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

Звіт

з лабораторної роботи №5 з дисципліни
«Аналіз даних в інформаційних системах»

„Регресійні моделі”

Виконав(ла) *ІП-11 Тарасьонок Дмитро Євгенович*
(шифр, прізвище, ім'я, по батькові)

Перевірила *Ліхоузова Т. А.*
(прізвище, ім'я, по батькові)

Київ 2023

ЗМІСТ

1 Мета лабораторної роботи 3

2 Завдання 4

2.1 Основне завдання 4

2.2 Додаткове завдання 4

3 Виконання основного завдання 5

3.1 Дослідити дані, підготувати їх для побудови регресійної моделі 5

3.2 Розділити дані на навчальну та тестову вибірки 7

3.3 Побудувати декілька регресійних моделей для прогнозу якості вина (12 - quality). Використати лінійну регресію та поліноміальну регресію обраного вами виду 8

3.4 Використовуючи тестову вибірку, з'ясувати яка з моделей краща 10

4 Виконання додаткового завдання 11

4.1 Дослідити дані, сказати чи є мультиколінеарність, побудувати діаграми розсіювання 11

4.2 Побудувати декілька регресійних моделей (використати лінійну регресію та поліноміальну регресію обраного вами виду) 14

4.3 Використовуючи тестову вибірку з файлу Data4t.csv, з'ясувати яка з моделей краща 17

5 Висновок 18

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – ознайомитись з різновидами регресійних моделей.

2 ЗАВДАННЯ

2.1 Основне завдання

Скачати дані із файлу Data2.csv

1. Дослідити дані, підготувати їх для побудови регресійної моделі
2. Розділити дані на навчальну та тестову вибірки
3. Побудувати декілька регресійних моделей для прогнозу якості вина (12 - quality). Використати лінійну регресію та поліноміальну регресію обраного вами виду
4. Використовуючи тестову вибірку, з'ясувати яка з моделей краща

2.2 Додаткове завдання

Завантажити дані файлу Data4.csv

1. Дослідити дані, сказати чи є мультиколінеарність, побудувати діаграми розсіювання
2. Побудувати декілька регресійних моделей (використати лінійну регресію та поліноміальну регресію обраного вами виду)
3. Використовуючи тестову вибірку з файлу Data4t.csv, з'ясувати яка з моделей краща

3 ВИКОНАННЯ ОСНОВНОГО ЗАВДАННЯ

3.1 Дослідити дані, підготувати їх для побудови регресійної моделі

Для початку завантажимо дані, замінимо назви стовпців, щоб надалі до них було простіше звертатися.

```
In 2  wine = pd.read_csv('winequality-red.csv')
      wine.columns = wine.columns.str.replace(' ', '_')
      wine
      Executed in 59ms, 5 Apr at 23:02:42
```

Out 2 — 1599 rows × 12 columns pd.Dataframe

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	qua
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	
5	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	
6	7.9	0.600	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	
7	7.3	0.650	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	
8	7.8	0.580	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	
9	7.5	0.500	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	
10	6.7	0.580	0.08	1.8	0.097	15.0	65.0	0.9959	3.28	0.54	9.2	
11	7.5	0.500	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	
12	5.6	0.615	0.00	1.6	0.089	16.0	59.0	0.9943	3.58	0.52	9.9	

Рис 3.1 – Завантаження даних, перейменовування стовпців

Після цього виведемо теплову карту кореляції між різними стовпцями.

