Deakin University

Machine Learning

OnTrack Submission

Task 2.1P

Submitted By:
Xueying Feng
s224270349
2025/07/21 18:23

Tutor:
Shashank Gupta

July 21, 2025

Cell 01

```
import pandas as pd

# Read the CSV file
df = pd.read_csv("microclimate-sensors-data.csv")

# Count the number of missing values in each column
missing_counts = df.isnull().sum()
print("Missing values per feature:")
print(missing_counts)
```

```
Missing values per feature:
Device_id                   0
Time                        0
SensorLocation           6143
LatLong                 11483
MinimumWindDirection    40395
AverageWindDirection      507
MaximumWindDirection    40553
MinimumWindSpeed        40553
AverageWindSpeed          507
GustWindSpeed           40553
AirTemperature            507
RelativeHumidity          507
AtmosphericPressure       507
PM25                    19130
PM10                    19130
Noise                   19130
dtype: int64
```

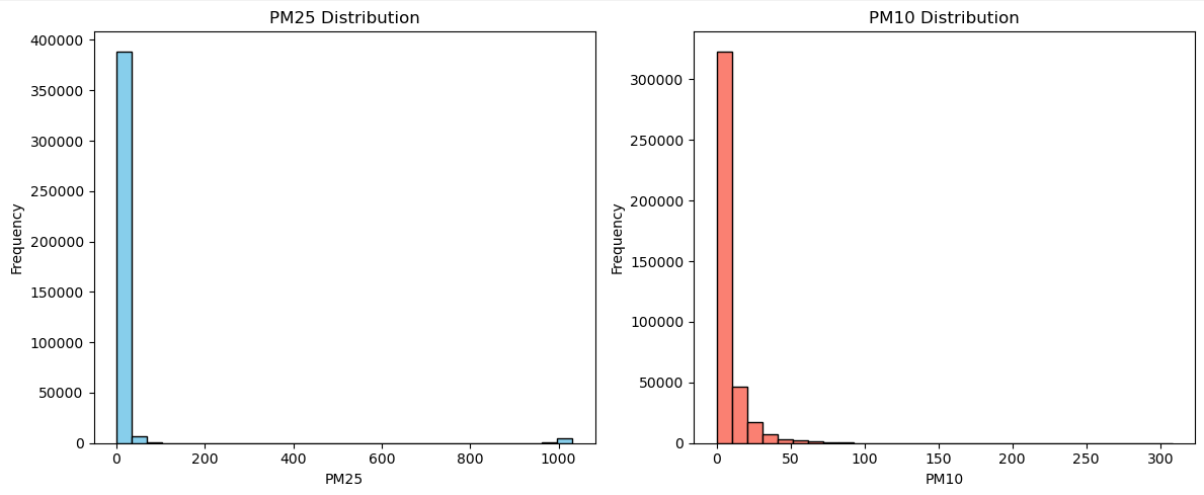
Cell 02

```
import pandas as pd

df = pd.read_csv("microclimate-sensors-data.csv")

for column in df.columns:
    if df[column].isnull().sum() > 0:
        if df[column].dtype == 'O':  # non-numeric column
            mode_value = df[column].mode()[0]
            df[column] = df[column].fillna(mode_value)
            print(f"Missing values in non-numeric column '{column}' have been filled with
    mode value '{mode_value}'")
        else:  # numeric column
            median_value = df[column].median()
            df[column] = df[column].fillna(median_value)
            print(f"Missing values in numeric column '{column}' have been filled with
    median value {median_value}")
```

```
Missing values in non-numeric column 'SensorLocation' have been filled with mode value '1
     Treasury Place'
Missing values in non-numeric column 'LatLong' have been filled with mode value '-37.8185931,
     144.9716404'
Missing values in numeric column 'MinimumWindDirection' have been filled with median value 0.0
Missing values in numeric column 'AverageWindDirection' have been filled with median value
     159.0
Missing values in numeric column 'MaximumWindDirection' have been filled with median value
     353.0
Missing values in numeric column 'MinimumWindSpeed' have been filled with median value 0.0
Missing values in numeric column 'AverageWindSpeed' have been filled with median value 0.8
Missing values in numeric column 'GustWindSpeed' have been filled with median value 2.8
Missing values in numeric column 'AirTemperature' have been filled with median value 15.7
Missing values in numeric column 'RelativeHumidity' have been filled with median value 68.4
Missing values in numeric column 'AtmosphericPressure' have been filled with median value
     1014.6
Missing values in numeric column 'PM25' have been filled with median value 3.0
Missing values in numeric column 'PM10' have been filled with median value 5.0
Missing values in numeric column 'Noise' have been filled with median value 68.3
```

