

## Лабораторна робота № 21 (курсова робота)

### Тема: «Пошуки регресії за поліноміальною функцією втрат»

#### Мета:

1. Отримати уявлення про математичну основу навчання алгоритмів пошуку регресії.
2. Засвоїти математичну модель пошуку регресії за лінійною функцією втрат.
3. Самостійно отримати математичну модель для пошуку регресії за квадратичною функцією втрат.
4. Програмно реалізувати обидва підходи до пошуку регресії для об'єктів з однією числовою ознакою.
5. Розібратися із наданим кодом, що реалізує побудову поліноміальний регресор довільного ступеню загальний.

### Теоретичний мінімум

Регресія – це прогнозування значення цільової функції на підставі значень вектору ознак. Розглянемо простий випадок, коли об'єкт має одну числову ознаку і цільова функція функцією однієї змінної. Отримані результати легко адаптувати на випадок вектору ознак більш загального вигляду.

Квадратична функція втрат має вид

$$\mathcal{L}(a, x) = (a(x) - y^*(x))^2$$

де  $y^*(x)$  – невідома цільова функція;

$a(x)$  – модель алгоритму, що її наближає;

$x$  – числова ознака (об'єкти та їх ознаки ми ототожнюємо!).

Навчальна вибірка у цьому випадку буде множиною значень ознак

$$X^l = x_1, x_2, \dots, x_l,$$

А множина прецедентів -  $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$ ,

Функціонал якості алгоритму  $a$  за вибіркою  $X^l$  буде мати вигляд

$$Q(a, X^l) = \frac{1}{l} \sum_{i=1}^l \mathcal{L}(a, x_i) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2$$

У якості моделі алгоритму вибираємо лінійну модель у вигляді виразу, що легко алгоритмізується

$$\theta(x, \theta_0, \theta_1) = \theta_0 + \theta_1 x,$$

А функціонал якості моделі прийме вигляд

$$Q(\theta, X^l) = \frac{1}{l} \sum_{i=1}^l \mathcal{L}(\theta, x_i) = \frac{1}{l} \sum_{i=1}^l (\theta_0 + \theta_1 x_i - y_i)^2$$

Навчання моделі алгоритму полягає у знаходження таких значень параметрів  $\theta_0, \theta_1$ , при яких функціонал якості мінімізується.

Необхідна умова екстремуму (мінімуму у нашому випадку), це рівність нулю частинних похідних від функціоналу якості за параметрами  $\theta_0, \theta_1$

$$\frac{\partial Q}{\partial \theta_0} = \frac{2}{l} \sum_{i=1}^l (\theta_0 + \theta_1 x_i - y_i) = 0$$

$$\frac{\partial Q}{\partial \theta_1} = \frac{2}{l} \sum_{i=1}^l x_i (\theta_0 + \theta_1 x_i - y_i) = 0$$

Після перетворень, отримаємо

$$\begin{aligned} \theta_0 l + \theta_1 \sum_{i=1}^l x_i &= \sum_{i=1}^l y_i \\ \theta_0 \sum_{i=1}^l x_i + \theta_1 \sum_{i=1}^l x_i^2 &= \sum_{i=1}^l x_i y_i \end{aligned}$$

Ці вирази є системою лінійних рівнянь відносно значень параметрів  $\theta_0, \theta_1$ , при яких функціонал якості має мінімальне значення. Якщо ці значення підстави у вираз для моделі алгоритму, то отримаємо навчений алгоритм

$$a(x) = \theta_0 + \theta_1 x,$$

який наближає, на підставі лінійної моделі, шукану цільову функцію.

### **ЗАВДАННЯ:**

**А)** Поновити код із Лабораторної роботи 14 (див. Посібник), що реалізує наведену вище модель із лінійною функцією втрат.

**Б)** Побудувати, за аналогією, модель із квадратичною функцією втрат.

**В)** Знайти у пакеті **Numpy** модуль, що реалізує модель із квадратичною функцією втрат. Адаптувати код із попереднього пункту на випадок такої моделі.

- Для налагодження програм із п.п. А) та В) використати тренувальний набір `data_singlevar_regr.txt`

- Знайти інформаційні джерела і описати зміст метрик якості, що використовуються.

**Г)** Застосувати отриманий код для побудови лінійної та квадратичної регресії на даних каротажних досліджень, що відображає залежність температури гірського масиву від глибини занурення датчиків.

- Файл даних `data_1.txt` і `data_2.txt`

- Візуалізувати результати, виділивши різними кольорами тренувальний, тестовий набори даних та лінії регресії.

- Отримати метрики якості і зробити висновок, що до адекватності моделей.

**Д)** Наданий у Додатку код **Lab 21 multi regrsor** реалізує процес навчання регресору, побудованого на підставі моделі функції втрат ц вигляді поліному довільного ступеня.

- Поновити код

- Описати процеси, що реалізує цей код

- Налагодити його за допомогою наданих даних `data_mult_regressor.txt`

- Побудувати лінійну і поліноміальну модель регресії на даних каротажних досліджень і візуалізувати дані.

```
Lab 21 multi regressor.py - C:\Users\User\AppData\Local\Programs\Python\5 CS\Lab 22 multi regressor\Lab 21 multi...
File Edit Format Run Options Window Help
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
from sklearn.preprocessing import PolynomialFeatures

#Load data
input_file = 'data_multivar_regr.txt'
data = np.loadtxt(input_file, delimiter = ',')
X, y = data[:, :-1], data[:, -1]

# learning and test sets
num_training = int(0.8*len(X))
num_test = len(X) - num_training

X_train, y_train = X[:num_training], y[:num_training]
X_test, y_test = X[num_training:], y[num_training:]

# create model linear regressor
linear_regressor = linear_model.LinearRegression()

# learning of model linear regressor
linear_regressor.fit(X_train, y_train)

# prediction of result model learning
y_test_pred = linear_regressor.predict(X_test)

# example of using of learned regressor
b = np.array([[7.75, 6.35, 5.56]])
a = linear_regressor.predict(b)

# metrix of learning
print('Linear Regressor performance:')
print('Mean absolute error= ', round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print('Mean squared error= ', round(sm.mean_squared_error(y_test, y_test_pred), 2))
print('Median absolute error= ', round(sm.median_absolute_error(y_test, y_test_pred), 2))
print('Explained variance score =', round(sm.explained_variance_score(y_test, y_test_pred), 2))
print('R2 score =', round(sm.r2_score(y_test, y_test_pred), 2))

# polynomial regressor
polynomial = PolynomialFeatures(degree=10)
X_train_transformed = polynomial.fit_transform(X_train)
datapoint = [[7.75, 6.35, 5.56]]
poly_datapoint = polynomial.fit_transform(datapoint)

poly_linear_model = linear_model.LinearRegression()
poly_linear_model.fit(X_train_transformed, y_train)
print("\nLinear regression:\n", linear_regressor.predict(datapoint))
print("\nPolynomial regression:\n", poly_linear_model.predict(poly_datapoint))
```

Ln: 1 Col: 0

```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\User\AppData\Local\Programs\Python\5 CS\Lab 22 multi regressor\Lab 21 multi regressor.py
Linear Regressor performance:
Mean absolute error= 3.8
Mean squared error= 22.36
Median absolute error= 3.13
Explained variance score = 0.87
R2 score = 0.87

Linear regression:
[36.05286276]

Polynomial regression:
[41.45846104]
>>>
```