue = 7.312'),
 Text(0.10688060227524589, 0.609375, 'x[0] <= 0.382\nsquared_error =
0.373\nsamples = 15\nvalue = 7.4'),
 Text(0.10595044745533622, 0.578125, 'x[6] <= 0.079\nsquared_error =
0.122\nsamples = 7\nvalue = 7.143'),
 Text(0.10548537004538139, 0.546875, 'squared_error = 0.0\nsamples =
6\nvalue = 7.0'),
 Text(0.10641552486529106, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.10781075709515557, 0.578125, 'x[6] <= 0.075\nsquared_error =
0.484\nsamples = 8\nvalue = 7.625'),
 Text(0.10734567968520073, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.1082758345051104, 0.546875, 'x[3] <= 0.251\nsquared_error =
0.245\nsamples = 7\nvalue = 7.429'),
 Text(0.10781075709515557, 0.515625, 'x[3] <= 0.234\nsquared_error =
0.188\nsamples = 4\nvalue = 7.75'),
 Text(0.10734567968520073, 0.484375, 'x[6] <= 0.08\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.10688060227524589, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.10781075709515557, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.1082758345051104, 0.484375, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
 Text(0.10874091191506524, 0.515625, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.10781075709515557, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.11542639968316601, 0.640625, 'x[6] <= 0.074\nsquared_error =
3.006\nsamples = 26\nvalue = 8.385'),
 Text(0.11496132227321118, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 12.0'),
```

```
 Text(0.11589147709312085, 0.609375, 'x[0] <= 0.382\nsquared_error =
2.582\nsamples = 25\nvalue = 8.24'),
 Text(0.1132754166621249, 0.578125, 'x[3] <= 0.244\nsquared_error =
2.817\nsamples = 17\nvalue = 8.647'),
 Text(0.11129883766981684, 0.546875, 'x[1] <= 0.282\nsquared_error =
1.75\nsamples = 8\nvalue = 7.5'),
 Text(0.11013614414492975, 0.515625, 'x[4] <= 0.084\nsquared_error =
1.25\nsamples = 4\nvalue = 8.5'),
 Text(0.10920598932502007, 0.484375, 'x[4] <= 0.081\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.10874091191506524, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.10967106673497491, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.11106629896483942, 0.484375, 'x[2] <= 0.1\nsquared_error =
0.25\nsamples = 2\nvalue = 9.5'),
 Text(0.11060122155488458, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.11153137637479425, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.11246153119470394, 0.515625, 'x[3] <= 0.239\nsquared_error =
0.25\nsamples = 4\nvalue = 6.5'),
 Text(0.1119964537847491, 0.484375, 'squared_error = 0.0\nsamples = 2\
nvalue = 6.0'),
 Text(0.11292660860465877, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.11525199565443295, 0.546875, 'x[3] <= 0.267\nsquared_error =
1.556\nsamples = 9\nvalue = 9.667'),
 Text(0.11432184083452328, 0.515625, 'x[5] <= 0.058\nsquared_error =
0.556\nsamples = 6\nvalue = 10.333'),
 Text(0.11385676342456844, 0.484375, 'x[6] <= 0.081\nsquared_error =
0.5\nsamples = 4\nvalue = 10.0'),
 Text(0.11339168601461361, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.11432184083452328, 0.453125, 'x[5] <= 0.049\nsquared_error =
0.222\nsamples = 3\nvalue = 10.333'),
 Text(0.11385676342456844, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.11478691824447812, 0.421875, 'squared_error = 0.0\nsamples =
2\nvalue = 10.0'),
 Text(0.11478691824447812, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 11.0'),
 Text(0.11618215047434262, 0.515625, 'x[5] <= 0.056\nsquared_error =
0.889\nsamples = 3\nvalue = 8.333'),
 Text(0.11571707306438779, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.11664722788429746, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.1185075375241168, 0.578125, 'x[2] <= 0.098\nsquared_error =
```

```
0.984\nsamples = 8\nvalue = 7.375'),
 Text(0.11757738270420713, 0.546875, 'x[6] <= 0.082\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.1171123052942523, 0.515625, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.11804246011416197, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.11943769234402649, 0.546875, 'x[2] <= 0.102\nsquared_error =
0.667\nsamples = 6\nvalue = 7.0'),
 Text(0.11897261493407164, 0.515625, 'x[0] <= 0.387\nsquared_error =
0.25\nsamples = 4\nvalue = 6.5'),
 Text(0.1185075375241168, 0.484375, 'squared_error = 0.0\nsamples = 2\
nvalue = 6.0'),
 Text(0.11943769234402649, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.11990276975398131, 0.515625, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.11775178673294019, 0.703125, 'x[6] <= 0.078\nsquared_error =
0.688\nsamples = 12\nvalue = 6.75'),
 Text(0.11682163191303052, 0.671875, 'x[5] <= 0.059\nsquared_error =
0.188\nsamples = 4\nvalue = 5.75'),
 Text(0.1163565545030757, 0.640625, 'squared_error = 0.0\nsamples = 3\
nvalue = 6.0'),
 Text(0.11728670932298536, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 5.0'),
 Text(0.11868194155284988, 0.671875, 'x[2] <= 0.088\nsquared_error =
0.188\nsamples = 8\nvalue = 7.25'),
 Text(0.11821686414289503, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.1191470189628047, 0.640625, 'x[3] <= 0.285\nsquared_error =
0.122\nsamples = 7\nvalue = 7.143'),
 Text(0.11868194155284988, 0.609375, 'squared_error = 0.0\nsamples =
6\nvalue = 7.0'),
 Text(0.11961209637275955, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.12115266529323494, 0.796875, 'x[6] <= 0.089\nsquared_error =
4.995\nsamples = 28\nvalue = 8.929'),
 Text(0.1191470189628047, 0.765625, 'x[1] <= 0.293\nsquared_error =
0.25\nsamples = 2\nvalue = 12.5'),
 Text(0.11868194155284988, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 12.0'),
 Text(0.11961209637275955, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.12315831162366517, 0.765625, 'x[5] <= 0.047\nsquared_error =
4.303\nsamples = 26\nvalue = 8.654'),
 Text(0.12054225119266922, 0.734375, 'x[3] <= 0.178\nsquared_error =
0.4\nsamples = 5\nvalue = 7.0'),
 Text(0.12007717378271437, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.12100732860262406, 0.703125, 'x[0] <= 0.363\nsquared_error =
```

```
0.188\nsamples = 4\nvalue = 7.25'),
 Text(0.12054225119266922, 0.671875, 'x[3] <= 0.221\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.12007717378271437, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.12100732860262406, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.12147240601257889, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.12577437205466113, 0.734375, 'x[0] <= 0.422\nsquared_error =
4.426\nsamples = 21\nvalue = 9.048'),
 Text(0.1253092946447063, 0.703125, 'x[6] <= 0.091\nsquared_error =
3.812\nsamples = 19\nvalue = 9.368'),
 Text(0.1228676382424434, 0.671875, 'x[4] <= 0.113\nsquared_error =
6.16\nsamples = 5\nvalue = 10.8'),
 Text(0.12193748342253373, 0.640625, 'x[5] <= 0.058\nsquared_error =
0.889\nsamples = 3\nvalue = 12.667'),
 Text(0.12147240601257889, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 14.0'),
 Text(0.12240256083248856, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 12.0'),
 Text(0.12379779306235307, 0.640625, 'x[3] <= 0.298\nsquared_error =
1.0\nsamples = 2\nvalue = 8.0'),
 Text(0.12333271565239824, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.12426287047230791, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.12775095104696918, 0.671875, 'x[3] <= 0.284\nsquared_error =
1.98\nsamples = 14\nvalue = 8.857'),
 Text(0.12612318011212725, 0.640625, 'x[0] <= 0.4\nsquared_error =
1.0\nsamples = 8\nvalue = 8.0'),
 Text(0.1251930252922176, 0.609375, 'x[6] <= 0.095\nsquared_error =
0.24\nsamples = 5\nvalue = 8.6'),
 Text(0.12472794788226274, 0.578125, 'x[2] <= 0.1\nsquared_error =
0.222\nsamples = 3\nvalue = 8.333'),
 Text(0.12426287047230791, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.1251930252922176, 0.546875, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
 Text(0.12565810270217243, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.12705333493203694, 0.609375, 'x[5] <= 0.064\nsquared_error =
0.667\nsamples = 3\nvalue = 7.0'),
 Text(0.12658825752208208, 0.578125, 'x[3] <= 0.281\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.12612318011212725, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.12705333493203694, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
```

