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: «Передбачення віку краба на основі
його фізичних параметрів»

Студента 2 курсу ІП-13 групи
Спеціальності: 121
«Інженерія програмного забезпечення»
Шевцової Анастасії Андріївни

«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 ІП-13 Семестр 4

ЗАВДАННЯ

на курсову роботу студента

Шевцової Анастасії Андріївни

1.Тема роботи Передбачення віку краба на основі його фізичних параметрів

2.Строк здачі студентом закінченої роботи 08.06.2022

3. Вхідні дані до роботи методичні вказівки до курсової робота, обрані дані з сайту https://www.kaggle.com/datasets/sidhus/crab-age-prediction

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

1.Постановка задачі

2.Аналіз предметної області

3.Розробка сховища даних

4.Інтелектуальний аналіз даних

5.Висновки

4.Перелік посилань

4.Додаток А

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

6.Дата видачі завдання 30.03.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1.	Отримання теми курсової роботи	30.03.2023	
2.	Визначення зовнішніх джерел даних	01.05.2023	
3.	Пошук та вивчення літератури з питань курсової роботи	01.05.2023	
4.	Обґрунтування методів інтелектуального аналізу даних	01.05.2023	
5.	Застосування та порівняння ефективності методів інтелектуального аналізу даних	08.06.2023	
6.	Підготовка пояснювальної записки	08.06.2023	
7.	Здача курсової роботи на перевірку	08.06.2023	
8.	Захист курсової роботи	08.06.2023	

Студент _____ Шевцова А. А.

(підпис) (прізвище, ім'я, по батькові)

Керівник _____ доц. Ліхоузова Т.А

(підпис) (прізвище, ім'я, по батькові)

Керівник _____ доц. Олійник Ю.О.

(підпис) (прізвище, ім'я, по батькові)

"08" червня 2023 р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 24 сторінок, 11 рисунки, 5 посилань.

Об’єкт дослідження: інтелектуальний аналіз даних.

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

Мета роботи: розробка моделі машинного навчання, яка здатна передбачати вік крабів на основі їх фізичних характеристик

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.

МОДЕЛЬ ПРОГНОЗУВАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, МОДЕЛЬ LINEAR REGRESSION, МОДЕЛЬ RANDOM FOREST, МОДЕЛЬ K-NEAREST NEIGHBORS REGRESSION.

ЗМІСТ

ВСТУП

Визначення віку тварин є важливою задачею в багатьох наукових і промислових галузях, таких як аквакультура, екологія та риболовля. Існують різні методи визначення віку тварин, проте деякі з них можуть бути часо- та працезатратними. Тому пошук альтернативних методів, зокрема використання моделей машинного навчання, є актуальною та перспективною задачею.

У цьому дослідженні ми зосереджуємося на передбаченні віку крабів на основі їх фізичних параметрів. Краби є важливими морськими організмами, які використовуються в комерційних та наукових цілях. Вік крабів є критичною характеристикою для багатьох досліджень, таких як популяційні структури, вікові розподіли та розвиток рибальських стратегій.

Метою цієї роботи є розробка моделі машинного навчання, яка здатна передбачати вік краба на основі його фізичних параметрів. Для досягнення цієї мети будуть використані наявні дані про фізичні характеристики крабів, такі як довжина, діаметр, висота та вага, а також дані про вагу шкаралупи, вагу м'яса та вагу внутрішніх органів. Застосовуючи методи машинного навчання, такі як лінійна регресія, дерева рішень або нейронні мережі, буде розроблена модель, яка навчатиметься передбачати вік краба на основі цих фізичних параметрів.

Для реалізації застосунку були використана мова програмування Python3[2] та бібліотеки Pandas[3], Matplotlib[4], Sklearn[5].

1.ПОСТАНОВКА ЗАДАЧІ

У даній курсовій роботі метою є розробка моделі машинного навчання для передбачення віку краба на основі його фізичних параметрів. Для досягнення цієї мети потрібно виконати наступні кроки.

Початковим етапом є збір та підготовка даних. Завданням є отримання набору даних, що містить інформацію про фізичні характеристики крабів, такі як довжина, діаметр, висота, вага, вага шкаралупи, вага м'яса та вага внутрішніх органів. Необхідно провести аналіз даних, виявити та вирішити можливі проблеми, такі як відсутні значення, викиди або неоднорідність даних.

Наступним етапом є вибір підходу до моделювання. Розглядаються різні моделі машинного навчання, такі як лінійна регресія, дерева рішень, випадковий ліс або нейронні мережі. Метою є вибір найбільш підходящої моделі для передбачення віку краба на основі фізичних параметрів.

Після вибору моделі, необхідно розбити набір даних на тренувальну та тестову вибірки. Тренувальна вибірка буде використана для навчання моделі, тестова - для оцінки її ефективності та точності передбачень.

