

Міністерство освіти і науки України  
Львівський національний університет імені Івана Франка  
Факультет електроніки та комп'ютерних технологій  
Кафедра системного проектування

Звіт

Про виконання лабораторної роботи №1  
З курсу «Системи машинного навчання»  
Вступ в машинне навчання та Scikit-learn

**Виконала:**

Студентка групи ФЕС-32  
Філь Дарина

**Перевірив:**

Доцент Колич І.І.

**Мета:** Ознайомитись з базовими поняттями машинного навчання та бібліотекою Skilit-learn.

### Теоретичні відомості:

Основні поняття:

1. **Залежна змінна (target, response):** Це змінна, яку ми намагаємося передбачити або пояснити.
2. **Незалежні змінні (predictors, features):** Це змінні, які ми використовуємо для передбачення значення залежної змінної.

### Формула лінійної регресії Одновимірна лінійна регресія:

Формула одновимірної (простої) лінійної регресії виглядає так:

$$y = \beta_0 + \beta_1 x$$

де:

- $y$  — залежна змінна;
- $\beta_0$  — вільний член (**intercept**);
- $\beta_1$  — коефіцієнт нахилу (**slope**);
- $x$  — незалежна змінна;

### Множинна лінійна регресія:

Формула множинної лінійної регресії (з більш ніж однією незалежною змінною) виглядає так:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

де:

- $y$  — залежна змінна;
- $\beta_0$  — вільний член (**intercept**);
- $\beta_1, \dots, \beta_n$  — коефіцієнти регресії для кожної незалежної змінної;
- $x_1, \dots, x_n$  — незалежні змінні;

## Метод найменших квадратів (Ordinary Least Squares, OLS)

Метод найменших квадратів (OLS) використовується для знаходження оптимальних значень коефіцієнтів регресії  $\beta$ , які мінімізують суму квадратів різниць між передбаченими значеннями та фактичними значеннями залежної змінної.

**Формула для обчислення коефіцієнтів регресії:**

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

де:

- $\beta$  — вектор оцінених коефіцієнтів;
- $X$  — матриця незалежних змінних;
- $y$  — вектор залежної змінної;
- $X^T$  — транспонована матриця ( $X$ );
- $(X^T X)^{-1} X^T$  — обернена матриця до  $(X^T X)$ ;

## Оцінка моделі

Після навчання моделі лінійної регресії важливо оцінити її продуктивність. Ось деякі ключові метрики для оцінки моделі:

### 1. Середньоквадратична помилка (Mean Squared Error, MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

де:

- $n$  — кількість спостережень;
- $y_i$  — фактичне значення;
- $\hat{y}_i$  — передбачене значення;

### 2. Коефіцієнт детермінації ( $R^2$ ):

$$\left[ R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \right] \text{ де:}$$

- $\bar{y}$  — середнє значення залежної змінної.

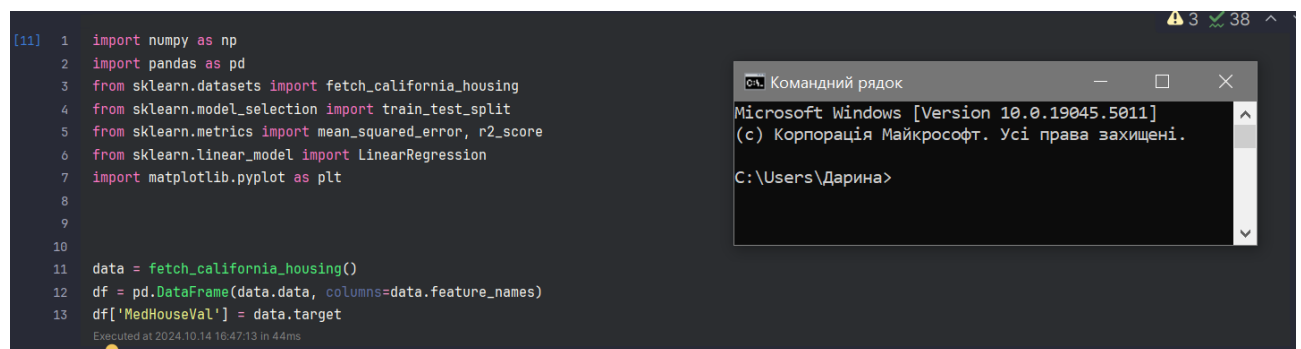
$R^2$  показує, яка частка варіації залежної змінної може бути пояснена незалежними, змінними моделі. Значення  $R^2$  варіюється від 0 до 1, де значення, близьке до 1, вказує на хорошу модель.

