EXPERIMENT NO. 2

Implement simple, multivariate and polynomial linear regression models

Aim: Implement simple, multivariate and polynomial linear regression models.

Theory:

REGRESSION MODELS: Regression models are tools used to predict a continuous outcome based on one or more input variables. They help find the relationship between inputs (features) and an output (target) by fitting a function, usually a line or curve, to the data.

- Simple Regression: Predicts using one feature.
- Multivariate Regression: Uses multiple features to predict.
- Polynomial Regression: Uses curved lines (polynomials) to capture more complex relationships.

1. Simple Linear Regression:

Theory:

- Simple linear regression models the relationship between **one independent variable** (feature) and **one dependent variable** (target).
- The goal is to fit a **straight line** that best predicts the target variable from the feature.
- The model has the form:

$$y=\beta 0+\beta 1x+\varepsilon$$

Where:

- y is the dependent variable,
- x is the independent variable,
- β0 is the intercept (value of y when x = 0),
- β1 is the slope (rate of change of y with respect to x),
- ε is the error term capturing noise or unexplained variation.

Use case: Predicting sales based on advertising budget, predicting height based on age, etc.

2. Multivariate Linear Regression (Multiple Linear Regression)

Theory:

- Multivariate linear regression extends simple linear regression to **multiple independent variables**.
- The model finds a **hyperplane** that best fits the data points in multidimensional space.
- The equation is:

$$y=\beta 0+\beta 1x1+\beta 2x2+\cdots+\beta nxn+\varepsilon$$

Where:

- $x1,x2,\ldots,xn$ are multiple independent variables,
- $\beta 1,\beta 2,\ldots, \beta n$ are coefficients representing the impact of each feature,
- $\beta 0$is the intercept.

Use case: Predicting house prices based on size, number of bedrooms, location; predicting sales from multiple marketing channels.

3. Polynomial Regression

Theory:

- Polynomial regression is a type of linear regression that models the relationship between the independent variable(s) and the dependent variable as an **nth degree polynomial**.
- It allows for **non-linear relationships** between the features and the target but is still linear in the coefficients.
- For one feature, the model is:

$$y = \beta 0 + \beta 1x + \beta 2x^2 + \beta 3x^3+\cdots+\beta dx^d + \varepsilon$$

Where:

- d is the degree of the polynomial,
- The model includes powers of the feature x to capture curvature.
- Polynomial regression can be extended to multiple variables by including polynomial terms and interaction terms of the features.

Use case: Modeling growth rates, curvilinear relationships like acceleration vs time, temperature variation, etc.

CONCLUSION:

Regression models are powerful tools for understanding and predicting relationships between variables. Simple linear regression helps model the effect of one predictor on a target variable, while multivariate regression extends this to multiple predictors, capturing more complex influences. Polynomial regression further enhances modeling capabilities by allowing for non-linear relationships. Together, these models provide a flexible framework for analyzing and forecasting continuous outcomes in various fields. Implementing and comparing these techniques helps in selecting the best model suited to the data and the problem at hand.

VIVA QUESTIONS:

1. What is the difference between simple linear regression and multivariate linear regression?
2. Why do we use polynomial regression instead of simple linear regression?
3. What does the term 'coefficient' in a regression model represent?
4. How do you evaluate the performance of a regression model?
5. Can regression models be used for classification problems? Why or why not?

PROGRAM:

```
# Before running, install required packages:
# pip install scikit-learn numpy pandas
```

```
from sklearn.linear_model import LinearRegression
import numpy as np
# Feature: Humidity only
X = np.array([[30], [35], [40], [45], [50], [55], [60], [65], [70], [75]])
# Target: Temperature
y = np.array([25, 24, 22, 21, 20, 18, 17, 15, 14, 13])
# Create and train the model
model = LinearRegression()
model.fit(X, y)
# Predict temperature for a new humidity value
humidity_new = np.array([[68]])
temp_pred = model.predict(humidity_new)
print(f"Predicted temperature at {humidity_new[0][0]}% humidity is {temp_pred[0]:.2f}°C")
# Print model parameters
print(f"Intercept (β0): {model.intercept_:.2f}")
print(f"Slope (β1): {model.coef_[0]:.2f}")
```