Далі, модель машинного навчання навчається на тренувальних даних, використовуючи відповідну архітектуру та алгоритми. Оптимізуються параметри моделі, щоб мінімізувати помилку передбачення віку краба.

Після навчання моделі, оцінюється її ефективність на тестовій вибірці. Використовуються метрики, такі як Коефіцієнт детермінації (R^2) та середньоквадратична помилка (MSE), для вимірювання точності та якості передбачень моделі.

Завершальним етапом є порівняння результатів розробленої моделі з існуючими методами визначення віку крабів, такими як підрахунок років за кільцями на раковині. Оцінюються переваги та недоліки розробленої моделі в порівнянні з традиційними методами.

2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Визначення віку краба є важливим завданням у біологічних дослідженнях, рибальстві та аквакультурі. Традиційно вік краба визначають, спираючись на кільця на його раковині. Однак, цей метод може бути складним, часоємким та піддається помилкам. Тому розвиток автоматизованих методів, зокрема моделей машинного навчання, для передбачення віку крабів на основі їх фізичних параметрів є актуальним напрямом досліджень.

Фізичні параметри крабів, такі як довжина, діаметр, висота та вага, можуть бути корисними ознаками для передбачення їх віку. Науковці виявили, що ці параметри можуть відрізнятися в залежності від віку краба. Наприклад, зі зростанням віку, краби можуть збільшувати свою довжину та вагу. Також відомо, що стать краба може впливати на його фізичні характеристики. Чоловічі краби, як правило, мають більші розміри ніж жіночі.

Застосування моделей машинного навчання для передбачення віку краба на основі фізичних параметрів може мати кілька переваг. По-перше, це може бути швидшим та ефективнішим способом визначення віку, порівняно з традиційними методами. По-друге, моделі машинного навчання можуть виявити складні неоднорідності у взаємозв'язку між фізичними параметрами та віком, які можуть бути складні для спостереження людським оком.

У програмному забезпеченні буде реалізовано наступну функціональність, що включає в себе:

- інтелектуальний аналіз даних;
- використання декількох моделей прогнозування даних;
- прогнозування віку краба;
- графічне відображення отриманих результатів та їх аналіз.

3. РОБОТА З ДАНИМИ

3.1 Опис обраних даних

Для виконання курсової роботи було обрано джерело відкритих даних на сайті https://www.kaggle.com//:

– Фізичні ознаки крабів, знайдених в районі Бостона: https://www.kaggle.com/datasets/sidhus/crab-age-prediction

Даний набір складається з 3843 рядків даних, які є даними фізичних ознак крабів. Даний датасет містить в собі таблицю, що складається з 9 стовпців: Sex, Length, Diameter, Height, Weight, Shucked Weight, Viscera Weight, Shell Weight, Age. Дані несуть в собі наступну інформацію:

- Sex – стать краба - чоловіча, жіноча та невизначена.
- Length – довжина краба (у футах; 1 фут = 30,48 см);
- Diameter – діаметр краба (у футах; 1 фут = 30,48 см);
- Height – висота краба (у футах; 1 фут = 30,48 см);
- Weight – вага краба (в унціях; 1 фунт = 16 унцій);
- Shucked Weight – вага без оболонки (в унціях; 1 фунт = 16 унцій);
- Viscera Weight – це вага, яка оточує органи черевної порожнини в глибині тіла (в унціях; 1 фунт = 16 унцій);
- Shell Weight – вага оболонки (в унціях; 1 фунт = 16 унцій);
- Age – вік краба (у місяцях).

3.2 Перевірка даних

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

Використаємо наступний скрипт для початкового аналізу датасету.

```
d_path = r'C:\Users\nasty\OneDrive - kpi.ua\KPI\курсова 4
сем\CrabAgePrediction.csv'
data_frame = pd.read_csv(d_path, sep=',', decimal='.')

# аналіз початкового датасету
print('\nІнформація про датасет:')
data_frame.info()
print('\nПерші 5 рядків:')
print(data_frame.head())
```

```
print('\nКолонки з пропущеними значеннями:')
print(data_frame.isna().any())
print('\nОпис датасету:')
print(data_frame.describe())
```

Виведемо основну інформацію про датасет.

```
Інформація про датасет:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3893 entries, 0 to 3892
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Sex             3893 non-null   object
 1   Length          3893 non-null   float64
 2   Diameter        3893 non-null   float64
 3   Height          3893 non-null   float64
 4   Weight          3893 non-null   float64
 5   Shucked Weight  3893 non-null   float64
 6   Viscera Weight  3893 non-null   float64
 7   Shell Weight    3893 non-null   float64
 8   Age             3893 non-null   int64
dtypes: float64(7), int64(1), object(1)
memory usage: 273.9+ KB
```

Рисунок 3.1 – Інформація про датасет

З результатів бачимо, що тип колонок визначено правильно та всі дані non-null.

