Investigate Factors Influencing the Housing Price using Multiple Regression

Vipin Sharma
Dept. of Computing
National College of Ireland
Dublin, Ireland
x22207406@student.ncirl.ie

I. Introduction

The task involves investigating the various factors influencing the sale price of the house through the use of a multiple linear regression model on the given dataset `housing.csv`. This `housing.csv` dataset includes a range of variables that help us to influence the housing sale price.

The main aim of this analysis is how different variables interact with the sale price and how they affect the sale price of the house. Using the multiple linear regression analysis, offering valuable insights for market stakeholders and policy-makers.

Multiple Linear Regression: It is a regression-based model that examines the relationship between the dependent variable (Y) and two or more independent variables ($X_1, X_2, X_3, X_4, \ldots, X_n$). The aim of the multiple linear regression model is to describe how the DV is related to the independent variables.

The variable obtained in this housing data is `Sales_Price`, which is the dependent variable (Y), and (`Lot Frontage`, `Lot Area`, `Building Type`, `House Style`, `Overall Condition`, `Year Built`, `Exterior Condition`, `Total Basement Surface Area`, `First Floor Surface Area`, `Second Floor Surface Area`, `Full Bathroom`, `Half Bathroom`, `Bedroom Above Ground Floor`, `Kitchen Above Ground Floor`, `Fireplaces`, `Longitude`, and `Latitude`) are the independent variables.

The Multi-Linear Regression Line equation is expressed by:

$$Y = B_0 + B_1X_1 + B_2X_2 + B_3X_3 + \ldots + B_nX_n + C$$

II. Dataset Description and Variable Overview

The dataset used in this project consists of 1689 rows and 18 columns/features. In the housing dataset, Have a single dependent variable, `Sales_Price`, and 17 other independent variables.

The Variables Explained in the following manner :

- `Lot Frontage`: The distance in feet that the road has to the property.
- `Lot Area`: Plot dimensions expressed in square feet.
- `Building Type`: The kind of property.
- `House Style`: The style of the property.
- `Overall Condition`: Overall, the house's condition.
- `Year Built`: The year in which the house was constructed.
- `Exterior Condition`: The material condition on the exterior of the house.
- `Total Basement Area`: Area of the entire basement in square feet.
- `First Floor Surface`: Area of the ground floor in square feet.
- `Second Floor Surface`: Area of the first floor in square feet.
- `Full Bath`: Number of full bathrooms.
- `Half Bath`: Number of half bathrooms.
- `Bedroom Above Ground Floor`: Number of bedrooms on the ground floor or above.
- `Kitchen Above Ground Floor`: Number of kitchens on the ground floor or above.
- `Fireplaces`: Number of fireplaces.
- `Longitude`: Longitude of the plot.
- `Latitude`: Latitude of the plot.

III. Exploratory Data Analysis

The first step in any analysis is to understand the data before diving into the multiple-linear regression first understand the housing data. This section provides information about the housing data, like the continuous and categorical columns, the five-number summary (minimum value, maximum value, median, quantiles, standard deviation), types of data types, null values, outliers, and missing values.

This author used the Pandas library to find the shape of the data and then describe a five-number summary (fig-1), which includes:

- **Count**: This gives the total number of data in the column.
- **Mean**: Average of the sample.
- **Standard Deviation**: Dispersion of the data around the mean value.
- **Minimum and Maximum**: Minimum and maximum values of the samples.
- **Quantiles**: Various quantiles of the sample.

