Name - Gaurang A Raorane Div - D15A Roll no - 49 Batch - C

**Experiment - 5**

**Aim:-** To perform regression analysis on a dataset.

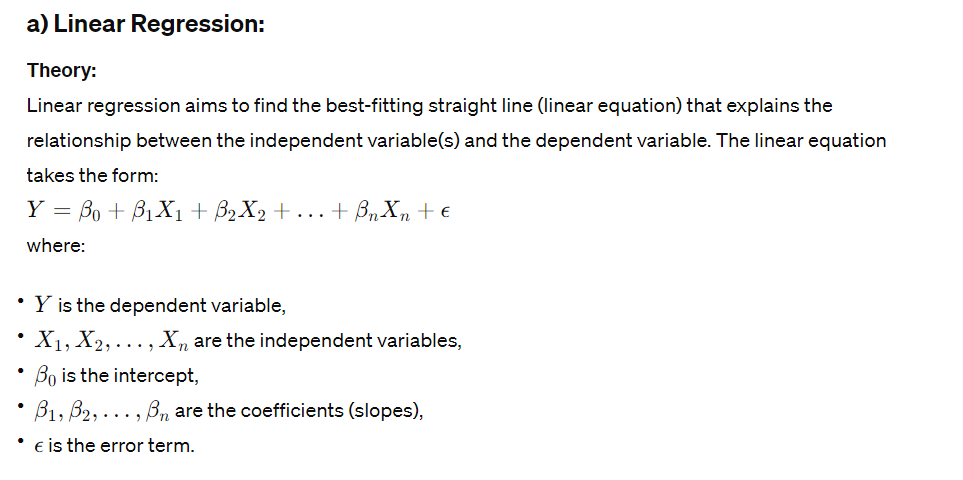
**Problem Statement:-**

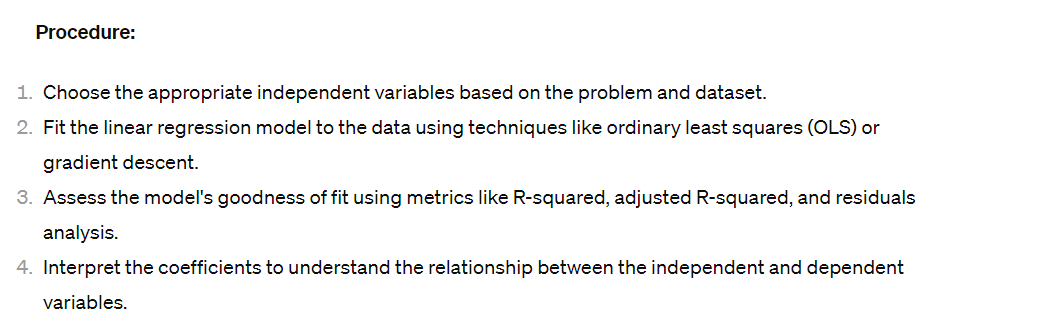
Regression Analysis

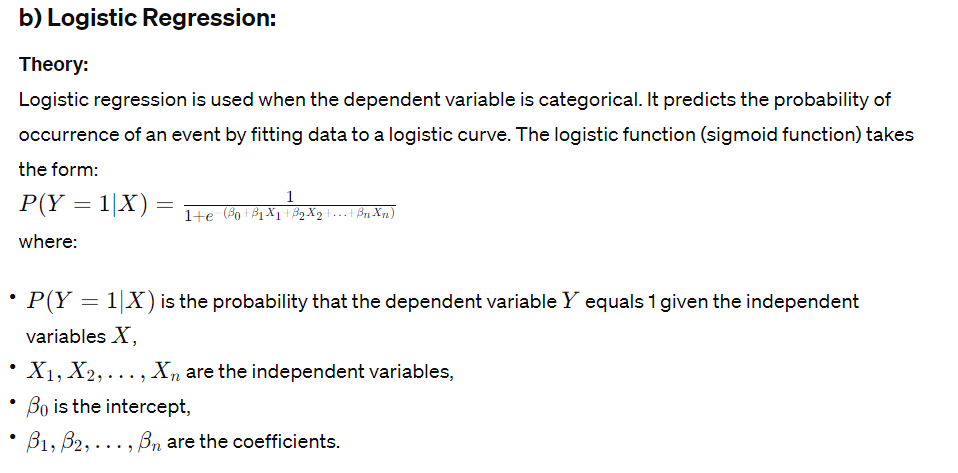
a) Perform Linear and Logistic regression to find out relation between variables

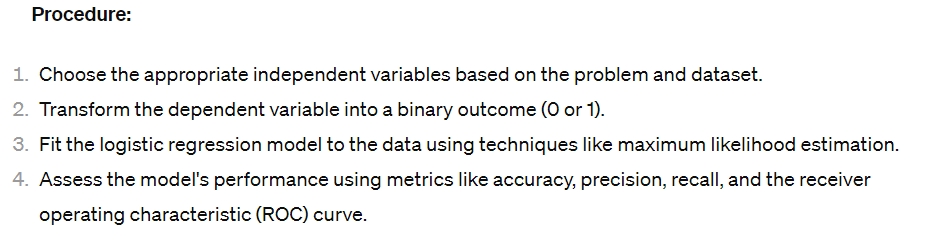
b) Apply regression model technique to predict the data on above dataset.