Виведемо перші 5 рядків датасету для того щоб переглянути їхній вміст.

```
Перші 5 рядків:
  Sex  Length  Diameter  Height     Weight  Shucked Weight  Viscera Weight  \
0   F  1.4375    1.1750  0.4125  24.635715       12.332033        5.584852
1   M  0.8875    0.6500  0.2125   5.400580        2.296310        1.374951
2   I  1.0375    0.7750  0.2500   7.952035        3.231843        1.601747
3   F  1.1750    0.8875  0.2500  13.480187        4.748541        2.282135
4   I  0.8875    0.6625  0.2125   6.903103        3.458639        1.488349

   Shell Weight  Age
0      6.747181    9
1      1.559222    6
2      2.764076    6
3      5.244657   10
4      1.700970    6
```

Рисунок 3.2 – Перші 5 рядків

Перевіримо чи є колонки з пропущеними значеннями.

```
Колонки з пропущеними значеннями:
Sex               False
Length            False
Diameter          False
Height            False
Weight            False
Shucked Weight    False
Viscera Weight    False
Shell Weight      False
Age               False
dtype: bool
```

Рисунок 3.3 – Колонки з пропущеними значеннями

Як бачимо таких колонок немає.

Виведемо загальний опис датасету.

```
Опис датасету:
            Length     Diameter       Height       Weight  Shucked Weight
count  3893.000000  3893.000000  3893.000000  3893.000000     3893.000000  \
mean      1.311306     1.020893     0.349374    23.567275       10.207342
std       0.300431     0.248233     0.104976    13.891201        6.275275
min       0.187500     0.137500     0.000000     0.056699        0.028349
25%       1.125000     0.875000     0.287500    12.672227        5.343881
50%       1.362500     1.062500     0.362500    22.792998        9.539607
75%       1.537500     1.200000     0.412500    32.786197       14.273973
max       2.037500     1.625000     2.825000    80.101512       42.184056

       Viscera Weight  Shell Weight          Age
count     3893.000000   3893.000000  3893.000000
mean         5.136546      6.795844     9.954791
std          3.104133      3.943392     3.220967
min          0.014175      0.042524     1.000000
25%          2.664853      3.713785     8.000000
50%          4.861939      6.662133    10.000000
75%          7.200773      9.355335    11.000000
max         21.545620     28.491248    29.000000
```

Рисунок 3.4 – Опис датасету

Замінимо колонки з рядковими значеннями на числові за допомогою наступного скрипту.

```
def replace_with_unique_numbers(data, columns):
    for column in columns:
        data[column], _ = pd.factorize(data[column])
    return data
```

```
# заміняємо колони зі строковими значеннями на числові
columns_to_replace = ['Sex']
data_frame = replace_with_unique_numbers(data_frame, columns_to_replace)
print('\nПерші 5 рядків:')
print(data_frame.head())
```

Після заміни датасет має наступний вигляд.

```
Перші 5 рядків:
   Sex  Length  Diameter  Height     Weight  Shucked Weight  Viscera Weight  \
0    0  1.4375    1.1750  0.4125  24.635715       12.332033        5.584852
1    1  0.8875    0.6500  0.2125   5.400580        2.296310        1.374951
2    2  1.0375    0.7750  0.2500   7.952035        3.231843        1.601747
3    0  1.1750    0.8875  0.2500  13.480187        4.748541        2.282135
4    2  0.8875    0.6625  0.2125   6.903103        3.458639        1.488349

   Shell Weight  Age
0      6.747181    9
1      1.559222    6
2      2.764076    6
3      5.244657   10
4      1.700970    6
```

Рисунок 3.5 – Перші 5 рядків

3.3 Поділ даних

Розділимо дані на навчальну і тестову вибірки. Для цього напишемо наступний скрипт.

```
# розділення на навчальну і тестову вибірки
x = data_frame.drop(columns='Age')
y = data_frame['Age']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42)
```

Для уникнення перенавчання (overfitting) ми провели поділ нашого набору даних на тренувальну та тестову вибірки в співвідношенні 80% до 20% відповідно. Це дозволяє нам оцінити ефективність наших методів на тестовій вибірці, яка є невидимою під час навчання. Такий підхід дозволяє краще зрозуміти, наскільки наші методи будуть коректно працювати для вирішення поставленої задачі.

4.ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

4.1 Обґрунтування вибору методів інтелектуального аналізу даних

Лінійна регресія (Linear Regression) є одним з найпростіших та найбільш поширених методів для моделювання залежності між залежною та незалежними змінними. У нашому випадку, ми хочемо передбачити вік краба на основі його фізичних параметрів. Лінійна регресія може допомогти нам знайти лінійну залежність між цими параметрами і віком. Вона проста у реалізації та інтерпретації, що дозволяє нам зрозуміти, як саме кожен параметр впливає на вік краба.

