

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И
РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

Отчет по лабораторной работе №2

“Нелинейная регрессия”

по дисциплине

Статистические основы индукционного вывода

Выполнил:

Якимович И.В.

Студент группы 121703

Проверил:

Русина Н.В.

Минск, 2023

1. Цели

Приобрести навыки построения моделей нелинейной регрессии

2. Задачи

1. Построить модель для нелинейной регрессии
2. Оценить точность модели и получить её уравнение

Ход выполнения:

1. Импортирование необходимых библиотек:

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

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
```

2. Настройка и построение графика зависимости

```

COLUMN_X = "area"
COLUMN_Y = "price"

sns.set_style('darkgrid')
sns.set_palette('Set2')

sns.lineplot(
    x=dataset[COLUMN_X],
    y=dataset[COLUMN_Y],
)
plt.title('График зависимости')
plt.xlabel('Возраст')
plt.ylabel('Площадь')
plt.show()

```



Этот код создает график зависимости между двумя переменными: площадь и цена. Первые две строки определяют имена столбцов данных 'area' и 'price'. Затем устанавливаются стили для графика с помощью функций `sns.set_style('darkgrid')` и `sns.set_palette('Set2')`. Функция `sns.lineplot()` создает линейный график. Далее, устанавливаются заголовок графика с помощью `plt.title`, подписи оси x с помощью `plt.xlabel`, и подписи оси y с помощью `plt.ylabel`. Наконец, функция `plt.show()` отображает созданный график.

3. Построение модели

```

x = np.array(dataset[COLUMN_X]).reshape(-1, 1)
y = np.array(dataset[COLUMN_Y]).reshape(-1, 1)

DEGREES = 2
regression = make_pipeline(PolynomialFeatures(DEGREES), LinearRegression())
regression.fit(x, y)
predictions = regression.predict(x)
mean_squared_error = np.mean((predictions - np.array(y)) ** 2)
print(f'Среднеквадратическая ошибка = {mean_squared_error}')

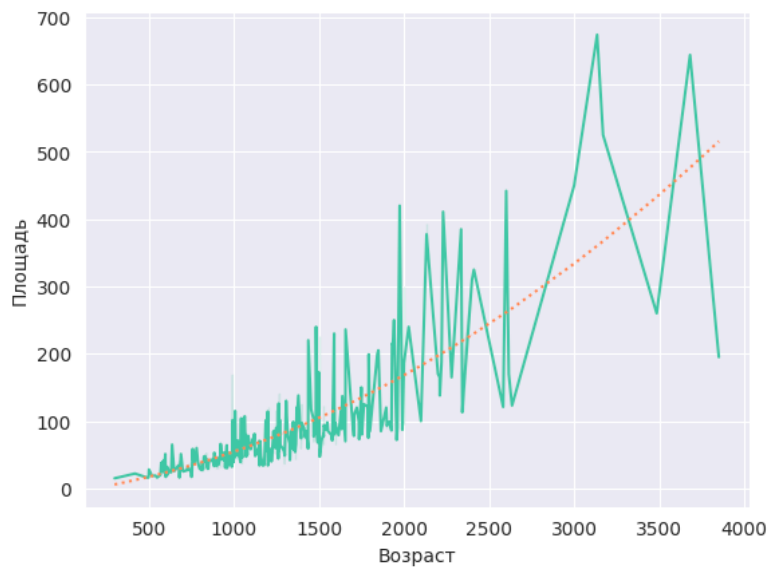
```

Среднеквадратическая ошибка = 2085.6516361398003

```

sns.lineplot(
    x=dataset[COLUMN_X],
    y=dataset[COLUMN_Y],
    linestyle="solid"
)
sns.lineplot(
    x=dataset[COLUMN_X],
    y=predictions.reshape(-1),
    linestyle="dotted"
)
plt.xlabel('Возраст')
plt.ylabel('Площадь')
plt.show()

```



```

x_parameters = np.append(
    regression['linearregression'].intercept_[0],
    regression['linearregression'].coef_[0][1:]
)
x_parameters

array([-7.28715641e+00,  3.57178815e-02,  2.60019879e-05])

```

1. Создаем массив x, содержащий значения столбца 'area' из dataset.
2. Создаем массив y, содержащий значения столбца 'price' из dataset.
3. Они преобразуются в двумерные массивы с помощью reshape() для использования в модели линейной регрессии.
4. Задаем переменную DEGREES для указания степени полинома, используемого в регрессии.
5. Создаем объект regression с использованием make_pipeline, который включает PolynomialFeatures(DEGREES) для создания полиномиальных признаков и LinearRegression() для выполнения регрессии.
6. Применяем метод fit к regression, передавая x и y, чтобы обучить.

7. Создаем прогнозы `predictions` с использованием обученной модели регрессии на основе `x`.
8. Вычисляем среднеквадратическую ошибку `mean_squared_error`, сравнивая прогнозы `predictions` с фактическими значениями `y`.
9. Выводим значение среднеквадратической ошибки на экран.
10. Создаем график, используя `sns.lineplot`, где значения из столбца `'area'` используются для оси `x`, значения из столбца `'price'` используются для оси `y`, а `linestyle="solid"` задает сплошную линию для первого графика.
11. Создаем второй график, используя `sns.lineplot`, где значения из столбца `'area'` используются для оси `x`, значения прогнозов `predictions` используются для оси `y`, а `linestyle="dotted"` задает пунктирную линию для второго графика.
12. извлекаем коэффициенты модели (параметры) и выводится уравнение модели.