Cell 03

```
import matplotlib.pyplot as plt

# Plot histograms
plt.figure(figsize=(12,5))

plt.subplot(1, 2, 1)
plt.hist(df['PM25'], bins=30, color='skyblue', edgecolor='black')
plt.title('PM25 Distribution')
plt.xlabel('PM25')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist(df['PM10'], bins=30, color='salmon', edgecolor='black')
plt.title('PM10 Distribution')
plt.xlabel('PM10')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

# Calculate correlation
correlation = df['PM25'].corr(df['PM10'])
print(f"Correlation between PM25 and PM10: {correlation:.2f}")
```




```
Correlation between PM25 and PM10: 0.05
```

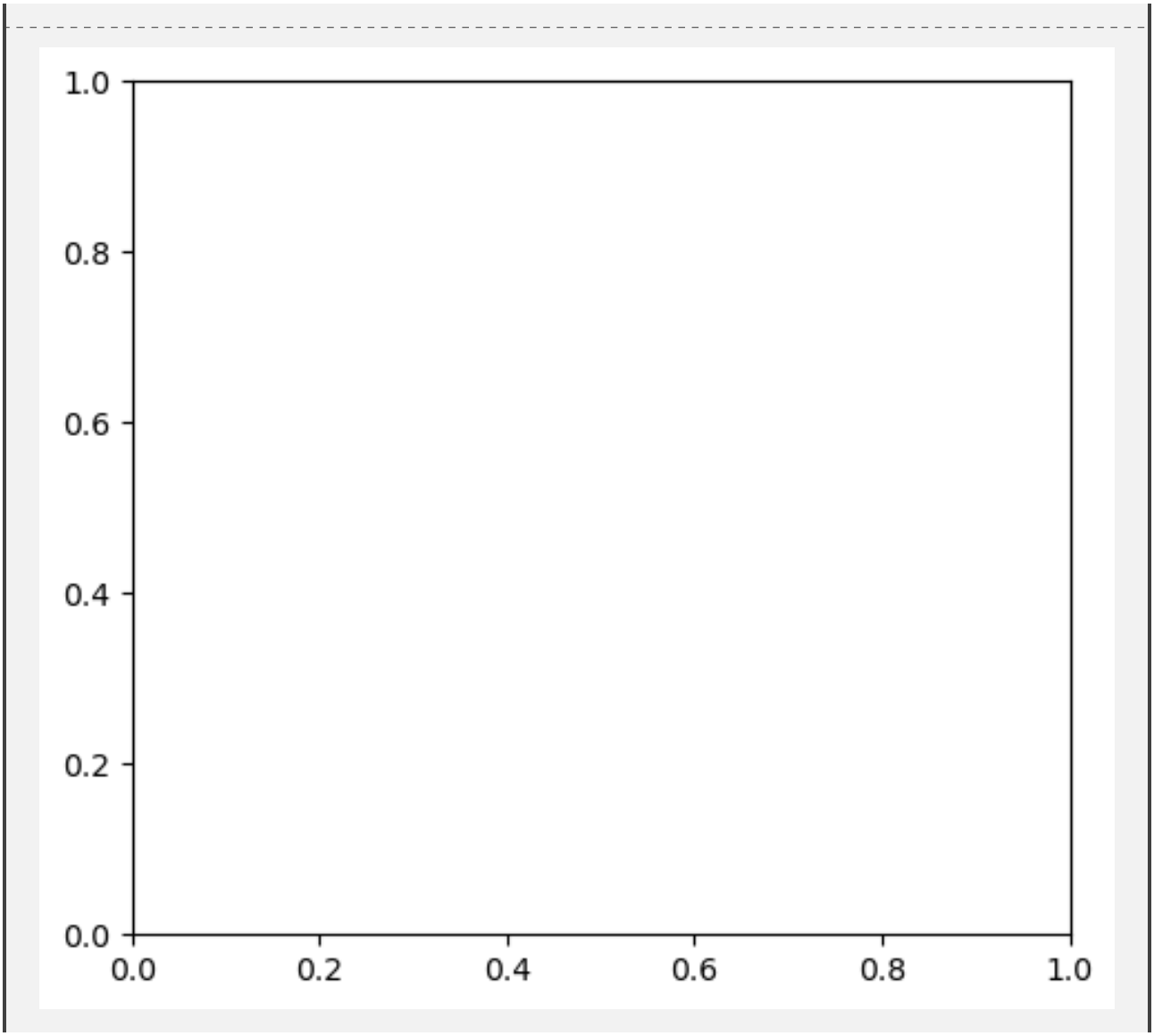
Cell 04

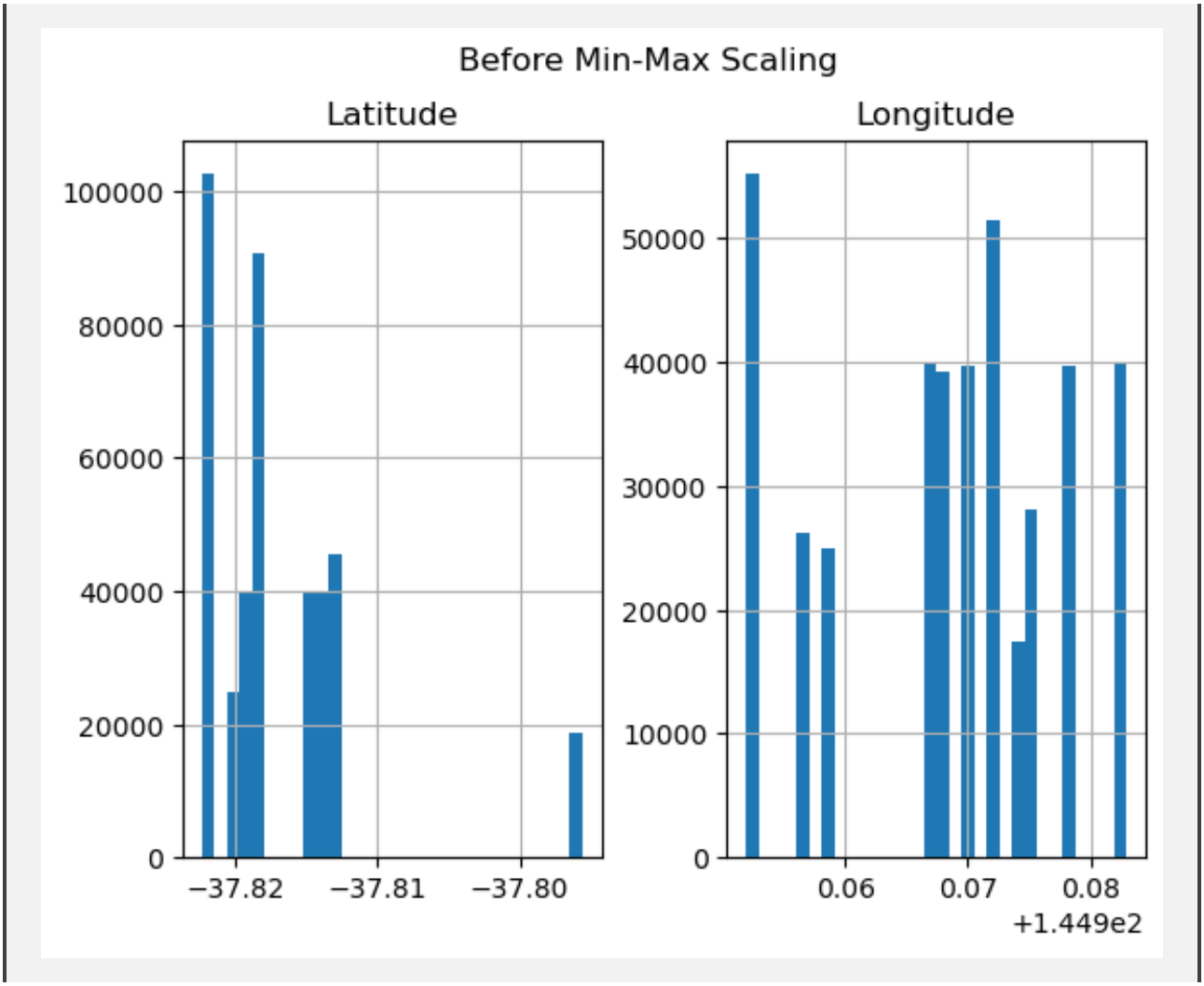
```
import pandas as pd

# Load data (assuming df is already loaded and cleaned)
# df = pd.read_csv("microclimate-sensors-data.csv")

# Split 'LatLong' column into two separate columns: 'Latitude' and 'Longitude'
df[['Latitude', 'Longitude']] = df['LatLong'].str.split(',', expand=True)

# Convert the new columns to float type
df['Latitude'] = df['Latitude'].astype(float)
df['Longitude'] = df['Longitude'].astype(float)

# Display the first few rows of new columns
print(df[['Latitude', 'Longitude']].head())
```

