Задание 9

Данные представлены в таблице 1.

Таблица 1. Исходная выборка

x_i	3	10	7	3	5	2	1	9	2	1
y_i	5	1	8	6	9	3	1	8	10	8
c_i	Б	Б	Ч	Ч	К	Б	Б	Ч	К	К

Построим график по данным (рис. 1).

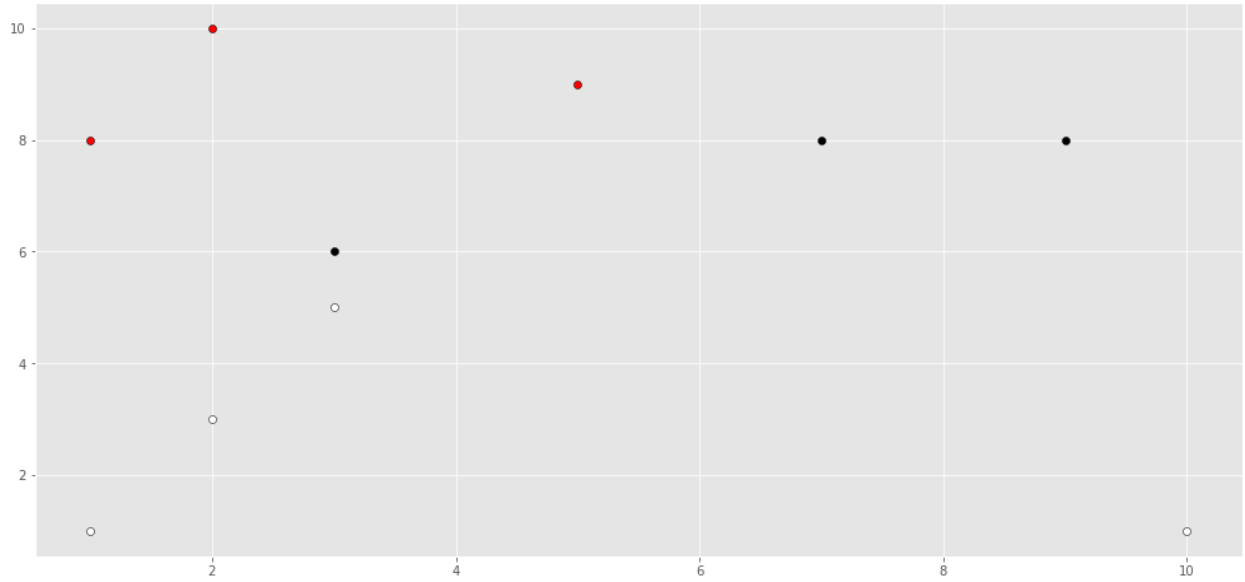

Рисунок 1 – Исходные данные

Проводить действительно полный перебор различных подмножеств вычислительно затратно, поэтому поступлю проще: визуально на графике видно, что если произвести расщепление по признаку y на уровне 5.5, то мы получим чистое подмножество для $y \leq 5.5$. Это расщепление и даст минимальный индекс Джини:

$$\begin{aligned} Gini_A(D) &= \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \\ &= \frac{|D_1|}{|D|}\left(1 - \sum_{j=1}^{3}\left[\frac{|C_{j,D_1}|}{|D_1|}\right]^2\right) \\ &+ \frac{|D_2|}{|D|}\left(1 - \sum_{j=1}^{3}\left[\frac{|C_{j,D_2}|}{|D_2|}\right]^2\right) \end{aligned} \quad (1)$$

$$Gini_y(D) = \frac{4}{10}\left(1 - \left[\frac{4}{4}\right]^2 - \left[\frac{0}{4}\right]^2 - \left[\frac{0}{4}\right]^2\right) + \frac{6}{10}\left(1 - \left[\frac{0}{6}\right]^2 - \left[\frac{3}{6}\right]^2 - \left[\frac{3}{6}\right]^2\right) = 0.3 \quad (2)$$

Далее возможно расщепить выборку y > 0.5 по признаку x на уровнях 2.5, 4 и 6. Ясно, что расщепления на уровнях 2.5 и 6 дадут одинаковый индекс Джини, так как эти расщепления в некотором смысле симметричны. Сравним поэтому только два расщепления – по 2.5 и по 4. Индекс Джини для расщепления по 2.5

$$Gini_x(D) = \frac{2}{6}\left(1 - \left[\frac{0}{2}\right]^2 - \left[\frac{0}{2}\right]^2 - \left[\frac{2}{2}\right]^2\right) + \frac{4}{6}\left(1 - - \left[\frac{0}{4}\right]^2 - \left[\frac{3}{4}\right]^2 - \left[\frac{1}{4}\right]^2\right) = 0.25 \quad (3)$$

Индекс Джини для расщепления по 4:

$$Gini_x(D) = \frac{3}{6}\left(1 - \left[\frac{0}{3}\right]^2 - \left[\frac{1}{3}\right]^2 - \left[\frac{2}{3}\right]^2\right) + \frac{3}{6}\left(1 - - \left[\frac{0}{3}\right]^2 - \left[\frac{2}{3}\right]^2 - \left[\frac{1}{3}\right]^2\right) = 0.(4) \quad (4)$$

Выбираем расщепление по 2.5, так как индекс Джини меньше. Теперь вновь производим расщепление по y, но уже на уровне 8.5. В данном случае получаются чистые подмножества и $Gini_y(D) = 0$. Все указанные расщепления показаны на рис 2.