	count	mean	std	min	25%	50%	75%	max
Lot_Frontage	1689.0	55.654825	33.780198	0.000000	39.000000	60.000000	77.000000	313.000000
Lot_Area	1689.0	10247.835998	9264.293635	1477.000000	7480.000000	9405.000000	11435.000000	215245.000000
Year_Built	1689.0	1969.081113	29.851607	1875.000000	1952.000000	1971.000000	1998.000000	2010.000000
Total_Bsmt_SF	1689.0	1031.267022	402.598411	0.000000	794.000000	973.000000	1243.000000	3206.000000
First_Flr_SF	1689.0	1133.587922	358.878294	334.000000	868.000000	1062.000000	1360.000000	2696.000000
Second_Flr_SF	1689.0	343.272351	425.485117	0.000000	0.000000	0.000000	702.000000	1872.000000
Full_Bath	1689.0	1.542925	0.543781	0.000000	1.000000	2.000000	2.000000	4.000000
Half_Bath	1689.0	0.375962	0.497782	0.000000	0.000000	0.000000	1.000000	2.000000
Bedroom_AbvGr	1689.0	2.859680	0.810090	0.000000	2.000000	3.000000	3.000000	6.000000
Kitchen_AbvGr	1689.0	1.041445	0.205232	0.000000	1.000000	1.000000	1.000000	3.000000
Fireplaces	1689.0	0.604500	0.652105	0.000000	0.000000	1.000000	1.000000	4.000000
Longitude	1689.0	-93.642418	0.026599	-93.693153	-93.662162	-93.641053	-93.620491	-93.577427
Latitude	1689.0	42.033300	0.017983	41.986498	42.021386	42.034414	42.046962	42.063381
Sale_Price	1689.0	175799.336886	71560.766518	35000.000000	130000.000000	158000.000000	207000.000000	755000.000000

Fig. 1. Five Number Summary

After this, using Pandas `.isnull().sum()`, functions that give information about the housing data that have how many null values, In Fig 2 saw that there are no missing values in the data.

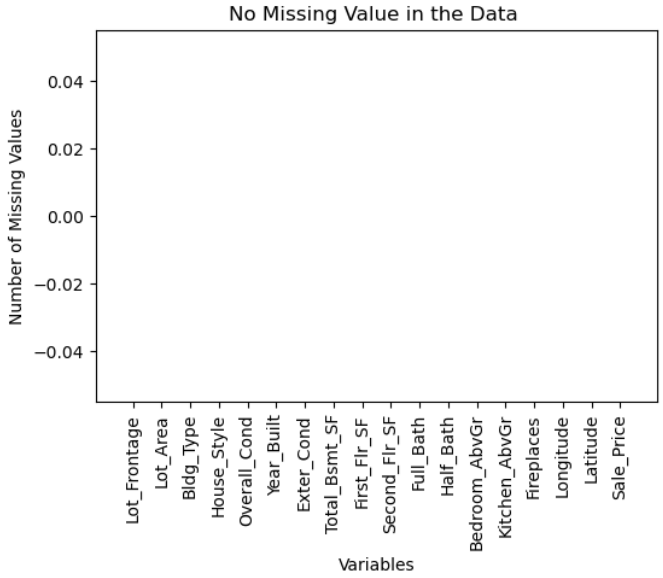

Fig. 2. Missing value

The `.info()` function gives data types and total non-null values in the housing data. The author also found the total unique values in the housing data using the `.nunique()` function, it will give all the nunique values for each column as shown in Fig 3. In the figure x-axis has all the variables and the y-axis has the number of unique values. The longitude and the Latitude have a large number of unique values and after this, the Lot area has the large unique values and so on.

The Exploratory Data Analysis (EDA) also describes the level of measurement (ordinal_data, nominal_data, continuous, discrete) categorical data, and continuous data after doing this author check the outliers and correlation between the variables, The outliers are the value in any dataset which is very high and low values as compared to other values, In Fig 4 author using the box plot chart it shows all the values which is different from the other's values and make the box plot for continous

