STEPS TO BE FOLLOWED:

1. Import required Libraries
   HINT: import keyword and from keyword some cases.

2. Basic EDA to understand dataset.
   2.1. Select more than five columns for EDA.

   Step1: Each plot must Use the subplots
   Plot 1: Histogram or bar, line and use suitable plots

   HINT: import matplotlib.pyplot and import seaborn aswell. And use above mentioned plots.

   Plot2: Box plot.

   HINT: import seaborn library and use boxplot

   Step2: Write the inference is it data skewed or not.

   HINT: Get inference from the distribution plot for numerical variables(Histograms).

   Step3: Write the inference is it data having outliers or not.

   HINT: Get inference from distribution(box) plots for numerical variables.

3. Plot the heat map.

HINT: use heatmap method from seaborn library.

Step1: Write the inference on the dependent variable and independent variable having the most (Positive and Negative) correlation columns.
HINT : Look at the correlation values between dependent variable and independents variables.

Give inference level wise

Ex: 1.Positive – Low, Moderate, and Strongly.
2. Negative – Low, Moderate, and Strongly.

Step2: Write the inference on the correlation among independent variables

HINT : Look at the correlation values between independent variables.(Multi correlation)

Give inference level wise

Ex: 1.Positive – Low, Moderate, and Strongly.
2. Negative – Low, Moderate, and Strongly.

4. Split dataset into train and test,Scaling:

1. Split the dataset in dependent variables and independent.

HINT: One method is that

* Drop the target variable and save into new dataframe let's X.

* use slicing to get y variable and store it in y dataframe let's y.

2. Convert categorical variables to numeric variables

HINT: Use encoding methods like get_dummies, Label Encoding, Ordinal Encoding, Frequency Encoding, Target Encoding, Hash and Binary Encoding etc.

3. Scale the features
   HINT: Import scaler classes from sklearn library
   Ex: Apply the scaler method on the variables.
4. Split dataset into train(70%) and test(30%).How would you ascertain this statistically.

   HINT: use train and test split method from the sklearn library, Set the train_size as 70% or test_size = 30%.

5. Model Building and Feature Selection using RFE(Recursive feature elimination.

   1. Use Linear Regression:

      HINT: 1. Import Linear Regression
            2. .fit() the model by passing training data.

   2. Evaluation metrics like Rsqaure,Standard error/RMSE. Write the inference.

      HINT:
      1. RSquare -> import Rscore from sklearn metrics and Get the Rsquare.

2. Standard Error/ RMSE – Use mean_square_error method to get the average squared error and take root of that.

3. Feature selection using RFE:

HINT:

Step 1: Import RFE.

Step 2: Declare the lists for storing the results like Rscore = [], RMSE = [] for train and test.

Step 3: Use for loop for finding the optimal number of features. EX: for n_features in range(1,7):

Step 4: Create an object for Linear Regression to get the metrics.

Step 5: Create an object for RFE by setting following required parameter. Like a. Estimator = ;

b. n_features_to_select = ;

etc…

Step 6: .fit() the model with train data.

Step 7: get the rank == 1 features.

Step 8: Use the RFE features for Linear Regression model. And fit Linear model using RFE features.

Step 9. Get the predicted values and get metrics like RMSE and Rsquare scores for Training and Testing. And append them to respective lists which created at step 2.

4. Build a model with optimal Features.

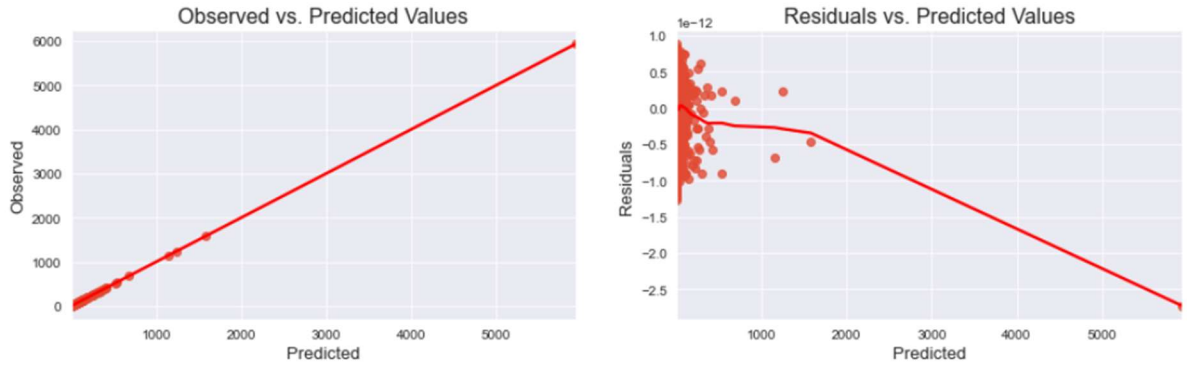
HINT: select the RFE model futures which has High Rsquare score and Less RMSE.

