UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

(FSKTM)

SEMESTER II 2024/2025

DATA MINING

BIT 33603

SECTION 03

LAB ASSIGNMENT 09+10

TITLE

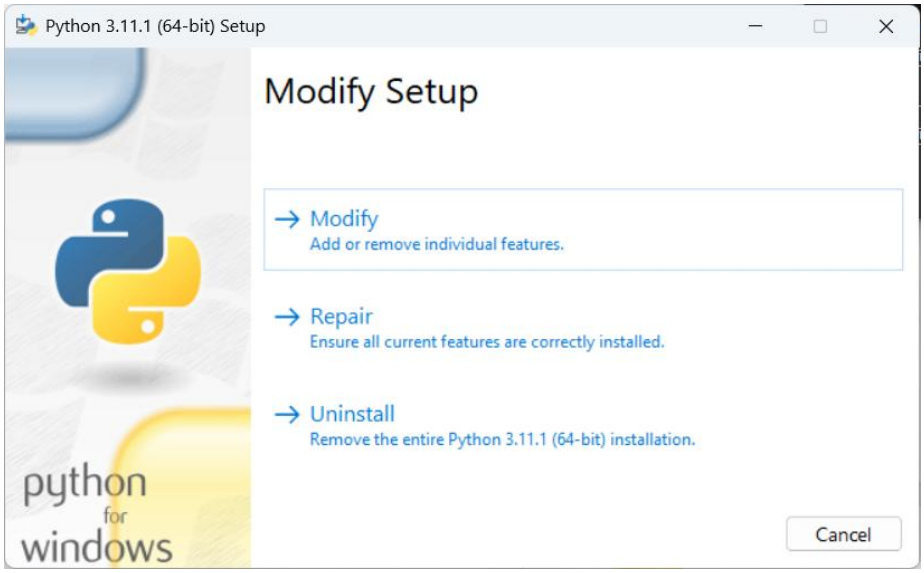
DEEP LEARNING CNN

LECTURER'S NAME

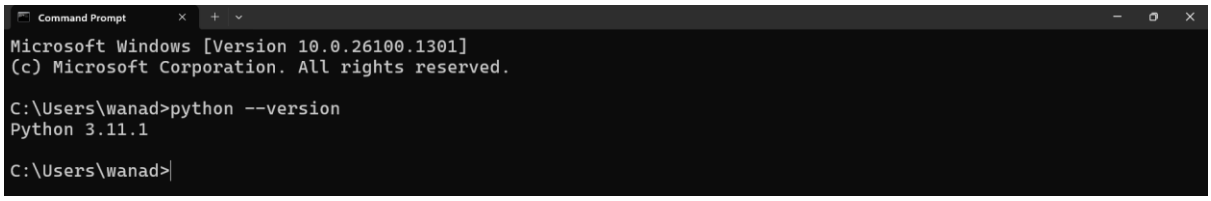
DR. ROZITA BINTI ABDUL JALIL