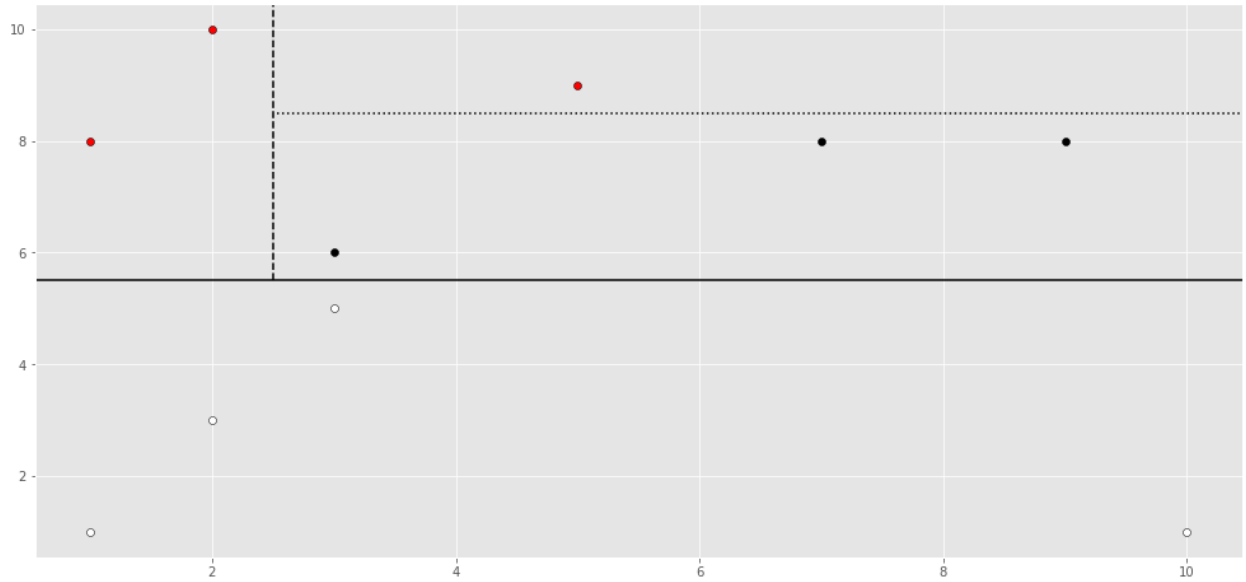

Рисунок 2 – Выборка с показанными расщеплениями

Дерево решений изображено на рис. 3а. На рис. 3б изображено усеченное до двух уровней дерево решений.

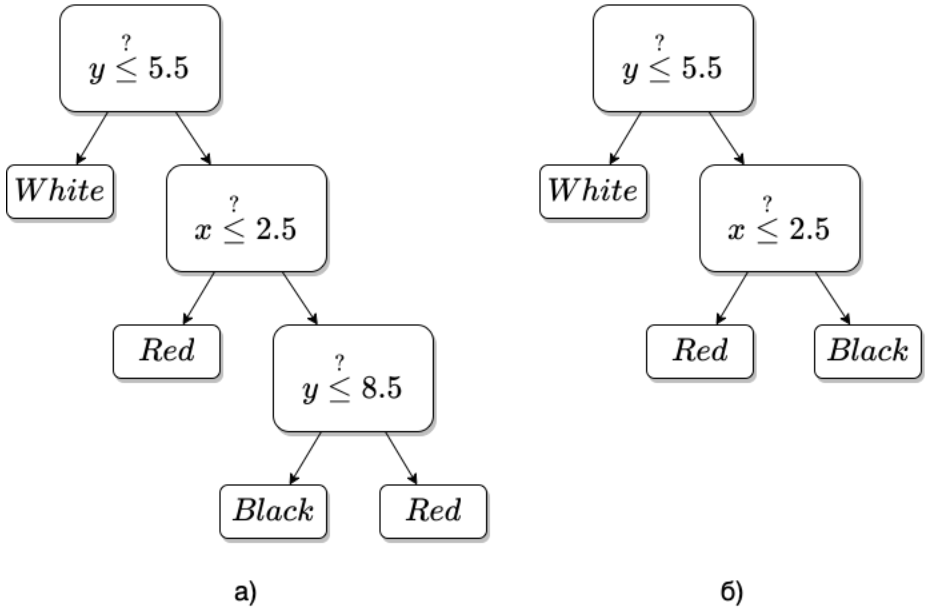

Рисунок 3 – а) Дерево решений; б) усеченное дерево решений

Рассчитаем метрики accuracy и error rate для усеченного дерева. Для начала для каждой точки выберем с помощью усеченного дерева класс (табл. 2).

Таблица 2. Классы точек и предсказанные моделью классы

x_i	3	10	7	3	5	2	1	9	2	1
y_i	5	1	8	6	9	3	1	8	10	8
c_i	Б	Б	Ч	Ч	К	Б	Б	Ч	К	К
$\hat{c}_i$	Б	Б	Ч	Ч	Ч	Б	Б	Ч	К	К

Видно, что усеченное дерево дало одну ошибку, поэтому: $accuracy = 0.9, error\_rate = 0.1$.

Ответ: А) Построенное дерево изображено на рис. 3а, а соответствующие разбиения – на рис. 2. Б) Усеченное дерево изображено на рис. 3б. Значения метрик: $accuracy = 0.9, error\_rate = 0.1$.

Задание 10

Построим график исходных данных и тестовой выборки (рис. 4).

Суть метода в том, чтобы выбрать некоторую r-окрестность исследуемой точки так, чтобы в нее попало k точек, класс которых известны (то есть точек тестовой выборки). Рассмотрим, к примеру, точку (3, 6). На рис. 5 изображены r-окрестности этой точки для k, равного 1 (черный), 3 (синий) и 5 (зеленый). Радиусы окружностей, задающих r-окрестности равны соответственно: $r_1 = 2, r_3 = \sqrt{10}, r_5 = \sqrt{17}$.