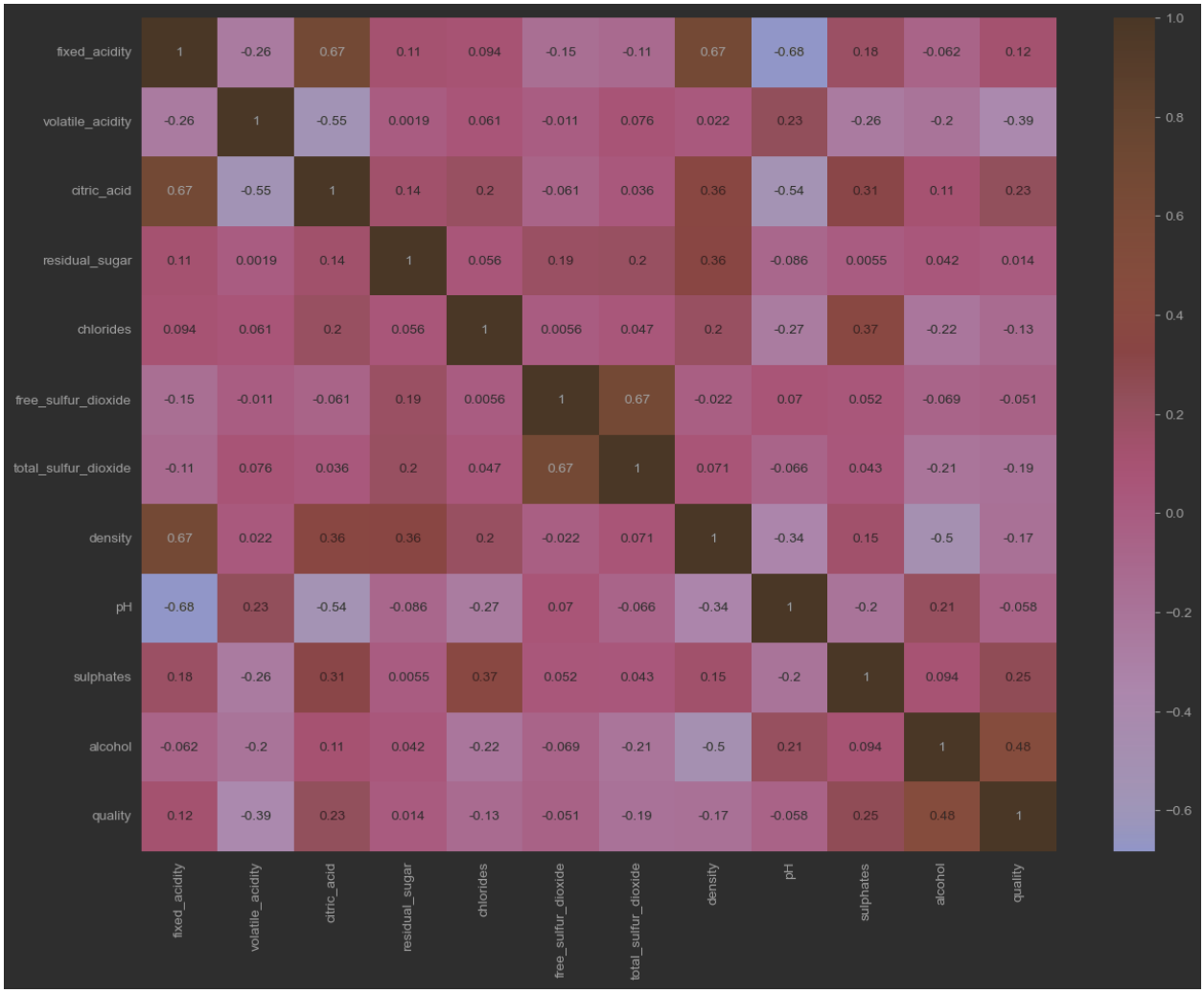
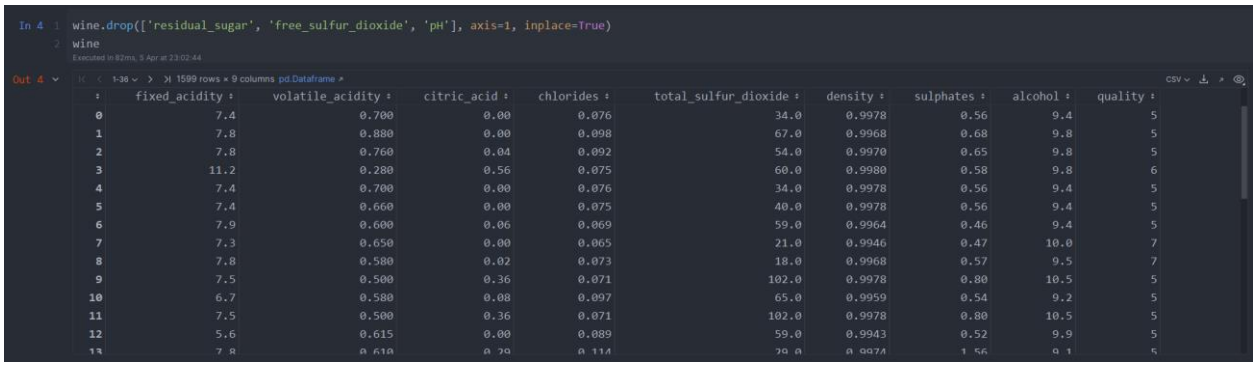


Рис 3.2 – Теплова карта кореляції

Бачимо, що на якість деякі стовпці впливають більше, а деякі – менше. Для збільшення точності регресії відкинемо стовпці pH (коефіцієнт кореляції = -0.058), free_sulfur_dioxide (-0.051) та residual_sugar (0.014).



```
In 4  wine.drop(['residual_sugar', 'free_sulfur_dioxide', 'pH'], axis=1, inplace=True)
      wine
```

Executed in 82ms, 5 Apr at 23:02:44

Out 4 — 1599 rows × 9 columns pd.Dataframe

	fixed_acidity	volatile_acidity	citric_acid	chlorides	total_sulfur_dioxide	density	sulphates	alcohol	quality
0	7.4	0.700	0.00	0.076	34.0	0.9978	0.56	9.4	5
1	7.8	0.880	0.00	0.098	67.0	0.9968	0.68	9.8	5
2	7.8	0.760	0.04	0.092	54.0	0.9970	0.65	9.8	5
3	11.2	0.280	0.56	0.075	60.0	0.9980	0.58	9.8	6
4	7.4	0.700	0.00	0.076	34.0	0.9978	0.56	9.4	5
5	7.4	0.660	0.00	0.075	40.0	0.9978	0.56	9.4	5
6	7.9	0.600	0.06	0.069	59.0	0.9964	0.46	9.4	5
7	7.3	0.650	0.00	0.065	21.0	0.9946	0.47	10.0	7
8	7.8	0.580	0.02	0.073	18.0	0.9968	0.57	9.5	7
9	7.5	0.500	0.36	0.071	102.0	0.9978	0.80	10.5	5
10	6.7	0.580	0.08	0.097	65.0	0.9959	0.54	9.2	5
11	7.5	0.500	0.36	0.071	102.0	0.9978	0.80	10.5	5
12	5.6	0.615	0.00	0.089	59.0	0.9943	0.52	9.9	5
13	7.8	0.610	0.29	0.114	29.0	0.9974	1.56	9.1	5

Рис 3.3 – Відкидання малозначущих стовпців

Також можемо вивести матрицю розсіювання.