## Хід роботи:

### Завдання

1. Завантаження готових наборів даних з Scikit-learn:
  - Завантажити набір `sklearn.datasets.fetch_california_housing`
  - Виведіть перших 7 рядків, використовуючи бібліотеку Pandas
2. Поділ даних на тренувальну та тестову вибірки:
  - Поділити дані на тренувальний 70% та тестовий 30% набори.
3. Написання функції для формули множинної лінійної регресії:
  - Вихід залежна змінна (ціна будинку), вхід вектор незалежних змінних та вектор коефіцієнтів.
4. Випадковий підбір коефіцієнтів:
  - Задати затравку (seed) для numpy використовуючи наступний код з заміною surname на прізвище автора.  

```
my_str = "surname"  
res = ".join(format(ord(i), '08b') for i in my_str)  
my_seed = int(res) % 12345
```
  - Написання цикл з генерацією випадкового вектору коефіцієнтів та вибрати вектор з найменшою середньою квадратичною помилкою
  - Оцінити коефіцієнт детермінації
5. Навчання та оцінка простої моделі (наприклад, лінійна регресія):
  - Навчити моделі та тренувальних даних за допомогою лінійної регресії. Оцінити продуктивності моделі на тестових даних.
6. Оформити звіт.



```
[11] 1 import numpy as np  
2 import pandas as pd  
3 from sklearn.datasets import fetch_california_housing  
4 from sklearn.model_selection import train_test_split  
5 from sklearn.metrics import mean_squared_error, r2_score  
6 from sklearn.linear_model import LinearRegression  
7 import matplotlib.pyplot as plt  
8  
9  
10  
11 data = fetch_california_housing()  
12 df = pd.DataFrame(data.data, columns=data.feature_names)  
13 df['MedHouseVal'] = data.target  
Executed at 2024.10.14 16:47:13 in 44ms
```

Рис. 1 Ініціалізація усіх бібліотек та дата фрейму

```
data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['MedHouseVal'] = data.target
df.head(7)
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422
5	4.0368	52.0	4.761658	1.103627	413.0	2.139896	37.85	-122.25	2.697
6	3.6591	52.0	4.931907	0.951362	1094.0	2.128405	37.84	-122.25	2.992

Рис. 2 Перші 7 значень дата фрейму

```
X = df.drop('MedHouseVal', axis=1)
y = df['MedHouseVal']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
print(f"Розмір тренувальної вибірки: {X_train.shape}")
print(f"Розмір тестової вибірки: {X_test.shape}")

def predict(X, coefficients):
    return np.dot(X, coefficients)
```

Розмір тренувальної вибірки: (14448, 8)  
Розмір тестової вибірки: (6192, 8)

Рис. 3 Поділ даних з дата фрейму на тестувальний та тренувальний. Також написана функція для множинної регресії

```
my_str = "Fil"
res = ''.join(format(ord(i), '08b') for i in my_str)
my_seed = int(res) % 12345
np.random.seed(my_seed)

best_mse = float('inf')
best_coefficients = None

for _ in range(100000):
    coefficients = np.random.rand(X_train.shape[1])
    predictions = predict(X_train, coefficients)
    mse = mean_squared_error(y_train, predictions)

    if mse < best_mse:
        best_mse = mse
        best_coefficients = coefficients