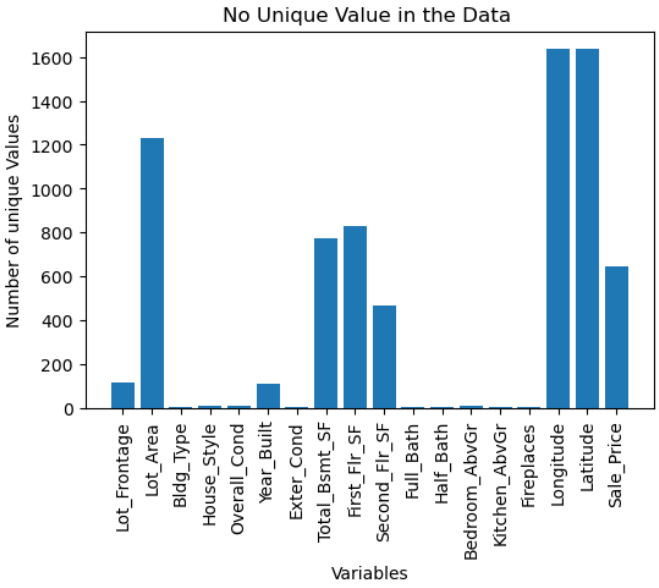

Fig. 3. Number of unique Values

data variables and Continous variables.

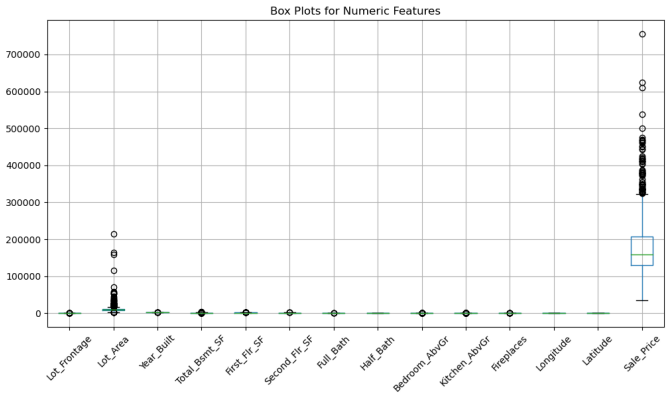

Fig. 4. Outliers

The correlation is defined as how the two variables are dependent on each other, if the value of one variable is changes then how much the other variable values affected or changed? For the correlation use the `.corr()` function and heatmap for the visualization as shown in Fig 5. After doing all of these steps author went ahead with the data preparation.

IV. Data Preparation

Data preparation is an essential part of machine learning because here the author prepares our data for modeling. The first step the author does in the data preparation is to Label Encoder, the label encoder is a method in machine learning that converts the categorical data into a which is the machine algorithm can understand and do the preprocessing it is a very important step when dealing with the categorical data, so for Label encoder, author use the `sklearn` library and

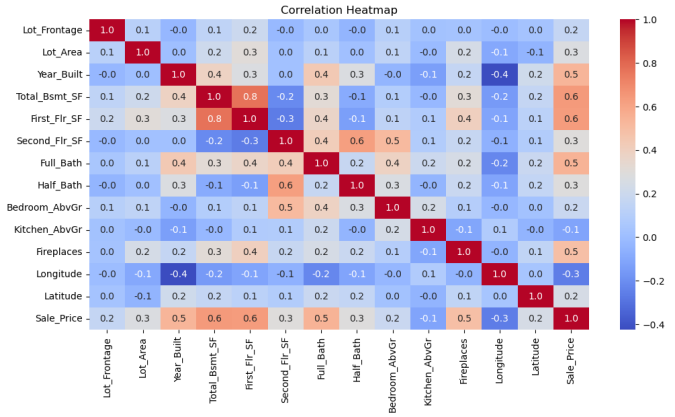

Fig. 5. Correation

import the `LabelEncoder` package and make the object of `LabelEncoder` class, denoted as le.

Secondly, author provide features. A technique called scaling is applied to normalize and standardize the independent variables, or features, in the dataset. The main use of feature scaling is to ensure that all the independent variables have the same scale so that our algorithm performs well. In the feature scaling, there are two common methods by which the user performs this operation Min-Max Scaling and Standard Scaler for our `housing_data` set this author use the Standard scaler method for this author has to import the `StandardScaler` package from the `sklearn` library and create the object of this package `sc` and perform on the `x_train` and `x_test`.