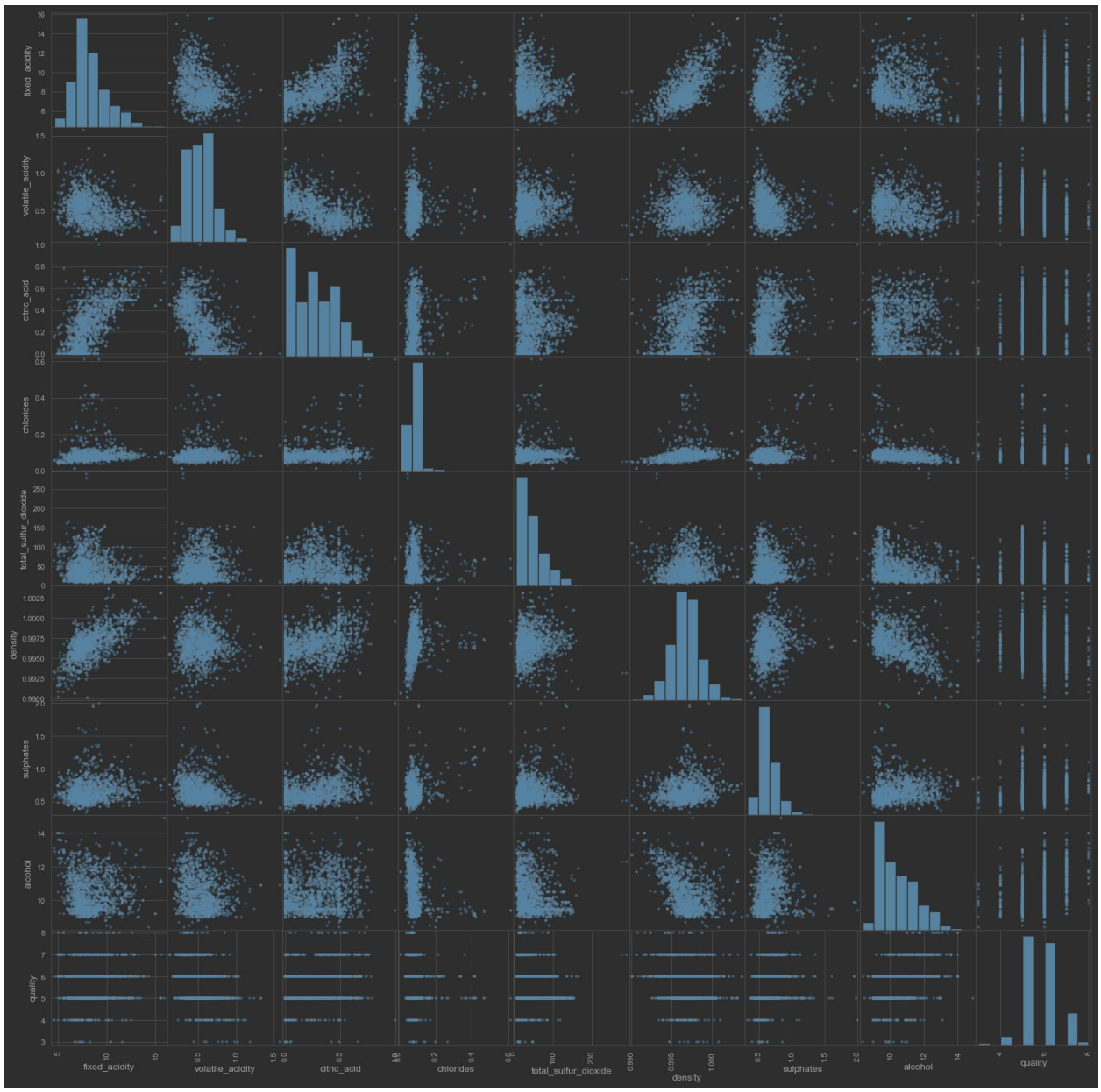


Рис 3.4 – Матриця розсіювання

3.2 Розділити дані на навчальну та тестову вибірки

Для того, щоб розділити дані на навчальну та тестову вибірки, можемо використати функцію .train_test_split() пакету scikit-learn. За замовчуванням, ця функція поділяє дані по відношенню 75% та 25%. Виведемо розмір кожної з вибірок.

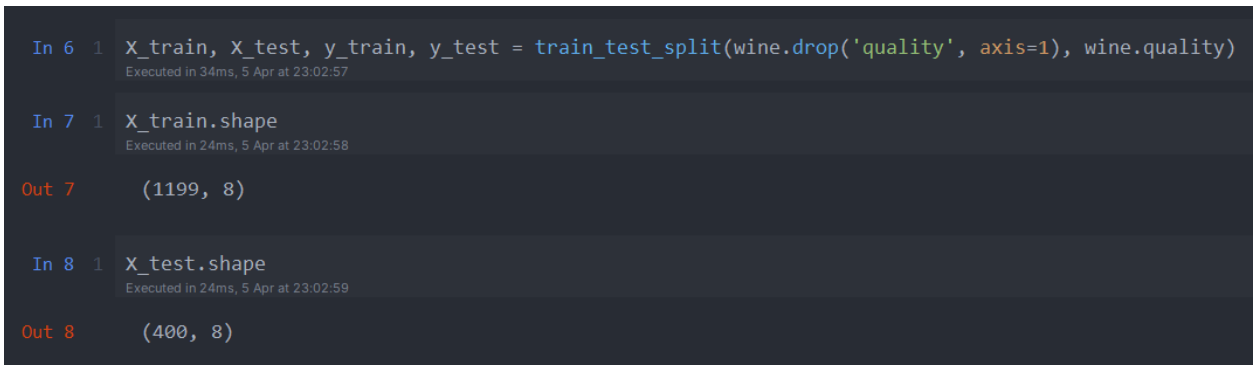


Рис 3.5 – Розділення даних на навчальну та тестову вибірки

3.3 Побудувати декілька регресійних моделей для прогнозу якості вина (12 - quality). Використати лінійну регресію та поліноміальну регресію обраного вами виду

Для початку спробуємо побудувати звичайну лінійну регресію, навчимо її та виведемо діаграму, що відобразити результати.

Як бачимо, регресія не дала високого результату: для всіх вин, що мають якість 7, було спрогнозовано якість від 5.5 до 6.5. Обрахуємо точність моделі за допомогою середньоквадратичної помилки.

