

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

КУРСОВА РОБОТА

з дисципліни «Аналіз даних в інформаційних системах»

на тему: «Прогнозування якості

вин на основі фізичних якостей»

Студента 2 курсу ІП-13 групи

Спеціальності: 121

«Інженерія програмного забезпечення»

Замковий Дмитро Володимирович

«ПРИЙНЯВ» з оцінкою

доц. Ліхоузова Т.А. / доц. Олійник Ю.О.

Підпис

Дата

Київ – 2023 рік

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Кафедра інформатики та програмної інженерії

Дисципліна Аналіз даних в інформаційно-управляючих системах

Спеціальність 121 "Інженерія програмного забезпечення"

Курс 2 Група ІІІ-13

Семестр 4

ЗАВДАННЯ

на курсову роботу студента

Замковий Дмитро Володимирович

1.Тема роботи Прогнозування якості вин на основі їх фізичних якостей. Методи Extra Trees Regressor, Random Forest Regressor, Linear Regression

2.Строк здачі студентом закінченої роботи 07.06.2022

3. Вхідні дані до роботи методичні вказівки, дані обрані з сайту
<https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009>

4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

1.Постановка задачі

2.Аналіз предметної області

3.Робота з даними

4.Інтелектуальний аналіз даних

5.Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

6.Дата видачі завдання 16.04.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1.	Отримання теми курсової роботи	07.06.2023	
2.	Визначення зовнішніх джерел даних	07.06.2023	
3.	Пошук та вивчення літератури з питань курсової роботи	07.06.2023	
4.	Обґрунтування методів інтелектуального аналізу даних	07.06.2023	
5.	Застосування та порівняння ефективності методів інтелектуального аналізу даних	07.06.2023	
6.	Підготовка пояснювальної записки	07.06.2023	
7.	Здача курсової роботи на перевірку	07.06.2023	

Студент

(підпис)

Замковий Д.В.

(прізвище, ім'я, по батькові)

Керівник

(підпис)

доц. Ліхоузова Т.А

(прізвище, ім'я, по батькові)

Керівник

(підпис)

доц. Олійник Ю.О.

(прізвище, ім'я, по батькові)

"07" червня 2023 р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 26 сторінок, 8 рисунків.

Об'єкт дослідження: інтелектуальний аналіз даних.

Предмет дослідження: створення програмного забезпечення, що проводить аналіз даних з подальшим прогнозуванням та графічним відображенням результатів.

Мета роботи: реалізація програмного забезпечення для отримання прогнозу якості вина по його смаковим характеристикам.

Дана курсова робота включає в себе: опис проектування, опис створення програмного забезпечення для інтелектуального аналізу даних, їх графічного відображення та прогнозування за допомогою різних моделей.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, МОДЕЛЬ EXTRA TREES REGRESSOR, МОДЕЛЬ RANDOM FOREST REGRESSOR, МОДЕЛЬ LINEAR REGRESSOR, ФІЗИЧНІ ЯКОСТІ ВИН, ПРОГНОЗУВАННЯ ДАНИХ.

ЗМІСТ

ВСТУП	6
1.ПОСТАНОВКА ЗАДАЧІ	7
2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
3.РОБОТА З ДАНИМИ	10
3.1. Опис обраних даних.....	10
3.2. Перевірка даних	10
3.3 Поділ даних.....	12
4.ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ.....	13
4.1. Обґрунтування вибору методів інтелектуального аналізу даних.....	13
4.1.1. Extra Trees Regressor:	13
4.1.2. Random Forest Regressor:	13
4.1.3. Linear Regression:	14
4.2. Аналіз отриманих результатів для методу Extra Trees Regressor ...	14
4.3. Аналіз отриманих результатів для методу Random Forest Regressor	16
4.4. Аналіз отриманих результатів для методу Linear Regression.....	17
4.5. Порівняння отриманих результатів методів	19
ВИСНОВКИ.....	21
ПЕРЕЛІК ПОСИЛАНЬ	22
ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ	23

ВСТУП

Вино є одним з найстаріших та найпопулярніших алкогольних напоїв, який має велике значення як в культурному, так і в економічному аспектах. В останні роки споживання вина значно зросло, і все більше споживачів стають вимогливими до якості цього напою. Щоб забезпечити задоволення споживачів та конкурентоспроможність на ринку, винороби повинні зосередитися на якості своїх вин.

