МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Практикум №5

з курсу «Програмування інтелектуальних інформаційних систем»

на тему: «Регресійні моделі»

Викладач:
Курченко О.А.

Виконав:
Хільчук А.В.
студент 3 курсу
групи ІП-14 ФІОТ

Київ-2023

Практична робота №5

Тема: Регресійні моделі
Завдання:
1. Підготувати навчальну вибірку у вигляді таблиць MS Excel і зберегти їх як персональні файли. Для підготовки даних використовувати тематичні сайти Інтернет, результати проходження практик, довідники і каталоги.
2. Дослідити дані, сказати чи є мультиколінеарність, побудувати діаграми розсіювання
3. Побудувати декілька регресійних моделей для прогнозу. Використати лінійну одномірну та багатомірну регресію та поліноміальну регресію обраного виду(3-5 моделей)
4. Отримати коефіцієнти регресійних моделей та проаналізувати/ проінтерпритувати їх
5. Використовуючи тестову вибірку, з'ясувати яка з моделей краща
6. Провести декілька експериментів
7. Зробити висновки.

Предметна бласть – передбачення вартості одиниці площі нерухомості

Виконання:

1. Відповідно до заданого варіанту підготувати необхідні дані у вигляді таблиць MS Excel і зберегти їх як персональні файли. Для підготовки даних використовувати тематичні сайти Інтернет, результати проходження практик, довідники і каталоги.

   Для виконання даної лабораторної роботи було взято даний набір даних:
   https://www.kaggle.com/datasets/quantbruce/real-estate-price-prediction
   Даний датасет містить записи про ціну одиниці площі нерухомості в Синдіанському районі міста Новий Тайбей Тайваню. Для кожного запису фіксуються такі значення: номер запису, дата купівлі будинку в роках, вік будинку, відстань до найближчої станції масового переміщення (наприклад, метро), кількість магазинів в пішій доступності, довгота й широта розташування будинку, а також вартість площі в десятках тисяч Тайванських доларів на один пінг, де пінг – місцева одиниця виміру площі, що рівна приблизно 3.3 метрам квадратним.

2. Дослідити дані, сказати чи є мультиколінеарність, побудувати діаграми розсіювання.

Код програми наведено в **додатку А.**

Переглядаємо дані:
Перші 5 значень:

	No	transaction date	house age	distance to the nearest MRT station	\
0	1	2012.917	32.0	84.87882	
1	2	2012.917	19.5	306.59470	
2	3	2013.583	13.3	561.98450	
3	4	2013.500	13.3	561.98450	
4	5	2012.833	5.0	390.56840	

	number of convenience stores	latitude	longitude	house price of unit area
0	10	24.98298	121.54024	37.9
1	9	24.98034	121.53951	42.2
2	5	24.98746	121.54391	47.3
3	5	24.98746	121.54391	54.8
4	5	24.97937	121.54245	43.1