```
 Text(0.12751841234199177, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.1293787219818111, 0.640625, 'x[1] <= 0.322\nsquared_error =
1.0\nsamples = 6\nvalue = 10.0'),
 Text(0.12891364457185628, 0.609375, 'x[1] <= 0.308\nsquared_error =
0.24\nsamples = 5\nvalue = 10.4'),
 Text(0.12844856716190145, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 10.0'),
 Text(0.1293787219818111, 0.578125, 'x[6] <= 0.098\nsquared_error =
0.222\nsamples = 3\nvalue = 10.667'),
 Text(0.12891364457185628, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.12984379939176596, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 11.0'),
 Text(0.12984379939176596, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.12623944946461596, 0.703125, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.14426846593489642, 0.828125, 'x[6] <= 0.094\nsquared_error =
1.344\nsamples = 145\nvalue = 7.034'),
 Text(0.13671822485766089, 0.796875, 'x[5] <= 0.035\nsquared_error =
1.149\nsamples = 79\nvalue = 6.797'),
 Text(0.13625314744770603, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.13718330226761571, 0.765625, 'x[4] <= 0.138\nsquared_error =
1.03\nsamples = 78\nvalue = 6.756'),
 Text(0.13318654327581633, 0.734375, 'x[3] <= 0.256\nsquared_error =
1.138\nsamples = 28\nvalue = 7.071'),
 Text(0.13123903162163045, 0.703125, 'x[1] <= 0.295\nsquared_error =
0.4\nsamples = 5\nvalue = 6.0'),
 Text(0.13077395421167562, 0.671875, 'x[1] <= 0.28\nsquared_error =
0.188\nsamples = 4\nvalue = 6.25'),
 Text(0.1303088768017208, 0.640625, 'squared_error = 0.0\nsamples = 3\
nvalue = 6.0'),
 Text(0.13123903162163045, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.1317041090315853, 0.671875, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.13513405493000222, 0.703125, 'x[3] <= 0.315\nsquared_error =
0.994\nsamples = 23\nvalue = 7.304'),
 Text(0.13368068802389335, 0.671875, 'x[4] <= 0.125\nsquared_error =
0.916\nsamples = 21\nvalue = 7.19'),
 Text(0.13216918644154013, 0.640625, 'x[2] <= 0.098\nsquared_error =
1.36\nsamples = 5\nvalue = 7.8'),
 Text(0.1317041090315853, 0.609375, 'x[6] <= 0.076\nsquared_error =
0.667\nsamples = 3\nvalue = 7.0'),
 Text(0.13123903162163045, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.13216918644154013, 0.578125, 'x[0] <= 0.392\nsquared_error =
```

```
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.1317041090315853, 0.546875, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.13263426385149496, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.13263426385149496, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.13519218960624657, 0.640625, 'x[0] <= 0.392\nsquared_error =
0.625\nsamples = 16\nvalue = 7.0'),
 Text(0.13356441867140464, 0.609375, 'x[5] <= 0.054\nsquared_error =
0.25\nsamples = 6\nvalue = 7.5'),
 Text(0.13309934126144982, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.13402949608135947, 0.578125, 'x[6] <= 0.075\nsquared_error =
0.188\nsamples = 4\nvalue = 7.25'),
 Text(0.13356441867140464, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.13449457349131433, 0.546875, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.1368199605410885, 0.609375, 'x[4] <= 0.13\nsquared_error =
0.61\nsamples = 10\nvalue = 6.7'),
 Text(0.1358898057211788, 0.578125, 'x[1] <= 0.267\nsquared_error =
0.222\nsamples = 6\nvalue = 6.333'),
 Text(0.13542472831122399, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.13635488313113367, 0.546875, 'x[0] <= 0.407\nsquared_error =
0.16\nsamples = 5\nvalue = 6.2'),
 Text(0.1358898057211788, 0.515625, 'squared_error = 0.0\nsamples = 3\
nvalue = 6.0'),
 Text(0.1368199605410885, 0.515625, 'x[6] <= 0.088\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.13635488313113367, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.13728503795104333, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.13775011536099818, 0.578125, 'x[5] <= 0.062\nsquared_error =
0.688\nsamples = 4\nvalue = 7.25'),
 Text(0.13728503795104333, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.138215192770953, 0.546875, 'x[2] <= 0.093\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.13775011536099818, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.13868027018090784, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.13658742183611108, 0.671875, 'x[6] <= 0.091\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.13612234442615626, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
```

```
 Text(0.1370524992460659, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.1411800612594151, 0.734375, 'x[3] <= 0.274\nsquared_error =
0.884\nsamples = 50\nvalue = 6.58'),
 Text(0.14071498384946027, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.14164513866936992, 0.703125, 'x[1] <= 0.278\nsquared_error =
0.78\nsamples = 49\nvalue = 6.531'),
 Text(0.14030804111574977, 0.671875, 'x[3] <= 0.302\nsquared_error =
0.222\nsamples = 3\nvalue = 5.333'),
 Text(0.13984296370579494, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 5.0'),
 Text(0.14077311852570462, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.14298223622299008, 0.671875, 'x[0] <= 0.392\nsquared_error =
0.716\nsamples = 46\nvalue = 6.609'),
 Text(0.14170327334561428, 0.640625, 'x[3] <= 0.308\nsquared_error =
0.615\nsamples = 13\nvalue = 7.0'),
 Text(0.14054057982072718, 0.609375, 'x[6] <= 0.083\nsquared_error =
0.889\nsamples = 6\nvalue = 7.333'),
 Text(0.13961042500081752, 0.578125, 'x[6] <= 0.081\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.1391453475908627, 0.546875, 'squared_error = 0.0\nsamples = 3\
nvalue = 7.0'),
 Text(0.14007550241077235, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.14147073464063686, 0.578125, 'x[6] <= 0.084\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.14100565723068204, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.1419358120505917, 0.546875, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.14286596687050138, 0.609375, 'x[4] <= 0.145\nsquared_error =
0.204\nsamples = 7\nvalue = 6.714'),
 Text(0.14240088946054655, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.1433310442804562, 0.578125, 'squared_error = 0.0\nsamples = 5\
nvalue = 7.0'),
 Text(0.1442611991003659, 0.640625, 'x[3] <= 0.283\nsquared_error =
0.672\nsamples = 33\nvalue = 6.455'),
 Text(0.14379612169041106, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.14472627651032072, 0.609375, 'x[6] <= 0.078\nsquared_error =
0.616\nsamples = 32\nvalue = 6.406'),
 Text(0.1442611991003659, 0.578125, 'squared_error = 0.0\nsamples = 3\
nvalue = 5.0'),
 Text(0.14519135392027555, 0.578125, 'x[0] <= 0.398\nsquared_error =
0.454\nsamples = 29\nvalue = 6.552'),
 Text(0.1442611991003659, 0.546875, 'x[3] <= 0.304\nsquared_error =
```

```
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.14379612169041106, 0.515625, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.14472627651032072, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.14612150874018523, 0.546875, 'x[4] <= 0.18\nsquared_error =
0.402\nsamples = 26\nvalue = 6.462'),
 Text(0.1456564313302304, 0.515625, 'x[4] <= 0.175\nsquared_error =
0.435\nsamples = 21\nvalue = 6.571'),
 Text(0.14472627651032072, 0.484375, 'x[0] <= 0.422\nsquared_error =
0.358\nsamples = 18\nvalue = 6.444'),
 Text(0.1442611991003659, 0.453125, 'x[6] <= 0.089\nsquared_error =
0.347\nsamples = 14\nvalue = 6.286'),
 Text(0.14379612169041106, 0.421875, 'x[5] <= 0.071\nsquared_error =
0.531\nsamples = 7\nvalue = 6.571'),
 Text(0.14286596687050138, 0.390625, 'x[3] <= 0.304\nsquared_error =
0.16\nsamples = 5\nvalue = 6.2'),
 Text(0.14240088946054655, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.1433310442804562, 0.359375, 'squared_error = 0.0\nsamples = 4\
nvalue = 6.0'),
 Text(0.14472627651032072, 0.390625, 'x[5] <= 0.075\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.1442611991003659, 0.359375, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.14519135392027555, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.14472627651032072, 0.421875, 'squared_error = 0.0\nsamples =
7\nvalue = 6.0'),
 Text(0.14519135392027555, 0.453125, 'squared_error = 0.0\nsamples =
4\nvalue = 7.0'),
 Text(0.14658658615014006, 0.484375, 'x[6] <= 0.086\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.14612150874018523, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.14705166356009491, 0.453125, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.14658658615014006, 0.515625, 'squared_error = 0.0\nsamples =
5\nvalue = 6.0'),
 Text(0.151818707012132, 0.796875, 'x[3] <= 0.365\nsquared_error =
1.429\nsamples = 66\nvalue = 7.318'),
 Text(0.14798181838000457, 0.765625, 'x[4] <= 0.142\nsquared_error =
1.644\nsamples = 37\nvalue = 7.757'),
 Text(0.14530762327276428, 0.734375, 'x[2] <= 0.098\nsquared_error =
0.375\nsamples = 16\nvalue = 7.0'),
 Text(0.1443774684528546, 0.703125, 'x[5] <= 0.062\nsquared_error =
0.25\nsamples = 4\nvalue = 7.5'),
 Text(0.14391239104289977, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
```