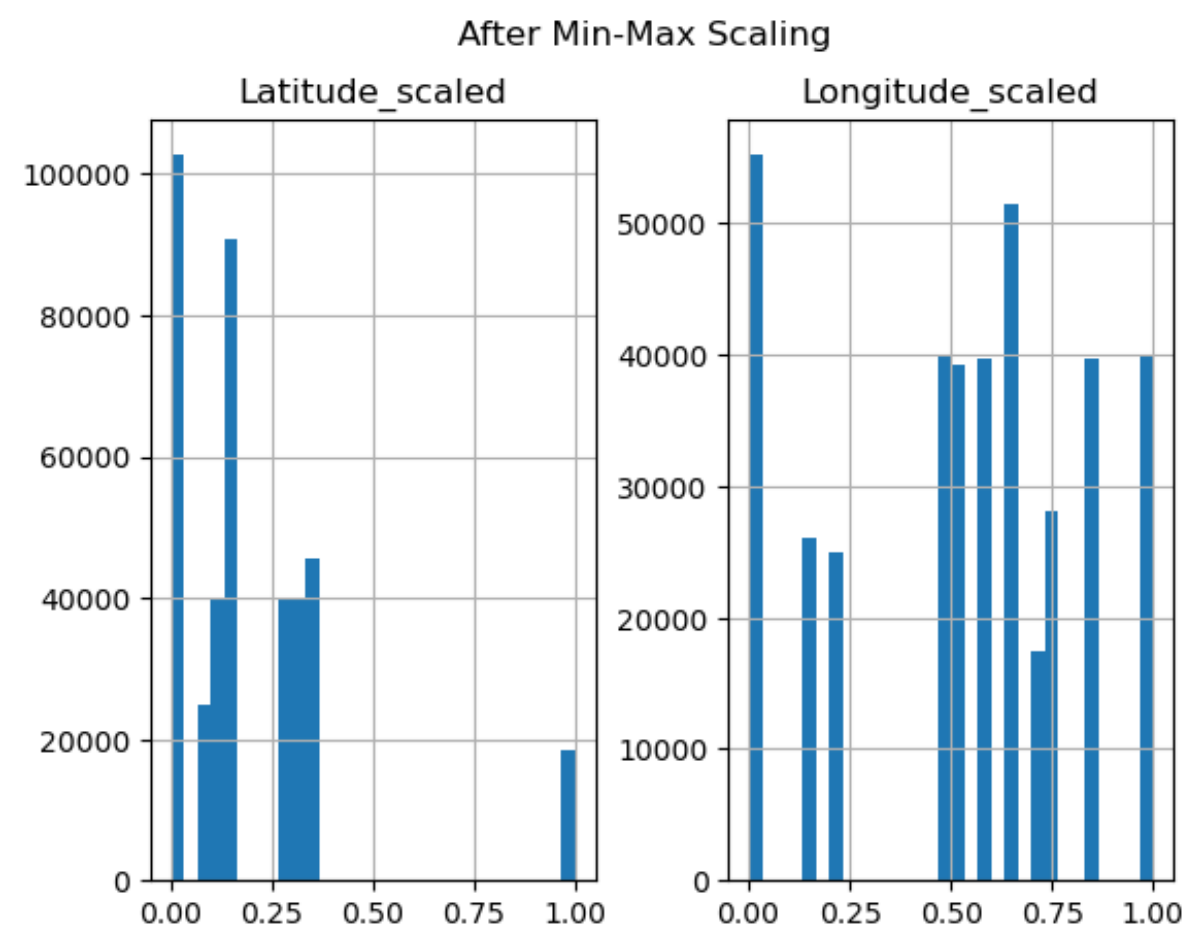
```
  Latitude   Longitude
0 -37.820408  144.959119
1 -37.812888  144.975086
2 -37.818593  144.971640
3 -37.822234  144.982941
4 -37.812888  144.975086
```