When feature scaling is done, start the treatment of outliers on `x_train` and `y_train` data for this author first concatenate `x_train` and `y_train` into the new variables `x_y_train` then by using the `z_scores` author treat the outliers based on a threshold value 3 it will remove the rows where the `z_scores` is greater than the threshold value. After treating the outliers find the column name in `x_train` which has a high correlation by using the `.corr()` function for this author has to create a `high_corr_column` blank list and by using the for loop author stores the variables in the blank column which has the threshold value greater than 0.5 doing this gets the column which has the high correlation.

After finding the high_corr_columns find the f-score, p-value, and the VIF (variance_inflation_factor) for the f-score and p-value author import the (f_regression) and for VIF. after this import the (variance_inflation_factor) package from `sklearn` library then author drop the features who have the high VIF. The high VIF means the vif has a value greater than 5 and this is the last step in the data preparation after this go to the modeling process and create our model.

V. Modelling

The modeling steps are performed after doing the Exploratory Data Analysis and Data Preparation. When the author starts doing the modeling author first to import the LinearRegression model from `sklearn.linear_model` library and then create the object of the LinearRegression model name as regressor after this make our model on `x_train` and `y_train` by using the `.fit()` function . Pass both parameters `x_train` and `y_train` in the `.fit()` function then our model is created.

Once our model is created then use the stats models library to perform the ordinary least squares(OLS) regression which gives the summary of our model like R-squared, Adj. R-squared, p-value, t-statistic, and coef. The t-statistic and the p-values help to get the statistical importance of each coefficient and coefficients give the direction and relationship between the dependent variables and Independent variables.

Our (`model0`) gives the R-Squared 0.772 and the Adj. R-Squared 0.770 but when print the summary of the (`model0`) by using the `model0.summary()` function author saw the column had p values greater than the significance value 0.05 so dropped those features and made another (`model1`) on `y_train` and `x_train`. The model1 also gives the same R-Squared 0.772 and the Adj. R-Squared 0.770 but now all the features are significant and the p-value is less than 0.05.

OLS Regression Results

Dep. Variable:	Sale_Price	**R-squared:**	0.772
Model:	OLS	**Adj. R-squared:**	0.770
Method:	Least Squares	**F-statistic:**	347.7

Fig. 6. Model0 OLS Regression Result

OLS Regression Results

Dep. Variable:	Sale_Price	**R-squared:**	0.772
Model:	OLS	**Adj. R-squared:**	0.770
Method:	Least Squares	**F-statistic:**	401.6

Fig. 7. Model1 OLS Regression Result

The next step is to predict so the author do the prediction on the `x_test` data and then compare the results prediction of `x_test` with the `y_test` values for this author created the `y_pred` variable and used the .predict() function to predict all the predictions and store in the `y_pred` variable.

VI. Interpretation

In this section, author created the model for our data and looked at what interpretations are coming for our dataset. The interpretation is important for checking the models because created the two models `model0` and `model1` both give a different F-statistic but there is no difference between the R Squared and Residual R Squared. The Durbin-Watson value of both the models is a little bit different,`model0` gives the value 2.033 and the `model1` value is 2.031.