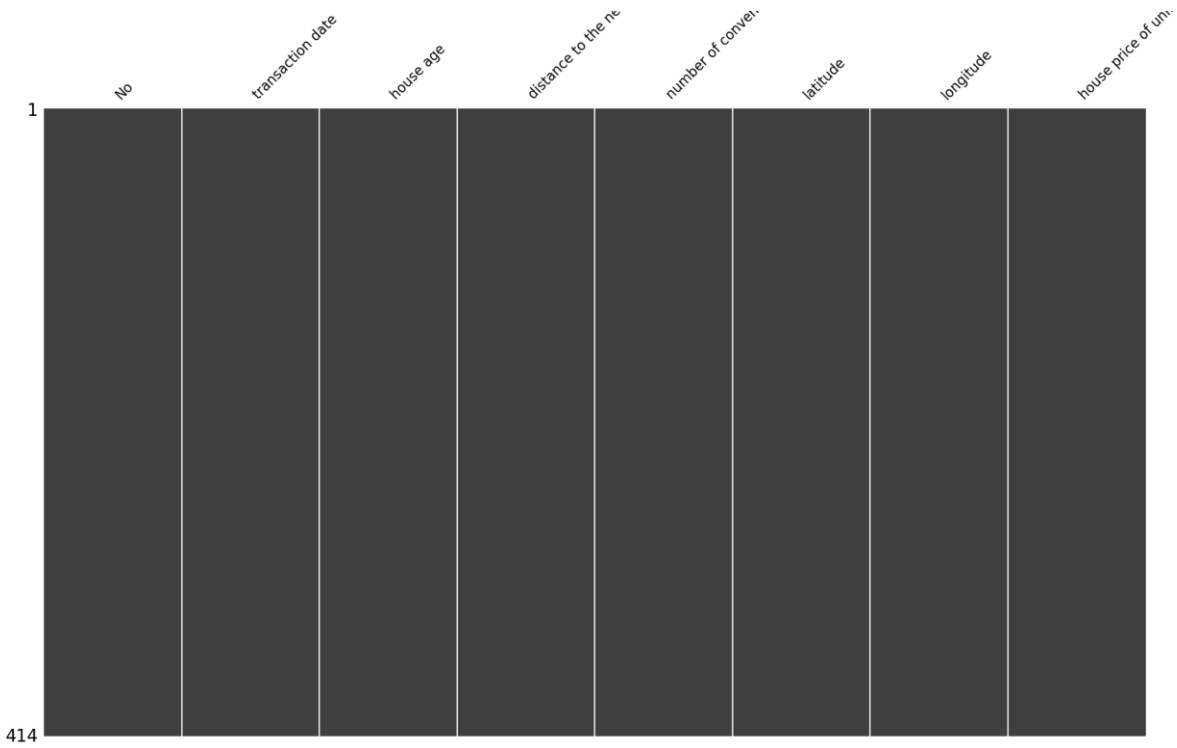
Переглядаємо опис даних:

	No	transaction date	house age	\
count	414.000000	414.000000	414.000000	
mean	207.500000	2013.148971	17.712560	
std	119.655756	0.281967	11.392485	
min	1.000000	2012.667000	0.000000	
25%	104.250000	2012.917000	9.025000	
50%	207.500000	2013.167000	16.100000	
75%	310.750000	2013.417000	28.150000	
max	414.000000	2013.583000	43.800000	

	distance to the nearest MRT station	number of convenience stores	\
count	414.000000	414.000000	
mean	1083.885689	4.094203	
std	1262.109595	2.945562	
min	23.382840	0.000000	
25%	289.324800	1.000000	
50%	492.231300	4.000000	
75%	1454.279000	6.000000	
max	6488.021000	10.000000	