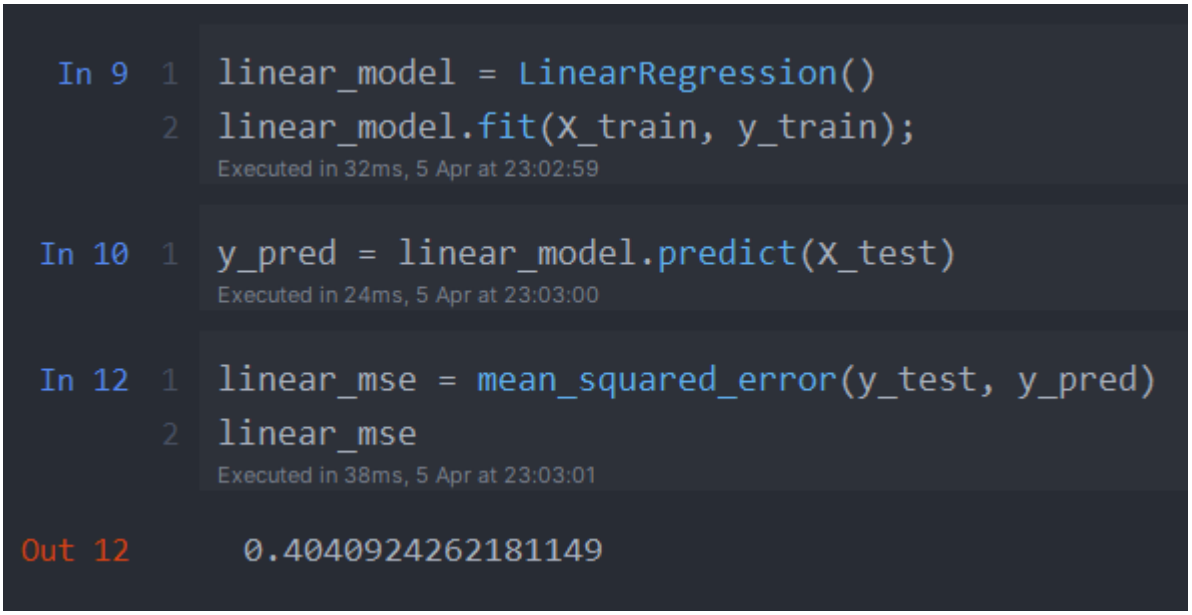


Рис 3.6 – Лінійна регресія

Отримали значення 0.4.

Далі спробуємо використати поліноміальну регресію 1, 2, 3, 4 порядків. Для кожної моделі обрахуємо середньоквадратичну помилку.

In 13

```
degrees = []
mse_list = []
predictions = []

for degree in range(1, 5):
    polynomial_feautures = PolynomialFeatures(degree=degree)

    X_train_polynomial = polynomial_feautures.fit_transform(X_train)
    X_test_polynomial = polynomial_feautures.transform(X_test)

    polynomial_model = LinearRegression()
    polynomial_model.fit(X_train_polynomial, y_train)

    y_pred_polynomial = polynomial_model.predict(X_test_polynomial)

    mse = mean_squared_error(y_test, y_pred_polynomial)

    degrees.append(degree)
    mse_list.append(mse)
    predictions.append(y_pred_polynomial)

poly_df = pd.DataFrame.from_dict({
    'degree': degrees,
    'mse': mse_list,
    'predictions': predictions
})

poly_df.set_index('degree', inplace=True)
poly_df['linear_mse_difference'] = poly_df.mse - linear_mse
poly_df
```

Executed in 299ms, 5 Apr at 23:03:03

Out 13

4 rows × 3 columns pd.Dataframe

degree	mse	predictions	linear_mse_difference
1	0.404092	[5.621657386523987, 5.202717128298772, 5.705981824055218, 5.5302…	-1.054712e-15
2	0.397727	[5.572312988640988, 5.118692113947873, 5.612649590801993, 5.5764…	-6.365436e-03
3	0.722136	[5.607020687653858, 5.211749505797343, 5.597347010472731, 5.7271…	3.180434e-01
4	26.006818	[5.468821479204053, 4.984726058282831, 5.912479274986254, 5.6664…	2.560273e+01

Рис 3.7 – Поліноміальна регресія

Як бачимо, найменша помилка була в поліноміальній регресії першого порядку, яка являє собою просто лінійну регресію. Можемо відобразити отримані дані на діаграмі.