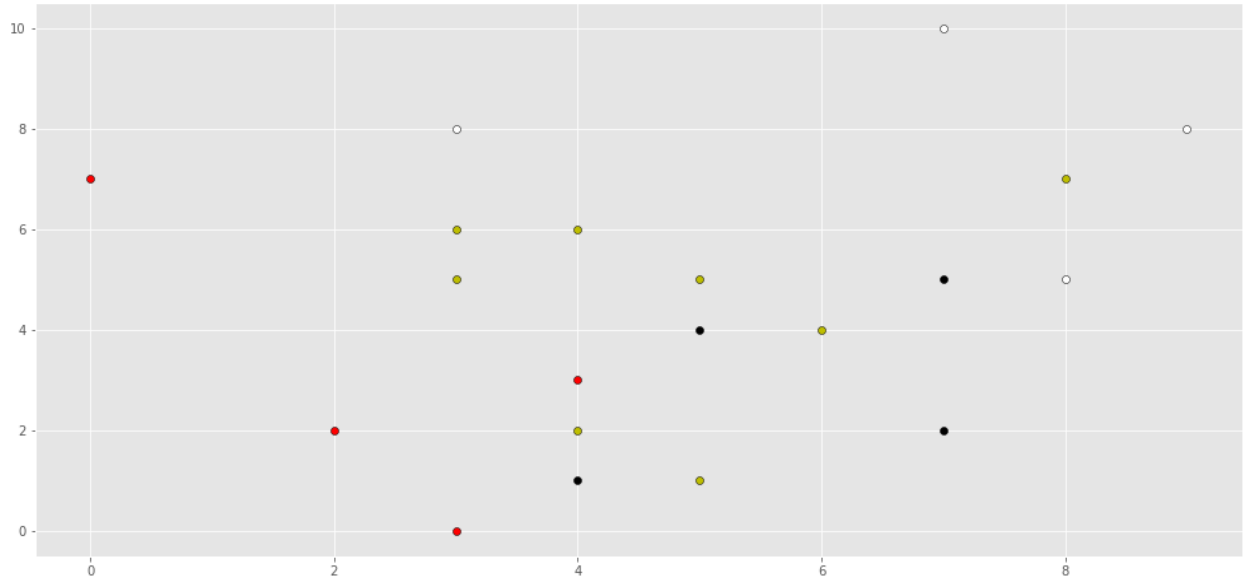

Рисунок 4 – Исходные данные (черный, белый и красный) и тестовая выборка (желтый)

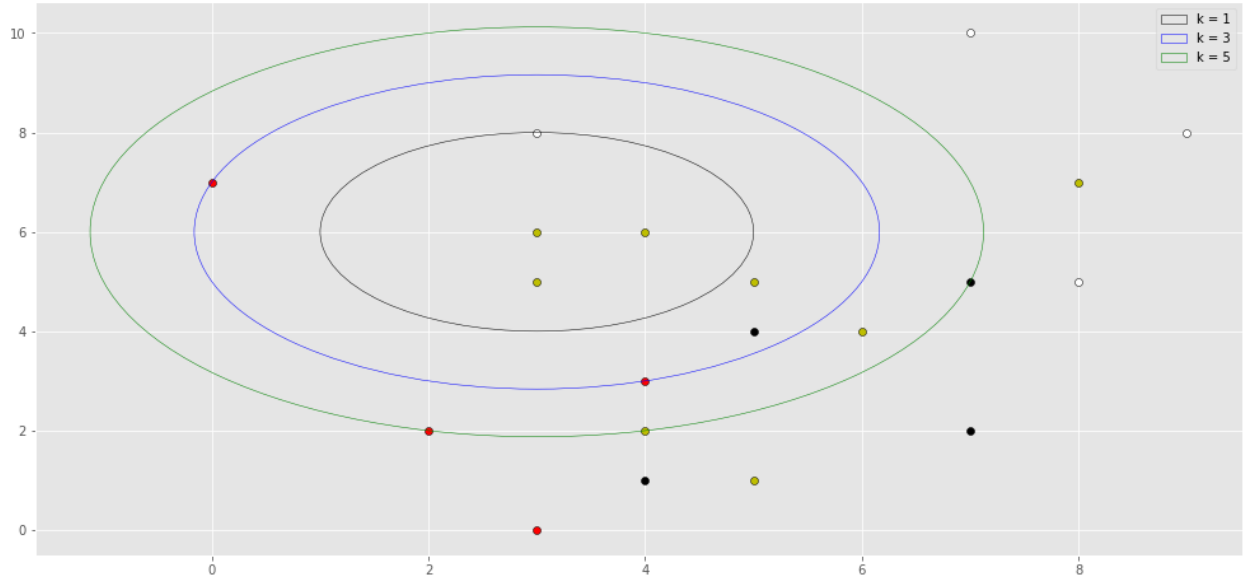

Рисунок 5 – r-окрестности точки (3, 6)

При этом несложно заметить, что при $k = 1$ и $k = 3$ точка классифицируется как белая (при $k = 3$ окружность немного не доходит до красной точки (4, 3)), а при $k = 5$ – как красная.

Несложные вычисления для каждой точки при каждом k приводят к результатам, отраженным в таблице 3.

Таблица 3. Классификация точек при различных k

Точка	$k = 1$	$k = 3$	$k = 5$
(6, 4)	Ч	Ч	Ч
(3, 5)	Ч	Б	К
(8, 7)	Б	Б	Б
(5, 1)	Ч	Ч	Ч

Окончание таблицы 3.