```
 Text(0.14484254586280942, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.14623777809267394, 0.703125, 'x[3] <= 0.303\nsquared_error =
0.306\nsamples = 12\nvalue = 6.833'),
 Text(0.1457727006827191, 0.671875, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.14670285550262877, 0.671875, 'x[5] <= 0.07\nsquared_error =
0.198\nsamples = 11\nvalue = 6.727'),
 Text(0.14623777809267394, 0.640625, 'squared_error = 0.0\nsamples =
5\nvalue = 7.0'),
 Text(0.14716793291258362, 0.640625, 'x[5] <= 0.072\nsquared_error =
0.25\nsamples = 6\nvalue = 6.5'),
 Text(0.14670285550262877, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.14763301032253845, 0.609375, 'x[3] <= 0.337\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.14716793291258362, 0.578125, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.14809808773249328, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.1506560134872449, 0.734375, 'x[4] <= 0.145\nsquared_error =
1.841\nsamples = 21\nvalue = 8.333'),
 Text(0.15019093607729006, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 12.0'),
 Text(0.15112109089719972, 0.703125, 'x[5] <= 0.066\nsquared_error =
1.228\nsamples = 20\nvalue = 8.15'),
 Text(0.14995839737231265, 0.671875, 'x[0] <= 0.417\nsquared_error =
1.333\nsamples = 6\nvalue = 9.0'),
 Text(0.14902824255240296, 0.640625, 'x[4] <= 0.164\nsquared_error =
0.25\nsamples = 2\nvalue = 10.5'),
 Text(0.14856316514244813, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.1494933199623578, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 11.0'),
 Text(0.1508885521922223, 0.640625, 'x[4] <= 0.149\nsquared_error =
0.188\nsamples = 4\nvalue = 8.25'),
 Text(0.15042347478226747, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.15135362960217713, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.15228378442208682, 0.671875, 'x[0] <= 0.407\nsquared_error =
0.74\nsamples = 14\nvalue = 7.786'),
 Text(0.151818707012132, 0.640625, 'squared_error = 0.0\nsamples = 3\
nvalue = 9.0'),
 Text(0.15274886183204164, 0.640625, 'x[4] <= 0.159\nsquared_error =
0.43\nsamples = 11\nvalue = 7.455'),
 Text(0.15228378442208682, 0.609375, 'squared_error = 0.0\nsamples =
6\nvalue = 7.0'),
 Text(0.1532139392419965, 0.609375, 'x[5] <= 0.084\nsquared_error =
```

```
0.4\nsamples = 5\nvalue = 8.0'),
 Text(0.15274886183204164, 0.578125, 'x[6] <= 0.104\nsquared_error =
0.188\nsamples = 4\nvalue = 7.75'),
 Text(0.15228378442208682, 0.546875, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.1532139392419965, 0.546875, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.15367901665195133, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.15565559564425938, 0.765625, 'x[4] <= 0.167\nsquared_error =
0.597\nsamples = 29\nvalue = 6.759'),
 Text(0.15414409406190616, 0.734375, 'x[0] <= 0.433\nsquared_error =
0.188\nsamples = 4\nvalue = 5.75'),
 Text(0.15367901665195133, 0.703125, 'squared_error = 0.0\nsamples =
3\nvalue = 6.0'),
 Text(0.154609171471861, 0.703125, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.1571670972266126, 0.734375, 'x[3] <= 0.392\nsquared_error =
0.474\nsamples = 25\nvalue = 6.92'),
 Text(0.15553932629177067, 0.703125, 'x[0] <= 0.453\nsquared_error =
0.331\nsamples = 13\nvalue = 7.231'),
 Text(0.15507424888181584, 0.671875, 'x[1] <= 0.345\nsquared_error =
0.2\nsamples = 10\nvalue = 7.0'),
 Text(0.154609171471861, 0.640625, 'x[6] <= 0.104\nsquared_error =
0.099\nsamples = 9\nvalue = 7.111'),
 Text(0.15414409406190616, 0.609375, 'squared_error = 0.0\nsamples =
6\nvalue = 7.0'),
 Text(0.15507424888181584, 0.609375, 'x[4] <= 0.174\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.154609171471861, 0.578125, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.15553932629177067, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.15553932629177067, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.1560044037017255, 0.671875, 'squared_error = 0.0\nsamples = 3\
nvalue = 8.0'),
 Text(0.15879486816145452, 0.703125, 'x[5] <= 0.087\nsquared_error =
0.41\nsamples = 12\nvalue = 6.583'),
 Text(0.15786471334154487, 0.671875, 'x[4] <= 0.264\nsquared_error =
0.204\nsamples = 7\nvalue = 6.286'),
 Text(0.15739963593159, 0.640625, 'x[6] <= 0.104\nsquared_error =
0.139\nsamples = 6\nvalue = 6.167'),
 Text(0.15693455852163518, 0.609375, 'x[5] <= 0.078\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.15646948111168035, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.15739963593159, 0.578125, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.0'),
```

```
 Text(0.15786471334154487, 0.609375, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.1583297907514997, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.1597250229813642, 0.671875, 'x[4] <= 0.191\nsquared_error =
0.4\nsamples = 5\nvalue = 7.0'),
 Text(0.15925994557140938, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.16019010039131903, 0.640625, 'x[3] <= 0.418\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.1597250229813642, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.0'),
 Text(0.16065517780127386, 0.609375, 'squared_error = 0.0\nsamples =
3\nvalue = 7.0'),
 Text(0.18105681574577706, 0.859375, 'x[4] <= 0.151\nsquared_error =
3.048\nsamples = 106\nvalue = 8.094'),
 Text(0.1747746372941215, 0.828125, 'x[6] <= 0.105\nsquared_error =
2.983\nsamples = 57\nvalue = 8.561'),
 Text(0.172151310028595, 0.796875, 'x[4] <= 0.138\nsquared_error =
2.405\nsamples = 52\nvalue = 8.308'),
 Text(0.16876496513736133, 0.765625, 'x[4] <= 0.131\nsquared_error =
2.037\nsamples = 44\nvalue = 8.091'),
 Text(0.16478273981462305, 0.734375, 'x[5] <= 0.046\nsquared_error =
2.054\nsamples = 35\nvalue = 8.343'),
 Text(0.16205041003113838, 0.703125, 'x[3] <= 0.22\nsquared_error =
1.5\nsamples = 4\nvalue = 7.0'),
 Text(0.16158533262118355, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.16251548744109323, 0.671875, 'x[0] <= 0.368\nsquared_error =
0.222\nsamples = 3\nvalue = 6.333'),
 Text(0.16205041003113838, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.16298056485104806, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.16751506959810772, 0.703125, 'x[1] <= 0.328\nsquared_error =
1.863\nsamples = 31\nvalue = 8.516'),
 Text(0.1670499921881529, 0.671875, 'x[3] <= 0.298\nsquared_error =
1.712\nsamples = 30\nvalue = 8.433'),
 Text(0.16391071967095774, 0.640625, 'x[1] <= 0.285\nsquared_error =
1.396\nsamples = 19\nvalue = 8.842'),
 Text(0.16158533262118355, 0.609375, 'x[5] <= 0.049\nsquared_error =
1.859\nsamples = 8\nvalue = 8.125'),
 Text(0.16112025521122872, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.16205041003113838, 0.578125, 'x[6] <= 0.082\nsquared_error =
1.551\nsamples = 7\nvalue = 7.857'),
 Text(0.16112025521122872, 0.546875, 'x[4] <= 0.074\nsquared_error =
1.188\nsamples = 4\nvalue = 7.25'),
 Text(0.16065517780127386, 0.515625, 'squared_error = 0.0\nsamples =
```

```
1\nvalue = 9.0'),
 Text(0.16158533262118355, 0.515625, 'x[0] <= 0.367\nsquared_error =
0.222\nsamples = 3\nvalue = 6.667'),
 Text(0.16112025521122872, 0.484375, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.16205041003113838, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.16298056485104806, 0.546875, 'x[3] <= 0.251\nsquared_error =
0.889\nsamples = 3\nvalue = 8.667'),
 Text(0.16251548744109323, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.1634456422610029, 0.515625, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
 Text(0.1662361067207319, 0.609375, 'x[5] <= 0.067\nsquared_error =
0.413\nsamples = 11\nvalue = 9.364'),
 Text(0.16530595190082223, 0.578125, 'x[1] <= 0.292\nsquared_error =
0.234\nsamples = 8\nvalue = 9.625'),
 Text(0.1648408744908674, 0.546875, 'x[5] <= 0.066\nsquared_error =
0.188\nsamples = 4\nvalue = 9.25'),
 Text(0.16437579708091257, 0.515625, 'squared_error = 0.0\nsamples =
3\nvalue = 9.0'),
 Text(0.16530595190082223, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.16577102931077708, 0.546875, 'squared_error = 0.0\nsamples =
4\nvalue = 10.0'),
 Text(0.1671662615406416, 0.578125, 'x[0] <= 0.388\nsquared_error =
0.222\nsamples = 3\nvalue = 8.667'),
 Text(0.16670118413068674, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.16763133895059643, 0.546875, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.17018926470534804, 0.640625, 'x[3] <= 0.31\nsquared_error =
1.471\nsamples = 11\nvalue = 7.727'),
 Text(0.1685614937705061, 0.609375, 'x[6] <= 0.09\nsquared_error =
0.4\nsamples = 5\nvalue = 7.0'),
 Text(0.16809641636055125, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.16902657118046094, 0.578125, 'x[3] <= 0.307\nsquared_error =
0.188\nsamples = 4\nvalue = 6.75'),
 Text(0.1685614937705061, 0.546875, 'squared_error = 0.0\nsamples = 3\
nvalue = 7.0'),
 Text(0.16949164859041577, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.17181703564018996, 0.609375, 'x[5] <= 0.075\nsquared_error =
1.556\nsamples = 6\nvalue = 8.333'),
 Text(0.17088688082028028, 0.578125, 'x[5] <= 0.065\nsquared_error =
0.889\nsamples = 3\nvalue = 9.333'),
 Text(0.17042180341032545, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
```

