Лабораторная работа 8. Построение пайплайна полиномиальной регрессии

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

Подключение библиотек

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Загрузка данных и разделение на матрицу признаков и зависимую переменную

In [4]:
```
!wget https://raw.githubusercontent.com/AlexRepkin/Machine-Learning/main/Act-8-Data
d = pd.read_table("relics.csv", delimiter=";")
d.head()
```

```
--2024-03-27 04:28:51--  https://raw.githubusercontent.com/AlexRepkin/Machine-Lear
ning/main/Act-8-Data%20Analysis/relics.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.13
3, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.13
3|: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 04:28:51 (36.8 MB/s) - ‘relics.csv’ saved [442/442]
```

Out[4]:

	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 [6]:
```
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]
```

Обучение модел

Обучение линейной модели

In [7]:
```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)
```

Out[7]:
```
LinearRegression()
```

Обучение полиномиальной модели

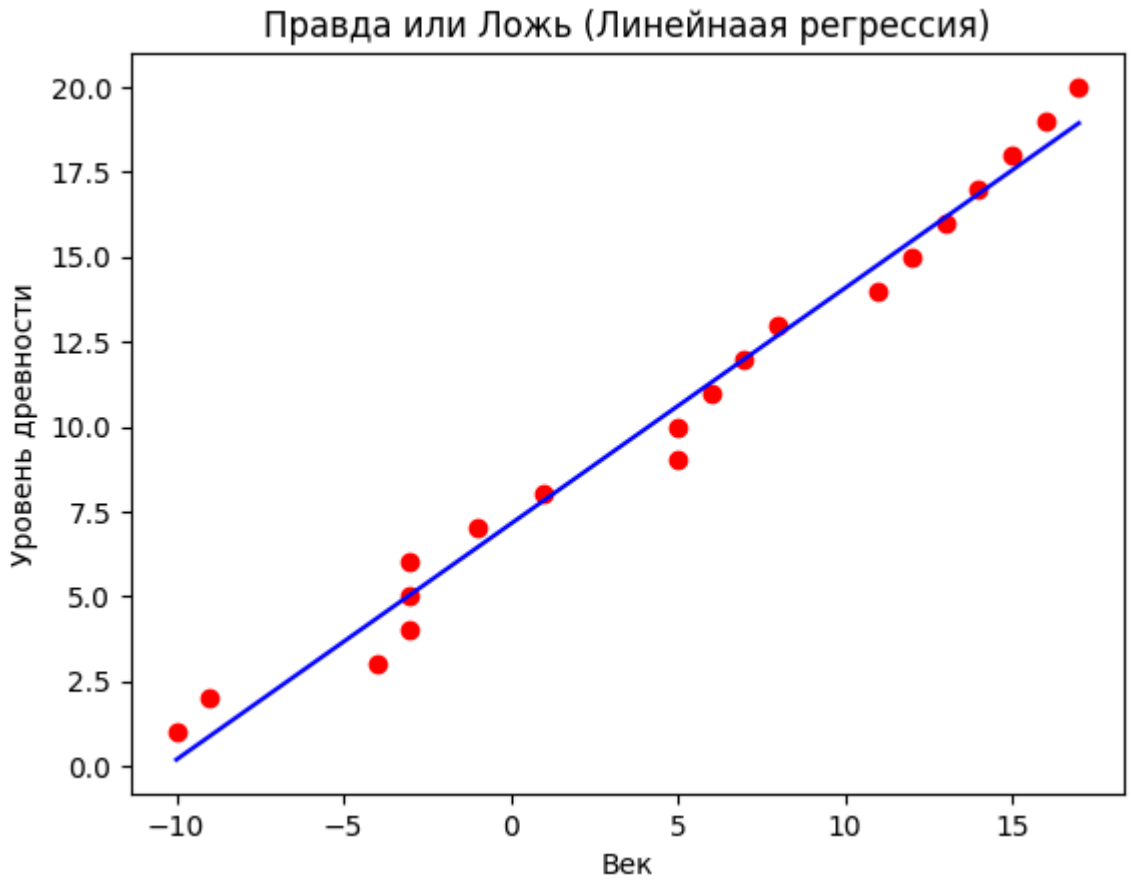
In [8]:
```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 10)
X_poly = poly_reg.fit_transform(X)
poly_reg.fit(X_poly, y)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

Out[8]:
```
LinearRegression()
```

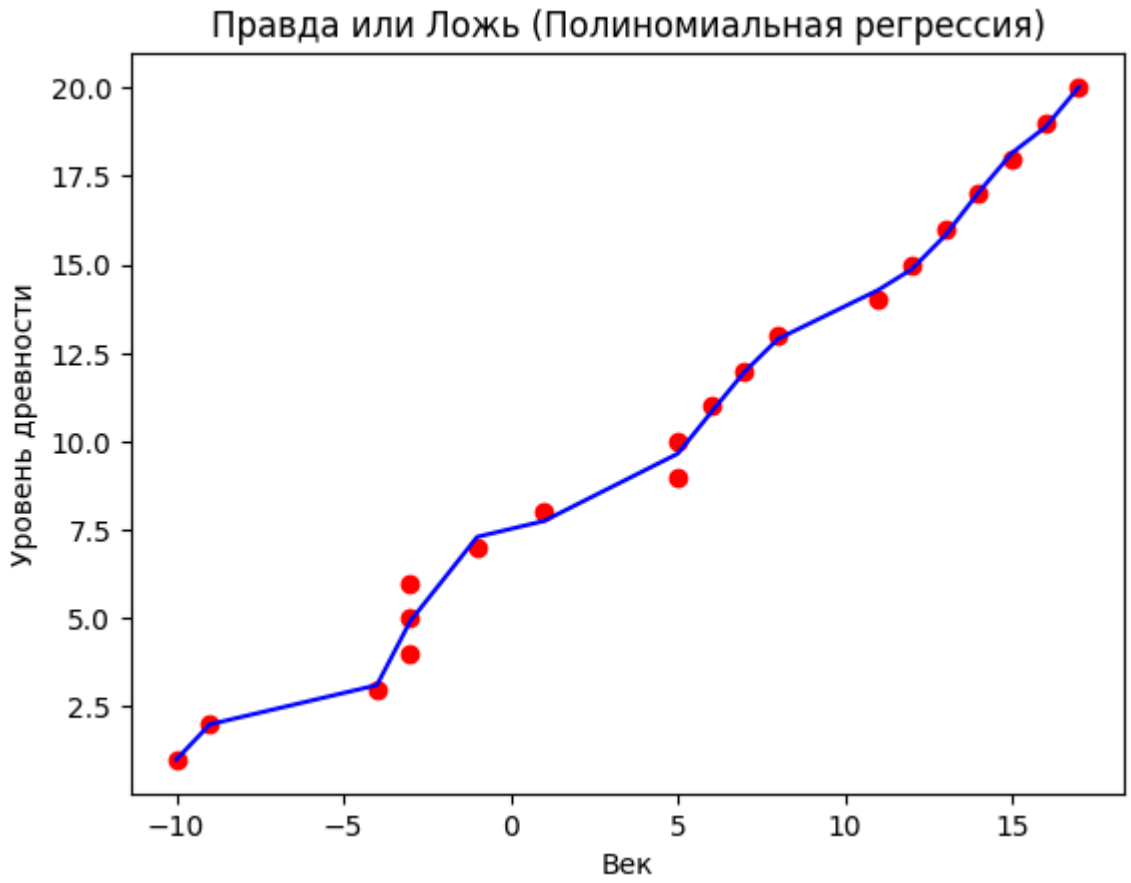
Предсказание, обработка и визуализация результатов

In [9]:
```
y_pred_lin = lin_reg.predict([[6.5]])
y_pred_poly = lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
print(y_pred_lin, y_pred_poly)
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg.predict(X), color = 'blue')
