

Лабораторна робота № 6. Дослідження методів класифікації: K-NN, Naive Bayes, SVM, Decision Trees, Logistic Regression

Мета: сформулювати практичні навички у застосуванні різних алгоритмів машинного навчання для вирішення задач класифікації. Здобувачі навчатимуться підбирати найбільш ефективний алгоритм для конкретного набору даних, а також здійснювати попередню обробку даних, налаштовувати гіперпараметри моделей та аналізувати результати класифікації.

Теоретичні відомості

Класифікація є однією з основних задач машинного навчання, яка полягає у визначенні категорії або класу для даного об'єкта з використанням набору ознак.

K-NN (K-Nearest Neighbors) – метод, який класифікує об'єкти на основі голосування найближчих сусідів. Він не має експліцитної тренувальної фази або вона дуже проста, що робить його простим у використанні, однак метод вимагає значного обчислювального ресурсу при класифікації нових об'єктів.

Naive Bayes – простий ймовірнісний класифікатор, який застосовує теорему Баєса з «наївним» припущенням про незалежність між ознаками. Попри свою простоту, Naive Bayes може бути дуже ефективним у деяких задачах, особливо при великій кількості ознак.

SVM (Support Vector Machine) – метод, який знаходить гіперплощину в багатовимірному просторі, що найкраще розділяє об'єкти різних класів. SVM ефективний у випадках, коли кількість ознак перевищує кількість спостережень.

Decision Trees – модель, яка використовує деревоподібну структуру для прийняття рішень. Кожен вузол дерева представляє ознаку, кожен лист – клас. Дереву рішень легко інтерпретувати, але схильні до перенавчання.

Logistic Regression – це статистичний метод для аналізу набору даних, в якому одна або кілька незалежних змінних визначають результат. Використовується для задач бінарної класифікації.

Приклад виконання роботи

Завдання

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Проаналізуйте датасет та визначте ознаки, які будуть використовуватися для класифікації. Як цільову змінну класифікації використовуйте клас заданий у таблиці.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується.
4. Виконайте тренування моделей: 1) K-NN; 2) Naïve Bayes; 3) SVM; 4) Decision Trees; 5) Logistic Regression. Виконайте оцінку та візуалізуйте результати за наступними метриками: Accuracy, Confusion matrix для кожної моделі.
5. Виконайте класифікацію за допомогою різних моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проєкспериментуйте з гіперпараметрами для моделей задля покращення результатів.

Контрольне запитання №1



Вкажіть результуюче значення Ассигасу для K-NN.

Відповідь: _____

Контрольне запитання №2



Вкажіть результуюче значення Ассигасу для SVM.

Відповідь: _____

Контрольне запитання №3

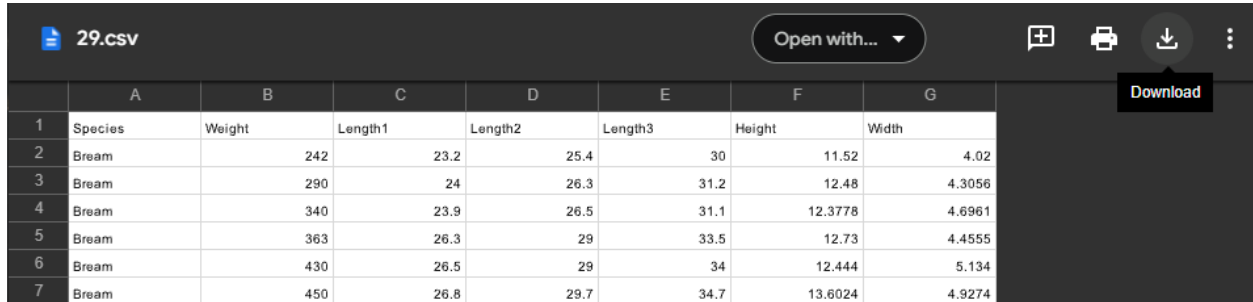


Вкажіть результуюче значення Accuracy для Logistic Regression.

Відповідь: _____

Хід роботи

1. Збережемо та імпортуємо набір даних за варіантом.