Ex: Build model with 5 Features using Linear Regression from Stats Library or sklearn Library.

5. Linear Regression Assumptions Validation.

1. Linearity of the model:

HINT: Plot the regplot for actual and predicted, And predicted and residual using subplots.



2.Homoscedasticity (equal variance) of residuals.

HINT: One way is that check the statistical tests. Or get inference from stats model linear regression.

Another way:

Plot the related plot.

→Get the residual and fitted values and residual Standardized.

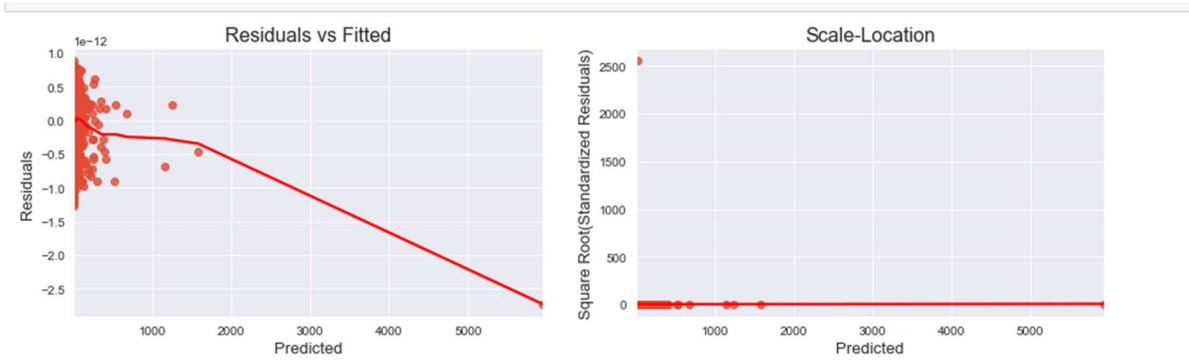
Ex:

fitted_vals = model.predict()

residuals = model.resid

resids_standardized = model.get_influence().resid_studentized_internal

➔ Take the absolute and square root for resids_standardized.
Ex: np.sqrt(np.abs(resids_standardized))
Plot regplot for fitted values and residuals, And fitted values and resids_standardized values using subplot method.
Ex: x=fitted_values, y = np.sqrt(np.abs(resids_standardized)) in the regplot.

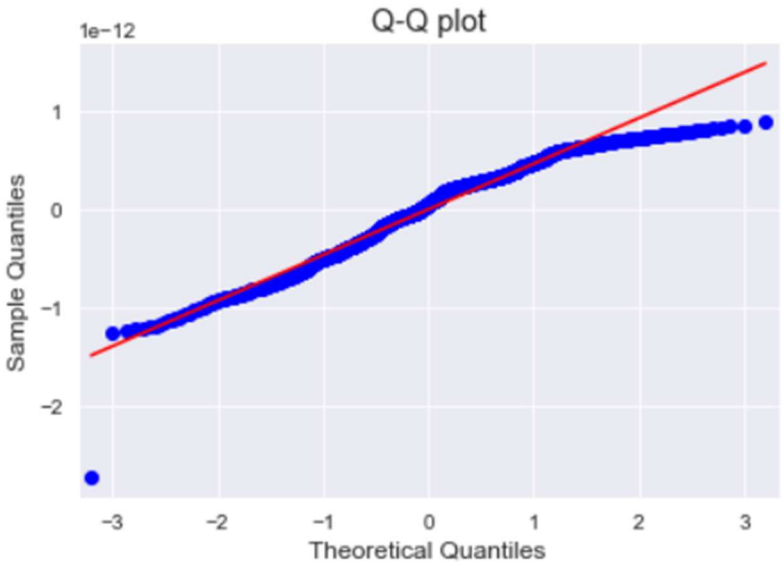


3. Normality of residuals.

HINT: Use statistical tests.

Or

Use related plots like QQ plot.



6.Rebuilding the Model: Feature Selection using RFE & K-Fold Cross Validation.

HINT:

Rebuild the model using RFE features.

And

Use K-Fold Cross validation for training data.

Note set the parameter accurately in the cross validation

EX:

Kfold = KFold(n_splits = 5) and cross_validation(score = kfold, etc)..

Last:

Write your inference..

Thank you.