```
 Text(0.1713519582302351, 0.546875, 'squared_error = 0.0\nsamples = 2\
nvalue = 10.0'),
 Text(0.17274719046009962, 0.578125, 'x[4] <= 0.128\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.1722821130501448, 0.546875, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.17321226787005448, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.16798014700806255, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.17274719046009962, 0.734375, 'x[5] <= 0.06\nsquared_error =
0.765\nsamples = 9\nvalue = 7.111'),
 Text(0.1722821130501448, 0.703125, 'squared_error = 0.0\nsamples = 3\
nvalue = 8.0'),
 Text(0.17321226787005448, 0.703125, 'x[2] <= 0.113\nsquared_error =
0.556\nsamples = 6\nvalue = 6.667'),
 Text(0.17274719046009962, 0.671875, 'x[6] <= 0.08\nsquared_error =
0.5\nsamples = 4\nvalue = 7.0'),
 Text(0.1722821130501448, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.0'),
 Text(0.17321226787005448, 0.640625, 'x[4] <= 0.134\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.17274719046009962, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.1736773452800093, 0.609375, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.1736773452800093, 0.671875, 'squared_error = 0.0\nsamples = 2\
nvalue = 6.0'),
 Text(0.17553765491982865, 0.765625, 'x[4] <= 0.138\nsquared_error =
2.75\nsamples = 8\nvalue = 9.5'),
 Text(0.174607500099919, 0.734375, 'x[0] <= 0.403\nsquared_error =
1.0\nsamples = 2\nvalue = 12.0'),
 Text(0.17414242268996413, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.17507257750987382, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.17646780973973833, 0.734375, 'x[4] <= 0.141\nsquared_error =
0.556\nsamples = 6\nvalue = 8.667'),
 Text(0.17600273232978347, 0.703125, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.17693288714969316, 0.703125, 'x[3] <= 0.336\nsquared_error =
0.222\nsamples = 3\nvalue = 9.333'),
 Text(0.17646780973973833, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.17739796455964799, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.17739796455964799, 0.796875, 'x[5] <= 0.074\nsquared_error =
1.36\nsamples = 5\nvalue = 11.2'),
 Text(0.17693288714969316, 0.765625, 'squared_error = 0.0\nsamples =
```

```
3\nvalue = 12.0'),
 Text(0.17786304196960284, 0.765625, 'x[4] <= 0.144\nsquared_error =
1.0\nsamples = 2\nvalue = 10.0'),
 Text(0.17739796455964799, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.17832811937955767, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.18733899419743263, 0.828125, 'x[2] <= 0.127\nsquared_error =
2.574\nsamples = 49\nvalue = 7.551'),
 Text(0.18449039506145926, 0.796875, 'x[3] <= 0.377\nsquared_error =
1.422\nsamples = 45\nvalue = 7.333'),
 Text(0.18193246930670764, 0.765625, 'x[3] <= 0.375\nsquared_error =
0.84\nsamples = 18\nvalue = 6.778'),
 Text(0.18146739189675282, 0.734375, 'x[5] <= 0.087\nsquared_error =
0.692\nsamples = 17\nvalue = 6.882'),
 Text(0.17949081290444477, 0.703125, 'x[1] <= 0.317\nsquared_error =
0.347\nsamples = 14\nvalue = 6.714'),
 Text(0.17832811937955767, 0.671875, 'x[2] <= 0.117\nsquared_error =
0.222\nsamples = 6\nvalue = 6.333'),
 Text(0.17786304196960284, 0.640625, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
 Text(0.1787931967895125, 0.640625, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.18065350642933184, 0.671875, 'x[0] <= 0.422\nsquared_error =
0.25\nsamples = 8\nvalue = 7.0'),
 Text(0.17972335160942218, 0.640625, 'x[2] <= 0.113\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.17925827419946735, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.180188429019377, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.18158366124924152, 0.640625, 'x[3] <= 0.366\nsquared_error =
0.139\nsamples = 6\nvalue = 6.833'),
 Text(0.1811185838392867, 0.609375, 'squared_error = 0.0\nsamples = 4\
nvalue = 7.0'),
 Text(0.18204873865919635, 0.609375, 'x[2] <= 0.113\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.18158366124924152, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.1825138160691512, 0.578125, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.18344397088906086, 0.703125, 'x[1] <= 0.325\nsquared_error =
1.556\nsamples = 3\nvalue = 7.667'),
 Text(0.18297889347910604, 0.671875, 'x[3] <= 0.363\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.1825138160691512, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.18344397088906086, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.18390904829901572, 0.671875, 'squared_error = 0.0\nsamples =
```

```
1\nvalue = 6.0'),
 Text(0.1823975467166625, 0.734375, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.18704832081621087, 0.765625, 'x[4] <= 0.172\nsquared_error =
1.468\nsamples = 27\nvalue = 7.704'),
 Text(0.18576935793883506, 0.734375, 'x[5] <= 0.1\nsquared_error =
2.472\nsamples = 6\nvalue = 8.833'),
 Text(0.1853042805288802, 0.703125, 'x[5] <= 0.091\nsquared_error =
0.56\nsamples = 5\nvalue = 8.2'),
 Text(0.18483920311892538, 0.671875, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.18576935793883506, 0.671875, 'x[6] <= 0.109\nsquared_error =
0.222\nsamples = 3\nvalue = 7.667'),
 Text(0.1853042805288802, 0.640625, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
 Text(0.1862344353487899, 0.640625, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.1862344353487899, 0.703125, 'squared_error = 0.0\nsamples = 1\
nvalue = 12.0'),
 Text(0.18832728369358664, 0.734375, 'x[1] <= 0.328\nsquared_error =
0.712\nsamples = 21\nvalue = 7.381'),
 Text(0.18716459016869957, 0.703125, 'x[5] <= 0.068\nsquared_error =
0.109\nsamples = 8\nvalue = 8.125'),
 Text(0.18669951275874472, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.1876296675786544, 0.671875, 'squared_error = 0.0\nsamples = 7\
nvalue = 8.0'),
 Text(0.18948997721847374, 0.703125, 'x[5] <= 0.095\nsquared_error =
0.533\nsamples = 13\nvalue = 6.923'),
 Text(0.1885598223985641, 0.671875, 'x[4] <= 0.223\nsquared_error =
0.188\nsamples = 8\nvalue = 7.25'),
 Text(0.18809474498860923, 0.640625, 'x[5] <= 0.091\nsquared_error =
0.122\nsamples = 7\nvalue = 7.143'),
 Text(0.1876296675786544, 0.609375, 'squared_error = 0.0\nsamples = 5\
nvalue = 7.0'),
 Text(0.1885598223985641, 0.609375, 'x[0] <= 0.44\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.18809474498860923, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.18902489980851891, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.18902489980851891, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.19042013203838343, 0.671875, 'x[0] <= 0.433\nsquared_error =
0.64\nsamples = 5\nvalue = 6.4'),
 Text(0.18995505462842857, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.19088520944833826, 0.640625, 'squared_error = 0.0\nsamples =
4\nvalue = 6.0'),
```

```
 Text(0.190187593333406, 0.796875, 'x[1] <= 0.343\nsquared_error =
9.0\nsamples = 4\nvalue = 10.0'),
 Text(0.18972251592345116, 0.765625, 'x[0] <= 0.425\nsquared_error =
0.889\nsamples = 3\nvalue = 8.333'),
 Text(0.18925743851349633, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.190187593333406, 0.734375, 'squared_error = 0.0\nsamples = 2\
nvalue = 9.0'),
 Text(0.19065267074336084, 0.765625, 'squared_error = 0.0\nsamples =
1\nvalue = 15.0'),
 Text(0.266634522278298, 0.890625, 'x[4] <= 0.218\nsquared_error =
4.191\nsamples = 495\nvalue = 8.802'),
 Text(0.23319601286956396, 0.859375, 'x[2] <= 0.122\nsquared_error =
4.961\nsamples = 301\nvalue = 9.229'),
 Text(0.21186206276364983, 0.828125, 'x[4] <= 0.123\nsquared_error =
3.872\nsamples = 212\nvalue = 8.877'),
 Text(0.196582407720285, 0.796875, 'x[3] <= 0.337\nsquared_error =
12.883\nsamples = 14\nvalue = 10.786'),
 Text(0.19483836743295438, 0.765625, 'x[2] <= 0.113\nsquared_error =
5.322\nsamples = 11\nvalue = 9.364'),
 Text(0.19367567390806728, 0.734375, 'x[2] <= 0.098\nsquared_error =
2.025\nsamples = 9\nvalue = 8.556'),
 Text(0.1927455190881576, 0.703125, 'x[3] <= 0.321\nsquared_error =
1.139\nsamples = 6\nvalue = 7.833'),
 Text(0.19228044167820277, 0.671875, 'x[5] <= 0.059\nsquared_error =
0.24\nsamples = 5\nvalue = 7.4'),
 Text(0.19181536426824794, 0.640625, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.1927455190881576, 0.640625, 'x[4] <= 0.115\nsquared_error =
0.222\nsamples = 3\nvalue = 7.667'),
 Text(0.19228044167820277, 0.609375, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.19321059649811245, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.19321059649811245, 0.671875, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.19460582872797694, 0.703125, 'x[6] <= 0.117\nsquared_error =
0.667\nsamples = 3\nvalue = 10.0'),
 Text(0.1941407513180221, 0.671875, 'x[6] <= 0.114\nsquared_error =
0.25\nsamples = 2\nvalue = 10.5'),
 Text(0.19367567390806728, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.19460582872797694, 0.640625, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.1950709061379318, 0.671875, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.19600106095784145, 0.734375, 'x[6] <= 0.117\nsquared_error =
4.0\nsamples = 2\nvalue = 13.0'),
 Text(0.19553598354788662, 0.703125, 'squared_error = 0.0\nsamples =
```