Випадковий (Random Forest) ліс є потужним ансамблевим методом машинного навчання, який поєднує декілька дерев рішень для досягнення кращих результатів передбачення. Враховуючи, що фізичні параметри крабів можуть мати складні та неоднорідні взаємозв'язки з їх віком, випадковий ліс може допомогти виявити ці залежності та забезпечити більш точні передбачення. Він також відповідає на нелінійні залежності між параметрами та віком краба.

Метод k-найближчих сусідів (k-Nearest Neighbors Regression) є непараметричним методом машинного навчання, що базується на концепції близькості об'єктів. Він використовує наближення до найближчих сусідів з тренувального набору даних для передбачення значень в тестовому наборі. Застосування методу k-найближчих сусідів до нашої задачі дозволяє знайти схожі краби з відомим віком на основі їх фізичних параметрів, що може допомогти у визначенні віку невідомих крабів.

4.2 Аналіз отриманих результатів для методу Linear Regression

Створення та навчання моделі лінійної регресії:

```
linear_reg = LinearRegression()
linear_reg.fit(x_train, y_train)
```

У цій частині створюється об'єкт linear_reg для моделі лінійної регресії. За допомогою методу fit модель навчається на тренувальних даних x_train та y_train.

Передбачення віку за допомогою навченої моделі:

```
y_pred_linear = linear_reg.predict(x_test)
```

Застосування методу predict до тестових даних x_test дозволяє отримати передбачені значення віку y_pred_linear.

Обчислення метрик помилки та оцінка результатів:

```
mse_lin = mean_squared_error(y_test, y_pred_linear)
r2_lin = r2_score(y_test, y_pred_linear)
print('\nLinear Regression')
print('MSE:', mse_lin)
print('R^2:', r2_lin, '\n')
```

В цій частині обчислюються середньоквадратична помилка (mse_lin) та коефіцієнт детермінації (r2_lin). Вивід результатів відбувається за допомогою функції print.

```
Linear Regression
MSE: 4.64871026652838
R^2: 0.5162360300862966
```

Рисунок 4.1 – Середньоквадратична помилка та коефіцієнт детермінації для лінійної регресії

Візуалізація результатів:

```
plt.scatter(y_test, y_pred_linear)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
plt.xlabel('Фізичні дані')
plt.ylabel('Вік')
plt.title('Лінійна регресія')
plt.show()
```

Цей код відображає діаграму розсіювання, де фактичні значення віку (y_test) відображаються на осі x, а передбачені значення віку (y_pred_linear) - на осі y. Також, за допомогою функції plot, додається лінія, що представляє ідеальну залежність між фактичними та передбаченими значеннями.