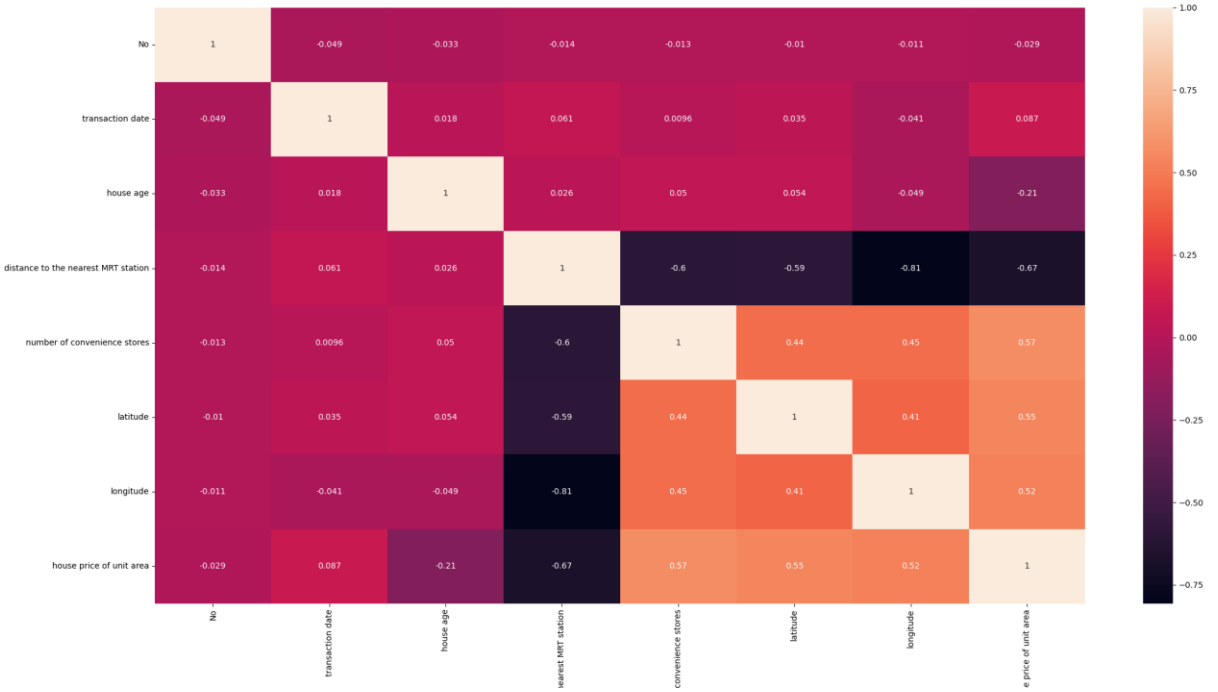
	latitude	longitude	house price of unit area
count	414.000000	414.000000	414.000000
mean	24.969030	121.533361	37.980193
std	0.012410	0.015347	13.606488
min	24.932070	121.473530	7.600000
25%	24.963000	121.528085	27.700000
50%	24.971100	121.538630	38.450000
75%	24.977455	121.543305	46.600000
max	25.014590	121.566270	117.500000

Проаналізуємо датафрейм на наявність відсутніх значень. Перш за все, побудуймо теплову карту відсутніх значень:

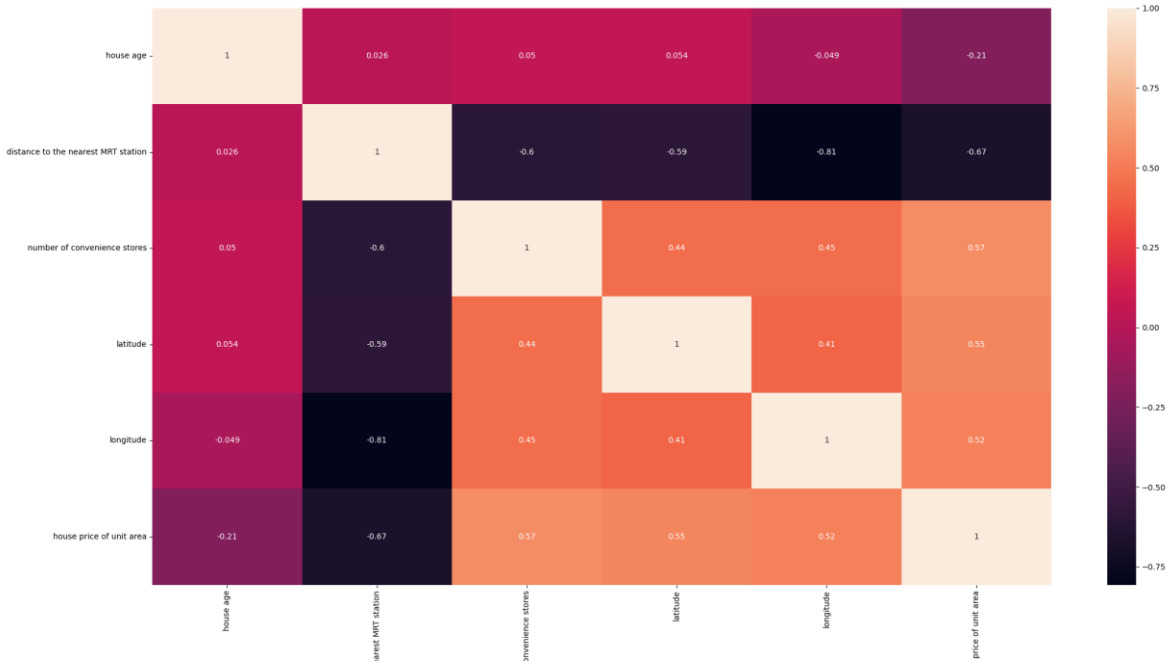


Як бачимо, відсутніх значень немає.