Точка	$k = 1$	$k = 3$	$k = 5$
(4, 2)	Ч	К	К
(3, 6)	Б	Б	К
(5, 5)	Ч	Ч	Ч
(4, 6)	Ч	Б	Ч

Графически результаты представлены на рисунках 6 – 8, где малыми точками обозначены элементы обучающей выборки, а большими – те элементы, классы которых требуется предсказать. Стоит отметить, что масштабы по осям не являются одинаковыми, из-за чего возникает ощущение, что некоторые точки располагаются ближе друг к другу, чем есть на самом деле.

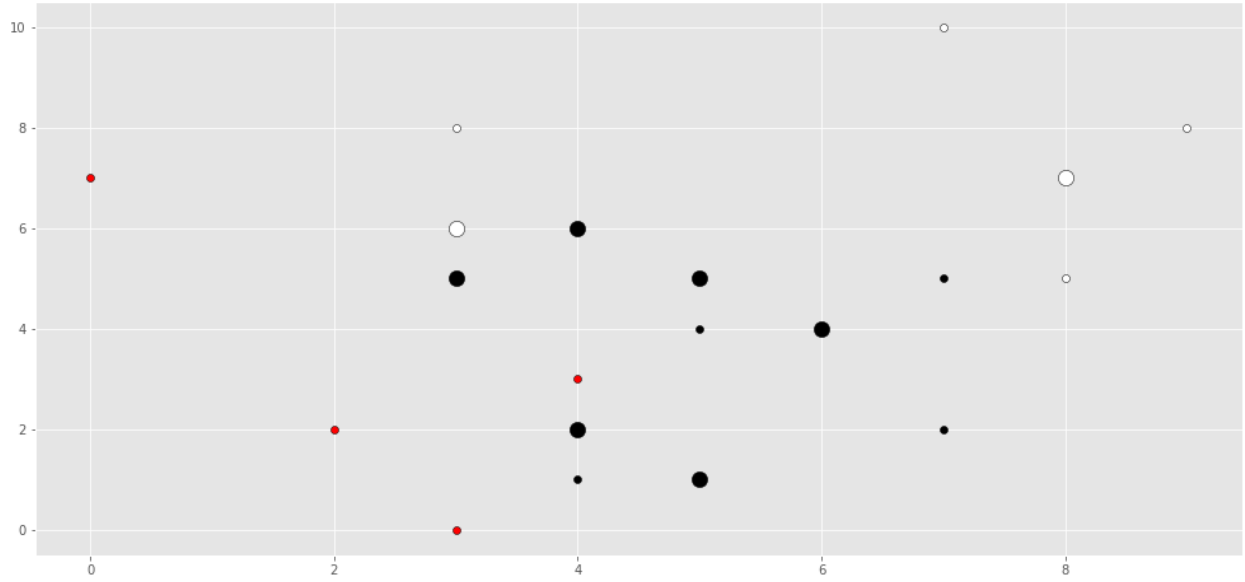

Рисунок 6 – Размеченные точки при $k = 1$

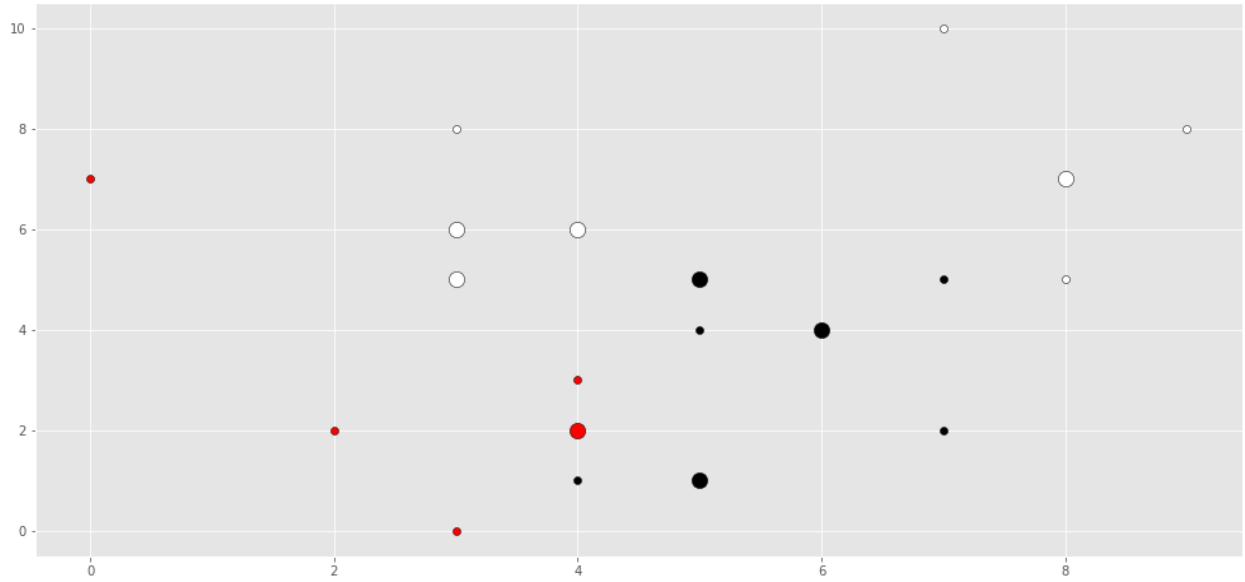

Рисунок 7 – Размеченные точки при $k = 3$

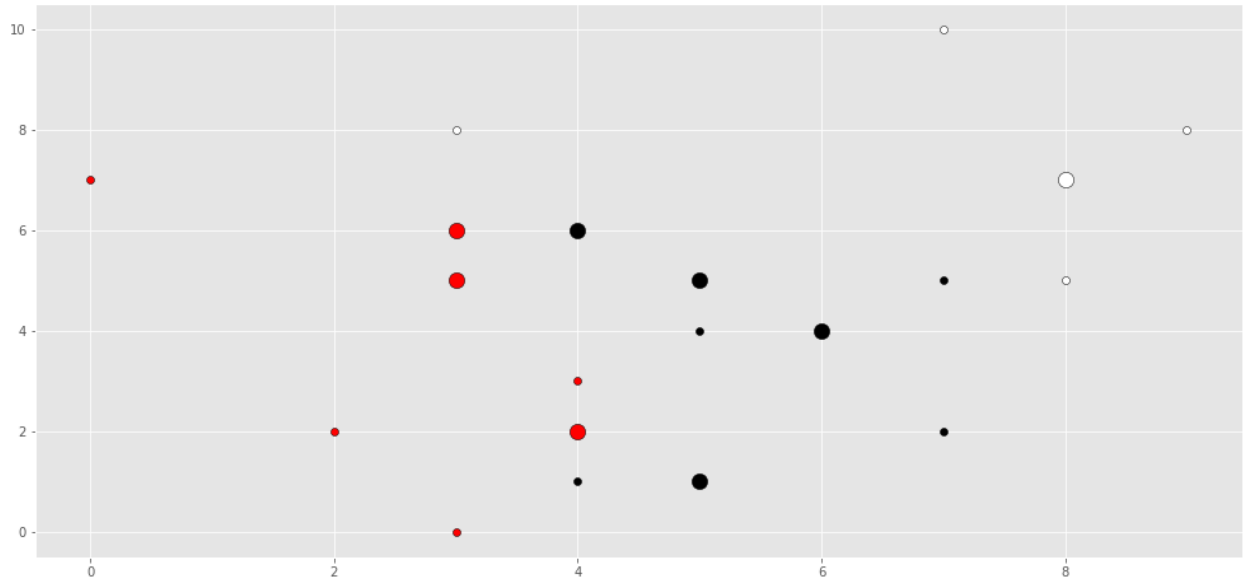

Рисунок 8 – Размеченные точки при $k = 5$

Ответ: см. табл. 3. (столбцы соответствуют различным k из пунктов задания).

Задание 11

Минимизируется количество ошибок на «тестовом» элементе: $LOO(k, X^l) = \sum_{j=1}^{l}[a(x_j, X^l \setminus \{x_j\}, k) \neq y_j] \to min$. Оценка производится на обучающей выборке, так как нам известны классы. Возьмем $k = 1$ и посчитаем, в скольких точках ответ будет дан неверно (табл. 4). Классификация точек производится аналогично заданию 10 (рис. 5).