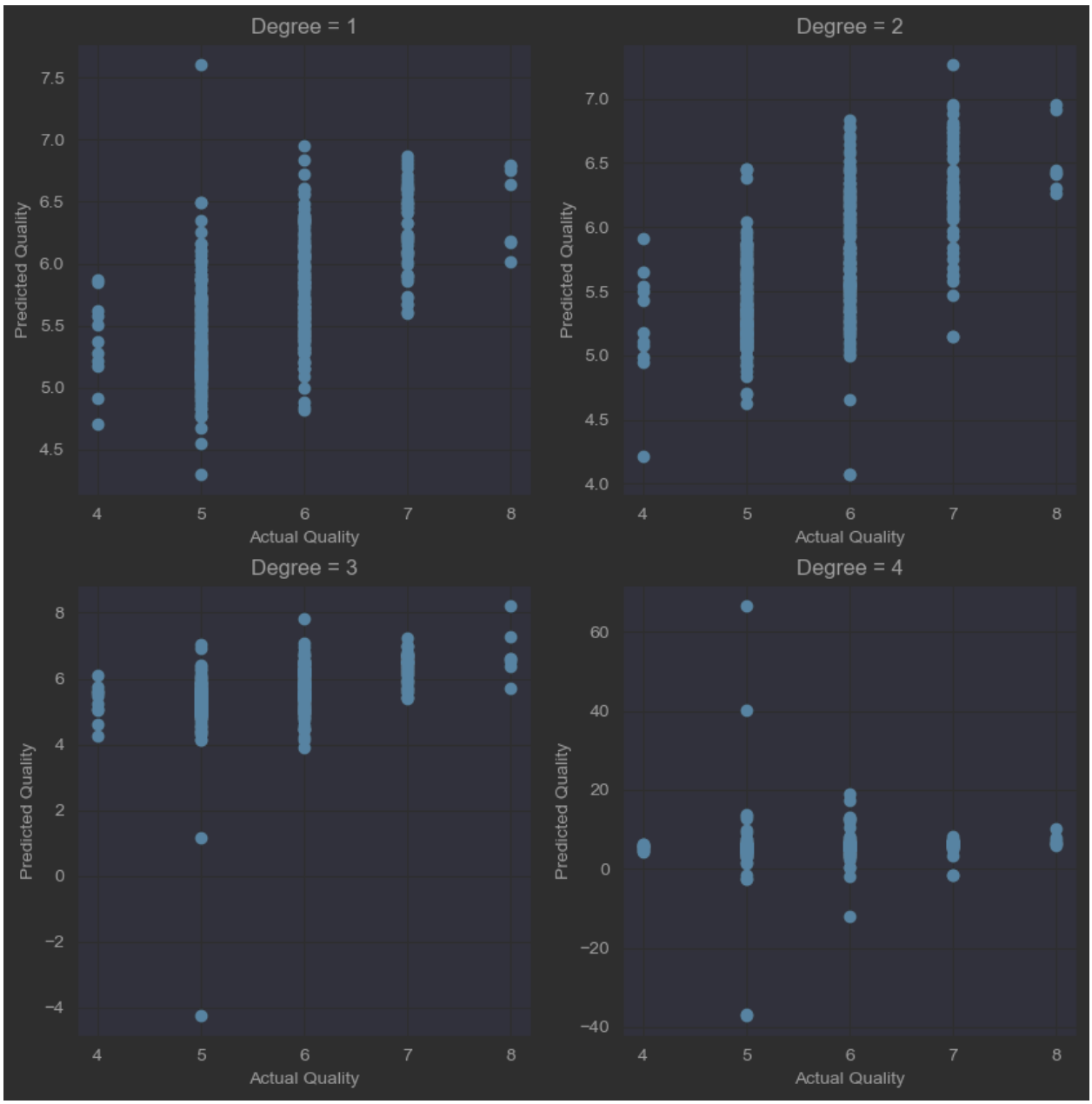


Рис 3.8 – Результати роботи поліноміальної регресії

3.4 Використовуючи тестову вибірку, з'ясувати яка з моделей краща

Як було зазначено в попередньому розділі, найвища точність моделі спостерігалася при використанні лінійної регресії.

4 ВИКОНАННЯ ДОДАТКОВОГО ЗАВДАННЯ

4.1 Дослідити дані, сказати чи є мультиколінеарність, побудувати діаграми розсіювання

Завантаживши дані, було помічено стандартну проблему: всі дійсні числа записані через кому. Виправимо це.

```
In 4
df = pd.read_csv('Data4.csv', delimiter=';', index_col=0, encoding='cp1251')
df
```

Executed in 31ms, 5 Apr at 22:20:21

Out 4 — 11 rows — 132 rows × 6 columns pd.Dataframe

	ISO	UA	Cql	Ie	Iec	Is
Albania	ALB	Албанія	0,97392353	0,605347614	0,538672856	0,510112666
Algeria	DZA	Алжир	0,782134498	0,58721932	0,348159396	0,497985576
Angola	AGO	Ангола	0,372343539	0,27439361	0,332117384	0,346906645
Argentina	ARG	Аргентина	0,883830062	0,699685109	0,28199471	0,518820368
Armenia	ARM	Вірменія	1,016498793	0,718326882	0,535647909	0,486498047
...	...	...	...	...	...	...
Uruguay	URY	Уругвай	1,198779297	0,612819394	0,568066795	0,648818585
Venezuela, Bolivarian Republic of	VEN	Венесуела, Боліварська Республіка	0,703459931	0,670452164	0,249923267	0,367243869
Viet Nam	VNM	В'єтнам	0,553255653	0,371843431	0,39385414	0,423359346
Yemen	YEM	Ємен	0,364579432	0,244049144	0,357659851	0,330743343
Zambia	ZMB	Замбія	0,689322125	0,495702141	0,442525331	0,424261463

```
In 5
df.Cql = df.Cql.astype(str).str.replace(',', '.').astype(float)
df.Ie = df.Ie.astype(str).str.replace(',', '.').astype(float)
df.Iec = df.Iec.astype(str).str.replace(',', '.').astype(float)
df.Is = df.Is.astype(str).str.replace(',', '.').astype(float)
df
```

Executed in 32ms, 5 Apr at 22:20:22

Out 5 — 11 rows — 132 rows × 6 columns pd.Dataframe

	ISO	UA	Cql	Ie	Iec	Is
Albania	ALB	Албанія	0.973924	0.605348	0.538673	0.510113
Algeria	DZA	Алжир	0.782134	0.587219	0.348159	0.497986
Angola	AGO	Ангола	0.372344	0.274394	0.332117	0.346907
Argentina	ARG	Аргентина	0.883830	0.699685	0.281995	0.518820
Armenia	ARM	Вірменія	1.016499	0.718327	0.535648	0.486498
...	...	...	...	...	...	...
Uruguay	URY	Уругвай	1.198779	0.612819	0.568067	0.648819
Venezuela, Bolivarian Republic of	VEN	Венесуела, Боліварська Республіка	0.703460	0.670452	0.249923	0.367244
Viet Nam	VNM	В'єтнам	0.553256	0.371843	0.393854	0.423359
Yemen	YEM	Ємен	0.364579	0.244049	0.357660	0.330743
Zambia	ZMB	Замбія	0.689322	0.495702	0.442525	0.424261

Рис 4.1 – Завантаження навчальних даних, виправлення помилок

Такі ж дії проведемо й для тестового файлу.

In 6
```
df_test = pd.read_csv('Data4t.csv', delimiter=';', index_col=0, encoding='cp1251')
df_test
```
Executed in 61ms, 5 Apr at 22:20:22

Out 6

5 rows × 6 columns pd.Dataframe

	ISO	UA	Cql	Ie	Iec	Is
Togo	TGO	Того	0,45349807	0,216806252	0,368234721	0,433950896
Tunisia	TUN	Туніс	0,899461952	0,659123985	0,418255976	0,514745939
Turkey	TUR	Туреччина	0,859283802	0,498840185	0,509228185	0,499453094
Uganda	UGA	Уганда	0,571284014	0,362945628	0,448731923	0,375726313
Ukraine	UKR	Україна	0,802203636	0,689164485	0,303554874	0,462744168

In 7
```
df_test.Cql = df_test.Cql.astype(str).str.replace(',', '.').astype(float)
df_test.Ie = df_test.Ie.astype(str).str.replace(',', '.').astype(float)
df_test.Iec = df_test.Iec.astype(str).str.replace(',', '.').astype(float)
df_test.Is = df_test.Is.astype(str).str.replace(',', '.').astype(float)
df_test
```
Executed in 71ms, 5 Apr at 22:20:23

Out 7

5 rows × 6 columns pd.Dataframe

	ISO	UA	Cql	Ie	Iec	Is
Togo	TGO	Того	0.453498	0.216806	0.368235	0.433951
Tunisia	TUN	Туніс	0.899462	0.659124	0.418256	0.514746
Turkey	TUR	Туреччина	0.859284	0.498840	0.509228	0.499453
Uganda	UGA	Уганда	0.571284	0.362946	0.448732	0.375726
Ukraine	UKR	Україна	0.802204	0.689164	0.303555	0.462744

Рис 4.2 – Завантаження тестових даних, виправлення помилок

Для дослідження мультиколінеарності можемо вивести теплову карту з коефіцієнтами кореляції.