В рамках даної курсової роботи були проаналізовані дані оцінки фізичних якостей вин та їх загальна якість. На основі отриманих даних з використанням декількох методів було побудовано та навчено моделі для прогнозування якості вина.

Дана курсова робота була розроблена з використанням технологій та бібліотек Python 3.7[1], sklearn[2], matplotlib[3], pandas[4]

1. ПОСТАНОВКА ЗАДАЧІ

Виконання даної курсової роботи потребує виконання декількох задач: аналізу предметної області;

- роботи з датасетом:
 - завантаження,
 - дослідження його структури та виправлення наявних помилок;
- вибір методів для прогнозування та обґрунтування даного вибору;
- аналіз отриманих результатів кожного з методів та порівняння отриманих результатів ефективності.

Створення застосунку, що поділяє отримані дані на тренувальні та тестові для перевірки декількох методів, у подальшому порівняння ефективності методів.

Для прогнозування буде використано методи Linear Regression, Random Forest Regressor, Extra Trees Regressor. Для кожного методу проаналізувати результати та в кінці порівняти результати роботи цих методів. Обрати найоптимальніший метод для прогнозування рейтингу вина.

Виконане дослідження можна буде використовувати для прогнозування можливого рейтингу вина.

Вхідними даними буде фіксована кислотність, летюча кислотність, лимонна кислота, залишковий цукор, хлориди, вільний діоксид сірки, загальний діоксид сірки, густина, рН, сульфати, спирт, якість вина.

2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Прогнозування якості вин є важливим завданням для виноробів та експертів з винарства. Фізико-хімічні параметри винограду та винного напою можуть бути використані для побудови моделей прогнозування, які допоможуть визначити якість вина.

Один з методів прогнозування якості вина - це використання регресійних моделей. Регресійний аналіз дозволяє знайти залежність між фізико-хімічними параметрами вина (наприклад, рівень кислотності, вміст алкоголю, рН-рівень тощо) та його якістю (наприклад, оцінкою експертів або органолептичними характеристиками).

Для прогнозування якості вина можна використовувати різні моделі регресії. Одними з них є Extra Trees Regressor, Random Forest Regressor та Linear Regression.

Extra Trees Regressor є ансамблевою моделлю, яка базується на використанні багатьох рішень (дерев рішень) та комбінуванні їх прогнозів для отримання кінцевого результату. Ця модель добре справляється з великою кількістю ознак і може бути ефективною в прогнозуванні якості вин на основі фізико-хімічних параметрів.

Random Forest Regressor також є ансамблевою моделлю, яка використовує дерева рішень. Вона випадковим чином вибирає підмножини ознак та будує декілька дерев рішень, а потім комбінує їх прогнози для отримання кінцевого результату. Random Forest Regressor може бути ефективним методом для прогнозування якості вин.

Linear Regression є простою лінійною моделлю, яка шукає лінійну залежність між фізико-хімічними параметрами вина та його якістю. Вона може

бути корисною, коли нам потрібно знайти пряму залежність між ознаками і цільовою змінною.

При розробці програмного забезпечення для прогнозування якості вина, можна використовувати ці моделі прогнозування. Спочатку необхідно завантажити та дослідити структуру датасету, що містить фізико-хімічні параметри вин та оцінки якості. Далі можна використовувати моделі Extra Trees Regressor, Random Forest Regressor та Linear Regression для прогнозування якості вин на основі цих параметрів.

Після прогнозування можна візуалізувати отримані результати та проаналізувати їх. Порівняння використаних методів також є важливим кроком, щоб визначити, яка модель прогнозування є найефективнішою для даної задачі прогнозування якості вина.

Прогнозування якості вин на основі фізико-хімічних якостей може бути корисним інструментом для виноробів, споживачів та експертів з винарства. Воно допомагає зрозуміти вплив фізико-хімічних параметрів на якість вина і може сприяти поліпшенню виробництва та вибору вин.

