NAME :- JOBA ADHIKARY

EMAIL:- joba.adhikary55@gmail.com

#1. What is Simple Linear Regression?

--Simple liner Regression is a statistical method used to model the relationship between two variables.

Those 2 variables are:-

- Independent variable:- the predictor or input variable
- Dependent variable:- the response or output variable

#2. What are the key assumptions of Simple Linear Regression?

--The key assumptions of Simple Linear Regression are:-

*Linearity

*Independence Error

*Homoscedasticity

*Normality of Errors

#3. What is heteroscedasticity, and why is it important to address in regression models?

--refers to a situation in regression analysis where the variance of the errors is not constant across all levels of the independent variable.

*The spread of residuals increases or decreases with X or the predicted Y values.

*This violates the homoscedasticity assumption of linear regression, which assumes constant error variance.

>>Heteroscedasticity is important to address in regression models because:-

*inefficient estimates

*Misleading model interpretation

*Biased Standard errors

#4. What is Multiple Linear Regression?

--Multiple Linear Regression is an extension of Simple Linear Regression that models the relationship between one dependent variable (Y) and two or more independent variables.

#5. What is polynomial regression, and how does it differ from linear regression?

--Polynomial Regression is a type of regression analysis where the relationship between the independent variable (X) and the dependent variable.

It differs from Linear Regression by:-

*Linear Regression:-

>>Straight line

>>It Captures linear trends

>>Its Complexity is low.

*Polynomial Regression:-

>>Curved line

>>Captures nonlinear trends

>>Increases with degree of polynomial

#6. Implement a Python program to fit a Simple Linear Regression model to the following sample data: ● X = [1, 2, 3, 4, 5] ● Y = [2.1, 4.3, 6.1, 7.9, 10.2] Plot the regression line over the data points.

--import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model

import LinearRegression

X = [1, 2, 3, 4, 5]

Y = [2.1, 4.3, 6.1, 7.9, 10.2]

x = np.array(X).reshape(-1, 1)

Y = np.array(Y)

model = LinearRegression()

model.fit(X, Y)

Y_pred = model.predict(X)

print(f"Intercept (β_0): {model.intercept_:.2f}")

print(f"Slope (β_1): {model.coef_[0]:.2f}")

plt.scatter(X, Y, color='blue', label='Actual Data')

plt.plot(X, Y_pred, color='red', label='Regression Line')

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Simple Linear Regression')

plt.legend()

plt.grid(True)

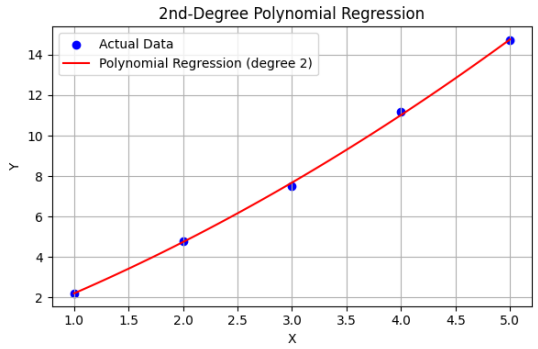
plt.show()

#7. Fit a Multiple Linear Regression model on this sample data: ● Area = [1200, 1500, 1800, 2000] ● Rooms = [2, 3, 3, 4] ● Price = [250000, 300000, 320000, 370000] Check for multicollinearity using VIF and report the results.

#8. Implement polynomial regression on the following data: ● X = [1, 2, 3, 4, 5] ● Y = [2.2, 4.8, 7.5, 11.2, 14.7] Fit a 2nd-degree polynomial and plot the resulting curve.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
Y = np.array([2.2, 4.8, 7.5, 11.2, 14.7])
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
model = LinearRegression()
```

```
model.fit(X_poly, Y)
X_curve = np.linspace(1, 5, 100).reshape(-1, 1)
X_curve_poly = poly.transform(X_curve)
Y_curve = model.predict(X_curve_poly)
plt.scatter(X, Y, color='blue', label='Actual Data')
plt.plot(X_curve, Y_curve, color='red', label='Polynomial Regression
(degree 2)')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('2nd-Degree Polynomial Regression')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



#9. Create a residuals plot for a regression model trained on this data: ● X = [10, 20, 30, 40, 50] ● Y = [15, 35, 40, 50, 65] Assess heteroscedasticity by examining the spread of residuals.

```
-- import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model
import LinearRegression
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
Y = np.array([5, 5, 0, 0, 5])
model = LinearRegression()
model.fit(X, Y)
Y_pred = model.predict(X)
residuals = Y - Y_pred
plt.figure(figsize=(6, 4))
plt.scatter(Y_pred, residuals, color='blue', label='Residuals')
plt.axhline(y=0, color='red', linestyle='--', label='Zero Error Line')
plt.xlabel('Predicted Values')
```

```
plt.ylabel('Residuals')
plt.title('Residual Plot for Linear Regression')
plt.grid(True)
plt.legend()
plt.show()
```