

Лабораторная работа 6. Построение пайплайна одномерной регрессии

Репкин Александр, студент 2 курса группы ИВТ-6-о-22-1

Подключение библиотек

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Загрузка данных и разделение на матрицу признаков и зависимую переменную

```
In [2]: !wget https://raw.githubusercontent.com/AlexRepkin/Machine-Learning/main/Act-8-Data
d = pd.read_table("relics.csv", delimiter=";")
d.head()
```

```
--2024-03-27 05:49:57-- https://raw.githubusercontent.com/AlexRepkin/Machine-Learning/main/Act-8-Data%20Analysis/relics.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 442 [text/plain]
Saving to: 'relics.csv'
```

```
relics.csv          100%[=====>]          442  --.-KB/s    in 0s
```

```
2024-03-27 05:49:57 (5.10 MB/s) - 'relics.csv' saved [442/442]
```

```
Out[2]:
```

	Relic	Creation Date	Antiquity Level
0	Ark of Berith	-10	1
1	Scepter of Solomon	-9	2
2	Altar of Zeus	-4	3
3	Climacteric	-3	4
4	Book of Thoth	-3	5

```
In [4]: X = d.iloc[:, 1:2].values
y = d.iloc[:, 2].values
print ("Матрица признаков"); print(X[:5])
print ("Зависимая переменная"); print(y[:5])
```

Матрица признаков

```
[[-10]  
 [ -9]  
 [ -4]  
 [ -3]  
 [ -3]]
```

Зависимая переменная

```
[1 2 3 4 5]
```

Обработка категориальных данных (если требуется)

Разделение выборки на тестовую и тренировочную

```
In [5]: from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/4, random_s
```

Обучение линейной модели регрессии

```
In [6]: from sklearn.linear_model import LinearRegression
```

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

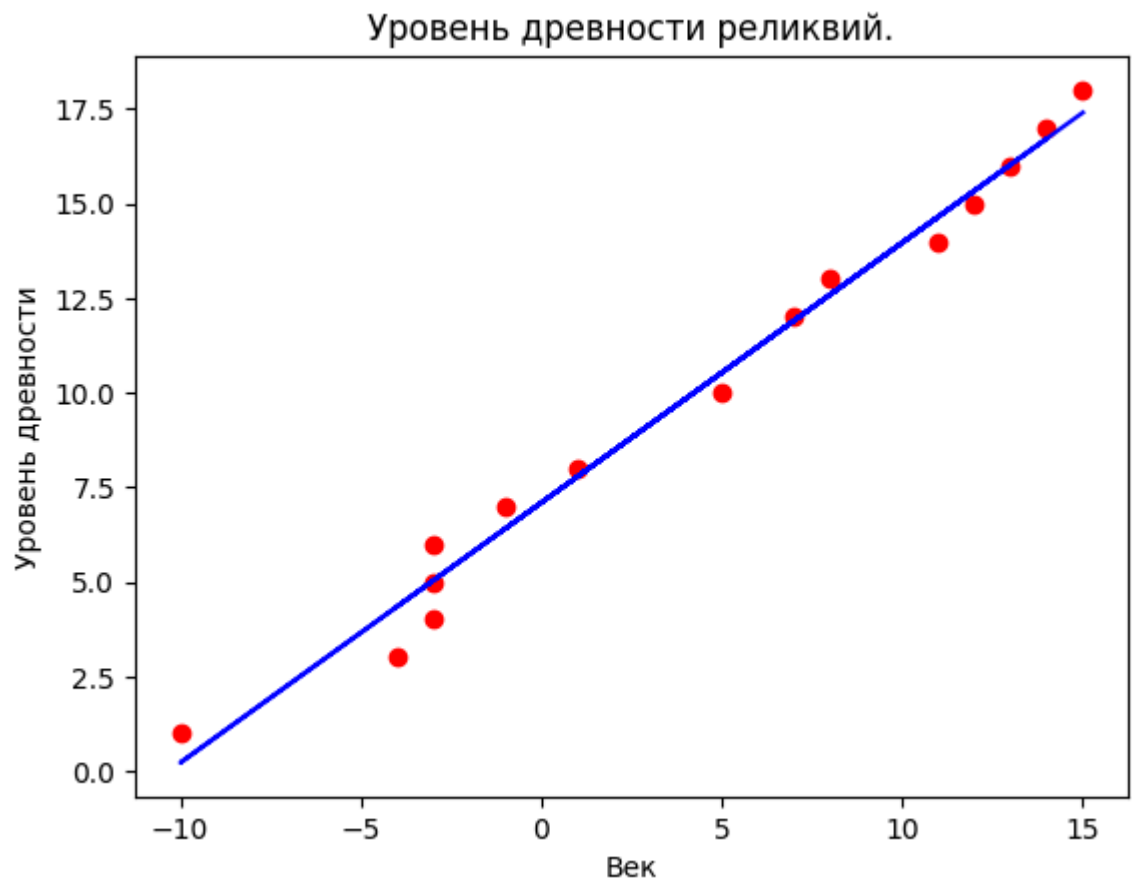
```
Out[6]: ▾ LinearRegression  
LinearRegression()
```

Предсказание, обработка и визуализация результатов

```
In [7]: y_pred = regressor.predict(X_test)  
print(y_pred)
```

```
[18.0908816  0.90503554 18.77831544 10.52910933 11.21654317]
```

```
In [10]: plt.scatter(X_train, y_train, color = 'red')  
plt.plot(X_train, regressor.predict(X_train), color = 'blue')  
plt.title('Уровень древности реликвий.')  
plt.ylabel('Уровень древности')  
plt.xlabel('Век')  
plt.show()
```



```
In [11]: plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Уровень древности реликвий.')
plt.ylabel('Уровень древности')
plt.xlabel('Век')
plt.show()
```

Уровень древности реликвий.