In `model0` this author take all the features means as shown in Fig 8 we take all features that have a p-value greater than 0.05 but in `model1` drop the features that have a high p-value

Dep. Variable:	Sale_Price	R-squared:	0.772
Model:	OLS	Adj. R-squared:	0.770
Method:	Least Squares	F-statistic:	347.7
Date:	Sun, 19 Nov 2023	Prob (F-statistic):	0.00
Time:	17:00:23	Log-Likelihood:	-18135.
No. Observations:	1554	AIC:	3.630e+04
Df Residuals:	1538	BIC:	3.639e+04
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Lot_Frontage	2716.5877	796.760	3.410	0.001	1153.737	4279.438
Lot_Area	9609.2992	1592.602	6.034	0.000	6485.397	1.27e+04
Bldg_Type	-616.2467	932.186	-0.661	0.509	-2444.737	1212.244
House_Style	1.502e+04	1192.813	12.589	0.000	1.27e+04	1.74e+04
Overall_Cond	6566.9386	791.829	8.293	0.000	5013.760	8120.117
Year_Built	1.477e+04	1135.354	13.006	0.000	1.25e+04	1.7e+04
Exter_Cond	-54.0450	776.130	-0.070	0.944	-1576.429	1468.339
Total_Bsmt_SF	1.856e+04	1483.613	12.511	0.000	1.57e+04	2.15e+04
First_Flr_SF	1.982e+04	1613.532	12.282	0.000	1.67e+04	2.3e+04
Full_Bath	1.148e+04	1082.509	10.605	0.000	9356.205	1.36e+04
Half_Bath	7470.1324	1029.836	7.254	0.000	5450.101	9490.164
Bedroom_AbvGr	-3062.9996	1007.336	-3.041	0.002	-5038.896	-1087.103
Kitchen_AbvGr	-9.14e+05	3798.090	-240.652	0.000	-9.21e+05	-9.07e+05
Fireplaces	7614.1176	908.771	8.378	0.000	5831.556	9396.679
Longitude	-1824.7227	825.643	-2.210	0.027	-3444.228	-205.217
Latitude	1251.4519	778.020	1.609	0.108	-274.640	2777.544

Omnibus:	181.040	Durbin-Watson:	2.033
Prob(Omnibus):	0.000	Jarque-Bera (JB):	371.368
Skew:	0.714	Prob(JB):	2.28e-81
Kurtosis:	4.923	Cond. No.	8.74

Fig. 8. Model0 Summary with P-value and F-Statistic

Dep. Variable:	Sale_Price	R-squared:	0.772
Model:	OLS	Adj. R-squared:	0.770
Method:	Least Squares	F-statistic:	401.6
Date:	Sun, 19 Nov 2023	Prob (F-statistic):	0.00
Time:	17:00:23	Log-Likelihood:	-18136.
No. Observations:	1554	AIC:	3.630e+04
Df Residuals:	1540	BIC:	3.637e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Lot_Frontage	2792.4716	787.823	3.545	0.000	1247.152	4337.791
Lot_Area	9982.0911	1488.967	6.704	0.000	7061.474	1.29e+04
House_Style	1.501e+04	1191.461	12.598	0.000	1.27e+04	1.73e+04
Overall_Cond	6598.1683	780.238	8.457	0.000	5067.727	8128.610
Year_Built	1.468e+04	1110.054	13.226	0.000	1.25e+04	1.69e+04
Total_Bsmt_SF	1.853e+04	1481.956	12.504	0.000	1.56e+04	2.14e+04
First_Flr_SF	1.979e+04	1611.536	12.280	0.000	1.66e+04	2.3e+04
Full_Bath	1.143e+04	1075.306	10.632	0.000	9323.254	1.35e+04
Half_Bath	7458.3071	1028.114	7.254	0.000	5441.655	9474.959
Bedroom_AbvGr	-2914.1711	979.527	-2.975	0.003	-4835.518	-992.824
Kitchen_AbvGr	-9.141e+05	3792.683	-241.024	0.000	-9.22e+05	-9.07e+05
Fireplaces	7612.7793	908.149	8.383	0.000	5831.439	9394.119
Longitude	-1859.9692	823.457	-2.259	0.024	-3475.185	-244.754
Latitude	1242.8535	776.242	1.601	0.110	-279.749	2765.456

Omnibus:	178.178	Durbin-Watson:	2.031
Prob(Omnibus):	0.000	Jarque-Bera (JB):	363.651
Skew:	0.706	Prob(JB):	1.08e-79
Kurtosis:	4.903	Cond. No.	8.62

Fig. 9. Model1 Summary with P-value and F-Statistic

and take only a significant value as shown in Fig 9 that's why `model0` has an high F-statistic is 347.7 and the `model1` have F-statistic is 401.6 which is used to test the significance level of our model. A greater F-Statistic value shows that the model is very statistically significant.