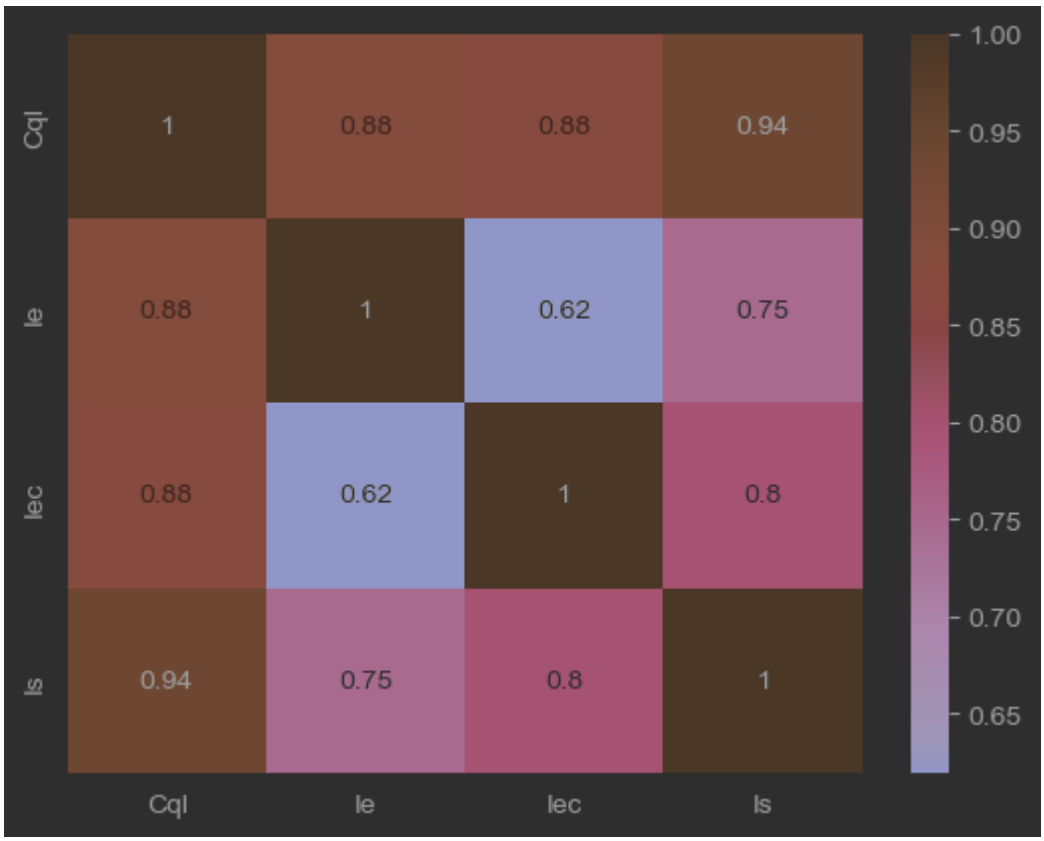


Рис 4.3 – Теплова карта кореляції

Як бачимо, у даному наборі даних є мультиколінеарність: стовпець Cql впливає на стовпці Ie та Iec – коефіцієнти кореляції аж 0.88. Виведемо діаграми розсіювання

