1.0 - Introduction

In this project, we have chosen a dataset that is simple and talks about height and weight measurements of individuals. By examining the relationship between height and weight, we can uncover patterns and trends that are crucial for health and fitness research. Here is a detailed overview of the dataset's content:

Dataset Link: https://www.kaggle.com/datasets/kkaranismm/heightweight-csv

Variables	*Description*
Index	A unique identifier for every individual in the dataset.
Height (Inches)	The height of the individual measured in inches. Heights in the dataset range from approximately 62 inches to 75 inches.
Weight (Pounds)	The weight of the individual measured in pounds. Weights in the dataset range from approximately 83 pounds to 168 pounds.

2.0 – Problem Statement

The primary objective of this analysis is to determine if there is a significant correlation between height and weight among the individuals in the dataset. By examining the relationship between these two variables, we aim to identify any patterns or trends that could inform health and fitness recommendations. Specifically, we seek to understand how height influences weight and vice versa, and whether this relationship can be used to develop predictive models for health assessments. This analysis will provide valuable insights that can be applied to improve health and fitness strategies, ultimately contributing to better overall well-being.

3.0 – Data Loading & Preprocessing

1. Data Loading and Preprocessing:

```
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv("C:/Users/harik/OneDrive/Documents/NWU DOCS/ML/week7/SOCR-HeightWeight.csv")

# Display basic information about the dataset
print(df.info())

# Summary statistics to understand data distribution
print(df.describe())

# Calculate Q1 (25th percentile) and Q3 (75th percentile) for outlier detection
Q1 = df[['Height(Inches)', 'Weight(Pounds)']].quantile(0.25)
Q3 = df[['Height(Inches)', 'Weight(Pounds)']].quantile(0.75)
IQR = Q3 - Q1

# Align the DataFrame and remove outliers using the IQR method
df_aligned, IQR_aligned = df.align(IQR, axis=1, copy=False)
df_filtered = df_aligned[~((df_aligned < (Q1 - 1.5 * IQR_aligned)) | (df_aligned > (Q3 + 1.5 * IQR_aligned))).any(axis=1)]

# Prepare the independent variable (X) and dependent variable (y)
X = df_filtered[['Height(Inches)']]  # Feature
y = df_filtered['Weight(Pounds)']    # Target

# Display the first few rows of X and y to verify the data
print(X.head())
print(y.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Index           25000 non-null  int64  
 1   Height(Inches)  25000 non-null  float64
 2   Weight(Pounds)  25000 non-null  float64
dtypes: float64(2), int64(1)
memory usage: 586.1 KB
None
              Index  Height(Inches)  Weight(Pounds)
count  25000.000000    25000.000000    25000.000000
mean   12500.500000       67.993114      127.079421
std     7217.022701        1.901679       11.660898
min        1.000000       60.278360       78.014760
25%     6250.750000       66.704397      119.308675
50%    12500.500000       67.995700      127.157750
75%    18750.250000       69.272958      134.892850
max    25000.000000       75.152800      170.924000
   Height(Inches)
0        65.78331
1        71.51521
2        69.39874
3        68.21660
4        67.78781
0    112.9925
1    136.4873
2    153.0269
3    142.3354
4    144.2971
Name: Weight(Pounds), dtype: float64
```

The output shows information about a dataset containing height and weight measurements. There are 25,000 people in the data, and the average height is 67.99 inches, while the average weight is 127.08 pounds. The data also shows the minimum, maximum, and median values for both height and weight.

4.0 – Model training

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Split the dataset into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model using the training data
model.fit(X_train, y_train)

# Display the model's coefficients and intercept
print("Model Coefficients (Slope):", model.coef_)
print("Model Intercept:", model.intercept_)
```

```
Model Coefficients (Slope): [2.93434123]
Model Intercept: -72.44748395731321
```

The output provides the coefficients of a linear regression model. The slope coefficient of 2.93434123 indicates that for every unit increase in the predictor variable, the predicted value increases by 2.93434123 units. The intercept coefficient of -72.44748395731321 represents the predicted value when the predictor variable is zero.

5.0 – Evaluation using Mean Squared Error (MSE)

```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

# Calculate additional evaluation metrics
mae = mean_absolute_error(y_test, y_pred)  # Mean Absolute Error
rmse = np.sqrt(mean_squared_error(y_test, y_pred))  # Root Mean Squared Error
r2 = r2_score(y_test, y_pred)  # R-squared

# Display all metrics
print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error (MSE): 94.37823026276934
Mean Absolute Error (MAE): 7.807889423289418
Root Mean Squared Error (RMSE): 9.714845869223522
R-squared: 0.24129269794463915
```

The output shows the evaluation metrics for a regression model. The Mean Squared Error (MSE) is 94.38, which indicates the average squared difference between the predicted and actual values. The Mean Absolute Error (MAE) is 7.81, which indicates the average absolute difference between the predicted and actual values. The Root Mean Squared Error (RMSE) is 9.71, which is the square root of the MSE and provides a measure of the average error in the same units as the target variable. The R-squared value is 0.24, which indicates that the model explains 24.13% of the variance in the target variable.

6.0 – Reflection on the Problem & Solution

Model Training

A linear regression model was used to analyze the relationship between height and weight. The model's coefficients indicate that for every unit increase in height, the predicted weight increases by approximately 2.93 pounds. The intercept suggests the predicted weight when height is zero, though this is more of a theoretical value.

Evaluation Metrics

The model's performance was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The R-squared value of 0.24 indicates that the model explains 24.13% of the variance in weight, suggesting that while there is a correlation, other factors also play a significant role.

Potential Improvements

The reflection highlights the potential for enhancing the model's predictive capability. This could involve incorporating additional relevant features, improving data preprocessing techniques, or experimenting with more complex models. Continuous refinement based on these evaluations can lead to a more robust and effective predictive model.

7.0 – Conclusion

In summary, while the regression model demonstrates a decent predictive capability, as indicated by the MSE, MAE, RMSE, and R-squared values, there is potential for enhancement. Factors such as additional relevant features, data preprocessing, or even experimenting with more complex models could lead to improved performance. Continuous refinement based on these evaluations can help create a more robust and effective predictive model in the healthcare context. By leveraging these metrics, we can systematically improve the model, ensuring it provides accurate and reliable predictions.