	A	B	C	D	E	F	G
1	Species	Weight	Length1	Length2	Length3	Height	Width
2	Bream	242	23.2	25.4	30	11.52	4.02
3	Bream	290	24	26.3	31.2	12.48	4.3056
4	Bream	340	23.9	26.5	31.1	12.3778	4.6961
5	Bream	363	26.3	29	33.5	12.73	4.4555
6	Bream	430	26.5	29	34	12.444	5.134
7	Bream	450	26.8	29.7	34.7	13.6024	4.9274

Рисунок 5.1. Датасет на Google Drive

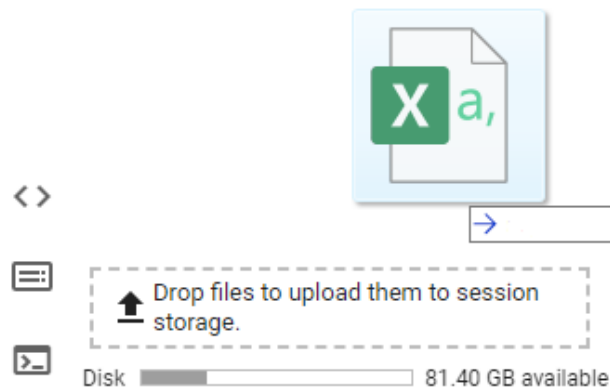


Рисунок 5.2. Завантаження датасету до Google Colab

2. Встановлення необхідних бібліотек та модулів

```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV,
cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, roc_curve,
auc, precision_recall_curve
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.utils import resample
from imblearn.over_sampling import SMOTE
```

```

from imblearn.under_sampling import RandomUnderSampler
from sklearn.datasets import load_iris
import ipywidgets as widgets
from IPython.display import display, clear_output

```

3. Ініціалізація набору даних в Google Colab.

```

# Завдання №1. # Завантаження датасету
data = pd.read_csv("/content/29.csv")
data.head()

```

	Species	Weight	Length1	Length2	Length3	Height	Width
0	Bream	242.0	23.2	25.4	30.0	11.5200	4.0200
1	Bream	290.0	24.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	23.9	26.5	31.1	12.3778	4.6961
3	Bream	363.0	26.3	29.0	33.5	12.7300	4.4555
4	Bream	430.0	26.5	29.0	34.0	12.4440	5.1340

Рисунок 5.3. Перші 5 рядків набору даних

4. Визначимо цільову змінну, як Species, ознаками будуть: Weight, Length1, Length2, Length3, Height, Width.

5. Виконаємо очищення даних

```

# Перевірка на відсутні значення
print(data.isnull().sum())

```

```

Species      0
Weight       0
Length1      0
Length2      0
Length3      0
Height       0
Width        0
dtype: int64

```

Рисунок 5.4. Пошук нульових значень у датасеті

Порожні значення відсутні, тож видалення або заповнення виконувати не треба.

```

# Видалення або заповнення відсутніх значень
# data.dropna(inplace=True) # Видалення
# data.fillna(data.mean(), inplace=True) # Заповнення середніми значеннями

```

Виконаємо візуалізацію однієї з ознак набору даних, а саме «Length2»

```
# Візуалізація викидів
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot before removing outliers')
plt.show()
```