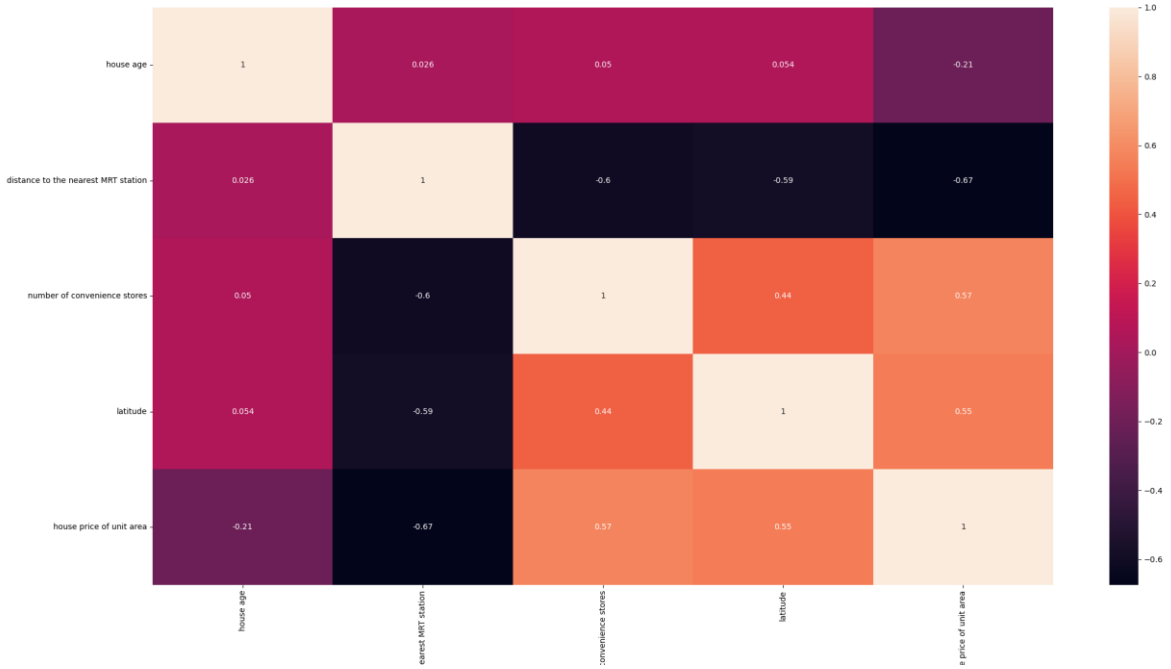
Тепер необхідно обрати предиктори. Для цього буде застосовано кореляційний аналіз:
Будуємо матрицю кореляцій та відображаємо її у вигляді теплової карти:



Номер будинку та дата купівлі не корелюють з ціною – прибираємо їх:

