TASK OF DAY- 6:-

1. *Write a simple Python program to train a linear regression model using any small dataset.*

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

file_path = r'C:\Users\Heet\Desktop\PROGRAMS\NEW LEARNINGS\Python\Salary_dataset.csv'

try:
    df = pd.read_csv(file_path)
    print("Dataset loaded successfully.")
    print(df.head())
except FileNotFoundError:
    print(f"Error: '{file_path}' not found. Please ensure the file is in the correct directory.")
    exit()
except Exception as e:
    print(f"An error occurred while loading the dataset: {e}")
    exit()

if 'YearsExperience' in df.columns and 'Salary' in df.columns:
    X = df[['YearsExperience']]
    y = df['Salary']
    print("\nFeatures (X) and Target (y) defined.")
else:
    print("Error: 'YearsExperience' or 'Salary' columns not found in the dataset.")
    print(f"Available columns: {df.columns.tolist()}")
    exit()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(f"\nData split into training (X_train: {X_train.shape}, y_train: {y_train.shape})")
print(f"and testing (X_test: {X_test.shape}, y_test: {y_test.shape}) sets.")

model = LinearRegression()
model.fit(X_train, y_train)
print("\nLinear Regression model trained successfully.")
```

```
print(f"Model Intercept: {model.intercept_:.2f}")
print(f"Model Coefficient (YearsExperience): {model.coef_[0]:.2f}")

y_pred = model.predict(X_test)
print("\nPredictions made on the test set.")

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("\nModel Evaluation Metrics:")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R2): {r2:.2f}")

new_years_experience = np.array([[6.5]])
predicted_salary = model.predict(new_years_experience)
print(f"\nPredicted Salary for {new_years_experience[0][0]} years of experience:
${predicted_salary[0]:.2f}")
```

OUTPUT :-

```
Dataset loaded successfully.
   Unnamed: 0  YearsExperience   Salary
0           0              1.2  39344.0
1           1              1.4  46206.0
2           2              1.6  37732.0
3           3              2.1  43526.0
4           4              2.3  39892.0

Features (X) and Target (y) defined.

Data split into training (X_train: (24, 1), y_train: (24,))
and testing (X_test: (6, 1), y_test: (6,)) sets.

Linear Regression model trained successfully.
Model Intercept: 24380.20
Model Coefficient (YearsExperience): 9423.82

Predictions made on the test set.

Model Evaluation Metrics:
Mean Absolute Error (MAE): 6286.45
Mean Squared Error (MSE): 49830096.86
Root Mean Squared Error (RMSE): 7059.04
R-squared (R2): 0.90
```

2. *Print model coefficients, intercept, and evaluate model performance.*

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

df = pd.read_csv(r'C:\Users\Heet\Desktop\PROGRAMS\NEW
LEARNINGS\Python\Salary_dataset.csv')

print("First 5 rows of the dataset:")
print(df.head().to_markdown(index=False, numalign="left", stralign="left"))

print("\nColumn information:")
print(df.info())

print("\nMissing values before dropping:")
print(df.isnull().sum().to_markdown(numalign="left", stralign="left"))

df.dropna(inplace=True)

X = df[['YearsExperience']]
y = df['Salary']

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

model = LinearRegression()

model.fit(X_train, y_train)

print("\nModel Coefficients:")
print(f"Coefficient: {model.coef_[0]}")
print(f"Intercept: {model.intercept_}")

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)

print("\nModel Performance Metrics:")
```

```
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"R-squared (R2): {r2}")
```

OUTPUT :-

```
First 5 rows of the dataset:
| Unnamed: 0 | YearsExperience | Salary |
|:-----------|:----------------|:-------|
| 0          | 1.2             | 39344  |
| 1          | 1.4             | 46206  |
| 2          | 1.6             | 37732  |
| 3          | 2.1             | 43526  |
| 4          | 2.3             | 39892  |

Column information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       30 non-null     int64
 1   YearsExperience  30 non-null     float64
 2   Salary           30 non-null     float64
dtypes: float64(2), int64(1)
memory usage: 852.0 bytes
None

Missing values before dropping:
|                | 0   |
|:---------------|:----|
| Unnamed: 0     | 0   |
| YearsExperience | 0   |
| Salary         | 0   |

Model Coefficients:
Coefficient: 9423.81532303098
Intercept: 24380.20147947369
```