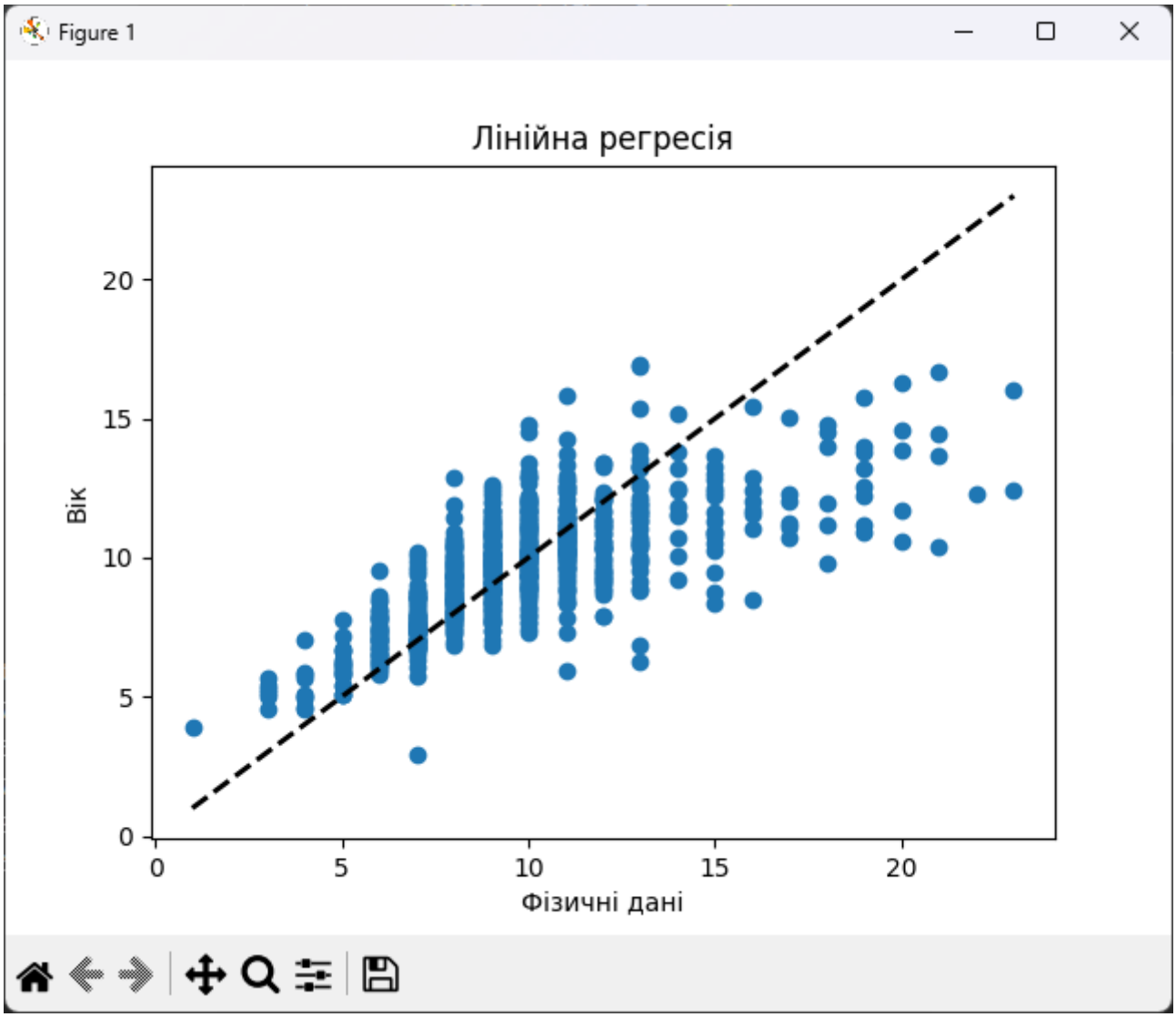

Рисунок 4.2 – Візуалізація лінійної регресії

4.3 Аналіз отриманих результатів для методу Random Forest

Створення та навчання моделі "Випадковий ліс":

```
random_forest_reg = RandomForestRegressor()
random_forest_reg.fit(x_train, y_train)
```

У цій частині створюється об'єкт random_forest_reg для моделі "Випадковий ліс". За допомогою методу fit модель навчається на тренувальних даних x_train та y_train.

Передбачення віку за допомогою навченої моделі:

```
y_pred_rand_forest = random_forest_reg.predict(x_test)
```

Застосування методу predict до тестових даних x_test дозволяє отримати передбачені значення віку y_pred_rand_forest.

Обчислення метрик помилки та оцінка результатів:

```
mse_rand_forest = mean_squared_error(y_test, y_pred_rand_forest)
r2_rand_forest = r2_score(y_test, y_pred_rand_forest)
print('Random Forest')
print('MSE:', mse_rand_forest)
print('R^2:', r2_rand_forest, '\n')
```

В цій частині обчислюються середньоквадратична помилка (mse_rand_forest) та коефіцієнт детермінації (r2_rand_forest). Вивід результатів відбувається за допомогою функції print.

```
Random Forest
MSE: 4.437500256739409
R^2: 0.5382154151120122
```

Рисунок 4.3 – Середньоквадратична помилка та коефіцієнт детермінації для Random Forest

Візуалізація результатів:

```
plt.scatter(y_test, y_pred_rand_forest)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
plt.xlabel('Фізичні дані')
plt.ylabel('Вік')
plt.title('Випадковий ліс')
plt.show()
```

Цей код відображає діаграму розсіювання, де фактичні значення віку (y_test) відображаються на осі x, а передбачені значення віку (y_pred_rand_forest) - на осі y. Також, за допомогою функції plot, додається лінія, що представляє ідеальну залежність між фактичними та передбаченими значеннями