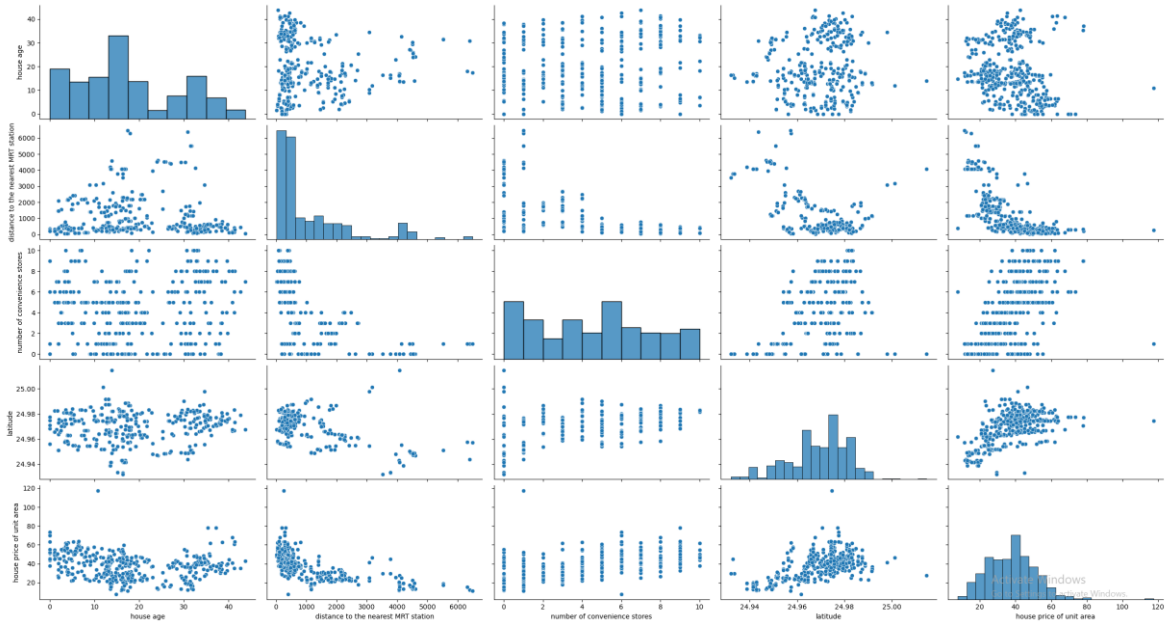


Критерієм мультиколінеарності будемо вважати кореляцію між факторами в більш ніж 0.7. Карта демонструє, що мультиколінеарність є. Прибираємо довготу:



Тепер, умовно, мультиколінеарність можна вважати ліквідованою.

Діаграма розсіювання даних:



3. Побудувати декілька регресійних моделей для прогнозу. Використати лінійну одномірну та багатомірну регресію та поліноміальну регресію обраного виду(3-5 моделей)

   Отож, у рамках даного підзвадання було побудовано такі моделі:

   1. Лінійну однофакторну, що спирається на найближчу точку масового швидкого транзиту
   2. Лінійну багатофакторну
   3. Поліноміальну багатофакторну другого ступеню

4. Побудувати Отримати коефіцієнти регресійних моделей та проаналізувати/проінтерпритувати їх

   У результаті маємо такі коефіцієнти:

```
Checking out coefficients:
Linear single-factor:
    Base constant value: 46.24269004533652
    Coefficient for distance to the nearest MRT station: -0.007409512934932927
polynomial regression of 1 degree:
    Base constant value: -5908.102022661176
    Coefficient for 1: 0.0
    Coefficient for house age: -0.2697330698545541
    Coefficient for distance to the nearest MRT station: -0.004297025800142601
    Coefficient for number of convenience stores: 1.1117695456787486
    Coefficient for latitude: 238.3364312922782
polynomial regression of 2 degree:
    Base constant value: 3394899.2512721615
    Coefficient for 1: 0.0
    Coefficient for house age: 128.50485801764
    Coefficient for distance to the nearest MRT station: 5.860382718000503
    Coefficient for number of convenience stores: 2703.3587984242777
    Coefficient for latitude: -272972.65957985923
    Coefficient for house age^2: 0.01822340977167155
    Coefficient for house age distance to the nearest MRT station: 5.227451135330137e-05
    Coefficient for house age number of convenience stores: 0.008194764364713914
    Coefficient for house age latitude: -5.188611011422059
    Coefficient for distance to the nearest MRT station^2: 2.397228892032814e-07
    Coefficient for distance to the nearest MRT station number of convenience stores: -0.000997316685339177
    Coefficient for distance to the nearest MRT station latitude: -0.23499067354434455
    Coefficient for number of convenience stores^2: 0.03509242976806062
    Coefficient for number of convenience stores latitude: -108.23079580968671
    Coefficient for latitude^2: 5487.208365397156
```