NAME	TUAN KHALIDAH SYAZWANA BINTI TUAN MOHD KASMAWI
MATRIC NUMBER	AI220118
DATE SUBMISSION	May 28, 2025

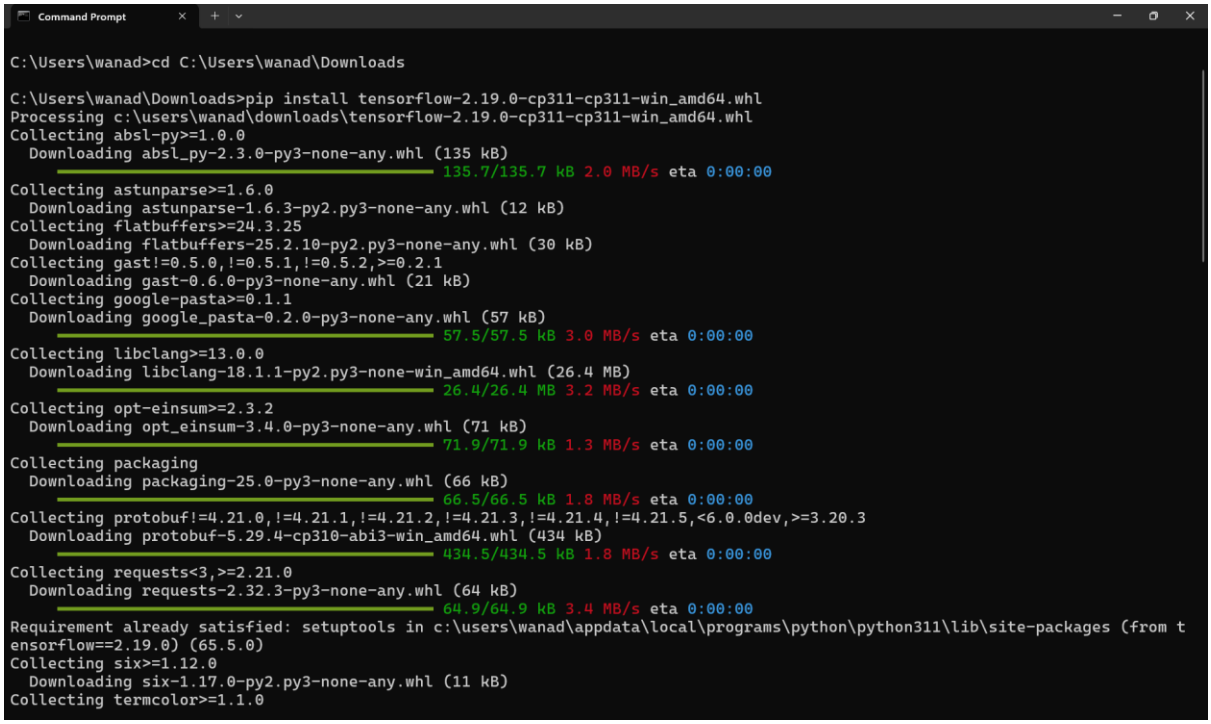
Python Installation



Installation Test

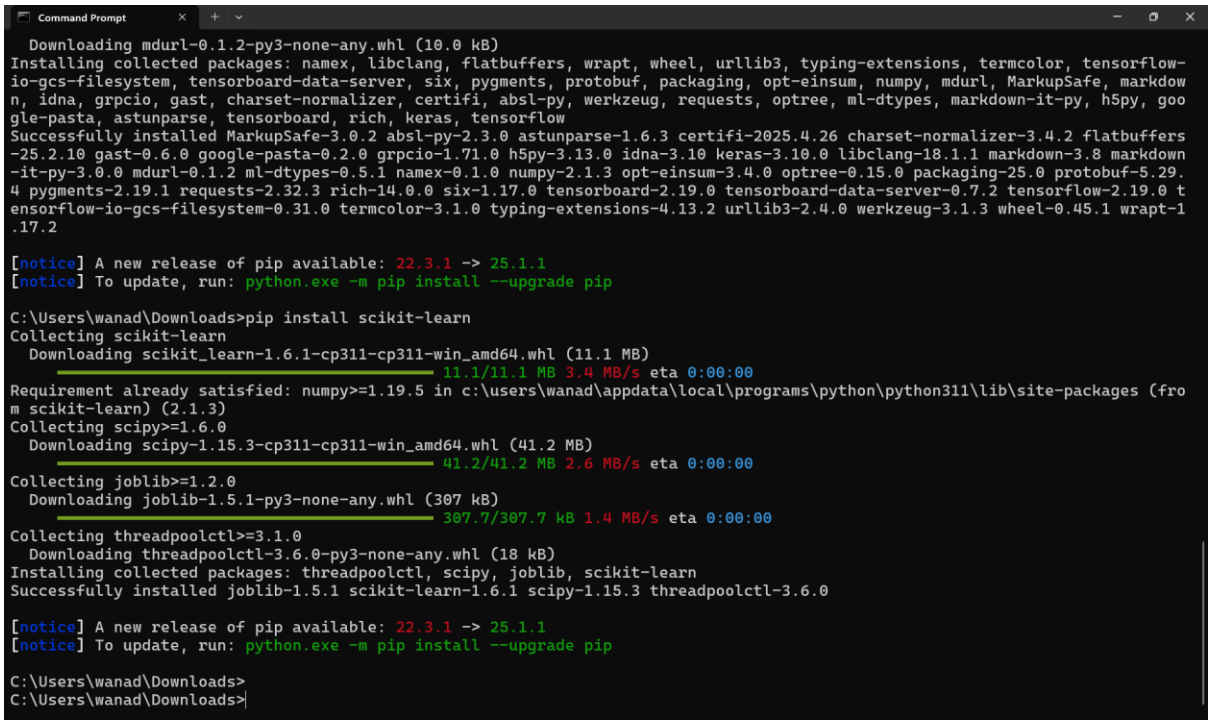


TensorFlow Installation

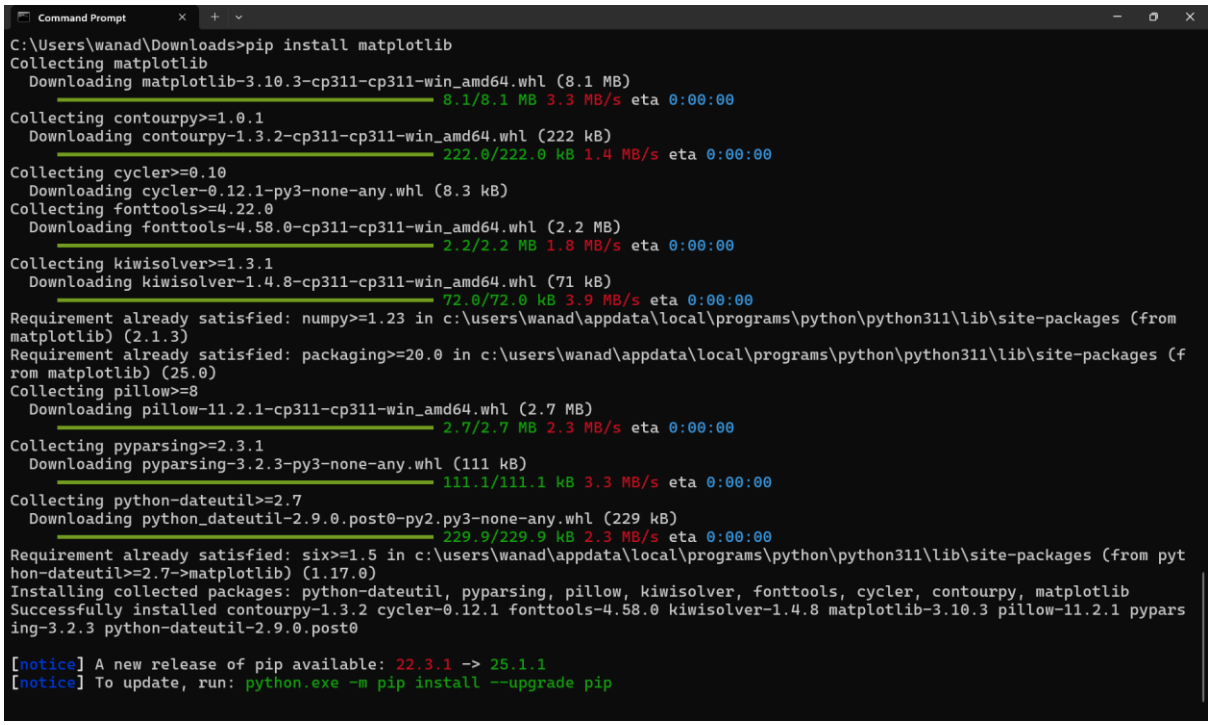


Related Packages Installation

1. Install this package for performance metrics: pip install scikit-learn



2. Install this package for plot auc :pip install matplotlib



Source Code

Lab9-10.py - C:/Users/wanad/AppData/Local/Programs/Python/Python311/Lab9-10.py (3.11.1)

