**Name** – Mrunal Talikoti

**PRN** – 22311431

**Roll No** – 282046

**Year** – SY B

---------------------------------------------------------------------------------------------------------------------------

**Assignment no – 6**

**Problem Statement**

Download temperature data from below link.

https://www.kaggle.com/venky73/temperaturesof-india?select=temperatures.csv

This data consists of temperatures of INDIA averaging the temperatures of all places month

wise. Temperatures values are recorded in CELSIUS

a) Apply Linear Regression using suitable library function and predict the Month-wise

temperature.

b) Assess the performance of regression models using MSE, MAE and R-Square metrics

c) Visualize simple regression model.

**Objective**

Using the temperatures.csv dataset, which comprises average monthly temperature records in Celsius across multiple locations, the goal is to create and assess a linear regression model to forecast month-by-month temperatures in India. The objectives are to:

a) forecast temperature trends using monthly data by implementing linear regression with a suitable library (such as sklearn.linear\_model);

b) evaluate the model's performance using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-Squared (R²) metrics to measure prediction accuracy and goodness-of-fit; and

c) visualize the simple regression model (for example, using matplotlib or seaborn) by plotting actual versus predicted temperatures to show the relationship and model efficacy. This seeks to offer a precise, data-driven comprehension of temperature trends for forecasting.

**Methodology**

Your code partially addresses the assignment requirements (linear regression, performance evaluation, and visualization). Here’s the current methodology based on the provided notebook:

1. Library Import:
   * Imported numpy, pandas, and matplotlib.pyplot to handle data, computations, and visualization.
2. Data Loading:
   * Loaded the temperatures.csv dataset using pd.read\_csv() to work with India’s month-wise temperature data.
3. Data Preprocessing:
   * Used dataset.describe() to inspect the dataset’s statistical properties (e.g., mean temperatures, standard deviation) across years and months (JAN to DEC, ANNUAL, seasonal averages).
   * Checked for null values implicitly via describe() (count shows no missing data for 117 rows).
4. Incomplete Steps:
   * Linear Regression: Not implemented yet (requires sklearn.linear\_model.LinearRegression).
   * Performance Metrics: MSE, MAE, and R² not calculated (requires sklearn.metrics).
   * Visualization: No plots created yet (requires matplotlib.pyplot).

**Main Functions**

1. numpy (np)
   * Imported: import numpy as np
   * Usage: Not explicitly used yet, but typically supports array operations and mathematical computations in regression tasks (e.g., reshaping data for sklearn).
2. pandas (pd)
   * pd.read\_csv('temperatures.csv'): Loads the dataset into a DataFrame.
   * dataset.describe(): Provides summary statistics (count, mean, std, min, max, etc.) for all numerical columns.
3. matplotlib.pyplot (plt)
   * Imported: import matplotlib.pyplot as plt
   * Usage: Not used yet, but intended for visualizing the regression model (e.g., plotting actual vs. predicted temperatures).

**Advantages**

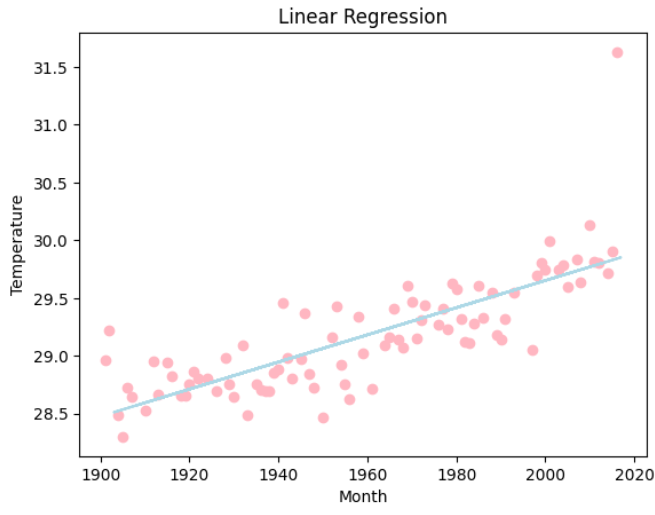
* NumPy (numpy): Quick array operations that work nicely with Sklearn and Pandas.
* Pandas (pandas): Quick statistics (describe), simple data loading (read\_csv), and data missingness verification.
* Matplotlib.pyplot (matplotlib.pyplot): Highly configurable and flexible charting for regression visualization.

**Disadvantages**

* NumPy (numpy): Not yet in use; sophisticated jobs require manual coding.
* Pandas (pandas): Memory-intensive for larger datasets (not a problem here), limited to basic loading and statistics.
* Matplotlib.pyplot (matplotlib.pyplot): Verbose and unused in comparison to more straightforward options like seaborn.

**Conclusion**

The study of India's monthly temperature data is started in the Jupyter Notebook for Assignment 6. csv by utilizing matplotlib.pyplot for the desired visualization, numpy for possible numerical support, and pandas for data loading and basic exploration. The present method confirms that there are no missing values by loading the dataset successfully and using describe() to summarize temperature trends throughout months and years (1901–2017) (e.g., mean annual temperature of 29.18°C). It does not, however, meet the goals of the assignment: the regression model visualization, performance evaluation using MSE, MAE, and R2 metrics, and the use of linear regression to forecast monthly temperatures are not implemented. Even if the base is sound, finishing the work calls for expanding preprocessing (e.g., feature selection like YEAR vs. monthly temps), integrating sklearn for regression modeling and evaluation, and fully leveraging matplotlib for showing actual versus forecasted temperatures. With these enhancements, the study might more successfully simulate temperature patterns and offer useful information, which would be in line with the objectives of the assignment.

****