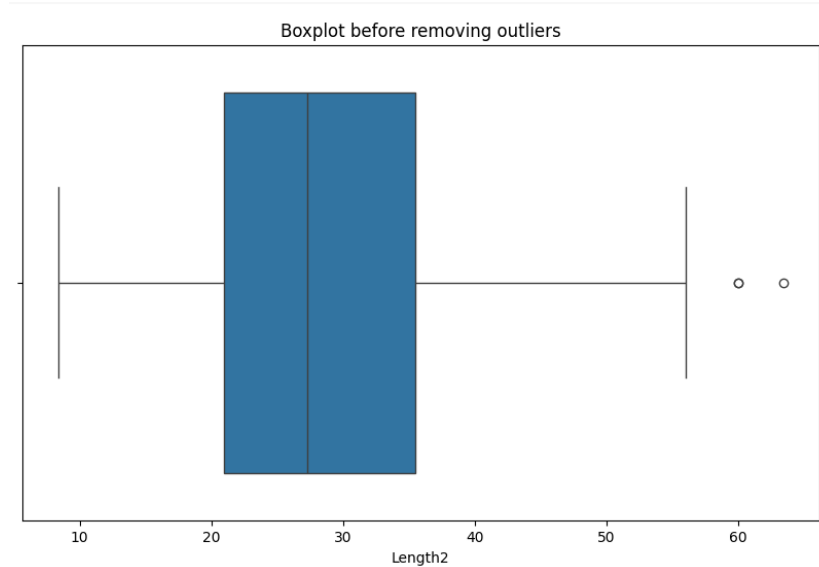


Рисунок 5.5. Діаграма викидів для Length2 перед очищенням

Виконаємо очищення викидів для «Length2».

```
# Видалення викидів
feature_processing = 'Length2'
q_low = data[feature_processing].quantile(0.01)
q_hi = data[feature_processing].quantile(0.99)
data = data[(data[feature_processing] < q_hi) & (data[feature_processing]
> q_low)]
```

```
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot after removing outliers')
plt.show()
```

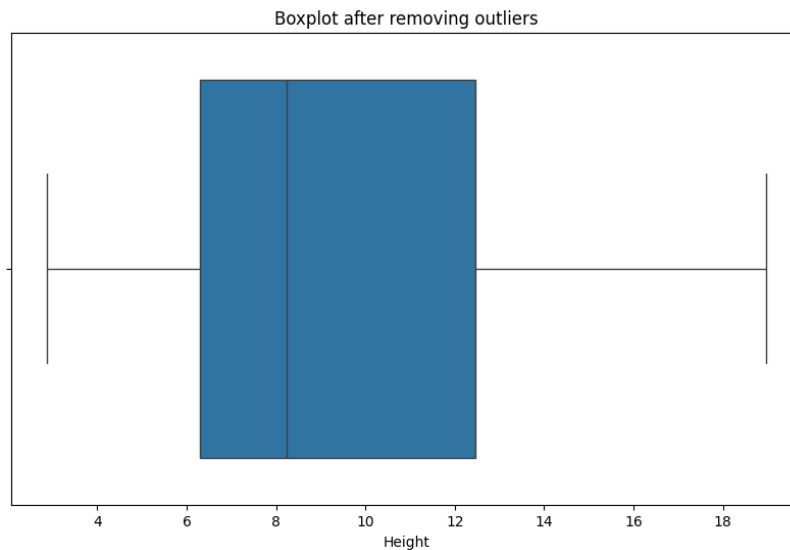


Рисунок 5.6. Діаграма викидів для Length2 після очищення

Подібні операції виконаємо для інших ознак.

6. Розподілимо дані на тренувальні та тестові набори з урахуванням ознак та цільової змінної. В тестовому наборі даних видалимо стовпець «Species», що класифікується, та виконаємо розподіл тестових та тренувальних наборів даних.

```
X = data.drop('Species', axis=1)
y = data['Species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

Виконаємо нормалізацію даних.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

7. Виконаємо тренування моделей K-NN, Naïve Bayes, SVM, Decision Trees, Logistic Regression та їх оцінку за показниками Accuracy та Confusion Matrix.

```
model_widget = widgets.Dropdown(
    options=['K-NN', 'Naive Bayes', 'SVM', 'Decision Trees', 'Logistic
Regression'],
    description='Model:',
)

hyperparameter_widget = widgets.IntSlider(
    value=3,
    min=1,
```

```

    max=10,
    step=1,
    description='Hyperparameter:',
)

@widgets.interact(model_choice=model_widget,
hyperparameter=hyperparameter_widget)
def train_model(model_choice, hyperparameter):
    print(f"Починаємо обробку з моделлю {model_choice} та гіперпараметром {hyperparameter}...")
    if model_choice == 'K-NN':
        model = KNeighborsClassifier(n_neighbors=hyperparameter)
    elif model_choice == 'SVM':
        model = SVC(kernel='linear', C=hyperparameter)
    elif model_choice == 'Decision Trees':
        model = DecisionTreeClassifier(max_depth=hyperparameter)
    elif model_choice == 'Logistic Regression':
        model = LogisticRegression(C=hyperparameter)
    else:
        model = GaussianNB()

