Regression & Its Evaluation Assignment

Question 1: What is Simple Linear Regression?

Answer: Simple linear regression is a statistical method used to measure the relation between one independent variable and one dependent variable using a straight line.
It aims to find the best-fitting line that describes how the dependent variable changes as the independent variable changes.

Question 2: What are the key assumptions of Simple Linear Regression?

Answer: Key assumptions are:
Linearity:

The relationship between the independent and dependent variables should be linear, meaning a straight line .

Independence:

The observations in the dataset should be independent of each other.

Homoscedasticity:

The spread of the data points around the regression line should be roughly the same for all values of the predictor variable.

Normality:

For any fixed value of the independent variable, the dependent variable should be normally distributed.

Question 3: What is heteroscedasticity, and why is it important to address in regression models?

Answer: The spread of the data points around the regression line should be roughly the same for all values of the predictor variable.

Heteroscedasticity, the uneven spread of data points, can distort the results of statistical analyses, leading to inaccurate conclusions about relationships between variables

Question 4: What is Multiple Linear Regression?

Answer: Multiple linear regression is a statistical method that analyzes the relationship between one dependent variable and two or more independent variables.

It aims to find the best-fitting linear equation that predicts the dependent variable based on the values of the independent variables.

Question 5: What is polynomial regression, and how does it differ from linear regression?

Answer: Polynomial regression is an extension of linear regression that can model non-linear relationships between variables by fitting a polynomial equation to the data.

Unlike linear regression, which assumes a straight-line relationship, polynomial regression can capture curves and other non-linear patterns.

Question 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.

Answer:

```
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([2.1, 4.3, 6.1, 7.9, 10.2])

model = LinearRegression()
model.fit(X, Y)

intercept = model.intercept_
coefficient = model.coef_[0]

print(f"Intercept (a): {intercept:.4f}")
print(f"Coefficient (b): {coefficient:.4f}")

# Predict Y values using the fitted model
Y_pred = model.predict(X)

# Plot the data points and the regression line
plt.figure(figsize=(8, 6))
plt.scatter(X, Y, color='blue', label='Data Points')
plt.plot(X, Y_pred, color='red', label='Regression Line')
plt.title('Simple Linear Regression')
plt.show()
```

Question 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.**

Answer:

Question 10: Imagine you are a data scientist working for a real estate company. You need to predict house prices using features

like area, number of rooms, and location. However, you detect heteroscedasticity and multicollinearity in your regression model. Explain the steps you would take to address these issues and ensure a robust model.

Answer:

1. Detecting Heteroscedasticity and Multicollinearity

2. Addressing Heteroscedasticity

3. Addressing Multicollinearity

4. Model Validation

Question 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**

Answer:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = pd.read_csv('headbrain3.csv')

sns.residplot(x='Head_size', y='Brain_weight', data=data)
plt.xlabel("Head Size")
plt.ylabel("Residuals")
plt.show()
```

Question 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**

Answer:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error

# Given data
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
Y = np.array([2.2, 4.8, 7.5, 11.2, 14.7])

# Create polynomial features (degree 2)
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Fit polynomial regression model
model = LinearRegression()
model.fit(X_poly, Y)

# Predict values using the fitted model
Y_pred = model.predict(X_poly)

# Calculate Mean Squared Error
mse = mean_squared_error(Y, Y_pred)
print(f"Mean Squared Error: {mse}")

# Plot the results
plt.scatter(X, Y, label="Original Data")
plt.plot(X, Y_pred, color='red', label="Polynomial Regression (Degree
2)")
plt.show()
```