3.РОБОТА З ДАНИМИ

3.1. Опис обраних даних

Для вирішення поставленої нами задачі був обраний датасет «Red Wine Quality». Даний набір складається з 1599 рядків даних, які є даними фізико-хімічних характеристик по червоному вину. Даний датасет містить в собі таблицю, що складається з 12 стовпців: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, quality. Дані несуть в собі наступну інформацію:

- fixed acidity – фіксована кислотність
- volatile acidity – летюча кислотність
- citric acid – лимонна кислотність
- residual sugar – залишковий цукор
- chlorides – хлориди
- free sulfur dioxide – вільний діоксид сірки
- total sulfur dioxide – загальний діоксид сірки
- density - густина
- pH – водневий показник
- sulphates – сульфати
- alcohol – спирт
- quality – якість

3.2. Перевірка даних

Для роботи з даними на мові Python ми використовуємо бібліотеку «pandas».

Для початку ми зчитуємо дані з файлу та виводимо основну інформацію про наш датафрейм (рис. 3.1).

```
path = r'C:\Users\Dima\source\PC\ADIS\KURsoVA\winequality-red.csv'
dataset = pd.read_csv(path, low_memory=False)
dataset.info()
```

Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	fixed acidity	1599 non-null	float64
1	volatile acidity	1599 non-null	float64
2	citric acid	1599 non-null	float64
3	residual sugar	1599 non-null	float64
4	chlorides	1599 non-null	float64
5	free sulfur dioxide	1599 non-null	float64
6	total sulfur dioxide	1599 non-null	float64
7	density	1599 non-null	float64
8	pH	1599 non-null	float64
9	sulphates	1599 non-null	float64
10	alcohol	1599 non-null	float64
11	quality	1599 non-null	int64

dtypes: float64(11), int64(1)
memory usage: 150.0 KB

Рисунок 3.1 – Загальна інформація про датафрейм

Як можемо побачити – всі дані визначились правильно.

Наступним кроком перевіримо написаними нами функціями, чи зустрічаються неправильні дані. Будемо перевіряти, чи менші наші дані за нуль (рис. 3.2).

```
col = list(ds.columns.values)
for i in col:
    if check_negative(ds, i):
        print('Значень менше нуля не знайдено в колонці', i)
```

```
Значень менше нуля не знайдено в колонці fixed acidity
Значень менше нуля не знайдено в колонці volatile acidity
Значень менше нуля не знайдено в колонці citric acid
Значень менше нуля не знайдено в колонці residual sugar
Значень менше нуля не знайдено в колонці chlorides
Значень менше нуля не знайдено в колонці free sulfur dioxide
Значень менше нуля не знайдено в колонці total sulfur dioxide
Значень менше нуля не знайдено в колонці density
Значень менше нуля не знайдено в колонці pH
Значень менше нуля не знайдено в колонці sulphates
Значень менше нуля не знайдено в колонці alcohol
Значень менше нуля не знайдено в колонці quality
```

Рисунок 3.2 – перевірка на числа менші нуля

Можемо побачити, що від’ємних значень немає, тому можемо перейти до наступного кроку.

Як можемо побачити – всі дані перебувають у гарному стані та не потребують для заміни.

3.3 Поділ даних

Для створення моделей та подальшого тренування розділимо датафрейм на тренувальну та тестову вибірки для подальшої роботи з методами.

```
train, test = train_test_split(ds, test_size=0.2, random_state=1)
X_train = train.drop(columns='quality')
y_train = train['quality']

X_test = test.drop(columns='quality')
y_test = test['quality']
```

Щоб уникнути оверфіту, ми розділили наш набір даних на навчальні та тестові, а саме на 80% тренувальних та 20% даних, на яких буде проводитися тестування. Це дасть нам краще уявлення про те, як наші методи працюють на етапі тестування. Таким чином наші методи тестуються на невидимих даних, для кращого розуміння їх коректності використання для розв’язання поставленої нами задачі.

4.ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

4.1. Обґрунтування вибору методів інтелектуального аналізу даних.

Використання методів Extra Trees Regressor, Random Forest Regressor і Linear Regression у моїй роботі має свої обґрунтування, оскільки кожен з цих методів має свої переваги та особливості.

4.1.1. Extra Trees Regressor:

- Extra Trees Regressor є одним із варіантів ансамблю дерев рішень (ensemble of decision trees) і використовується для завдань регресії.
- Він працює на основі випадкового лісу (random forest) і використовує випадкові підвибірки з дерев для побудови моделі.
- Цей метод дозволяє зменшити варіацію моделі, а також зменшити помилку прогнозування.
- Extra Trees Regressor також враховує важливість ознак (feature importance), що може допомогти визначити найбільш важливі змінні у вашому датасеті.