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy:.4f}")

    cm = confusion_matrix(y_test, y_pred)
    sns.heatmap(cm, annot=True, fmt="d")
    plt.show()

```

8. Виконаємо оцінювання моделей за показниками Accuracy та Confusion matrix.

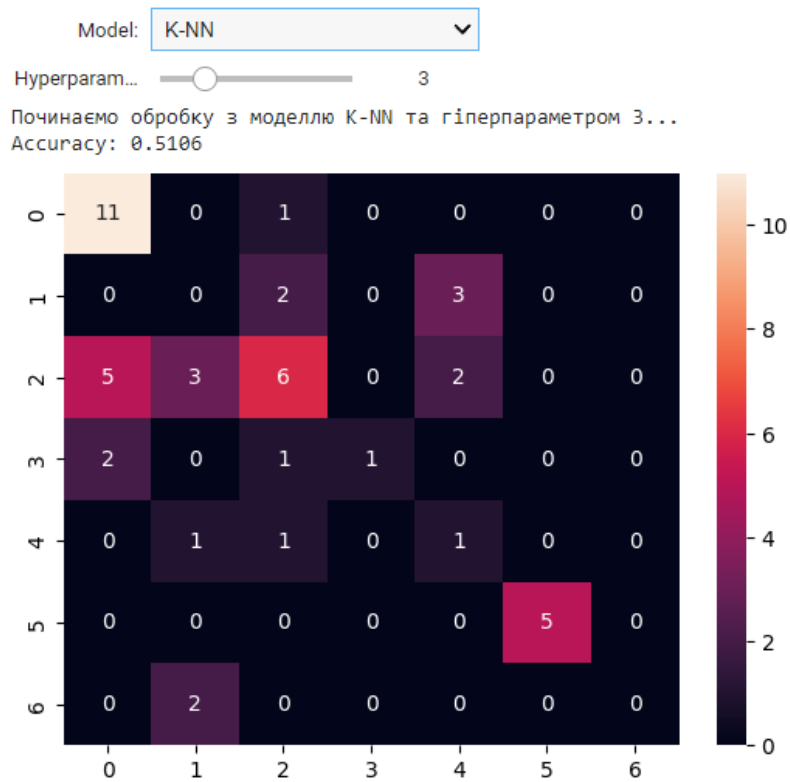


Рисунок 5.7. Результати моделі K-NN

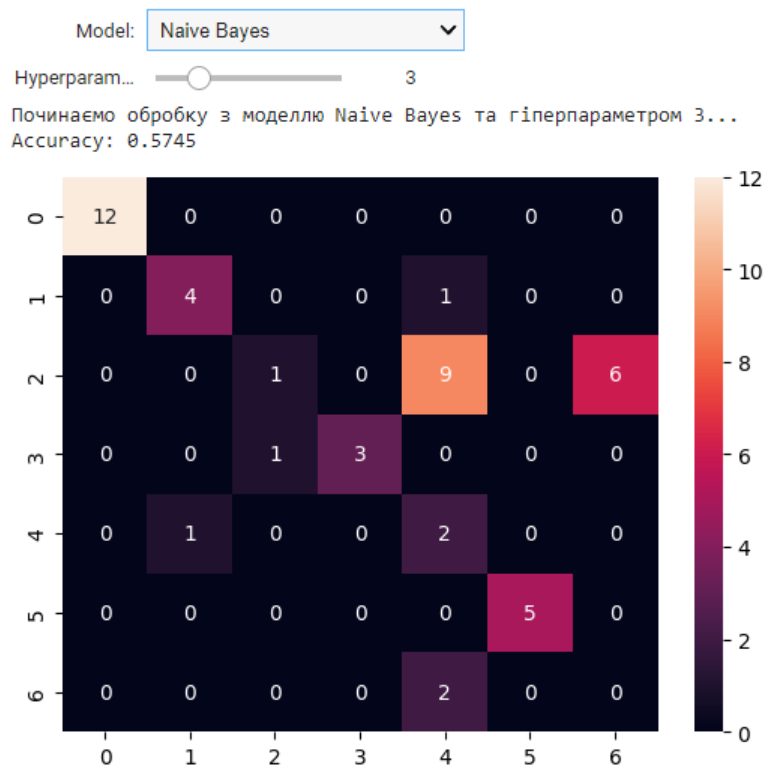


Рисунок 5.8. Результати моделі Naive Bayes

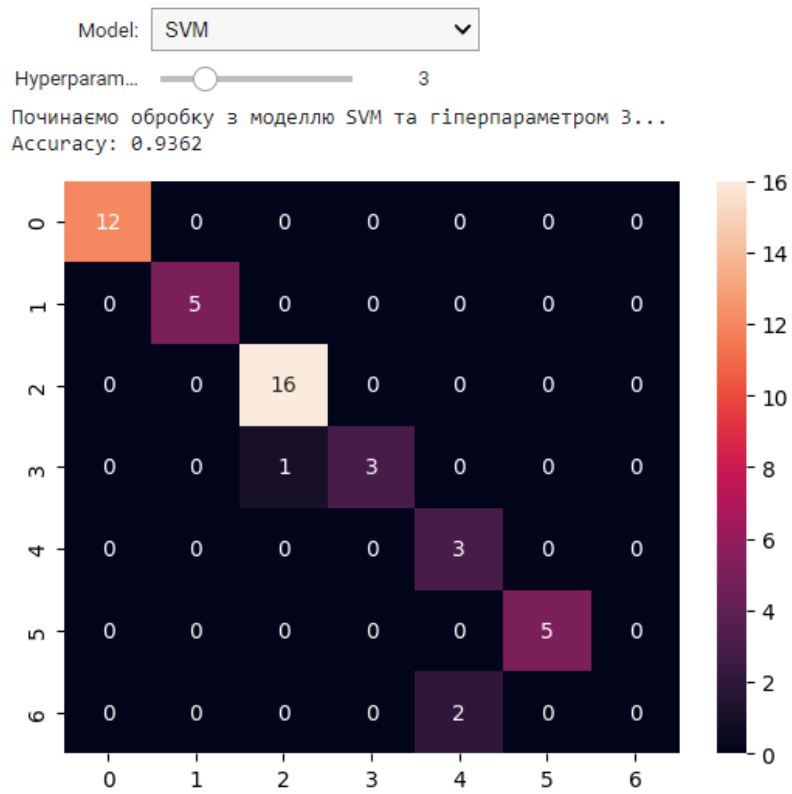


Рисунок 5.9. Результати моделі SVM