```
1\nvalue = 15.0'),
 Text(0.1964661383677963, 0.703125, 'squared_error = 0.0\nsamples = 1\
nvalue = 11.0'),
 Text(0.19832644800761565, 0.765625, 'x[6] <= 0.14\nsquared_error =
6.0\nsamples = 3\nvalue = 16.0'),
 Text(0.19786137059766082, 0.734375, 'x[4] <= 0.12\nsquared_error =
2.25\nsamples = 2\nvalue = 14.5'),
 Text(0.19739629318770596, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 16.0'),
 Text(0.19832644800761565, 0.703125, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.19879152541757047, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 19.0'),
 Text(0.22714171780701467, 0.796875, 'x[5] <= 0.114\nsquared_error =
2.959\nsamples = 198\nvalue = 8.742'),
 Text(0.2175445002779564, 0.765625, 'x[5] <= 0.048\nsquared_error =
2.474\nsamples = 182\nvalue = 8.582'),
 Text(0.21707942286800158, 0.734375, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.21800957768791127, 0.734375, 'x[6] <= 0.15\nsquared_error =
2.379\nsamples = 181\nvalue = 8.558'),
 Text(0.20480301428296327, 0.703125, 'x[4] <= 0.161\nsquared_error =
2.184\nsamples = 161\nvalue = 8.441'),
 Text(0.19717465473452436, 0.671875, 'x[6] <= 0.113\nsquared_error =
3.039\nsamples = 51\nvalue = 8.98'),
 Text(0.19553598354788662, 0.640625, 'x[3] <= 0.343\nsquared_error =
2.25\nsamples = 2\nvalue = 12.5'),
 Text(0.1950709061379318, 0.609375, 'squared_error = 0.0\nsamples = 1\
nvalue = 14.0'),
 Text(0.19600106095784145, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.1988133259211621, 0.640625, 'x[4] <= 0.158\nsquared_error =
2.545\nsamples = 49\nvalue = 8.837'),
 Text(0.19693121577775113, 0.609375, 'x[5] <= 0.06\nsquared_error =
2.146\nsamples = 45\nvalue = 8.622'),
 Text(0.1964661383677963, 0.578125, 'squared_error = 0.0\nsamples = 3\
nvalue = 7.0'),
 Text(0.19739629318770596, 0.578125, 'x[0] <= 0.468\nsquared_error =
2.098\nsamples = 42\nvalue = 8.738'),
 Text(0.1950273051307485, 0.546875, 'x[4] <= 0.136\nsquared_error =
1.994\nsamples = 40\nvalue = 8.825'),
 Text(0.19168456124669814, 0.515625, 'x[3] <= 0.362\nsquared_error =
4.556\nsamples = 6\nvalue = 9.667'),
 Text(0.19075440642678845, 0.484375, 'x[2] <= 0.085\nsquared_error =
0.222\nsamples = 3\nvalue = 7.667'),
 Text(0.19028932901683362, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.1912194838367433, 0.453125, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
```

```
 Text(0.1926147160666078, 0.484375, 'x[5] <= 0.088\nsquared_error =
0.889\nsamples = 3\nvalue = 11.667'),
 Text(0.19214963865665297, 0.453125, 'squared_error = 0.0\nsamples =
2\nvalue = 11.0'),
 Text(0.19307979347656265, 0.453125, 'squared_error = 0.0\nsamples =
1\nvalue = 13.0'),
 Text(0.1983700490147989, 0.515625, 'x[6] <= 0.125\nsquared_error =
1.395\nsamples = 34\nvalue = 8.676'),
 Text(0.1952889111738481, 0.484375, 'x[0] <= 0.422\nsquared_error =
0.848\nsamples = 21\nvalue = 8.238'),
 Text(0.1940099482964723, 0.453125, 'x[1] <= 0.317\nsquared_error =
0.222\nsamples = 3\nvalue = 9.333'),
 Text(0.19354487088651748, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.19447502570642716, 0.421875, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.19656787405122392, 0.453125, 'x[5] <= 0.095\nsquared_error =
0.719\nsamples = 18\nvalue = 8.056'),
 Text(0.19540518052633682, 0.421875, 'x[5] <= 0.088\nsquared_error =
0.454\nsamples = 14\nvalue = 7.786'),
 Text(0.19447502570642716, 0.390625, 'x[0] <= 0.455\nsquared_error =
0.167\nsamples = 12\nvalue = 8.0'),
 Text(0.1940099482964723, 0.359375, 'x[4] <= 0.153\nsquared_error =
0.083\nsamples = 11\nvalue = 7.909'),
 Text(0.19354487088651748, 0.328125, 'squared_error = 0.0\nsamples =
9\nvalue = 8.0'),
 Text(0.19447502570642716, 0.328125, 'x[4] <= 0.155\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.1940099482964723, 0.296875, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.194940103116382, 0.296875, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.194940103116382, 0.359375, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.1963353353462465, 0.390625, 'x[6] <= 0.117\nsquared_error =
0.25\nsamples = 2\nvalue = 6.5'),
 Text(0.19587025793629168, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.19680041275620133, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
 Text(0.19773056757611102, 0.421875, 'x[2] <= 0.105\nsquared_error =
0.5\nsamples = 4\nvalue = 9.0'),
 Text(0.19726549016615616, 0.390625, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.19819564498606584, 0.390625, 'x[1] <= 0.333\nsquared_error =
0.222\nsamples = 3\nvalue = 8.667'),
 Text(0.19773056757611102, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.19866072239602067, 0.359375, 'squared_error = 0.0\nsamples =
```

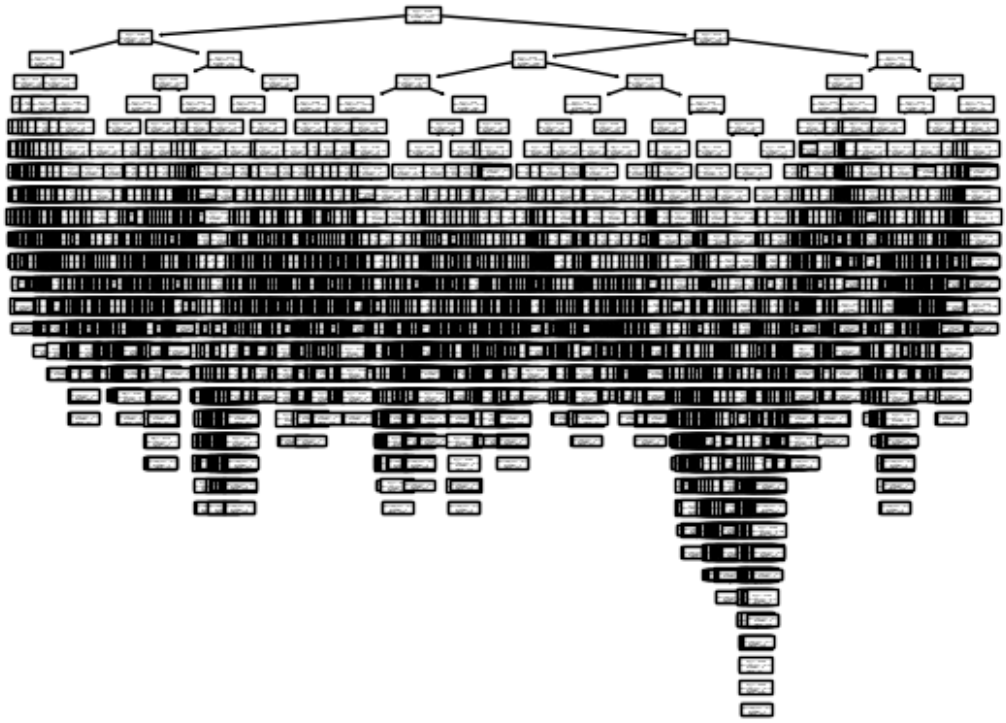
```
2\nvalue = 9.0'),
 Text(0.2014511868557497, 0.484375, 'x[3] <= 0.39\nsquared_error =
1.467\nsamples = 13\nvalue = 9.385'),
 Text(0.20005595462588519, 0.453125, 'x[5] <= 0.069\nsquared_error =
0.98\nsamples = 7\nvalue = 10.143'),
 Text(0.19959087721593036, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 12.0'),
 Text(0.20052103203584004, 0.421875, 'x[6] <= 0.127\nsquared_error =
0.472\nsamples = 6\nvalue = 9.833'),
 Text(0.20005595462588519, 0.390625, 'x[5] <= 0.089\nsquared_error =
0.188\nsamples = 4\nvalue = 10.25'),
 Text(0.19959087721593036, 0.359375, 'squared_error = 0.0\nsamples =
3\nvalue = 10.0'),
 Text(0.20052103203584004, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.20098610944579487, 0.390625, 'squared_error = 0.0\nsamples =
2\nvalue = 9.0'),
 Text(0.2028464190856142, 0.453125, 'x[0] <= 0.445\nsquared_error =
0.583\nsamples = 6\nvalue = 8.5'),
 Text(0.20238134167565938, 0.421875, 'x[6] <= 0.135\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.20191626426570453, 0.390625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.2028464190856142, 0.390625, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.20331149649556904, 0.421875, 'squared_error = 0.0\nsamples =
4\nvalue = 9.0'),
 Text(0.19976528124466342, 0.546875, 'x[6] <= 0.131\nsquared_error =
1.0\nsamples = 2\nvalue = 7.0'),
 Text(0.1993002038347086, 0.515625, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.20023035865461825, 0.515625, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.2006954360645731, 0.609375, 'x[3] <= 0.369\nsquared_error =
0.688\nsamples = 4\nvalue = 11.25'),
 Text(0.20023035865461825, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.20116051347452793, 0.578125, 'x[1] <= 0.34\nsquared_error =
0.222\nsamples = 3\nvalue = 11.667'),
 Text(0.2006954360645731, 0.546875, 'squared_error = 0.0\nsamples = 2\
nvalue = 12.0'),
 Text(0.20162559088448276, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.21243137383140218, 0.671875, 'x[0] <= 0.392\nsquared_error =
1.591\nsamples = 110\nvalue = 8.191'),
 Text(0.20674144239398598, 0.640625, 'x[3] <= 0.398\nsquared_error =
0.25\nsamples = 4\nvalue = 6.5'),
 Text(0.20627636498403112, 0.609375, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
```

