|  |  |
| --- | --- |
| Semester | T.E. Semester VI – Computer Engineering |
| Subject | QA |
| Subject Professor In-charge | Prof. Kavita Shirsat |

|  |  |  |
| --- | --- | --- |
| Group Members | Deep Salunkhee 21102A0014  Omkar Patil 21102A0003  Pranav Redij 21102A0005  Sukant Thombare 21102A0037  Shantanu Lagvankar 21102A0043 | |
| Grade and Subject  Teacher’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Mini Project Title | Multiple Regression | |
| Resources / Apparatus Required | Hardware:  Computer system | Software:  Python |
| Description | 1. **Theoretical Background:**    * **Multiple Linear Regression:** The code implements multiple linear regression, a statistical technique used to model the relationship between multiple independent variables (X1, X2, ...) and a dependent variable (Y).    * **Ordinary Least Squares (OLS):** The method used to estimate the parameters of the linear regression model is Ordinary Least Squares. OLS aims to minimize the sum of the squared differences between the observed and predicted values of the dependent variable.    * **Assumptions of Linear Regression:** The validity of the regression results relies on several assumptions, including linearity, independence of errors, homoscedasticity, and normality of errors. 2. **Mathematical Formulation:**    * **Model Equation:** The model equation for multiple linear regression is represented as:   Y = b0 + b1\*X1 + b2\*X2 + ... + bn\*Xn + ε  where Y is the dependent variable, X1, X2, ..., Xn are the independent variables, b0 is the intercept term, b1, b2, ..., bn are the regression coefficients, and ε is the error term.   * + **Coefficients Estimation:** The coefficients (b1, b2, ..., bn) are estimated using the method of Ordinary Least Squares (OLS) to minimize the sum of squared errors between the observed and predicted values of the dependent variable.   + **Model Summary:** The **model.summary()** function provides a detailed summary of the regression results, including coefficients, standard errors, t-values, p-values, and various statistics such as R-squared and adjusted R-squared.  1. **Statistical Metrics:**    * **R-squared (R^2):** R-squared is a measure of the proportion of variance in the dependent variable that is explained by the independent variables. It ranges from 0 to 1, where higher values indicate a better fit of the model to the data.    * **Total Sum of Squares (SST):** SST measures the total variance in the dependent variable.    * **Regression Sum of Squares (SSR):** SSR measures the variance explained by the regression model.    * **Error Sum of Squares (SSE):** SSE measures the unexplained variance or residual variance.    * **Mean Square Regression (MSR):** MSR is the average amount of variance explained by the regression model.    * **Mean Square Error (MSE):** MSE is the average amount of unexplained variance or residual variance.    * **Degrees of Freedom:** Degrees of freedom represent the number of independent pieces of information in the data used to estimate a statistic. In the context of regression, df\_model represents the degrees of freedom for the model, and df\_resid represents the degrees of freedom for the residuals. | |
| Program | import pandas as pd  import statsmodels.api as sm  *# Read data from Excel file*  data = pd.read\_excel("data.xlsx")  *# Separate independent variables (X) and dependent variable (Y)*  X = data[['X1', 'X2']]  Y = data['Y']  *# Add constant term for intercept*  X = sm.add\_constant(X)  *# Create and fit the regression model*  model = sm.OLS(Y, X).fit()  *# Print the model summary*  print(model.summary())  *# Calculate SST (Total Sum of Squares)*  y\_mean = Y.mean()  SST = ((Y - y\_mean) \*\* 2).sum()  *# Calculate SSR (Regression Sum of Squares)*  SSR = ((model.predict(X) - y\_mean) \*\* 2).sum()  *# Calculate SSE (Error Sum of Squares)*  SSE = ((Y - model.predict(X)) \*\* 2).sum()  *# Calculate R-squared*  R\_squared = SSR / SST  *# Calculate MSR (Mean Regression Sum of Squares)*  MSR = SSR / model.df\_model  *# Calculate MSE (Mean Error Sum of Squares)*  MSE = SSE / model.df\_resid  *# Print calculated values*  print("SST:", SST)  print("SSR:", SSR)  print("SSE:", SSE)  print("R^2:", R\_squared)  print("MSR:", MSR)  print("MSE:", MSE)  *# Print model equation*  print("Model Equation:")  print("Y = {:.2f} + {:.2f}\*X1 + {:.2f}\*X2".format(model.params[0], model.params[1], model.params[2])) | |
| Output |  | |