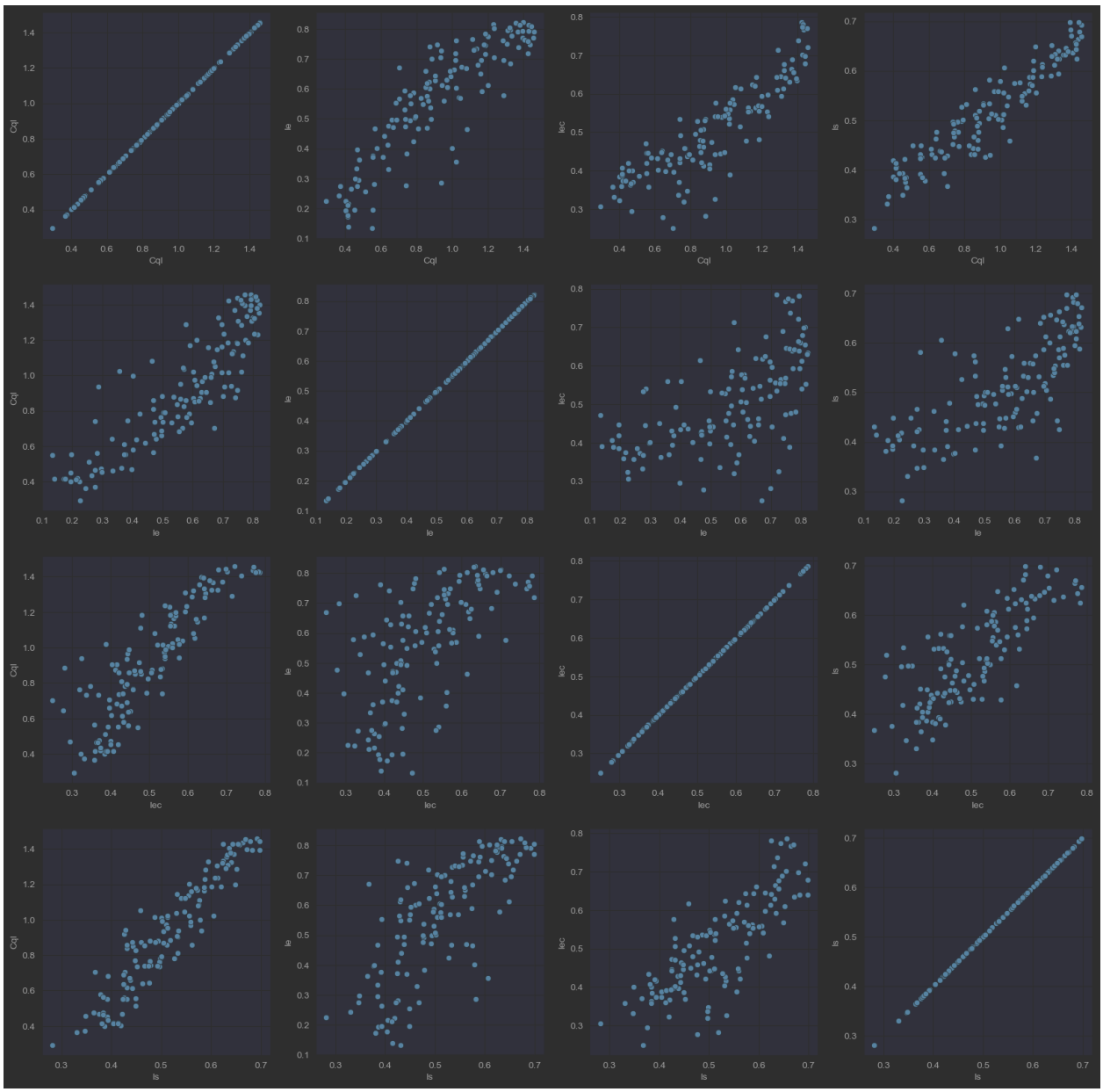


Рис 4.4 – Діаграми розсіювання

4.2 Побудувати декілька регресійних моделей (використати лінійну регресію та поліноміальну регресію обраного вами виду)

Я вирішив не будувати лінійну регресію окремо, а просто побудувати поліноміальну регресію порядку від 1 до 13, оскільки поліноміальна регресія першого порядку, як уже зазначалося, і є лінійною регресією. Для кожного порядку було записано точність, обчислену за допомогою методу .score() регресійної моделі, що використовує коефіцієнт детермінації (R^2).

In 28

```python
degrees = []
accuracies = []
predictions = []

for degree in range(1, 14):
    polynomial_feautures = PolynomialFeatures(degree=degree)

    X_train_polynomial = polynomial_feautures.fit_transform(X_train)
    X_test_polynomial = polynomial_feautures.transform(X_test)

    polynomial_model = LinearRegression()
    polynomial_model.fit(X_train_polynomial, y_train)

    y_pred = polynomial_model.predict(X_test_polynomial)

    degrees.append(degree)
    accuracies.append(polynomial_model.score(X_test_polynomial, y_test))
    predictions.append(y_pred)

poly_df = pd.DataFrame.from_dict({
    'degree': degrees,
    'accuracy': accuracies,
    'predictions': predictions
})

poly_df.set_index('degree', inplace=True)
poly_df
```

Executed in 84ms, 5 Apr at 22:35:39

Out 28

1-11 | 13 rows × 2 columns pd.Dataframe

degree	accuracy	predictions
1	0.899632	[0.41261352337175833, 0.5020957600309625, 0.49...
2	0.977882	[0.42550556733598643, 0.5018778754329956, 0.50...
3	0.985692	[0.4344886237231341, 0.5144497675528893, 0.501...
4	0.989287	[0.43231201171875, 0.51611328125, 0.5000915527...
5	0.990365	[0.43537425994873047, 0.5150671005249023, 0.49...
6	0.997348	[0.433502197265625, 0.5144824981689453, 0.4998...
7	0.982691	[0.4433259963989258, 0.5128030776977539, 0.498...
8	0.986385	[0.4314685598766488, 0.5137725384394258, 0.499...
9	0.950782	[0.4363802690032941, 0.5143344225090374, 0.499...
10	0.945087	[0.4384427513520137, 0.5139127763280982, 0.499...
11	0.971695	[0.43950841303536947, 0.5135539846434227, 0.49...

Рис 4.5 – Поліноміальна регресія

Також для наочності можемо відобразити отримані дані на графіку.