Cell 05

```
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler

# Assuming you already split 'LatLong' into 'Latitude' and 'Longitude' columns
# For example: df[['Latitude', 'Longitude']] = df['LatLong'].str.split(',', expand=True)

# Convert these columns from string to float
df['Latitude'] = df['Latitude'].astype(float)
df['Longitude'] = df['Longitude'].astype(float)

# Plot histograms before scaling
plt.figure(figsize=(12,5))

plt.subplot(1, 2, 1)
df[['Latitude', 'Longitude']].hist(bins=30)
plt.suptitle("Before Min-Max Scaling")

# Apply Min-Max scaling
scaler = MinMaxScaler()
df[['Latitude_scaled', 'Longitude_scaled']] = scaler.fit_transform(df[['Latitude',
    'Longitude']])

# Plot histograms after scaling
plt.subplot(1, 2, 2)
df[['Latitude_scaled', 'Longitude_scaled']].hist(bins=30)
plt.suptitle("After Min-Max Scaling")

plt.show()

print("Before scaling, Latitude and Longitude have their original range values.")
print("After Min-Max scaling, the values are transformed to a range between 0 and 1.")
print("This helps many machine learning algorithms work better by normalizing feature
    scales.")
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Before Min-Max Scaling
Latitude
Longitude
100000
80000
60000
40000
20000
0
−37.82
−37.81
−37.80
50000
40000
30000
20000
10000
0
0.06
0.07
0.08
+1.449e2