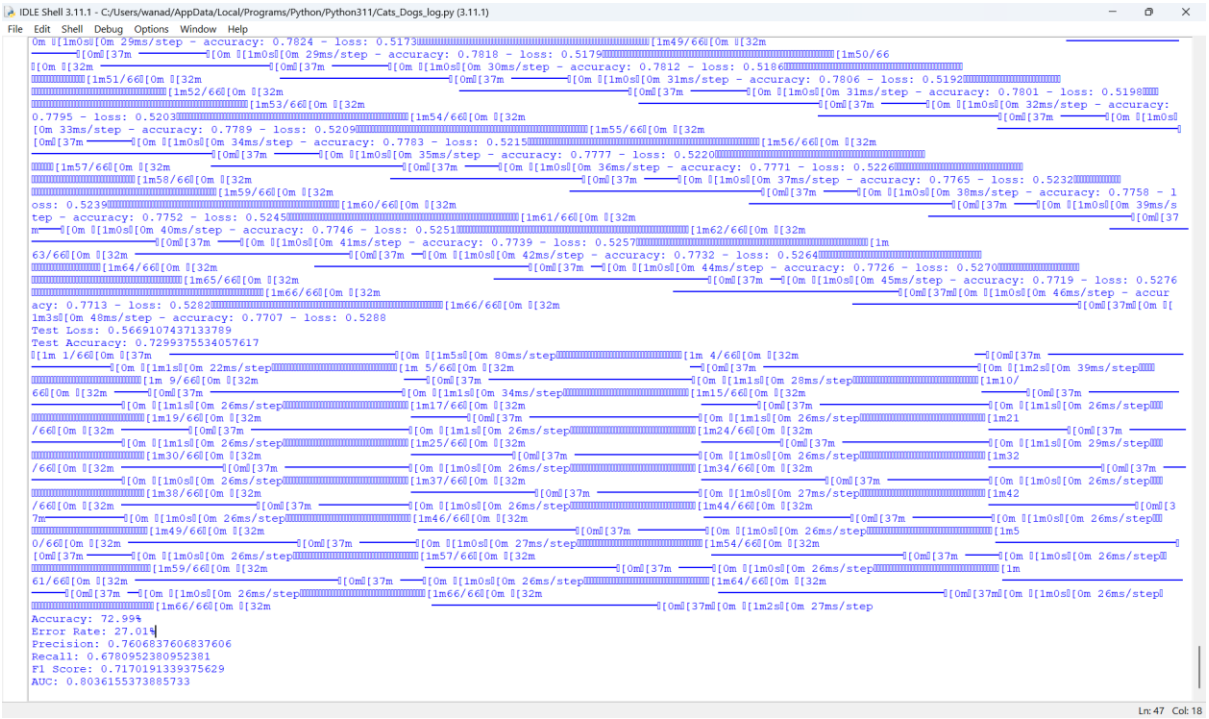
File Edit Format Run Options Window Help