4.1.2. Random Forest Regressor:

- Random Forest Regressor також є методом ансамблю дерев рішень і широко використовується для задач регресії.
- Він працює на основі багатьох рішень (decision trees) та використовує метод багатократного побудови дерев для отримання більш точного прогнозу.
- Random Forest Regressor дозволяє враховувати взаємодію між ознаками та автоматично обробляти відсутні дані (missing values) у датасеті.

- Цей метод також здатний показувати важливість ознак, що допомагає зрозуміти, які змінні найбільше впливають на прогнозування.

4.1.3. Linear Regression:

- Лінійна регресія є класичним методом для задач прогнозування, особливо коли ви працюєте з неперервними виходами.
- Вона заснована на припущенні, що існує лінійна залежність між вхідними ознаками та виходом.
- Лінійна регресія є простою і інтерпретованою моделлю, яка може дати вам загальне уявлення про вплив кожної змінної на вихід.
- Вона може бути основою для порівняння з більш складними моделями та дозволяє оцінити важливість кожної ознаки на основі коефіцієнтів регресії.

4.2. Аналіз отриманих результатів для методу Extra Trees Regressor

Спочатку ініціалізуємо модель

```
# Ініціалізація моделі Extra Trees Regressor
extra_tree_reg = ExtraTreesRegressor()
```

Далі задамо дані для тренування моделі

```
# Навчання моделі на тренувальних даних
extra_tree_reg.fit(X_train, y_train)
y_predict_lin = extra_tree_reg.predict(X_test)
```

Наступним кроком порахуємо MSE і R^2 для обрахунку точності навчання моделі. Результат обрахунку наведений на рисунку 4.1.

```
# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
print('Extra Trees Regressor')
print('MSE:', mse)
print('R^2:', r2, '\n')
```

```
Extra Trees Regressor
MSE: 0.294058125
R^2: 0.48310785340314133
```

Рисунок 4.1 – Результат роботи Extra Trees Regressor

Потім виведемо графік відповідності справжнім і передбаченим значенням. Графік наведений на рисунку 4.2.

```
fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()
```