Таблица 4. Классы точек и предсказанные моделью точки при $k = 1$

x_i	7	3	4	9	0	2	8	7	5	7	3	4
y_i	10	0	1	8	7	2	5	2	4	5	8	3
c_i	Б	К	Ч	Б	К	К	Б	Ч	Ч	Ч	Б	К
$\hat{c}_i$	Б	Ч	К	Б	Б	К	Ч	Ч	К	Б	К	Ч

Из табл. 4 видно, что при $k = 1$ количество ошибок равно 8. Аналогично рассчитывается для $k \in \{3, 5, 7, 9\}$ (таблицы 5 – 8).

Таблица 5. Классы точек и предсказанные моделью точки при $k = 3$

x_i	7	3	4	9	0	2	8	7	5	7	3	4
y_i	10	0	1	8	7	2	5	2	4	5	8	3
c_i	Б	К	Ч	Б	К	К	Б	Ч	Ч	Ч	Б	К
$\hat{c}_i$	Б	К	К	Б	К	К	Ч	Ч	Ч	Ч	Б	Ч

Таблица 6. Классы точек и предсказанные моделью точки при $k = 5$

x_i	7	3	4	9	0	2	8	7	5	7	3	4
y_i	10	0	1	8	7	2	5	2	4	5	8	3
c_i	Б	К	Ч	Б	К	К	Б	Ч	Ч	Ч	Б	К
$\hat{c}_i$	Б	Ч	К	Б	Ч	Ч	Ч	Ч	Ч	Б	Ч	Ч

Таблица 7. Классы точек и предсказанные моделью точки при $k = 7$

x_i	7	3	4	9	0	2	8	7	5	7	3	4
y_i	10	0	1	8	7	2	5	2	4	5	8	3
c_i	Б	К	Ч	Б	К	К	Б	Ч	Ч	Ч	Б	К
$\hat{c}_i$	Б	Ч	Ч	Б	Ч	Ч	Ч	Ч	Ч	Б	Б	Ч

Таблица 8. Классы точек и предсказанные моделью точки при $k = 9$

x_i	7	3	4	9	0	2	8	7	5	7	3	4
y_i	10	0	1	8	7	2	5	2	4	5	8	3
c_i	Б	К	Ч	Б	К	К	Б	Ч	Ч	Ч	Б	К
$\hat{c}_i$	Б	Ч	К	Ч	Б	Ч	Ч	Б	Б	Б	Б	Ч

Для наглядности построим зависимость $LOO(k, X^l)$ от k (рис. 9).