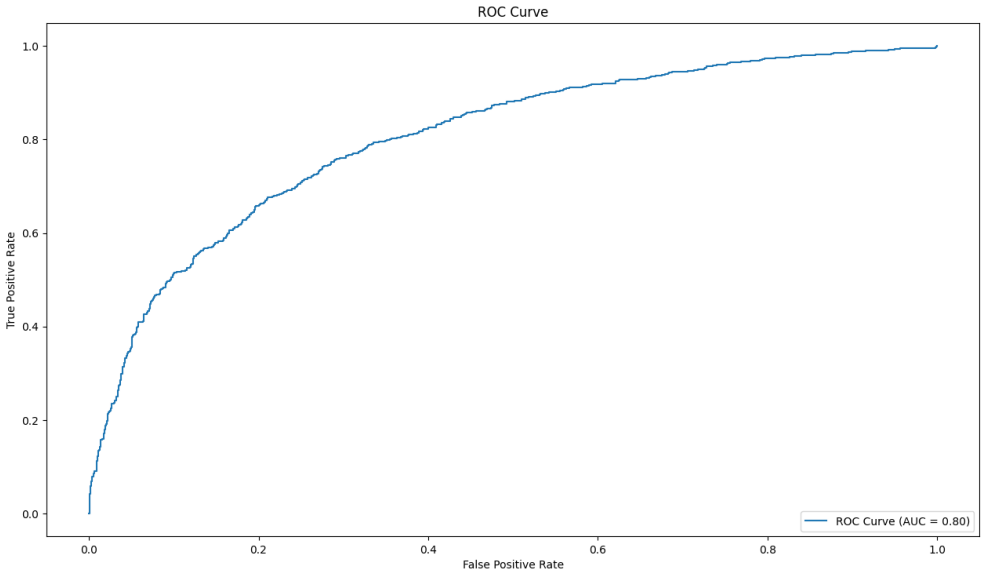
```python
#Python code
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

# Set the dataset path
path_to_data = 'C:\\Users\\wanad\\AppData\\Local\\Programs\\Python\\Python311\\Cats_Dogs'

# Splitting the dataset into training and testing
# Preparing the training data
train_data_generator = ImageDataGenerator(rescale=1.0 / 255)

train_data = train_data_generator.flow_from_directory(
    directory=os.path.join(path_to_data, "training_set"),
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary'
)

# Preparing the testing data
test_data_generator = ImageDataGenerator(rescale=1.0 / 255)

test_data = test_data_generator.flow_from_directory(
    directory=os.path.join(path_to_data, "test_set"),
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary',
    shuffle=False
)

# Building the CNN model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(
    optimizer=Adam(),
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Train the model
history = model.fit(
    train_data,
    steps_per_epoch=100,  # Adjust based on your dataset
    epochs=10,
    validation_data=test_data,
    validation_steps=50  # Adjust based on your validation dataset
)

# Evaluate the model on test data
model_evaluation = model.evaluate(
    test_data,
    steps=len(test_data)
)

test_loss, test_accuracy = model_evaluation
print(f"Test Loss: {test_loss}")
print(f"Test Accuracy: {test_accuracy}")

# Predictions on test data for additional metrics
predictions = model.predict(
    test_data,
    steps=len(test_data),
    verbose=1
)

# Process predictions and true labels
predictions = predictions.flatten()
predicted_classes = (predictions > 0.5).astype(int)
true_classes = test_data.classes

# Calculate accuracy and error rate
accuracy = np.mean(predicted_classes == true_classes)
error_rate = 1 - accuracy

# Calculate precision, recall, and F1 score
precision_value = precision_score(true_classes, predicted_classes)
recall_value = recall_score(true_classes, predicted_classes)
f1_score_value = f1_score(true_classes, predicted_classes)

# ROC and AUC
auc_value = roc_auc_score(true_classes, predictions)
fpr, tpr, _ = roc_curve(true_classes, predictions)

# Print additional metrics
print(f"Accuracy: {accuracy * 100:.2f}%")
print(f"Error Rate: {error_rate * 100:.2f}%")
print(f"Precision: {precision_value}")
print(f"Recall: {recall_value}")
print(f"F1 Score: {f1_score_value}")
print(f"AUC: {auc_value}")

# Plot ROC curve
plt.figure()
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {auc_value:.2f})")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc="lower right")
plt.show()
```

Ln: 4 Col: 30

Output


IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/wanad/AppData/Local/Programs/Python/Python311/Lab9-10.py ==
Found 8045 images belonging to 2 classes.
Found 2081 images belonging to 2 classes.
Warning (from warnings module):
File "C:\Users\wanad\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\layers\convolutional\base_conv.py", line 113
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in th
e model instead.
Warning (from warnings module):
File "C:\Users\wanad\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py", line 121
self._warn_if_super_not_called()
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_si
ze`. Do not pass these arguments to `fit()`, as they will be ignored.
Epoch 1/10
Ln: 13 Col: 0


IDLE Shell 3.11.1 - C:/Users/wanad/AppData/Local/Programs/Python/Python311/Cats_Dogs_log.py (3.11.1)
File Edit Shell Debug Options Window Help
Accuracy: 72.99%
Error Rate: 27.01%
Precision: 0.7606837606837606
Recall: 0.6780952380952381
F1 Score: 0.7170191339375629
AUC: 0.8036155373885733
Ln: 47 Col: 18


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC Curve (AUC = 0.80)