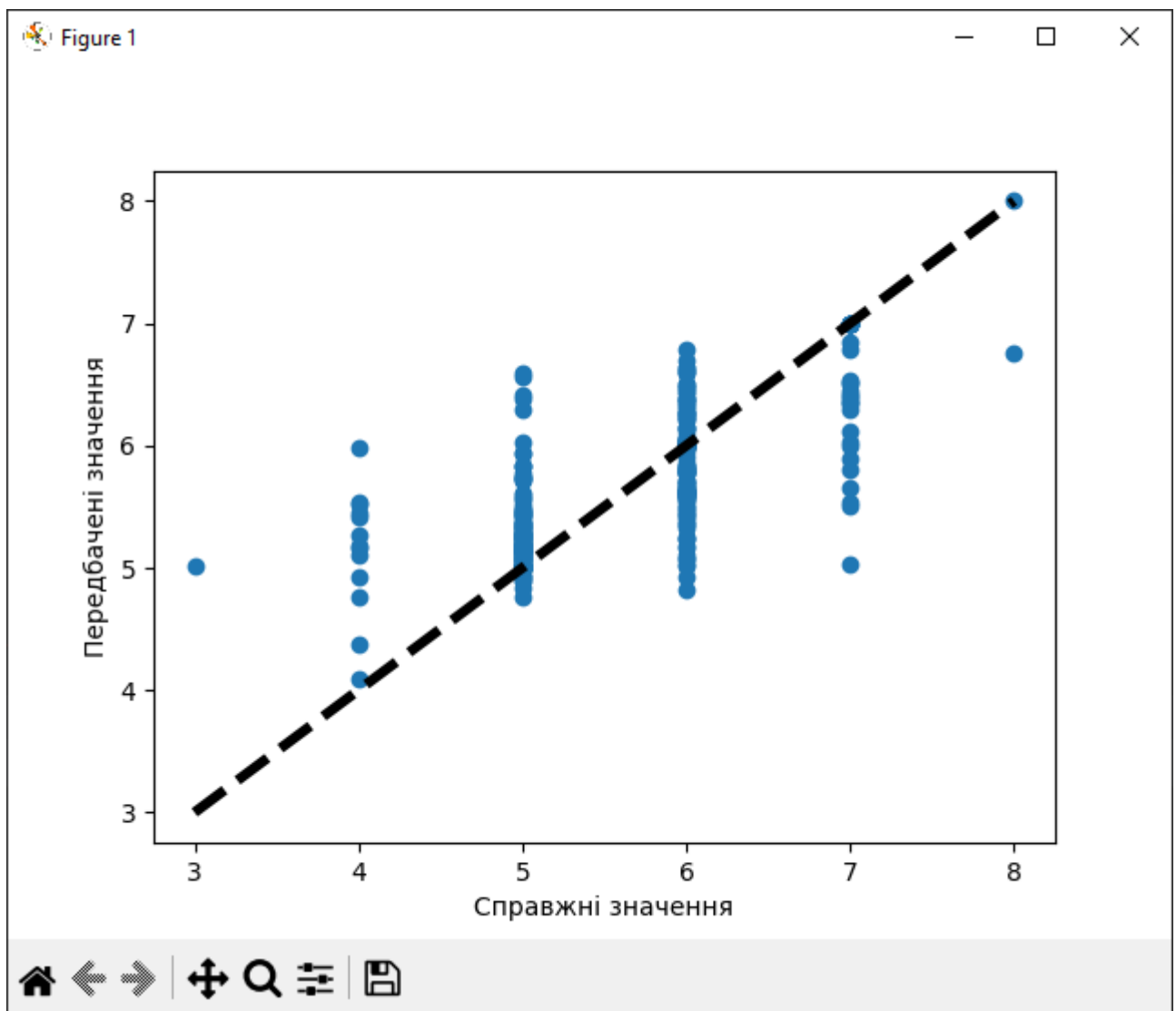


Рисунок 4.2 – Графік роботи моделі Extra Trees Regressor

4.3. Аналіз отриманих результатів для методу Random Forest Regressor

Спочатку ініціалізуємо модель

```
# Ініціалізація моделі Random Forest Regressor
rand_forest_reg = RandomForestRegressor()
```

Далі задамо дані для тренування моделі

```
# Навчання моделі на тренувальних даних
rand_forest_reg.fit(X_train, y_train)
y_predict_lin = rand_forest_reg.predict(X_test)
```

Наступним кроком порахуємо MSE і R^2 для обрахунку точності навчання моделі. Результат обрахунку наведений на рисунку 4.3.

```
# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
print('Random Forest Regressor')
print('MSE:', mse)
print('R^2:', r2, '\n')
```

```
Random Forest Regressor
MSE: 0.317150625
R^2: 0.4425161102051326
```

Рисунок 4.3 – Результат роботи Random Forest Regressor

Потім виведемо графік відповідності справжнім і передбаченим значенням. Графік наведений на рисунку 4.4.

```
fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()
```