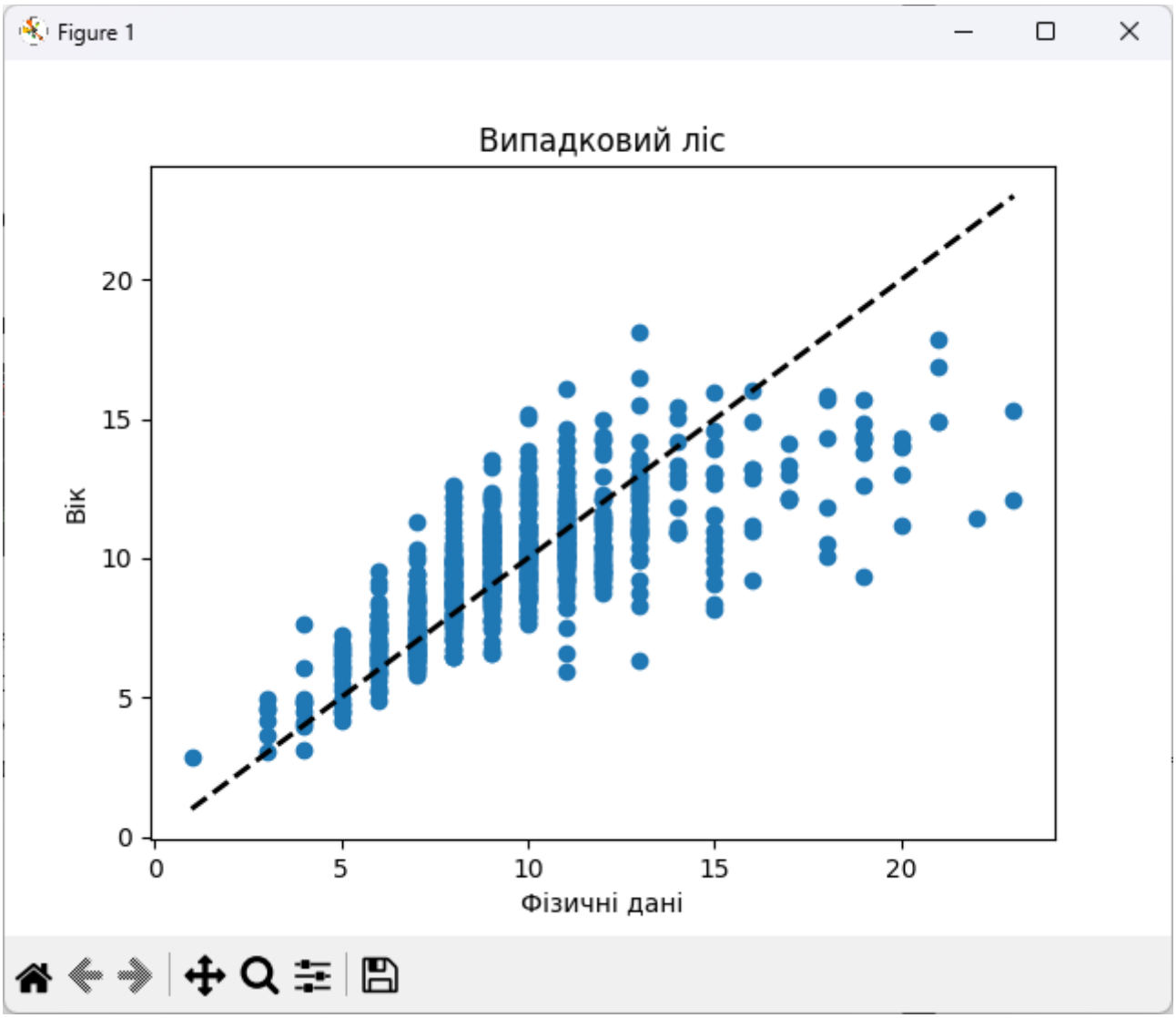

Рисунок 4.4 – Візуалізація Random Forest

4.4 Аналіз отриманих результатів для методу K-Nearest Neighbors Regression

Створення та навчання моделі KNN:

```
random_knn = KNeighborsRegressor()
random_knn.fit(x_train, y_train)
```

У цій частині створюється об'єкт random_knn для моделі KNN. За допомогою методу fit модель навчається на тренувальних даних x_train та y_train.

Передбачення віку за допомогою навченої моделі:

```
y_pred_knn = random_knn.predict(x_test)
```

Застосування методу predict до тестових даних x_test дозволяє отримати передбачені значення віку y_pred_knn.

Обчислення метрик помилки та оцінка результатів:

```
mse_knn = mean_squared_error(y_test, y_pred_knn)
r2_knn = r2_score(y_test, y_pred_knn)
print('KNN')
print('MSE:', mse_knn)
print('R^2:', r2_knn, '\n')
```

В цій частині обчислюються середньоквадратична помилка (mse_knn) та коефіцієнт детермінації (r2_knn). Вивід результатів відбувається за допомогою функції print.

```
KNN
MSE: 4.3452631578947365
R^2: 0.5478139881682212
```

Рисунок 4.5 – Середньоквадратична помилка та коефіцієнт детермінації для K-Nearest Neighbors Regression

Візуалізація результатів:

```
plt.scatter(y_test, y_pred_knn)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
plt.xlabel('Фізичні дані')
plt.ylabel('Вік')
plt.title('KNN')
plt.show()
```

Цей код відображає діаграму розсіювання, де фактичні значення віку (y_test) відображаються на осі x, а передбачені значення віку (y_pred_knn) - на осі y. Також, за допомогою функції plot, додається лінія, що представляє ідеальну залежність між фактичними та передбаченими значеннями.