plt.title('Правда или Ложь (Линейнаая регрессия)')
plt.xlabel("Век")
plt.ylabel("Уровень древности")
plt.show()
```

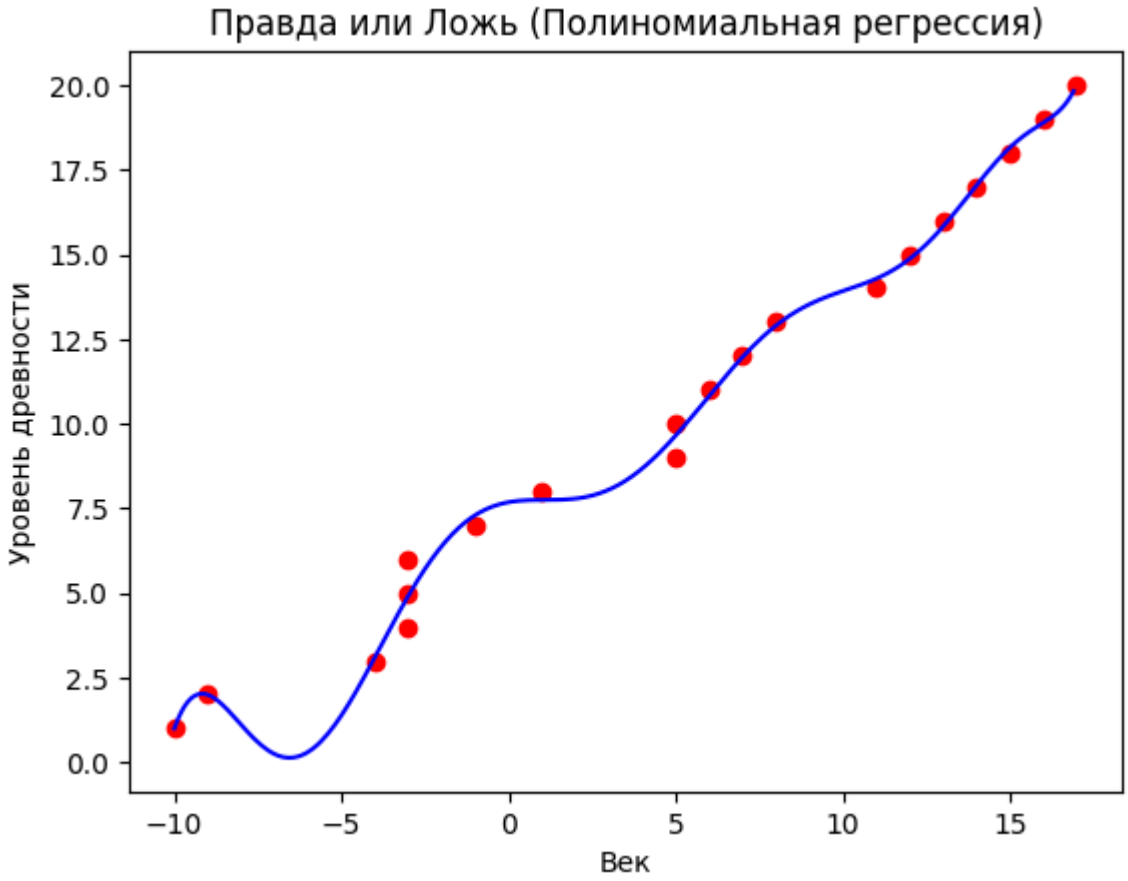
```
[11.64563877] [11.41272478]
```



In [10]:
```
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')
plt.title('Правда или Ложь (Полиномиальная регрессия)')
plt.xlabel("Век")
plt.ylabel("Уровень древности")
plt.show()
```



In [12]:
```
X_grid = np.arange(min(X)[0], max(X)[0], 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color='red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color='blue')
plt.title('Правда или Ложь (Полиномиальная регрессия)')
plt.xlabel("Век")
plt.ylabel("Уровень древности")
plt.show()
```


Правда или Ложь (Полиномиальная регрессия)
Уровень древности
Век
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−10
−5
0
5
10
15