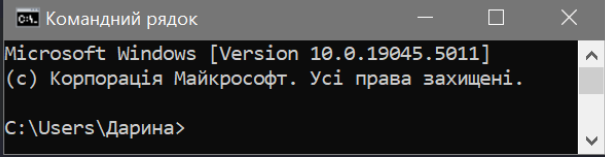
print("Найкращі коефіцієнти:", best_coefficients)
print("Найменша середньоквадратична помилка:", best_mse)
```

Найкращі коефіцієнти: [3.94151529e-01 2.38453075e-01 4.58399682e-01 3.22963085e-01  
1.50874472e-04 1.13638376e-01 2.19919437e-01 1.35180156e-01]  
Найменша середньоквадратична помилка: 13.022506917487856

Рис. 4 Створення seed та циклу для генерації випадкового вектору коефіцієнтів

```
y_pred_train = predict(X_train, best_coefficients)
r2_train = r2_score(y_train, y_pred_train)
print("Коефіцієнт детермінації на тренувальній вибірці:", r2_train)
Executed at 2024.10.14 16:48:16 in 9ms

Коефіцієнт детермінації на тренувальній вибірці: -8.72049415424987
```



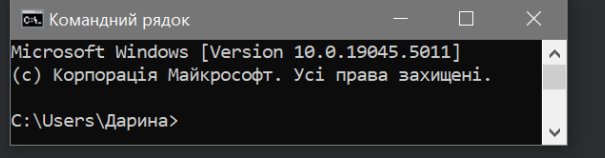
**Рис. 5** Тут зображено, що коефіцієнт детермінації на тренувальній вибірці дуже поганий, це пов'язано з тим, що вибір коефіцієнтів є випадковим

```
1 for _ in range(10000):
2     coefficients = np.random.rand(X_train.shape[1])
3     predictions = predict(X_train, coefficients)
4     mse = mean_squared_error(y_train, predictions)
5
6     if mse < best_mse:
7         best_mse = mse
8         best_coefficients = coefficients
9
10 print("Найкращі коефіцієнти:", best_coefficients)
11 print("Найменша середньоквадратична помилка:", best_mse)
Executed at 2024.10.26 19:55:58 in 10s 511ms

Найкращі коефіцієнти: [0.62224906 0.35120163 0.63928978 0.36301765 0.00123193 0.38057191
0.0940094 0.17964103]
Найменша середньоквадратична помилка: 42.51182768424148

1 y_pred_train = predict(X_train, best_coefficients)
2 r2_train = r2_score(y_train, y_pred_train)
3 print("Коефіцієнт детермінації на тренувальній вибірці:", r2_train)
Executed at 2024.10.26 19:55:58 in 11ms

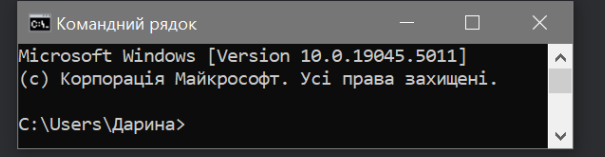
Коефіцієнт детермінації на тренувальній вибірці: -30.732444076202768
```



**Рис. 6** При зменшенні розміру вектора можемо побачити, що помилка росте, а точність коефіцієнту детермінації падає ще більше

```
1 model = LinearRegression()
2 model.fit(X_train, y_train)
3
4 y_pred_test = model.predict(X_test)
5 mse_test = mean_squared_error(y_test, y_pred_test)
6 r2_test = r2_score(y_test, y_pred_test)
7
8 print("Середньоквадратична помилка на тестовій вибірці:", mse_test)
9 print("Коефіцієнт детермінації на тестовій вибірці:", r2_test)
Executed at 2024.10.26 19:55:58 in 100ms

Середньоквадратична помилка на тестовій вибірці: 0.5305677824766757
Коефіцієнт детермінації на тестовій вибірці: 0.595770232606166
```



**Рис. 7** Результат навчання простої моделі лінійної регресії, можемо бачити, що коефіцієнт детермінації 0,59, що вказує на те, що більша частина варіації залежної змінної може бути пояснена незалежними змінними

**Висновок:** У цій лабораторній роботі я навчилась працювати з основами тренування простих моделей та бібліотекою `skikit-learn`, також провела наглядний приклад тренування моделі та оцінила результат тренування на основі коефіцієнту детермінації та середній квадратичній похибці.