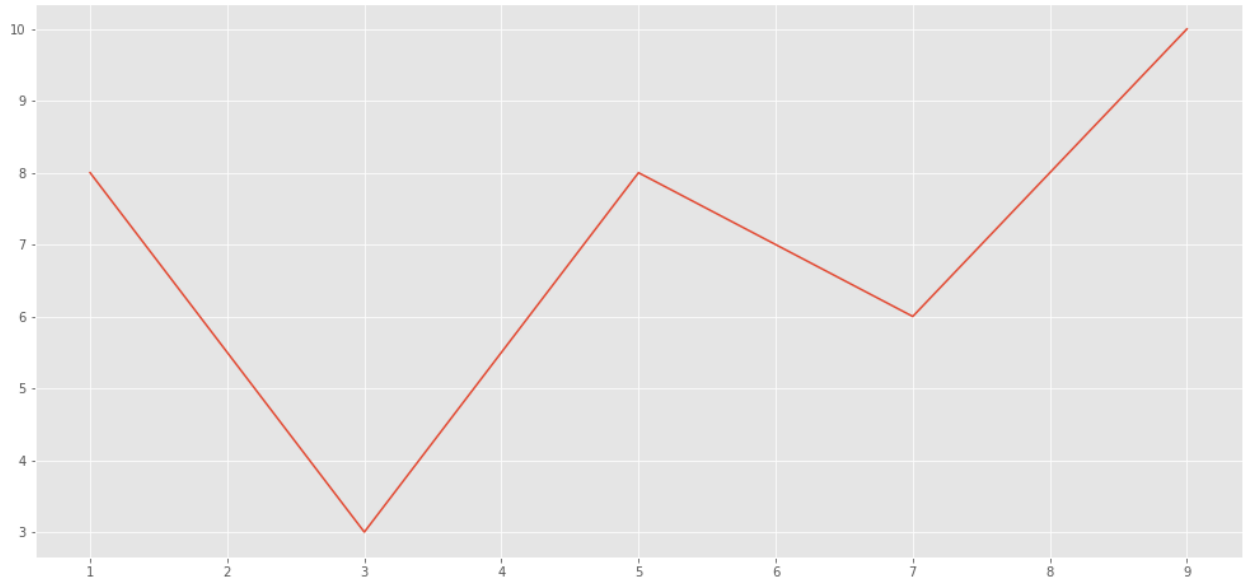

Рисунок 9 – Зависимость $LOO(k, X^l)$ от k

Видно, что минимум достигается при $k = 3$, это и будет оптимальным значением.

Ответ: $k = 3$.

Задание 12

Рассчитаем вероятности $\Pr[x_i|\text{да}]$ и $\Pr[x_i|\text{нет}]$, $i \in \{0,1,2,3\}$ для каждого из элементов тестовой выборки (таблицы 9, 10, 11). Данные вероятности рассчитываются по следующей формуле:

$$\Pr[x_i|\text{да}] = \frac{n_{i,\text{да}}}{n_{\text{да}}}$$
$$\Pr[x_i|\text{нет}] = \frac{n_{i,\text{нет}}}{n_{\text{нет}}} \quad (5)$$

где $n_{\text{да}}$ – кол-во элементов в обучающей выборке с целевой переменной, равной «да», аналогично определяется $n_{\text{нет}}$, $n_{i,\text{да}}$ – количество элементов в обучающей выборке с целевой переменной, равной «да», и признаком x_i, равному тому значению, которое представлено у текущего элемента тестовой выборки, аналогично определяется и $n_{i,\text{нет}}$. Иными словами, это частоты конкретного значения признака x_i среди положительных и отрицательных классов.

Таблица 9. Вероятности для первого объекта тестовой выборки

$\Pr[x_0\|\text{да}]$	$\Pr[x_0\|\text{нет}]$	$\Pr[x_1\|\text{да}]$	$\Pr[x_1\|\text{нет}]$
0.2	0.286	0.2	0.286
$\Pr[x_2\|\text{да}]$	$\Pr[x_2\|\text{нет}]$	$\Pr[x_3\|\text{да}]$	$\Pr[x_3\|\text{нет}]$
0.4	0.286	0.4	0.286

Таблица 10. Вероятности для первого объекта тестовой выборки

$\Pr[x_0\|\text{да}]$	$\Pr[x_0\|\text{нет}]$	$\Pr[x_1\|\text{да}]$	$\Pr[x_1\|\text{нет}]$
0.2	0.286	0.2	0.286
$\Pr[x_2\|\text{да}]$	$\Pr[x_2\|\text{нет}]$	$\Pr[x_3\|\text{да}]$	$\Pr[x_3\|\text{нет}]$
0.4	0.286	0.2	0.429

Таблица 11. Вероятности для первого объекта тестовой выборки

$\Pr[x_0\|\text{да}]$	$\Pr[x_0\|\text{нет}]$	$\Pr[x_1\|\text{да}]$	$\Pr[x_1\|\text{нет}]$
0.4	0.143	0.4	0.143
$\Pr[x_2\|\text{да}]$	$\Pr[x_2\|\text{нет}]$	$\Pr[x_3\|\text{да}]$	$\Pr[x_3\|\text{нет}]$
0.2	0.429	0.4	0.286

Далее рассчитаем апостериорные вероятности (а точнее пропорциональные им величины), используя теорему Байеса и априорные вероятности, рассчитанные как произведения соответствующих вероятностей из таблиц 9 – 11.

Также будем иметь в виду: $\Pr[\text{да}] = 0.41(6)$, $\Pr[\text{нет}] = 0.58(3)$.

Для первого элемента:

$$\Pr[\text{да}|\boldsymbol{X}] \sim \Pr[X|\text{да}] \cdot \Pr[\text{да}] = \Pr[\text{да}] \cdot \prod_{j=0}^{3} \Pr[x_i|\text{да}] =$$
$$= 0.41(6) \cdot 0.2 \cdot 0.2 \cdot 0.4 \cdot 0.4 = 0.002(6)$$
$$\Pr[\text{нет}|\boldsymbol{X}] \sim \Pr[\boldsymbol{X}|\text{нет}] \cdot \Pr[\text{нет}] = \Pr[\text{нет}] \cdot \prod_{j=0}^{3} \Pr[x_i|\text{нет}] = \quad (6)$$
$$= 0.58(3) \cdot 0.286 \cdot 0.286 \cdot 0.286 \approx 0.00389$$

Таким образом, для первого объекта получаем ответ «нет».