```
 Text(0.2072065198039408, 0.609375, 'x[2] <= 0.117\nsquared_error =
0.222\nsamples = 3\nvalue = 6.667'),
 Text(0.20674144239398598, 0.578125, 'squared_error = 0.0\nsamples =
2\nvalue = 7.0'),
 Text(0.20767159721389564, 0.578125, 'squared_error = 0.0\nsamples =
1\nvalue = 6.0'),
 Text(0.21812130526881837, 0.640625, 'x[6] <= 0.129\nsquared_error =
1.529\nsamples = 106\nvalue = 8.255'),
 Text(0.21153755318414522, 0.609375, 'x[0] <= 0.472\nsquared_error =
1.31\nsamples = 44\nvalue = 7.909'),
 Text(0.20860175203380532, 0.578125, 'x[6] <= 0.113\nsquared_error =
1.441\nsamples = 36\nvalue = 8.056'),
 Text(0.20813667462385047, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.20906682944376015, 0.546875, 'x[4] <= 0.204\nsquared_error =
1.228\nsamples = 35\nvalue = 7.971'),
 Text(0.20645076901276418, 0.515625, 'x[5] <= 0.076\nsquared_error =
1.229\nsamples = 27\nvalue = 7.741'),
 Text(0.20424165131547872, 0.484375, 'x[4] <= 0.165\nsquared_error =
1.673\nsamples = 7\nvalue = 8.571'),
 Text(0.2037765739055239, 0.453125, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.20470672872543355, 0.453125, 'x[3] <= 0.368\nsquared_error =
0.96\nsamples = 5\nvalue = 9.2'),
 Text(0.20424165131547872, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.2051718061353884, 0.421875, 'x[3] <= 0.375\nsquared_error =
0.188\nsamples = 4\nvalue = 8.75'),
 Text(0.20470672872543355, 0.390625, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.20563688354534324, 0.390625, 'squared_error = 0.0\nsamples =
3\nvalue = 9.0'),
 Text(0.20865988671004967, 0.484375, 'x[0] <= 0.447\nsquared_error =
0.747\nsamples = 20\nvalue = 7.45'),
 Text(0.20749719318516258, 0.453125, 'x[3] <= 0.444\nsquared_error =
0.667\nsamples = 9\nvalue = 7.0'),
 Text(0.20703211577520775, 0.421875, 'x[4] <= 0.182\nsquared_error =
0.188\nsamples = 8\nvalue = 6.75'),
 Text(0.2065670383652529, 0.390625, 'squared_error = 0.0\nsamples = 5\
nvalue = 7.0'),
 Text(0.20749719318516258, 0.390625, 'x[2] <= 0.117\nsquared_error =
0.222\nsamples = 3\nvalue = 6.333'),
 Text(0.20703211577520775, 0.359375, 'squared_error = 0.0\nsamples =
2\nvalue = 6.0'),
 Text(0.2079622705951174, 0.359375, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.2079622705951174, 0.421875, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.20982258023493677, 0.453125, 'x[3] <= 0.439\nsquared_error =
```

```
0.512\nsamples = 11\nvalue = 7.818'),
 Text(0.20935750282498192, 0.421875, 'x[0] <= 0.453\nsquared_error =
0.444\nsamples = 9\nvalue = 8.0'),
 Text(0.2088924254150271, 0.390625, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.20982258023493677, 0.390625, 'x[6] <= 0.125\nsquared_error =
0.359\nsamples = 8\nvalue = 7.875'),
 Text(0.2088924254150271, 0.359375, 'x[1] <= 0.352\nsquared_error =
0.222\nsamples = 6\nvalue = 7.667'),
 Text(0.20842734800507226, 0.328125, 'x[2] <= 0.107\nsquared_error =
0.222\nsamples = 3\nvalue = 7.333'),
 Text(0.2079622705951174, 0.296875, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.2088924254150271, 0.296875, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.20935750282498192, 0.328125, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.21075273505484643, 0.359375, 'x[4] <= 0.183\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.2102876576448916, 0.328125, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.21121781246480126, 0.328125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.2102876576448916, 0.421875, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.2116828898747561, 0.515625, 'x[4] <= 0.206\nsquared_error =
0.438\nsamples = 8\nvalue = 8.75'),
 Text(0.21121781246480126, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 10.0'),
 Text(0.21214796728471094, 0.484375, 'x[3] <= 0.442\nsquared_error =
0.245\nsamples = 7\nvalue = 8.571'),
 Text(0.2116828898747561, 0.453125, 'x[5] <= 0.084\nsquared_error =
0.188\nsamples = 4\nvalue = 8.25'),
 Text(0.21121781246480126, 0.421875, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.21214796728471094, 0.421875, 'squared_error = 0.0\nsamples =
3\nvalue = 8.0'),
 Text(0.21261304469466577, 0.453125, 'squared_error = 0.0\nsamples =
3\nvalue = 9.0'),
 Text(0.21447335433448514, 0.578125, 'x[5] <= 0.111\nsquared_error =
0.188\nsamples = 8\nvalue = 7.25'),
 Text(0.21400827692453028, 0.546875, 'x[4] <= 0.195\nsquared_error =
0.122\nsamples = 7\nvalue = 7.143'),
 Text(0.21354319951457545, 0.515625, 'x[1] <= 0.363\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.21307812210462063, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.21400827692453028, 0.484375, 'squared_error = 0.0\nsamples =
1\nvalue = 7.0'),
```

```
 Text(0.21447335433448514, 0.515625, 'squared_error = 0.0\nsamples =
5\nvalue = 7.0'),
 Text(0.21493843174443997, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.22470505735349153, 0.609375, 'x[4] <= 0.215\nsquared_error =
1.54\nsamples = 62\nvalue = 8.5'),
 Text(0.2242399799435367, 0.578125, 'x[4] <= 0.163\nsquared_error =
1.361\nsamples = 61\nvalue = 8.443'),
 Text(0.22377490253358187, 0.546875, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.22470505735349153, 0.546875, 'x[5] <= 0.106\nsquared_error =
1.273\nsamples = 60\nvalue = 8.4'),
 Text(0.220519360663898, 0.515625, 'x[3] <= 0.441\nsquared_error =
1.25\nsamples = 52\nvalue = 8.519'),
 Text(0.21691501073674802, 0.484375, 'x[4] <= 0.169\nsquared_error =
1.12\nsamples = 25\nvalue = 8.2'),
 Text(0.21470589303946255, 0.453125, 'x[2] <= 0.113\nsquared_error =
1.36\nsamples = 10\nvalue = 8.8'),
 Text(0.21307812210462063, 0.421875, 'x[0] <= 0.453\nsquared_error =
1.04\nsamples = 5\nvalue = 9.6'),
 Text(0.21214796728471094, 0.390625, 'x[0] <= 0.438\nsquared_error =
0.222\nsamples = 3\nvalue = 10.333'),
 Text(0.2116828898747561, 0.359375, 'squared_error = 0.0\nsamples = 2\
nvalue = 10.0'),
 Text(0.21261304469466577, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 11.0'),
 Text(0.21400827692453028, 0.390625, 'x[5] <= 0.103\nsquared_error =
0.25\nsamples = 2\nvalue = 8.5'),
 Text(0.21354319951457545, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.21447335433448514, 0.359375, 'squared_error = 0.0\nsamples =
1\nvalue = 9.0'),
 Text(0.21633366397430448, 0.421875, 'x[5] <= 0.094\nsquared_error =
0.4\nsamples = 5\nvalue = 8.0'),
 Text(0.21586858656434962, 0.390625, 'x[5] <= 0.082\nsquared_error =
0.188\nsamples = 4\nvalue = 7.75'),
 Text(0.2154035091543948, 0.359375, 'squared_error = 0.0\nsamples = 2\
nvalue = 8.0'),
 Text(0.21633366397430448, 0.359375, 'x[0] <= 0.45\nsquared_error =
0.25\nsamples = 2\nvalue = 7.5'),
 Text(0.21586858656434962, 0.328125, 'squared_error = 0.0\nsamples =
1\nvalue = 8.0'),
 Text(0.2167987413842593, 0.328125, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.2167987413842593, 0.390625, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.2191241284340335, 0.453125, 'x[6] <= 0.135\nsquared_error =
0.56\nsamples = 15\nvalue = 7.8'),
 Text(0.21819397361412382, 0.421875, 'x[3] <= 0.439\nsquared_error =
```

