

## Регрессия

### 1. Постановка задачи

Реализовать линейную, полиномиальную и экспоненциальную регрессию.

### 2. Используемые данные

Набор искусственно созданных внутри программы двумерных данных с использованием функции random.

### 3. Решение

Импорт используемых модулей:

```
import numpy as np - для работы с массивами
import matplotlib.pyplot as plt - для визуализации результатов
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from math import exp
from random import random - для рандомизации входных данных
```

Получение искусственных данных:

```
X = np.array([i for i in range(1, 10000)]).reshape(-1, 1)
Y = np.array([i*(1+0.1*random())*exp(i**(1/3)) for i in range(1, 10000)]).reshape(-1, 1)
```

Реализация регрессии:

#### 1. Линейной

```
reg_linear = LinearRegression()
reg_linear.fit(X, Y)
Y_linear = reg_linear.predict(X)
```

#### 2. Полиномиальной

```
preprocess_polynomial = PolynomialFeatures(degree=4)
reg_polynomial = LinearRegression()
X_poly = preprocess_polynomial.fit_transform(X)
reg_polynomial.fit(X_poly, Y)
Y_polynomial = reg_polynomial.predict(X_poly)
```

### 3. Экспоненциальной

```
expo = np.polyfit(X[:, 0], np.log(Y[:, 0]), 1, w=np.sqrt(Y[:, 0]))
```

```
Y_exponential = np.array([])
```

```
for x1 in X:
```

```
    y = exp(expo[1] + expo[0] * x1[0])
```

```
    Y_exponential = np.append(Y_exponential, [y], axis=0)
```

Визуализация полученных данных:

```
plt.figure(figsize=(10, 5))
```

```
plt.scatter(X, Y, s=1, c='blue')
```

```
plt.plot(X, Y_linear, c='red')
```

```
plt.plot(X, Y_polynomial, c='green')
```

```
plt.plot(X, Y_exponential, c='pink')
```

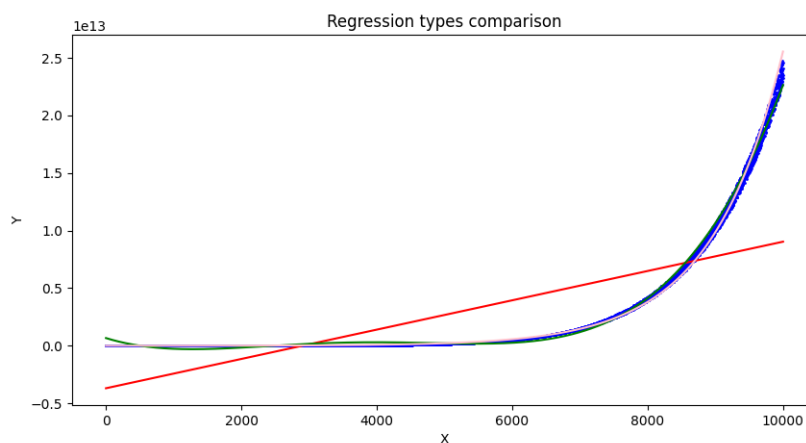
```
plt.title('Regression types comparison')
```

```
plt.xlabel('X')
```

```
plt.ylabel('Y')
```

```
plt.show()
```

Рисунок 1: результаты работы программы (синий - начальные данные, красный - линейная, зелёный - полиномиальная, розовый - экспоненциальная)



### 4. Выводы

Все три перечисленных типа регрессии реализованы с использованием библиотек.