

Supervised Learning using Linear Regression

```
# ➡ Predict Part 3 CGPA using Supervised Learning (Linear Regression)

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.metrics import mean_absolute_error, r2_score

# 1 Dataset (Sample CGPA Data)

data = pd.DataFrame({
    'Part1_CGPA': [2.5, 2.9, 3.2, 2.4, 2.2, 3.0, 2.5, 2.8, 3.1, 2.7,
                  3.0, 2.9, 2.6, 3.2, 2.8, 3.3, 3.4, 2.3, 2.9, 3.1],
    'Part2_CGPA': [2.9, 3.2, 3.3, 2.6, 2.6, 3.2, 2.6, 3.0, 3.2, 2.8,
                  3.1, 3.3, 2.7, 3.4, 2.9, 3.5, 3.6, 2.5, 3.1, 3.3],
    'Part3_CGPA': [3.0, 3.4, 3.4, 2.9, 2.9, 3.3, 2.7, 3.1, 3.3, 3.0,
                  3.2, 3.4, 2.8, 3.5, 3.0, 3.6, 3.7, 2.8, 3.3, 3.4]
})

# 2 Features and Target

X = data[['Part1_CGPA', 'Part2_CGPA']]
y = data['Part3_CGPA']

# 3 Train/Test Split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)

# 4 Train Model

model = LinearRegression()
model.fit(X_train, y_train)
```

```

# 5 Evaluate Model

y_pred = model.predict(X_test)
print("Model Accuracy (R2 Score):", round(r2_score(y_test, y_pred), 3))
print("Mean Absolute Error:", round(mean_absolute_error(y_test, y_pred),
3))

# 6 Predict New Student's Part 3 CGPA

print("\n➡ Enter your previous part CGPAs:")
p1 = float(input("Part 1 CGPA: "))
p2 = float(input("Part 2 CGPA: "))

new_data = pd.DataFrame([p1, p2], columns=['Part1_CGPA', 'Part2_CGPA'])
predicted_cgpa = model.predict(new_data)[0]

print(f"\n📊 Predicted Part 3 CGPA: {predicted_cgpa:.2f}")

# 7 Visualization (Graph)

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', label='Predicted vs Actual')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red', linestyle='--', label='Perfect Prediction Line')
plt.title("➡ Actual vs Predicted Part 3 CGPA")
plt.xlabel("Actual Part 3 CGPA")
plt.ylabel("Predicted Part 3 CGPA")
plt.legend()
plt.grid(True)
plt.show()

```

Output

Model Accuracy (R² Score): 0.937
Mean Absolute Error: 0.069

➡ Enter your previous part CGPAs:
Part 1 CGPA: 2.56
Part 2 CGPA: 2.91

📊 Predicted Part 3 CGPA: 3.12