В лінійній однофакторінй моделі найбільша вага призначена відстані до найближчої станції метро. Це означає, що з кожним збільшенням відстані до станції метро на одиницю, вартість площі одиниці нерухомості зменшується на 0.0074 одиниць.
В лінійній однофакторній найбільший наголос призначений широті: вона має найбільший позитивний вплив на вартість площі нерухомості. Крім того, кількість магазинів зручностей також має суттєвий голос у визначенні вартості. Вік будинку і відстань до станції метро мають вплив на вартість, але менший в порівнянні з іншими параметрами.
Для поліноміальної другого ступеня найбільший наголос призначений кількості магазинів зручностей, а також широті і віку будинку. Взаємодія між параметрами також має важливе значення. Ця модель враховує значення параметрів більш складно, враховуючи квадратичні та взаємодію між ними.

5. Використовуючи тестову вибірку, з'ясувати яка з моделей краща

У якості метрик якості було обрано такі:

- MSE – вона, як зрозуміло з назви, обраховує середнє суми квадратів різниць прогнозованих даних від дійсних. Чим вона менша – тим краще

- R^2 – вона використовується для виміру ступеню змінності залежної змінної, яка пояснюється моделлю, відносно загальної змінності в цій залежній змінній. Обчислюється за формулою:

$$1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

  Відповідно, чим ближча ця метрика до одиниці, тим краще.

Результат:

```
Checking models quality:

Model: Linear single-factor
    MSE: 77.34399178518458
    R-squared: 0.5389597665019772


Model: polynomial regression of 1 degree
    MSE: 54.30794499729841
    R-squared: 0.6762754667241218


Model: polynomial regression of 2 degree
    MSE: 41.029176740167564
    R-squared: 0.7554289507444077
```

6. Провести декілька експериментів
   Результат експериментів:

```
Preforming experiments:
Test number 1:
    House age: 4.0
    Distance to the nearest MRT station: 2147.376
    Number of convenience stores: 3.0
    Latitude: 24.96299
    Actual price: 30.7
    Linear single-factor regression says: 30.331679797171994
    Linear multifactor regression says: 34.5169746066631
    Polynomial multifactor of 2 degree regression says: 28.47823073528707
Test number 2:
    House age: 13.8
    Distance to the nearest MRT station: 4082.015
    Number of convenience stores: 0.0
    Latitude: 24.94155
    Actual price: 15.6
    Linear single-factor regression says: 15.996947102246292
    Linear multifactor regression says: 15.115155101182609
    Polynomial multifactor of 2 degree regression says: 20.86435883725062
Test number 3:
    House age: 26.9
    Distance to the nearest MRT station: 4449.27
    Number of convenience stores: 0.0
    Latitude: 24.94898
    Actual price: 15.5
    Linear single-factor regression says: 13.275766429327497
    Linear multifactor regression says: 11.77438736035947
    Polynomial multifactor of 2 degree regression says: 19.56409768247977
```

Висновок

Отож, у ході виконання лабораторної роботи було взято набір даних про вартість площі нерухомості в Синдіанському районі міста Новий Тайбей Тайваню, для кожного запису фіксуючи параметри: номер запису, дата купівлі будинку в роках, вік будинку, відстань до найближчої станції МШТ, кількість магазинів в пішій доступності, довгота й широта розташування будинку й вартість площі, та підготовлено їх для подальшого аналізу. У рамках підготовки було проведено кореляційний аналіз параметрів для відбору тих, що пов’язані з вартістю площи, та усунення мультиколінеарності. Опісля було побудовано моделі регресії: лінійну однофакторну, лінійну багатофакторну та поліноміальну багатофакторну другого ступеню,- і проаналізовано їх ефективність, базуючись на метриках MSE та R^2. Урешті-решт було вибрано декілька записів з тестового набору даних та проведено наочній експерименти за участю кожної з моделей. Набуто практичних навичок тренування й застосування моделей регресії мовою Python.