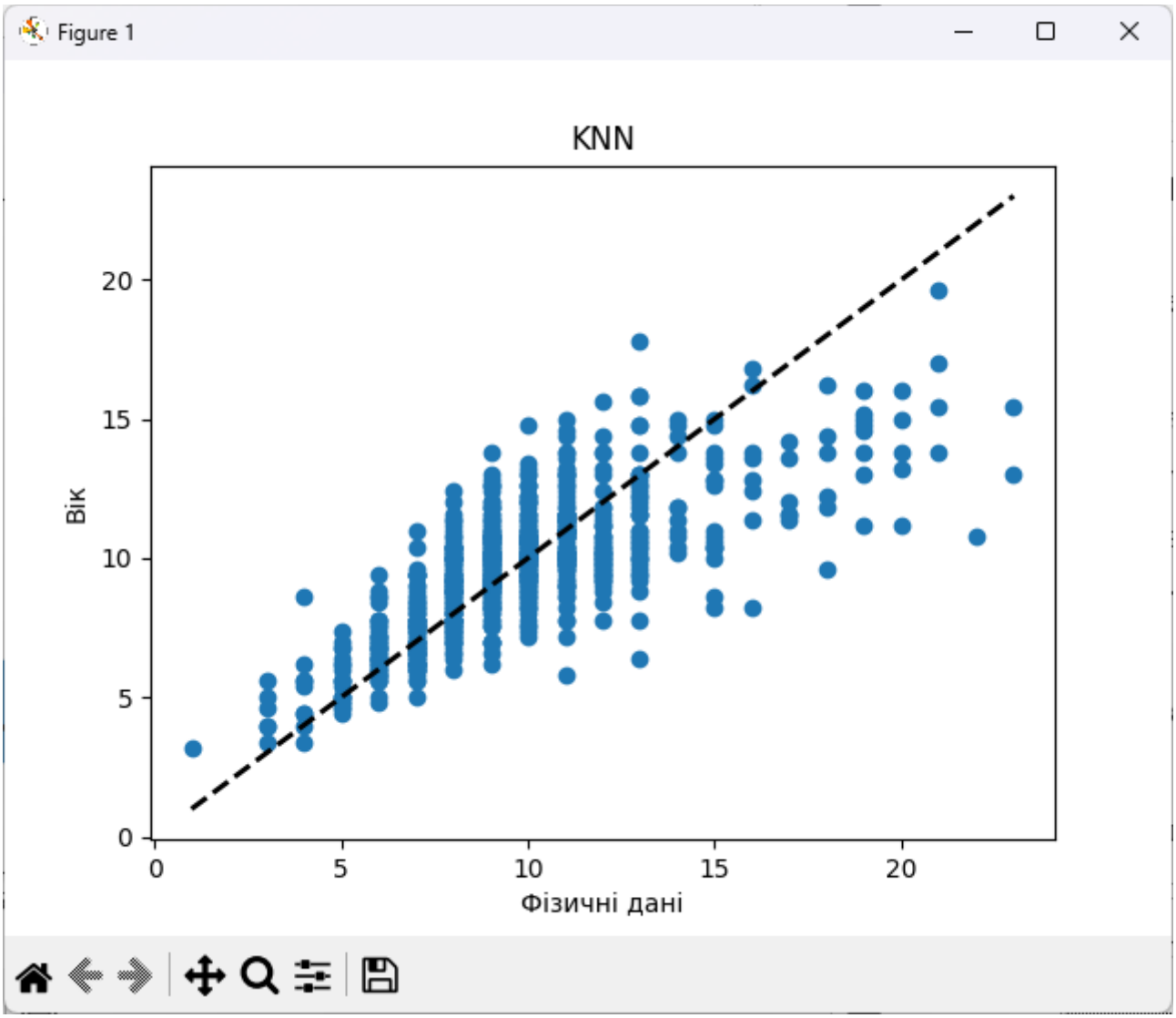

Рисунок 4.6 – Візуалізація K-Nearest Neighbors Regression

4.5 Порівняння отриманих результатів методів

За значеннями MSE (Mean Squared Error) можна зробити висновок, що KNN показує найнижчу помилку (4.345), що означає кращу точність передбачення віку крабів. У порівнянні з KNN, модель Random Forest також має невелику помилку (4.437), а модель Linear Regression має трохи вищу помилку (4.648). Чим нижче значення MSE, тим краще модель передбачає вік крабів.

Щодо R^2 (коефіцієнт детермінації), він вказує на те, наскільки добре модель відповідає даним. Чим ближче значення R^2 до 1, тим краще модель пояснює варіацію віку. В порівнянні з R^2, KNN також показує найкращий результат (0.548), Random Forest має трохи менше (0.538), а Linear Regression має найнижче значення (0.516).

Загалом, за обома метриками (MSE і R^2) KNN та Random Forest показують кращі результати в порівнянні з Linear Regression для передбачення віку крабів на основі їх фізичних характеристик.

ВИСНОВКИ

В результаті проведеного аналізу фізичних характеристик крабів та їх віку була виконана задача передбачення віку на основі цих параметрів. Для досягнення цієї мети були використані три методи інтелектуального аналізу даних: лінійна регресія, випадковий ліс та K ближчих сусідів регресія.

Під час виконання курсової роботи було проведено аналіз предметної області, зібрані дані про фізичні характеристики крабів та їх вік. Для уникнення оверфіту, набір даних був розділений на навчальну та тестову вибірки у співвідношенні 80% до 20% відповідно.