Аналогичные вычисления для второго объекта:

$$\Pr[\text{да}|\boldsymbol{X}] \sim \Pr[\boldsymbol{X}|\text{да}] \cdot \Pr[\text{да}] = 0.001(3)$$
$$\Pr[\text{нет}|\boldsymbol{X}] \sim \Pr[\boldsymbol{X}|\text{нет}] \cdot \Pr[\text{нет}] \approx 0.00583 \quad (7)$$

Для второго объекта – ответ «нет».

Аналогично для третьего объекта:

$$\Pr[\text{да}|\boldsymbol{X}] \sim \Pr[\boldsymbol{X}|\text{да}] \cdot \Pr[\text{да}] = 0.005(3)$$
$$\Pr[\text{нет}|\boldsymbol{X}] \sim \Pr[\boldsymbol{X}|\text{нет}] \cdot \Pr[\text{нет}] \approx 0.00146 \quad (8)$$

Для третьего объекта ответ «да».

Ответ: в ближайшее время расторгнуть договор может третий абонент из тестовой выборки.

Задание 13

Данные представлены в таблице 12.

Таблица 12. Исходная выборка

x_i	2	3	4	4	6	8	10
y_i	5	7	8	6	3	2	2

Построим график по данным (рис. 10).

Построим модель регрессии следующего вида:

$$Y = \beta_0 + \beta_1 x + \epsilon(x) \quad (9)$$

Коэффициенты регрессии (их МНК-оценки) вычисляются по следующим формулам:

$$\beta_0 = \bar{y} - \rho^*_{XY} \frac{\sigma^*_Y}{\sigma^*_X} \bar{x}$$
$$\beta_1 = \rho^*_{XY} \frac{\sigma^*_Y}{\sigma^*_X} \quad (10)$$

где ρ^*_{XY} – выборочный коэффициент корреляции Пирсона, σ^*_X – выборочное СКО для значений x, σ^*_Y – выборочное СКО для значений y.

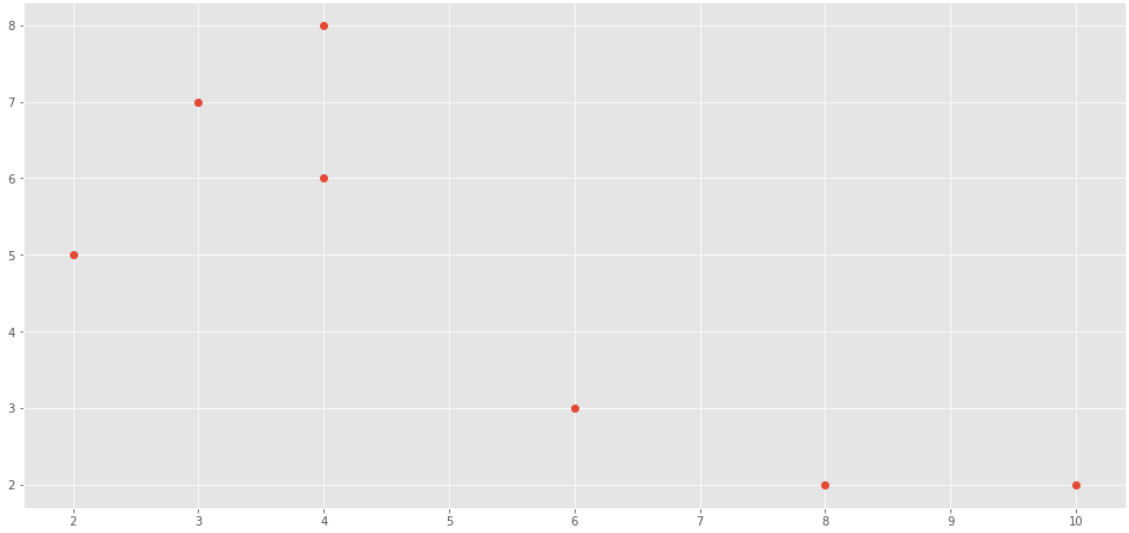

Рисунок 10 – Выборка

Рассчитаем коэффициент корреляции Пирсона (просто подставив значения из исходной выборки):

$$\rho^*_{XY} = \frac{k^*_{XY}}{\sigma^*_X \sigma^*_Y} = \frac{\sum_{j=1}^{7}(x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{7}(x_j - \bar{x})^2 \sum_{j=1}^{7}(y_j - \bar{y})^2}} = \\ = \frac{-33.42857142857143}{7.030545599636733 \cdot 5.9521904731427595} = \\ = -0.798825587720649 \tag{11}$$

где k^*_{XY} – выборочная ковариация значений x и y.

Далее, подставив в (11), несложными вычислениями получаем:

$$\beta_0 = 8.289017341040463$$
$$\beta_1 = -0.676300578034682 \tag{12}$$

Построим график полученной модели на одной картинке с исходными данными (рис. 11).