```
Before scaling, Latitude and Longitude have their original range values.
After Min-Max scaling, the values are transformed to a range between 0 and 1.
This helps many machine learning algorithms work better by normalizing feature scales.
```

Cell 06

SIT720 – Task 2.1 Summary Report

Student Name: Xueying Feng
Task: 2.1 – Summary Report (Weeks 1 & 2)
Date: July 2025

1. Summary of Weekly Content – Week 1 & 2

1. Overview of Week 1 and 2 Content

During the first two weeks of this unit, I have learned foundational concepts and practical skills essential for machine learning, focusing on Python programming and data wrangling.

- **Week 1** focused on introducing machine learning (ML) concepts, types of ML algorithms, and data representation. I explored how data is structured for ML, including features and labels, and the role of Python and libraries like Pandas and Scikit-learn in implementing ML workflows.
- **Week 2** introduced data wrangling techniques using Pandas and data preprocessing for ML using Scikit-learn. The emphasis was on handling messy real-world data, including missing values, inconsistent formats, and categorical encoding. Techniques such as filling missing values with mean or median, label encoding for categorical variables, and min-max scaling for feature normalisation were covered in detail.

The weekly learning also covered inspecting datasets with Pandas functions (`head()`, `info()`, `describe()`) and visualising data distributions using histograms, which are crucial for understanding data before modelling.

---

2. Summary of Reading and Reference Materials

Throughout Weeks 1 and 2, I consulted the following resources:

Internal Learning Materials

- SIT720 CloudDeakin Weekly Modules (Week 1 & 2)
- In-class tutorial notebooks and exercises

External Sources

- *Scikit-learn Documentation* – https://scikit-learn.org/stable/
- *NumPy Documentation* – https://numpy.org/doc/
- Articles on Towards Data Science (Medium)
  - "Understanding Supervised vs Unsupervised Learning"
  - "A Gentle Introduction to Probability in Machine Learning"
- YouTube video: *Python for Data Science – FreeCodeCamp*
- Kaggle tutorial: Exploratory Data Analysis (EDA) using Python

Books Referenced

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (2nd Ed.)
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (ISL)

---

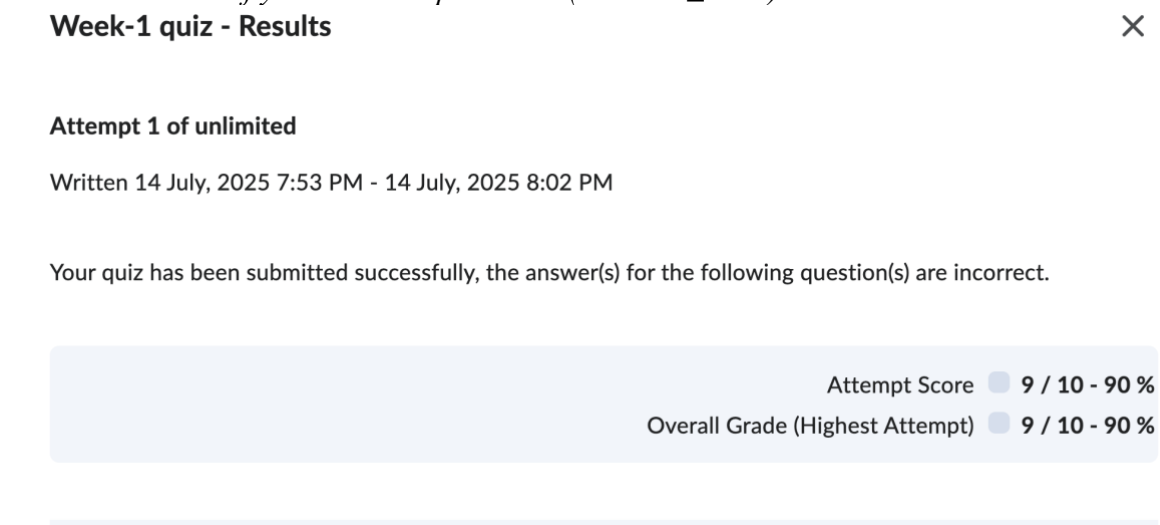
3. Reflection on Learning

By engaging with this material, I have deepened my understanding of the critical importance of data quality in ML. Clean, well-prepared data is essential for model accuracy and interpretability. I learned practical methods to deal with missing data using statistical imputation (mean vs median), encode categorical variables into numeric formats for model compatibility, and normalize feature values to a common scale to prevent bias during model training.

The hands-on coding exercises reinforced my ability to implement these preprocessing steps and interpret the data distribution, which lays the foundation for building robust ML models

✅ Week 1 – Pass Activity (1.28)

Insert screenshot of your Week 1 quiz score (must be ≥ 85%)



Week 2 – Pass Activity (2.14)

Insert screenshot of your Week 2 quiz score (must be ≥ 85%)