Додаток А

Текст скрипту для виконання лабораторної роботи

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
import random
import missingno as msno

if __name__=="__main__":
    pd.set_option('display.max_columns', None)
    data = pd.read_csv("data/Real estate.csv")

    print(data.head())
    print(data.describe())

    msno.matrix(data)
    plt.show()

    sns.heatmap(data.corr(), annot=True)
    plt.show()
    data.drop(columns=['No', 'transaction date'], inplace=True)

    sns.heatmap(data.corr(), annot=True)
    plt.show() #multicolinearity detected
    data.drop(columns=['longitude', ], inplace=True)

    sns.heatmap(data.corr(), annot=True)
    plt.show()# pretty much good

    #dispersion diagram
    sns.pairplot(data)
    plt.show()

    models={}

    #split data
    X_train, X_test, y_train, y_test = train_test_split(data.iloc[:, :-1],
data['house price of unit area'], test_size=0.2,random_state=42)

    # linear single-factor regression
    lin_reg = LinearRegression()
    lin_reg.fit(X_train['distance to the nearest MRT station'].values.reshape(-1,
1), y_train)

    models["Linear single-factor"] = (lin_reg, ["distance to the nearest MRT
station"], X_test['distance to the nearest MRT station'].values.reshape(-1, 1))

    #linear multi-factor and polynomial of 2 degree
    for degr in range(1, 3):
        poly = PolynomialFeatures(degree=degr)
        predictors_train_polyfeat = poly.fit_transform(X_train)
```

```
        poly_reg = LinearRegression()
        poly_reg.fit(predictors_train_polyfeat, y_train)

        models[f"polynomial regression of {degr} degree"]=(poly_reg,
poly.get_feature_names(X_test.columns), poly.transform(X_test))

    print("Checking out coefficients:")
    for name, tuple in models.items():
        print(f"{name}:")
        print(f"\tBase constant value: {tuple[0].intercept_}")
        for feature, coef in zip(tuple[1], tuple[0].coef_):
            print(f"\tCoefficient for {feature}: {coef}")

    print("\nChecking models quality:")
    for name, tuple in models.items():
        model, features, test_data = tuple

        y_pred = model.predict(test_data)  # make predictions

        # print the model name
        print(f"\nModel: {name}")

        # print model efficiency metrics
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)
        print(f"\tMSE: {mse}")
        print(f"\tR-squared: {r2}\n")

    #Perform random experiment:
    print("\nPreforming experiments:")
    for i in range(3):
        rec_index=random.randint(0,len(X_test) - 1)

        record_features= X_test.iloc[rec_index]
        record_res=y_test.iloc[rec_index]

        print(f"Test number {i+1}:")
        print(f"\tHouse age: {record_features['house age']}")
        print(f"\tDistance to the nearest MRT station: {record_features['distance
to the nearest MRT station']}")
        print(f"\tNumber of convenience stores: {record_features['number of
convenience stores']}")
        print(f"\tLatitude: {record_features['latitude']}")

        print(f"\tActual price: {record_res}")
        print(f"\tLinear single-factor regression says:
{lin_reg.predict(np.array([[record_features['distance to the nearest MRT
station']]]))[0]}")

        poly = PolynomialFeatures(degree=1)
        predictors_test_polyfeat =
poly.fit_transform(record_features.values.reshape(1, -1))

        print(f"\tLinear multifactor regression says: {models['polynomial
regression of 1 degree'][0].predict(predictors_test_polyfeat)[0]}")

        poly = PolynomialFeatures(degree=2)
        predictors_test_polyfeat =
poly.fit_transform(record_features.values.reshape(1, -1))
```

```
        print(f"\tPolynomial multifactor of 2 degree regression says:
{models['polynomial regression of 2
degree'][0].predict(predictors_test_polyfeat)[0]}")
```