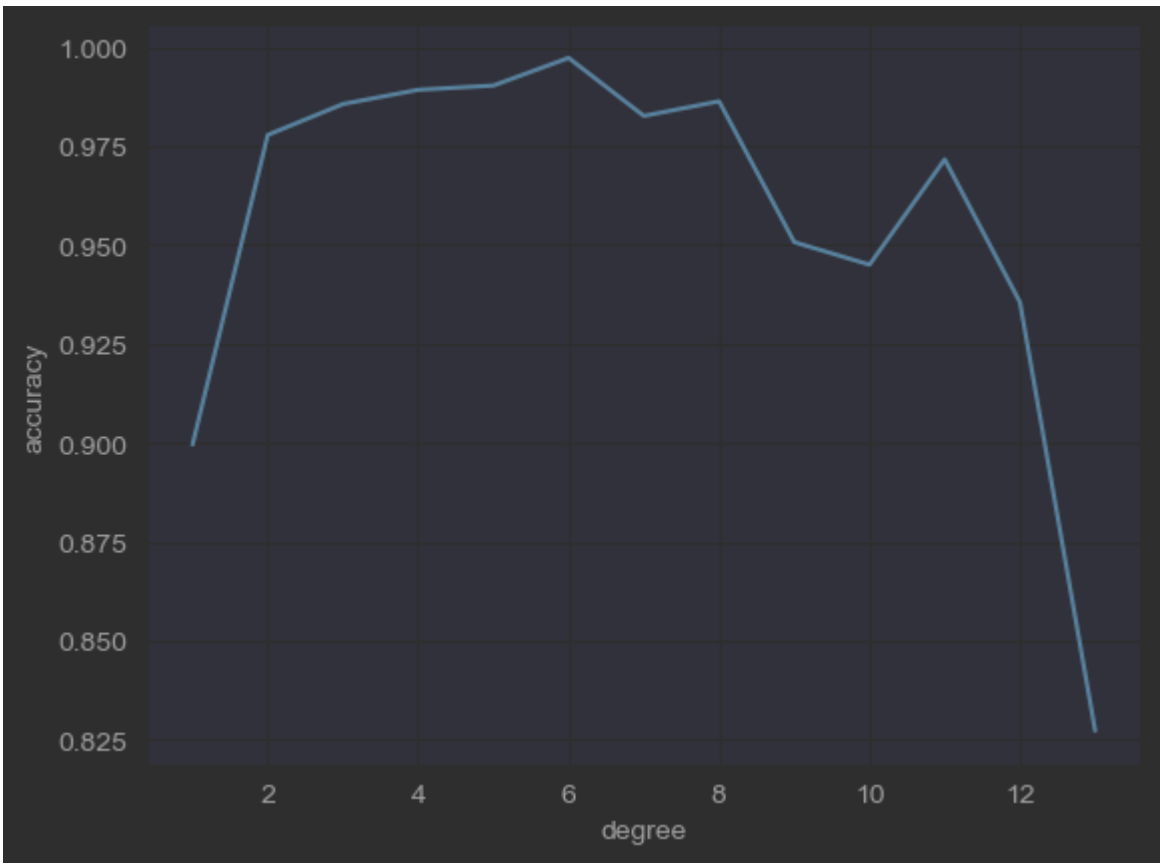


Рис 4.6 – Залежність точності регресійної моделі від порядку

Бачимо, що найвищу точність було отримано при використанні поліноміальної регресії шостого порядку – 99.73%. Можемо також побудувати діаграму розсіювання для отриманих даних.

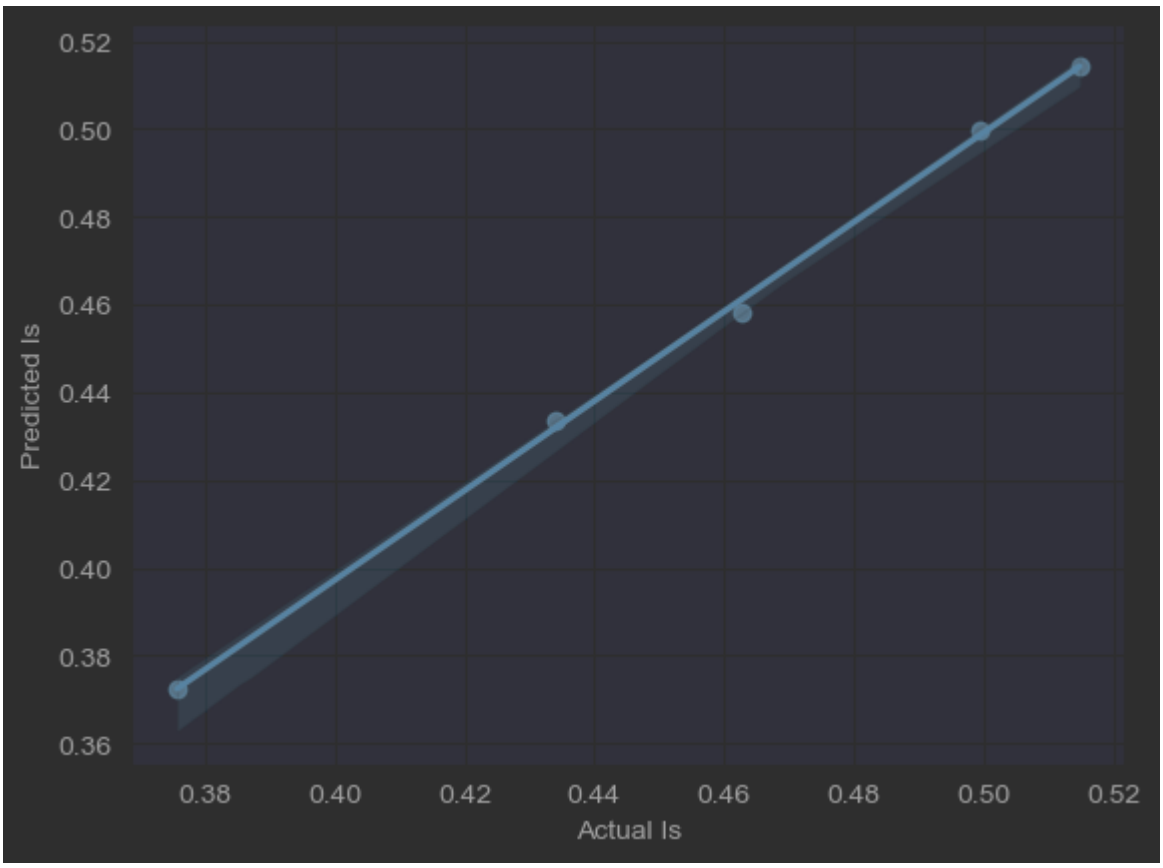


Рис 4.7 – Передбачення поліноміальної регресійної моделі шостого порядку

4.3 Використовуючи тестову вибірку з файлу Data4t.csv, з'ясувати яка з моделей краща

Використовуючи тестову вибірку, було встановлено, що найвищу точність для заданого набору даних має поліноміальна регресія шостого порядку – її оцінка становить 99.73%.

5 ВИСНОВОК

У ході даної лабораторної роботи було досліджено різні види регресії: лінійна та поліноміальна. Було досліджено набір даних якості вина, побудовано різні регресійні моделі для визначення якості вина за іншими показниками. Для тестування моделей набір даних було поділено на навчальну та тестову вибірку. При тестуванні було встановлено, що найкращі передбачення робить звичайна лінійна регресія: її середньоквадратична помилка має значення 0.4.

Далі було досліджено деякий набір даних за країнами на мультиколінеарність, встановлено, що вона присутня – коефіцієнт кореляції для двох пар стовпців дорівнює 0.88 – такий зв’язок вважається сильним. Було побудовано поліноміальні регресії порядку від 1 до 13 та порівняно їхню ефективність. Найкраще себе показала лінійна регресія шостого порядку – R^2 оцінка для неї становить 99.73%.