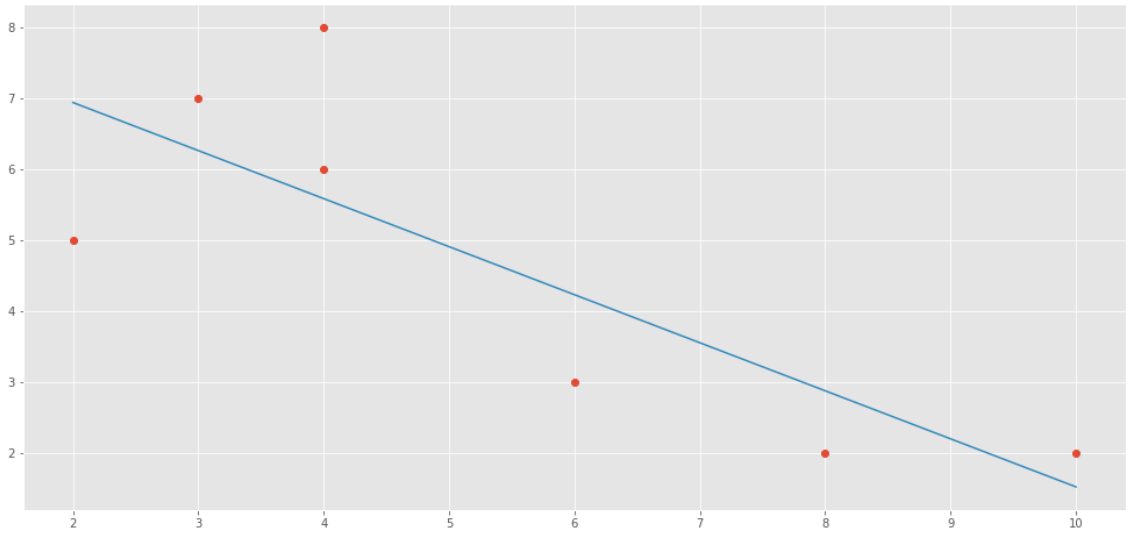

Рисунок 11 – График построенной модели

Ответ: регрессионная модель: $Y = 8.289017341040463 - 0.676300578034682 \cdot x$

Выборочный коэффициент корреляции Пирсона: $\rho^*_{XY} = -0.798825587720649$.

Задание 14

Результат классификации представлен в таблице 13.

Таблица 13. Результат классификации точек

Признак x_1	Признак x_2	Признак x_3	Признак x_4	Класс
0,000	0,562	-0,256	-2,505	-1
0,100	0,005	3,500	-0,270	-1
-0,150	-3,005	-0,013	3,537	-1
1,125	1,655	4,000	-0,235	+1
-2,250	-2,520	0,354	1,752	-1

Краткая суть метода: на каждой итерации на размеченном на текущий момент куске обучающей выборки обучаются разные модели, которые используются для разметки неразмеченной части. Если все модели единогласно проголосовали за конкретную метку, то она присваивается данному элементу.

Ответ: см. табл. 10.

Исходный код на языке Python версии 3.9:

```
from sklearn.tree import DecisionTreeClassifier

import pandas as pd
import numpy as np

df = pd.read_csv("train_14.csv", sep=';')

def voting(*predictions):
    result = []
    voting = [sum([p[i] for p in predictions])
              for i in range(len(predictions[0]))]

    for v in voting:
        if v >= 2:
            result.append(1)
        elif v <= -2:
            result.append(-1)
        else:
            result.append(0)

    return result

train_columns = [
    ['x1', 'x2', 'x3'],
    ['x2', 'x3', 'x4'],
    ['x1', 'x3', 'x4'],
]

last_labeled = 7
first_unknown = 8

while True:
    #
    # Разделяю трейн на размеченные и неразмеченные элементы
```

```
#

X_train     = df.loc[:last_labeled]
X_unlabeled = df.loc[first_unknown:].copy(deep=True)

#
# Отделю таргет
#

y_train = X_train['class']

#
# Обучу модели и сделаю предсказания для неразмеченного трейна
#

models = [DecisionTreeClassifier() for _ in range(3)]
[m.fit(X_train[columns], y_train) for m, columns in zip(models, train_columns)]
predictions = [m.predict(X_unlabeled[columns])
               for m, columns in zip(models, train_columns)]

#
# Голосование моделей. Решения должны быть приняты единогласно
#

voting_result = voting(*predictions)

#
# Если у нас нет единогласного решения ни для какого элемента,
# то я беру первый и для него выставляю метку, набравшую
# большинство голосов
#

if np.all(np.array(voting_result) == 0):
    #
    # Продвигаюсь на один элемент в трейне
    #

    shift = 1
    voting_result[0] = sum([p[0] for p in predictions])
else:
    #
    # Продвигаюсь далее по трейну на столько элементов, по
    # скольким было принято единогласное решение
    #

    shift = sum(list(map(abs, voting_result)))

last_labeled  += shift
first_unknown += shift

#
# Теперь я обновляю метки
#

for idx, i in enumerate(X_unlabeled.index):
    X_unlabeled.at[i, 'class'] = voting_result[idx]

#
# И заново соединяю трейн, после чего сортирую его и переиндксирую
#

df = X_train.append(X_unlabeled)                    \
    .sort_values(by=['class'], ascending=False,
                 key=lambda col: abs(col))          \
    .reset_index()                                  \
    .drop(["index"], axis=1)

#
# Если это был последний неразмеченный элемент, то по нему
# решение принято обязательно, а поэтому выходим
#
```

```
    if len(voting_result) == 1:
        break

#
# Теперь на полностью размеченной выборке обучу классификаторы
#

X_train = df.loc[:last_labeled]
y_train = X_train['class']

models = [DecisionTreeClassifier() for _ in range(3)]
[m.fit(X_train[columns], y_train) for m, columns in zip(models, train_columns)]

#
# Читаю данные, которые нужно классифицировать
#

X_real = pd.read_csv("target_data.csv", sep=';')

#
# Предсказания делаю
#

predict1, predict2, predict3 = [m.predict(X_real[columns])
                                for m, columns in zip(models, train_columns)]

#
# Голосование моделей и печать результата
#

result = []
for i in range(len(predict1)):
    v = predict1[i] + predict2[i] + predict3[i]

    if v > 0:
        result.append(1)
    elif v < 0:
        result.append(-1)
    else:
        result.append(0)

print(result)
```