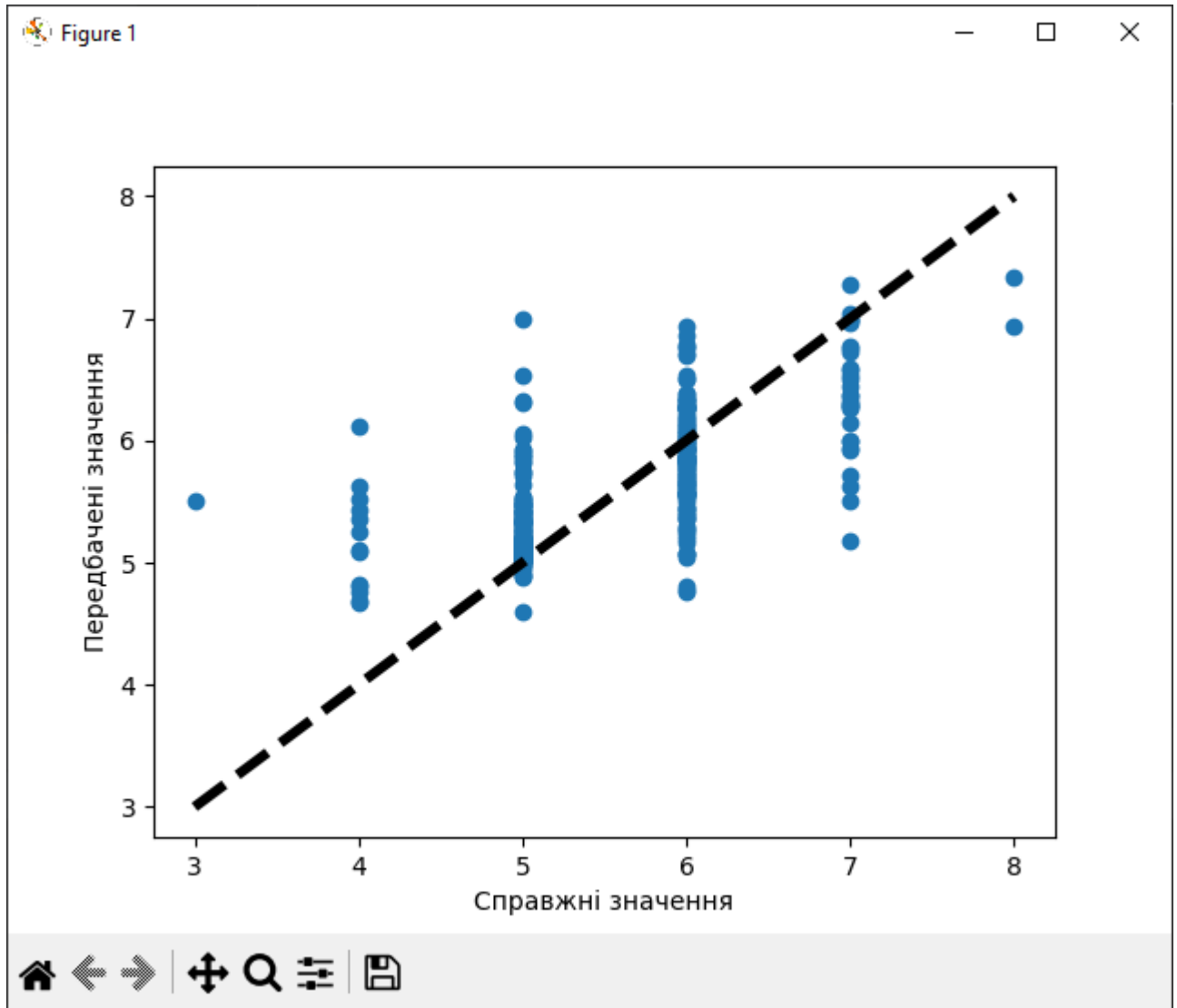



Рисунок 4.4 – Графік роботи моделі Random Forest Regressor

4.4. Аналіз отриманих результатів для методу Linear Regression

Спочатку ініціалізуємо модель

```
# Ініціалізація моделі Linear Regression
lin_reg = LinearRegression()
```

Далі задамо дані для тренування моделі

```
# Навчання моделі на тренувальних даних
rand_forest_reg.fit(X_train, y_train)
y_predict_lin = rand_forest_reg.predict(X_test)
```

Наступним кроком порахуємо MSE і R^2 для обрахунку точності навчання моделі. Результат обрахунку наведений на рисунку 4.5.

```
# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
```

```
print('Linear Regression')
print('MSE:', mse)
print('R^2:', r2, '\n')
```

```
Linear Regression
MSE: 0.38307198158142836
R^2: 0.32664027269868234
```

Рисунок 4.3 – Результат роботи Linear Regression

Потім виведемо графік відповідності справжнім і передбаченим значенням. Графік наведений на рисунку 4.6.

```
fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()
```