```
0.204\nsamples = 7\nvalue = 7.286'),
 Text(0.217728896204169, 0.390625, 'squared_error = 0.0\nsamples = 5\
nvalue = 7.0'),
 Text(0.21865905102407865, 0.390625, 'squared_error = 0.0\nsamples =
2\nvalue = 8.0'),
 Text(0.22005428325394316, 0.421875, 'x[5] <= 0.094\nsquared_error =
0.438\nsamples = 8\nvalue = 8.25'),
 Text(0.21958920584398833, 0.390625, 'x[4] <= 0.175\nsquared_error =
0.245\nsamples = 7\nvalue = 8.429'),
 Text(0.2191241284340335, 0.359375, 'squared_error = 0.0\nsamples = 3\
nvalue = 8.0'),
 ...]
```



```
import pandas as pd

url='https://raw.githubusercontent.com/likarajo/petrol_consumption/
master/data/petrol_consumption.csv'
dataframe=pd.read_csv(url)

dataframe.head()
print(dataframe.shape)
dataframe.describe()
```

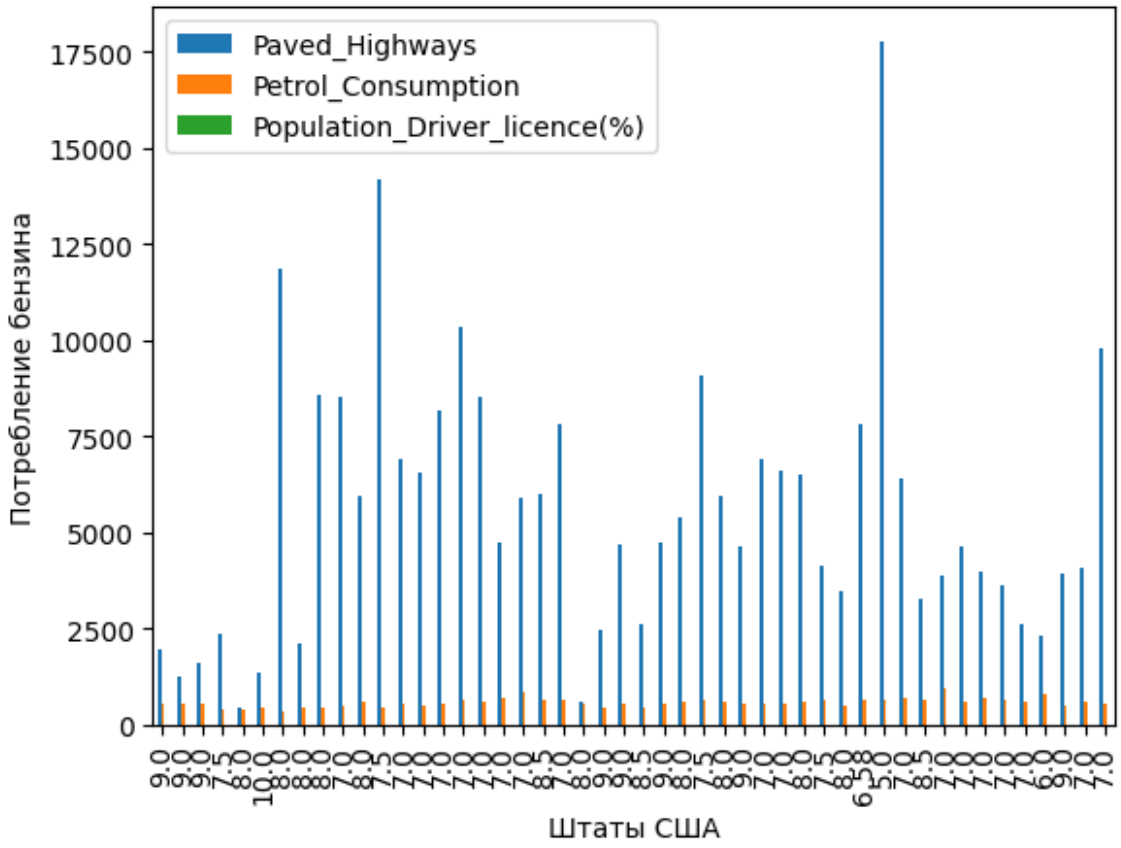
```
(48, 5)
```

```
       Petrol_tax  Average_income  Paved_Highways  \
count   48.000000       48.000000       48.000000
```

```
mean     7.668333     4241.833333     5565.416667
std      0.950770      573.623768     3491.507166
min      5.000000     3063.000000      431.000000
25%      7.000000     3739.000000     3110.250000
50%      7.500000     4298.000000     4735.500000
75%      8.125000     4578.750000     7156.000000
max     10.000000     5342.000000    17782.000000
```

```
       Population_Driver_licence(%)  Petrol_Consumption
count                     48.000000           48.000000
mean                       0.570333          576.770833
std                        0.055470          111.885816
min                        0.451000          344.000000
25%                        0.529750          509.500000
50%                        0.564500          568.500000
75%                        0.595250          632.750000
max                        0.724000          968.000000
```

```
import matplotlib.pyplot as plt
dataframe.plot(x="Petrol_tax",y=['Paved_Highways','Petrol_Consumption'
,'Population_Driver_licence(%)'],kind="bar")
plt.xlabel("Штаты США")
plt.ylabel("Потребление бензина")
plt.show()
```




```
X=dataframe[['Average_income','Population_Driver_licence(%)','Petrol_C
onsumption']]
y=dataframe['Petrol_tax']
print(X)
print(y)
```

	Average_income	Population_Driver_licence(%)	Petrol_Consumption
0	3571	0.525	541
1	4092	0.572	524
2	3865	0.580	561
3	4870	0.529	414
4	4399	0.544	410
5	5342	0.571	457
6	5319	0.451	344
7	5126	0.553	467
8	4447	0.529	464
9	4512	0.552	498
10	4391	0.530	580
11	5126	0.525	471
12	4817	0.574	525
13	4207	0.545	508
14	4332	0.608	566
15	4318	0.586	635

16	4206	0.572	603
17	3718	0.540	714
18	4716	0.724	865
19	4341	0.677	640
20	4593	0.663	649
21	4983	0.602	540
22	4897	0.511	464
23	4258	0.517	547
24	4574	0.551	460
25	3721	0.544	566
26	3448	0.548	577
27	3846	0.579	631
28	4188	0.563	574
29	3601	0.493	534
30	3640	0.518	571
31	3333	0.513	554
32	3063	0.578	577
33	3357	0.547	628
34	3528	0.487	487
35	3802	0.629	644
36	4045	0.566	640
37	3897	0.586	704
38	3635	0.663	648
39	4345	0.672	968
40	4449	0.626	587
41	3656	0.563	699
42	4300	0.603	632
43	3745	0.508	591
44	5215	0.672	782
45	4476	0.571	510
46	4296	0.623	610
47	5002	0.593	524

0	9.00
1	9.00
2	9.00
3	7.50
4	8.00
5	10.00
6	8.00
7	8.00
8	8.00
9	7.00
10	8.00
11	7.50
12	7.00
13	7.00
14	7.00
15	7.00
16	7.00

```
17     7.00
18     7.00
19     8.50
20     7.00
21     8.00
22     9.00
23     9.00
24     8.50
25     9.00
26     8.00
27     7.50
28     8.00
29     9.00
30     7.00
31     7.00
32     8.00
33     7.50
34     8.00
35     6.58
36     5.00
37     7.00
38     8.50
39     7.00
40     7.00
41     7.00
42     7.00
43     7.00
44     6.00
45     9.00
46     7.00
47     7.00
Name: Petrol_tax, dtype: float64
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rando
m_state=0)
```

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)
```

```
DecisionTreeRegressor()
```