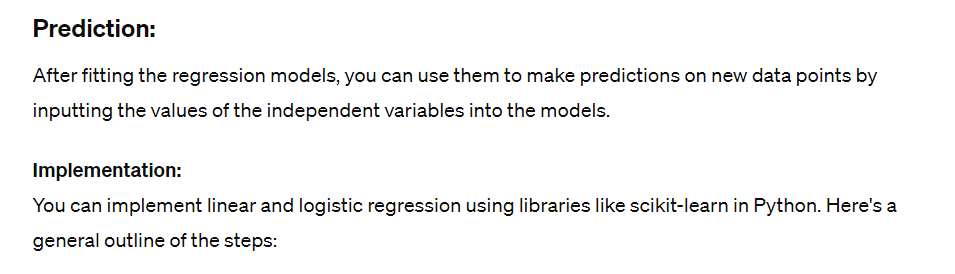
**Theory:-**

****

****







**Input:-**

X\_linear = df[['Latitude', 'Longitude', 'Precip', 'Pressure', 'Humidity\_2m', 'Temp\_2m']] # Features

y\_linear = df['MaxTemp\_2m'] # Target variable

X\_train\_linear, X\_test\_linear, y\_train\_linear, y\_test\_linear = train\_test\_split(X\_linear, y\_linear, test\_size=0.2, random\_state=42)

linear\_reg\_model = LinearRegression()

linear\_reg\_model.fit(X\_train\_linear, y\_train\_linear)

y\_pred\_linear = linear\_reg\_model.predict(X\_test\_linear)

mse\_linear = mean\_squared\_error(y\_test\_linear, y\_pred\_linear)

print("Linear Regression Mean Squared Error:", mse\_linear)

X\_logistic = df[['Latitude', 'Longitude', 'Precip', 'Pressure', 'Humidity\_2m', 'Temp\_2m']] # Features

y\_logistic = df['District'] # Target variable (assuming 'District' is categorical)

label\_encoder = LabelEncoder()

df['District\_binary'] = label\_encoder.fit\_transform(df['District'])

X\_train\_logistic, X\_test\_logistic, y\_train\_logistic, y\_test\_logistic = train\_test\_split(X\_logistic, df['District\_binary'], test\_size=0.2, random\_state=42)

logistic\_reg\_model = LogisticRegression()

logistic\_reg\_model.fit(X\_train\_logistic, y\_train\_logistic)

y\_pred\_logistic = logistic\_reg\_model.predict(X\_test\_logistic)

accuracy\_logistic = accuracy\_score(y\_test\_logistic, y\_pred\_logistic)

print("Logistic Regression Accuracy:", accuracy\_logistic)

**Output:-**

****

**Input:-**

attribute1 = 'Humidity\_2m'

attribute2 = 'MaxTemp\_2m'

plt.figure(figsize=(8, 6))

sns.scatterplot(x=attribute1, y=attribute2, data=df)

plt.title(f"Relationship between {attribute1} and {attribute2}")

plt.xlabel(attribute1)

plt.ylabel(attribute2)

plt.grid(True)

plt.show()

X = df[[attribute1]]

y = df[attribute2]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

plt.figure(figsize=(8, 6))

sns.scatterplot(x=attribute1, y=attribute2, data=df)

plt.plot(X\_test, y\_pred, color='red', linewidth=2)

plt.title(f"Linear Regression: {attribute1} vs {attribute2}")

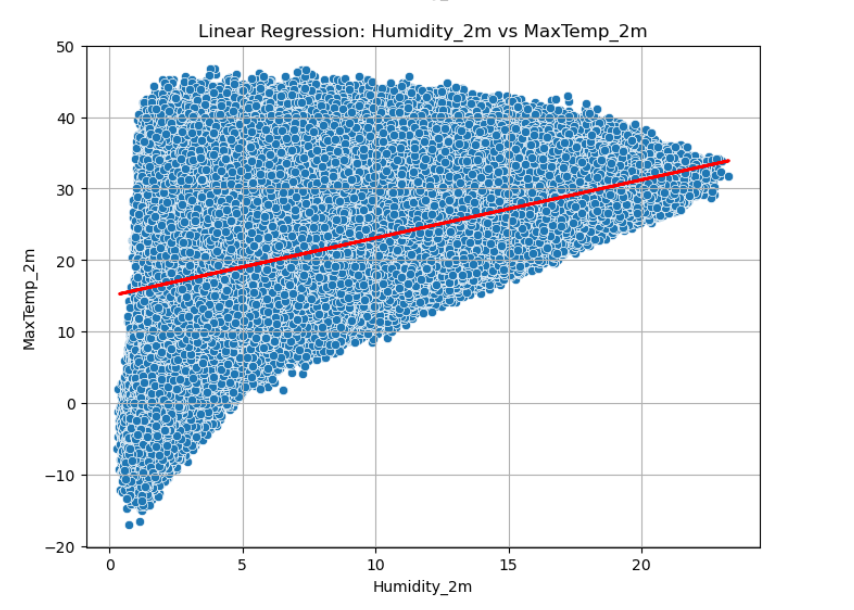
plt.xlabel(attribute1)

plt.ylabel(attribute2)

plt.grid(True)

plt.show()

**Output:-**

****

**Conclusion:-**

* Linear regression was used to model the relationship between numerical independent variables and a continuous dependent variable (e.g., predicting 'MaxTemp\_2m' based on weather attributes).
* We found that the linear regression model provided a mean squared error (MSE) of 0.6921908717594275, indicating **average** squared difference between the actual binary outcomes and the predicted probabilities.
* Logistic regression was employed to predict a binary outcome (e.g., predicting a categorical variable like 'District' based on weather attributes).