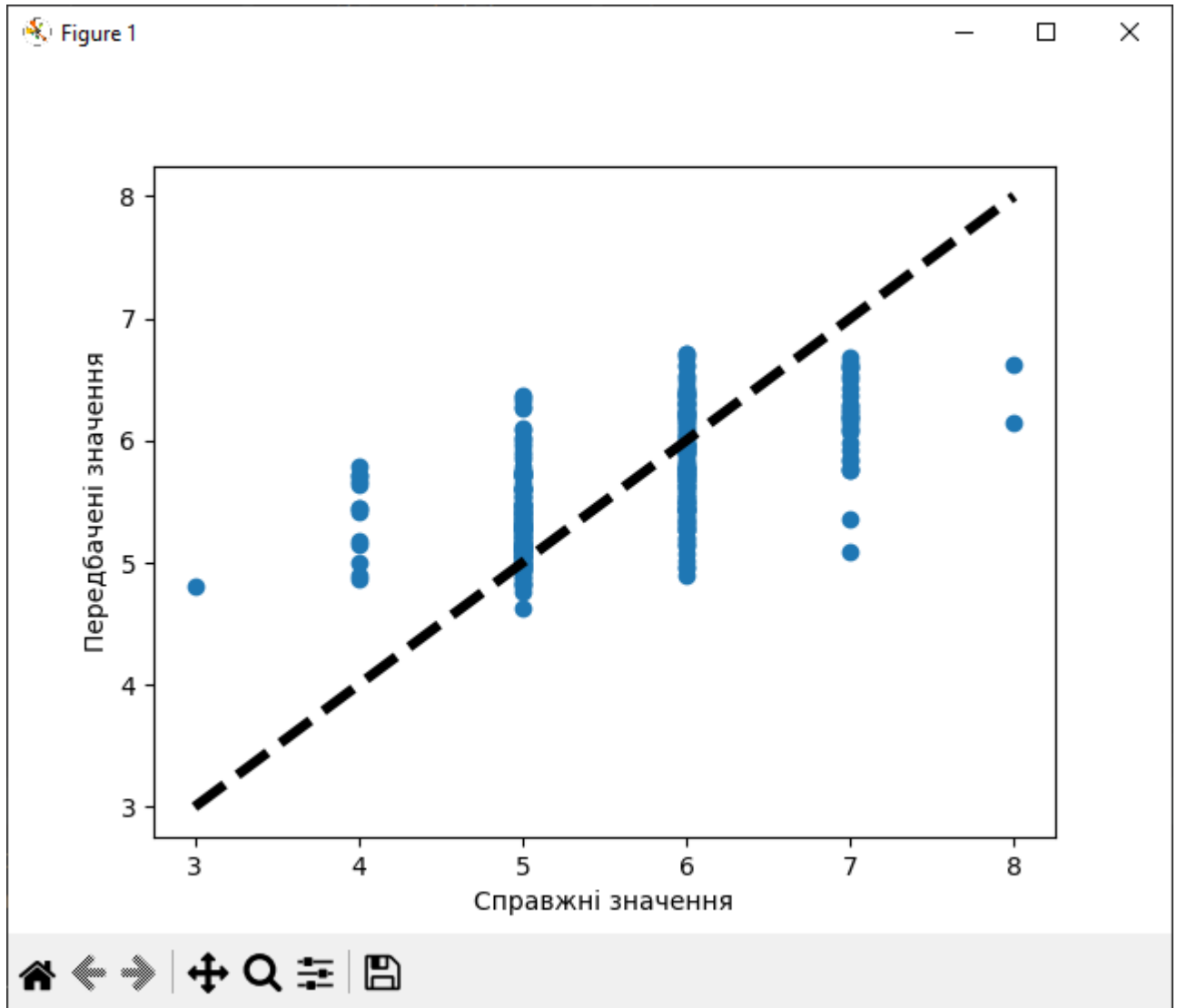


Рисунок 4.6 – Графік роботи моделі Linear Regression

4.5. Порівняння отриманих результатів методів

Проаналізувавши окремо кожен із методів, що були мною використані під час прогнозування загальної якості вина, варто провести порівняння даних методів.

З усіх трьох методів ми маємо найменшу MSE та найбільший R^2 у метода Extra Trees Regressor, а саме 0.2919759375 та 0.48676789975109436 відповідно.

На другому місці знаходить метод Random Forest Regressor в якого MSE та R^2 відповідно дорівнюють 0.317150625 та 0.4425161102051326.

Найгірше себе показав метод Linear Regression, значення MSE та R^2 якого дорівнюють 0.38307198158142836 та 0.32664027269868234 відповідно.

Також ми можемо оцінити графіки роботи методів та візуально визначити, що найкраще була натренована модель Extra Trees Regressor, на другому місці – Random Forest Regressor, і на останньому – Linear Regression, що в цілому і відповідає значенням MSE та R^2 кожної моделі.

Щодо графіків варто зазначити, що скупчення значень у вигляді вертикальних полосок обумовлене тим, що оцінка значень проходить по шкалі від 0 до 10 в цілих числах, а тобто і значення просто дуже «розтягнуті» по графіку.

ВИСНОВКИ

У даній курсовій роботі був проведений аналіз якості вин на основі їх фізичних характеристик з метою встановлення зв'язку між цими параметрами та загальною якістю вина. Для досягнення цієї мети були зібрані та проаналізовані дані про фізичні характеристики великої кількості вин.