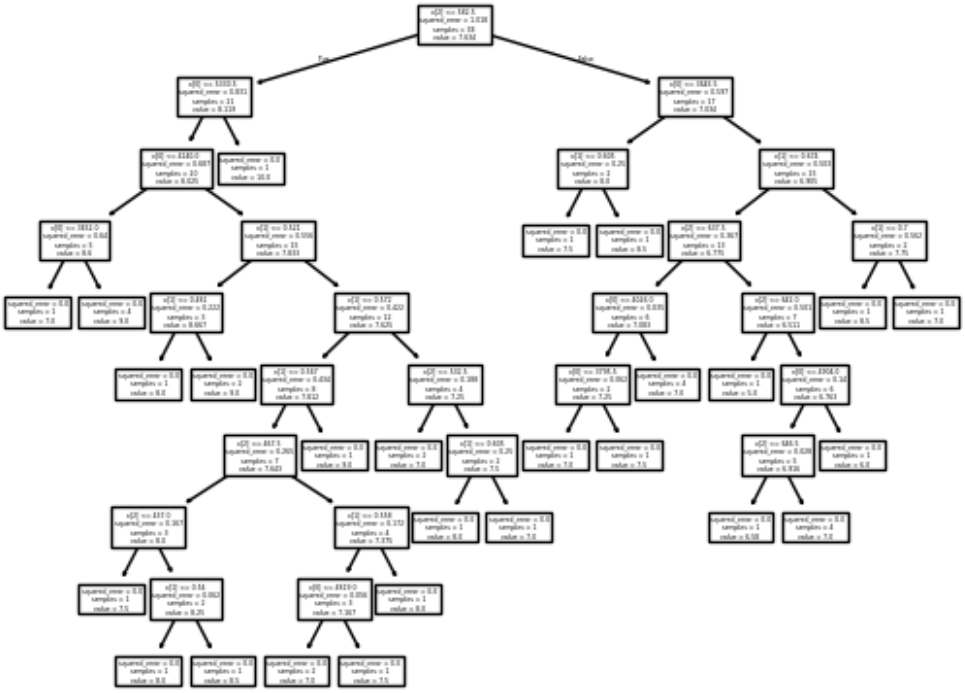
```
from sklearn import tree
tree.plot_tree(regressor)
```

```
[Text(0.47115384615384615, 0.95, 'x[2] <= 582.5\nsquared_error =
1.018\nsamples = 38\nvalue = 7.634'),
 Text(0.22115384615384615, 0.85, 'x[0] <= 5330.5\nsquared_error =
0.831\nsamples = 21\nvalue = 8.119'),
```

```
 Text(0.34615384615384615, 0.8999999999999999, 'True  '),
 Text(0.18269230769230768, 0.75, 'x[0] <= 4140.0\nsquared_error =
0.687\nsamples = 20\nvalue = 8.025'),
 Text(0.07692307692307693, 0.65, 'x[0] <= 3452.0\nsquared_error =
0.64\nsamples = 5\nvalue = 8.6'),
 Text(0.038461538461538464, 0.55, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.11538461538461539, 0.55, 'squared_error = 0.0\nsamples = 4\
nvalue = 9.0'),
 Text(0.28846153846153844, 0.65, 'x[1] <= 0.521\nsquared_error =
0.556\nsamples = 15\nvalue = 7.833'),
 Text(0.19230769230769232, 0.55, 'x[1] <= 0.481\nsquared_error =
0.222\nsamples = 3\nvalue = 8.667'),
 Text(0.15384615384615385, 0.45, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.23076923076923078, 0.45, 'squared_error = 0.0\nsamples = 2\
nvalue = 9.0'),
 Text(0.38461538461538464, 0.55, 'x[1] <= 0.572\nsquared_error =
0.422\nsamples = 12\nvalue = 7.625'),
 Text(0.3076923076923077, 0.45, 'x[1] <= 0.567\nsquared_error = 0.434\
nsamples = 8\nvalue = 7.812'),
 Text(0.2692307692307692, 0.35, 'x[2] <= 467.5\nsquared_error = 0.265\
nsamples = 7\nvalue = 7.643'),
 Text(0.15384615384615385, 0.25, 'x[2] <= 437.0\nsquared_error =
0.167\nsamples = 3\nvalue = 8.0'),
 Text(0.11538461538461539, 0.15, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.5'),
 Text(0.19230769230769232, 0.15, 'x[1] <= 0.54\nsquared_error = 0.062\
nsamples = 2\nvalue = 8.25'),
 Text(0.15384615384615385, 0.05, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.23076923076923078, 0.05, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.5'),
 Text(0.38461538461538464, 0.25, 'x[1] <= 0.558\nsquared_error =
0.172\nsamples = 4\nvalue = 7.375'),
 Text(0.34615384615384615, 0.15, 'x[0] <= 4819.0\nsquared_error =
0.056\nsamples = 3\nvalue = 7.167'),
 Text(0.3076923076923077, 0.05, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
 Text(0.38461538461538464, 0.05, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.5'),
 Text(0.4230769230769231, 0.15, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.34615384615384615, 0.35, 'squared_error = 0.0\nsamples = 1\
nvalue = 9.0'),
 Text(0.46153846153846156, 0.45, 'x[2] <= 532.5\nsquared_error =
0.188\nsamples = 4\nvalue = 7.25'),
 Text(0.4230769230769231, 0.35, 'squared_error = 0.0\nsamples = 2\
nvalue = 7.0'),
```

```
 Text(0.5, 0.35, 'x[1] <= 0.605\nsquared_error = 0.25\nsamples = 2\
nvalue = 7.5'),
 Text(0.46153846153846156, 0.25, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.0'),
 Text(0.5384615384615384, 0.25, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.25961538461538464, 0.75, 'squared_error = 0.0\nsamples = 1\
nvalue = 10.0'),
 Text(0.7211538461538461, 0.85, 'x[0] <= 3645.5\nsquared_error =
0.597\nsamples = 17\nvalue = 7.034'),
 Text(0.5961538461538461, 0.8999999999999999, '  False'),
 Text(0.6153846153846154, 0.75, 'x[1] <= 0.605\nsquared_error = 0.25\
nsamples = 2\nvalue = 8.0'),
 Text(0.5769230769230769, 0.65, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.5'),
 Text(0.6538461538461539, 0.65, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.5'),
 Text(0.8269230769230769, 0.75, 'x[1] <= 0.674\nsquared_error = 0.503\
nsamples = 15\nvalue = 6.905'),
 Text(0.7307692307692307, 0.65, 'x[2] <= 637.5\nsquared_error = 0.367\
nsamples = 13\nvalue = 6.775'),
 Text(0.6538461538461539, 0.55, 'x[0] <= 4026.0\nsquared_error =
0.035\nsamples = 6\nvalue = 7.083'),
 Text(0.6153846153846154, 0.45, 'x[0] <= 3795.5\nsquared_error =
0.062\nsamples = 2\nvalue = 7.25'),
 Text(0.5769230769230769, 0.35, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0'),
 Text(0.6538461538461539, 0.35, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.5'),
 Text(0.6923076923076923, 0.45, 'squared_error = 0.0\nsamples = 4\
nvalue = 7.0'),
 Text(0.8076923076923077, 0.55, 'x[2] <= 642.0\nsquared_error = 0.501\
nsamples = 7\nvalue = 6.511'),
 Text(0.7692307692307693, 0.45, 'squared_error = 0.0\nsamples = 1\
nvalue = 5.0'),
 Text(0.8461538461538461, 0.45, 'x[0] <= 4904.0\nsquared_error = 0.14\
nsamples = 6\nvalue = 6.763'),
 Text(0.8076923076923077, 0.35, 'x[2] <= 646.5\nsquared_error = 0.028\
nsamples = 5\nvalue = 6.916'),
 Text(0.7692307692307693, 0.25, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.58'),
 Text(0.8461538461538461, 0.25, 'squared_error = 0.0\nsamples = 4\
nvalue = 7.0'),
 Text(0.8846153846153846, 0.35, 'squared_error = 0.0\nsamples = 1\
nvalue = 6.0'),
 Text(0.9230769230769231, 0.65, 'x[1] <= 0.7\nsquared_error = 0.562\
nsamples = 2\nvalue = 7.75'),
 Text(0.8846153846153846, 0.55, 'squared_error = 0.0\nsamples = 1\
nvalue = 8.5'),
```

```
 Text(0.9615384615384616, 0.55, 'squared_error = 0.0\nsamples = 1\
nvalue = 7.0')]
```



```
y_pred = regressor.predict(X_test)
df=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
df
```

	Actual	Predicted
29	9.0	9.0
4	8.0	7.5
26	8.0	7.0
30	7.0	9.0
32	8.0	7.0
37	7.0	7.0
34	8.0	9.0
40	7.0	7.0
7	8.0	8.5
10	8.0	7.0

```
from sklearn import metrics
print('Корень средней квадратичной ошибки:',
metrics.root_mean_squared_error(y_test, y_pred))
print('Средняя абсолютная ошибка:',
metrics.mean_absolute_error(y_test, y_pred))
```

```
Корень средней квадратичной ошибки: 0.9219544457292888
Средняя абсолютная ошибка: 0.7
```

```
metrics.mean_absolute_error(y_test, y_pred) / np.average(y) * 100
```

```
np.float64(9.128450336883285)
```