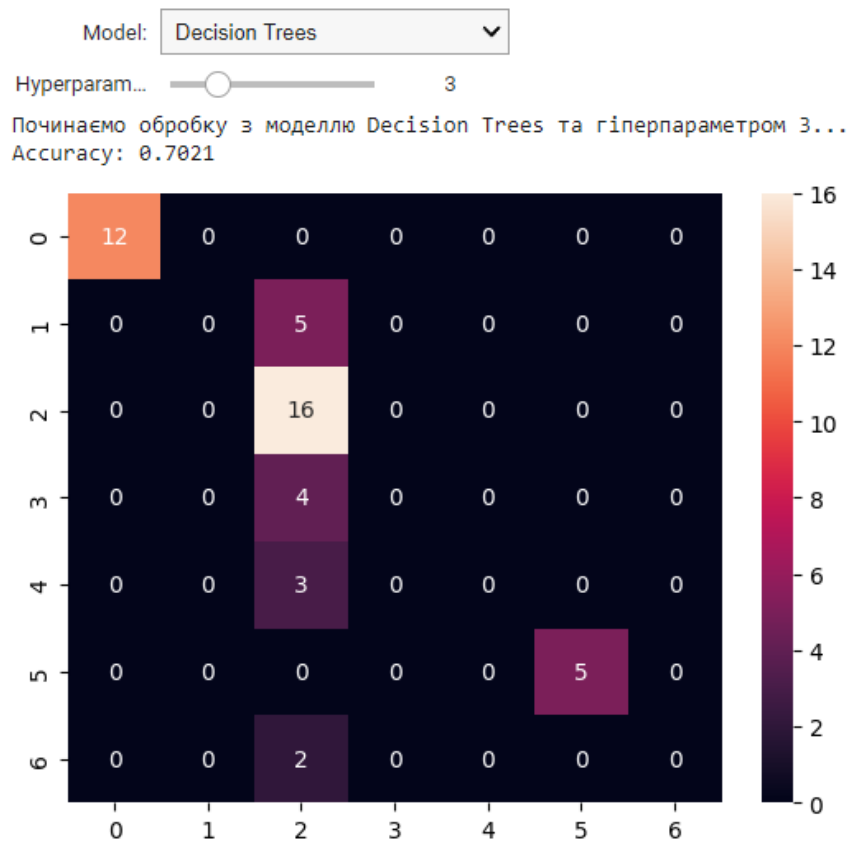


Рисунок 5.10. Результати моделі Decision Trees

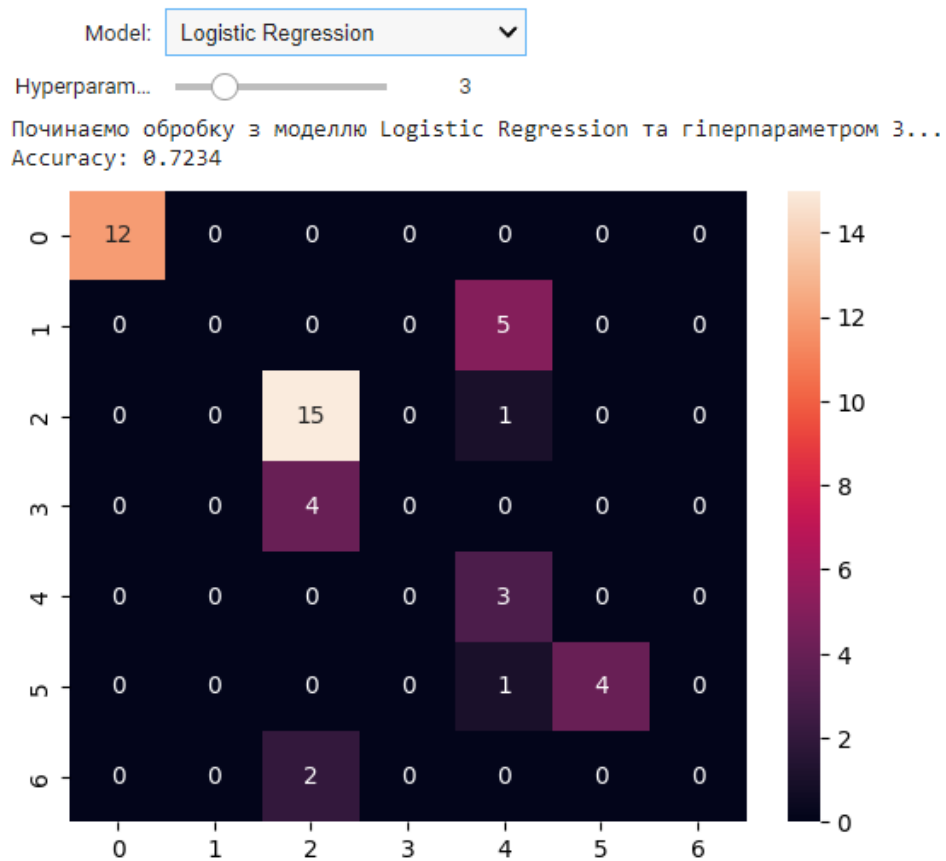


Рисунок 5.11. Результати моделі Logistic Regression

9. Виконаємо прогнозування на довільних даних