В результаті було натреновано три моделі – Extra Trees Regressor, Random Forest Regressor і Linear Regression і проаналізувавши їх роботу було визначено те, що найкраще із поставленою задачею справляється метод Extra Trees Regressor. Також було визначено, що загальна якість вина досить тісно залежить від його фізико-хімічних параметрів.

Отримані результати можуть мати практичне значення для виноробних підприємств, ресторанів, сомельє та споживачів вина. Вони допоможуть виноробам покращити якість своїх вин, забезпечуючи задоволення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Python Release Python 3.10.0. *Python.org*. URL: <https://www.python.org/downloads/release/python-3100/> (дата звернення: 07.06.2023).
2. scikit-learn: machine learning in Python – scikit-learn 1.2.2 documentation. *scikit-learn: machine learning in Python – scikit-learn 0.16.1 documentation*. URL: <https://scikit-learn.org/stable/> (дата звернення: 07.06.2023).
3. Matplotlib – Visualization with Python. *Matplotlib – Visualization with Python*. URL: <https://matplotlib.org/> (дата звернення: 07.06.2023).
4. pandas - Python Data Analysis Library. *pandas - Python Data Analysis Library*. URL: <https://pandas.pydata.org/> (дата звернення: 07.06.2023).
5. Red Wine Quality. *Kaggle: Your Machine Learning and Data Science Community*. URL: <https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009> (дата звернення: 07.06.2023).

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

*Тексти програмного коду прогнозування доходів та факторів
на популярність індустрії відеоігор*

(Найменування програми (документа))

Онлайн-репозиторій
(Вид носія даних)

3 аркуша
(Обсяг програми (документа). арк..)

студента групи ІП-13 ІІ курсу

Замкового Д. В.

```

from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import ExtraTreesRegressor,
RandomForestRegressor
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import pandas as pd
import time

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

def check_negative(ds, column):
    for i, row in ds.iterrows():
        if row[column] < 0:
            print('Знайдені значення менше нуля в
колонці', column)
            return False
    return True

def check_data(ds):
    ds.info()
    ds.hist(figsize=(8, 8))
    plt.show()
    col = list(ds.columns.values)
    for i in col:
        if check_negative(ds, i):
            print('Значень менше нуля не знайдено в
колонці', i)
    return ds

def train_model(ds):
    train, test = train_test_split(ds, test_size=0.2,
random_state=1)
    X_train = train.drop(columns='quality')
    y_train = train['quality']

    X_test = test.drop(columns='quality')
    y_test = test['quality']

    # Ініціалізація моделі Extra Trees Regressor
    extra_tree_reg = ExtraTreesRegressor()

```



```

# Навчання моделі на тренувальних даних
extra_tree_reg.fit(X_train, y_train)
y_predict_lin = extra_tree_reg.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
print('Extra Trees Regressor')
print('MSE:', mse)
print('R^2:', r2, '\n')

fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()

# Ініціалізація моделі Random Forest Regressor
rand_forest_reg = RandomForestRegressor()

# Навчання моделі на тренувальних даних
rand_forest_reg.fit(X_train, y_train)
y_predict_lin = rand_forest_reg.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
print('Random Forest Regressor')
print('MSE:', mse)
print('R^2:', r2, '\n')

fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()

# Ініціалізація моделі Linear Regression
lin_reg = LinearRegression()

# Навчання моделі на тренувальних даних

```

```

lin_reg.fit(X_train, y_train)
y_predict_lin = lin_reg.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_predict_lin)
r2 = r2_score(y_test, y_predict_lin)
print('Linear Regression')
print('MSE:', mse)
print('R^2:', r2, '\n')

fig, ax = plt.subplots()
ax.scatter(y_test, y_predict_lin)
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Справжні значення')
ax.set_ylabel('Передбачені значення')
plt.show()

return ds, extra_tree_reg, rand_forest_reg, lin_reg

if __name__ == '__main__':
    start = time.perf_counter()
    path =
r'C:\Users\Dima\source\PC\ADIS\KURsoVA\winequality-
red.csv'
    dataset = pd.read_csv(path, low_memory=False)
    dataset = check_data(dataset)

    dataset = train_model(dataset)[0]
    print(f"Time taken is {int(time.perf_counter() - start)}s")

```