Проведений аналіз результатів показав, що моделі лінійної регресії, випадкового лісу та K ближчих сусідів регресії мають різні показники точності передбачення. Найкращі результати були отримані за допомогою моделі K ближчих сусідів регресії, яка показала найнижчу середньоквадратичну помилку (MSE) та найвищий коефіцієнт детермінації (R^2). Далі слідує модель випадкового лісу, яка також показала низьку помилку та високу пояснювальну здатність. Модель лінійної регресії має трохи менші показники точності у порівнянні з іншими методами.

Отже, на основі результатів аналізу можна зробити висновок, що модель K ближчих сусідів регресії є найбільш ефективною для передбачення віку крабів на основі їх фізичних характеристик. Дана модель може бути використана для подальшого вивчення та дослідження впливу фізичних параметрів на вік крабів та може мати практичне застосування в галузі морського господарства та охорони довкілля.

ПЕРЕЛІК ПОСИЛАНЬ

1. Crab Age Prediction. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/datasets/sidhus/crab-age-prediction.
2. 3.11.4 Documentation. 3.11.4 Documentation. URL: https://docs.python.org/3/.
3. pandas documentation – pandas 2.0.2 documentation. pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/docs/.
4. Matplotlib documentation – Matplotlib 3.7.1 documentation. Matplotlib – Visualization with Python. URL: https://matplotlib.org/stable/.
5. User guide: contents. scikit-learn. URL: https://scikit-learn.org/stable/user_guide.html.

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду передбачення віку краба на основі його фізичних параметрів

(Найменування програми (документа))

SSD

(Вид носія даних)

2 арк, 4 Кб

(Обсяг програми (документа), арк.,

студента групи ІП-13 ІІ курсу

Шевцова А. А.

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

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


def replace_with_unique_numbers(data, columns):
    for column in columns:
        data[column], _ = pd.factorize(data[column])
    return data


if __name__ == "__main__":
    d_path = r'C:\Users\nasty\OneDrive - kpi.ua\KPI\курсова 4
сем\CrabAgePrediction.csv'
    data_frame = pd.read_csv(d_path, sep=',', decimal='.')

    # аналіз початкового датасету
    print('\nІнформація про датасет:')
    data_frame.info()
    print('\nПерші 5 рядків:')
    print(data_frame.head())
    print('\nКолонки з пропущеними значеннями:')
    print(data_frame.isna().any())
    print('\nОпис датасету:')
    print(data_frame.describe())

    # заміняємо колони зі строковими значеннями на числові
    columns_to_replace = ['Sex']
    data_frame = replace_with_unique_numbers(data_frame, columns_to_replace)
    print('\nПерші 5 рядків:')
    print(data_frame.head())

    # розділення на навчальну і тестову вибірки
    x = data_frame.drop(columns='Age')
    y = data_frame['Age']
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42)

    # побудова моделей

    # лінійна регресія
    linear_reg = LinearRegression()
    linear_reg.fit(x_train, y_train)
    y_pred_linear = linear_reg.predict(x_test)

    mse_lin = mean_squared_error(y_test, y_pred_linear)
    r2_lin = r2_score(y_test, y_pred_linear)
    print('\nLinear Regression')
    print('MSE:', mse_lin)
    print('R^2:', r2_lin, '\n')

    plt.scatter(y_test, y_pred_linear)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
    plt.xlabel('Фізичні дані')
    plt.ylabel('Вік')
    plt.title('Лінійна регресія')
    plt.show()
```

```
    # випадковий ліс
    random_forest_reg = RandomForestRegressor()
    random_forest_reg.fit(x_train, y_train)
    y_pred_rand_forest = random_forest_reg.predict(x_test)

    mse_rand_forest = mean_squared_error(y_test, y_pred_rand_forest)
    r2_rand_forest = r2_score(y_test, y_pred_rand_forest)
    print('Random Forest')
    print('MSE:', mse_rand_forest)
    print('R^2:', r2_rand_forest, '\n')

    plt.scatter(y_test, y_pred_rand_forest)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
    plt.xlabel('Фізичні дані')
    plt.ylabel('Вік')
    plt.title('Випадковий ліс')
    plt.show()

    # KNeighborsRegressor
    random_knn = KNeighborsRegressor()
    random_knn.fit(x_train, y_train)
    y_pred_knn = random_knn.predict(x_test)

    mse_knn = mean_squared_error(y_test, y_pred_knn)
    r2_knn = r2_score(y_test, y_pred_knn)
    print('KNN')
    print('MSE:', mse_knn)
    print('R^2:', r2_knn, '\n')

    plt.scatter(y_test, y_pred_knn)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
    plt.xlabel('Фізичні дані')
    plt.ylabel('Вік')
    plt.title('KNN')
    plt.show()
```