```
input_widgets = [widgets.FloatText(description=f'Feature {i+1}:') for i in
range(X_train.shape[1])]

classify_button = widgets.Button(description="Classify")

output = widgets.Output()

def classify(b):
    with output:
        output.clear_output()
        data_to_classify = [w.value for w in input_widgets]

        model_choice = model_widget.value
        hyperparameter = hyperparameter_widget.value
        model = prepare_model(model_choice, hyperparameter)

        prediction = model.predict([data_to_classify])
        print(f"Predicted class: {prediction[0]}")

classify_button.on_click(classify)
```

```
def prepare_model(model_choice, hyperparameter):
    if model_choice == 'K-NN':
        model = KNeighborsClassifier(n_neighbors=hyperparameter)
    elif model_choice == 'SVM':
        model = SVC(kernel='linear', C=hyperparameter)
    elif model_choice == 'Decision Trees':
        model = DecisionTreeClassifier(max_depth=hyperparameter)
    elif model_choice == 'Logistic Regression':
        model = LogisticRegression(C=hyperparameter)
    else:
        model = GaussianNB()
    model.fit(X_train, y_train)
    return model

display(*input_widgets, model_widget, hyperparameter_widget,
        classify_button, output)
```

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	Logistic Regression ▼
Hyperparam...	<input type="range"/> 3
<button>Classify</button>	
Predicted class: Bream	

Рисунок 5.12. Результати прогнозування для моделі Logistic Regression

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	SVM ▼
Hyperparam...	<input type="range"/> 3
<button>Classify</button>	
Predicted class: Bream	

Рисунок 5.13. Результати прогнозування для моделі SVM

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	K-NN
Hyperparam...	<input type="range"/> 3
Classify	
Predicted class: Perch	

Рисунок 5.14. Результати прогнозування для моделі K-NN

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	Naïve Bayes
Hyperparam...	<input type="range"/> 3
Classify	
Predicted class: Perch	

Рисунок 5.15. Результати прогнозування для моделі Naïve Bayes

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	Decision Trees
Hyperparam...	<input type="range"/> 3
Classify	
Predicted class: Perch	

Рисунок 5.16. Результати прогнозування для моделі Decision Trees

10. Виконаємо тестування гіперпараметрів

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	K-NN
Hyperparam...	<input type="range"/> 1
<button>Classify</button>	
Predicted class: Bream	

Рисунок 5.17. Результати прогнозування для моделі K-NN зі значенням гіперпараметрів «1»

Feature 1:	242
Feature 2:	23.2
Feature 3:	25.4
Feature 4:	30
Feature 5:	11.52
Feature 6:	4.02
Model:	Decision Trees
Hyperparam...	<input type="range"/> 6
<button>Classify</button>	
Predicted class: Bream	
Accuracy: 0.8085	

Рисунок 5.18. Результати прогнозування для моделі Decision Trees зі значенням гіперпараметрів «6»

Таким чином, аналізуючи отримані дані, можна зазначити, що використання моделі SVM продемонструвало найкращі результати за показником Accuracy (0.9362). Моделі Decision Trees (0.7021) та Logistic Regression (0.7234) надали близькі один до одного результати за показником Accuracy, що є достатнім та дозволяє виконувати задачу класифікації для даного набору даних. Проте зі стандартним налаштуванням гіперпараметрів модель Decision Trees надала некоректний результат класифікації на довільному прикладі. При збільшенні значення гіперпараметрів, модель почала надавати більшу точність (0.8085) та правильно класифікувала приклад. Що стосується моделей K-

NN (0.5106) та Naïve Bayes (0.5745), то їх показники точності є незадовільними та не дозволяють якісно виконувати класифікацію на заданому наборі даних, потребують доопрацювання.



Контрольне запитання №1

Вкажіть результуюче значення Accurasy для K-NN.

Відповідь: **0.5106**



Контрольне запитання №2

Вкажіть результуюче значення Accurasy для SVM.

Відповідь: **0.9362**



Контрольне запитання №3

Вкажіть результуюче значення Accurasy для Logistic Regression.

Відповідь: **0.7234**

Висновки: ...

Завдання для виконання лабораторної роботи

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Проаналізуйте датасет та визначте ознаки, які будуть використовуватися для класифікації. Як цільову змінну класифікації використовуйте клас заданий у таблиці.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується.
4. Виконайте тренування моделей: 1) K-NN; 2) Naïve Bayes; 3) SVM; 4) Decision Trees; 5) Logistic Regression. Виконайте оцінку та візуалізуйте результати за наступними метриками: Accuracy, Confusion matrix для кожної моделі.
5. Виконайте класифікацію за допомогою різних моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проєкспериментуйте з гіперпараметрами для моделей задля покращення результатів.

	Посилання на датасети				
№	1	5	9	13	17
Classification	Brand	label	Microcalcification	category	wifi
№	2	6	10	14	18
Classification	class	Type	diabetes	R	is_safe
№	3	7	11	15	19
Classification	POP	Species	Car	class	quality
№	4	8	12	16	20
Classification	ocean_proximity	CarName	poutcome	custcat	Species

Контрольне запитання №1



Вкажіть результуюче значення Ассигасу для K-NN.

Відповідь: _____

Контрольне запитання №2



Вкажіть результуюче значення Ассигасу для SVM.

Відповідь: _____



Контрольне запитання №3

Вкажіть результуюче значення Accuracy для Logistic Regression.

Відповідь: _____



Welcome to Google Colab

<https://colab.research.google.com/drive/1eAe0Z89Tb3HNyNbenUcn7RSUIqr00i5I?usp=sharing>