The F-statistics of `Model-0` and `Model-1` are different, with the `Model-1` having a larger value. Hence this author can say that Model-1 is statistically even more important than `Model-0`, supporting the model's overall importance and quality.

VII. Diagnostics

Applying the Gauss-Markov assumptions on `model1` and checking whether the `model1` follows all the assumptions or not. It is important to check how a good model satisfies these assumptions. There are many assumptions this author take the most important assumption like

1) **Linearity:** In this, author plot residuals using the predicted values (`y_pred`) and actual values (`y_test`) and check the linearity relationship for doing this use the `.axhline` from the `matplotlib` library and also display line with the red color.
2) **Homoscedasticity:** First, this author find the error or residuals between the `y_test` and `y_pred` then by using the scatter plot this author plot the residual that will fit all the residuals.
3) **Normality of Errors:** Plot the residuals and check whether they follow the normal distribution for doing this author can use the `seaborn displot`. Figure 9 shows whether the residuals follow the normal distribution or not. To confirm this parameter this author have to find the Shapiro-Wilk value this value tells us whether our residual follows the normal distribution or not. If the p-value is greater than the alpha then it follows the normal distribution if less than the alpha value it will not follow the normal distribution and our Shapiro-Wilk

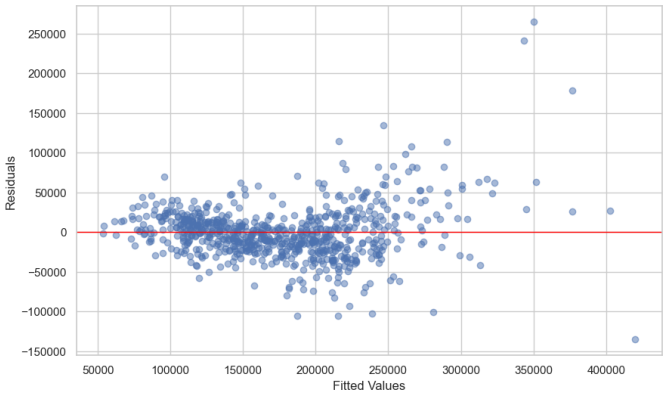

Fig. 10. Linearity Residual vs Fitted

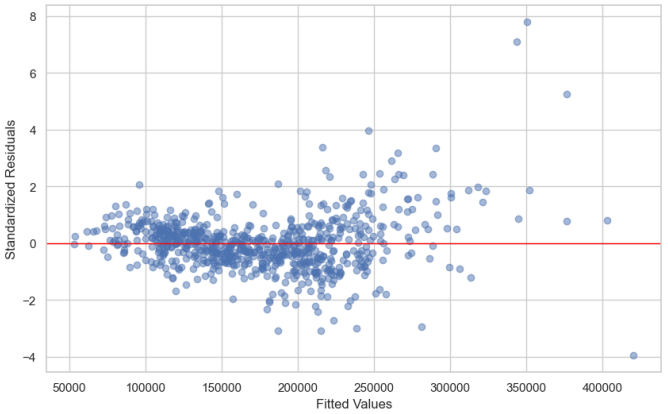

Fig. 11. Homoscedasticity

value is 0.946, which means the residuals do not follow the normal distribution.

4) **Multicollinearity:** For checking the Multicollinearity this author use the Durbin-Watson statistic value and put the condition if the value is less than 1.5 then it is positive autocorrelation is present and if the Durbin-Watson statistic is greater than 2.5 then it is negative correlation if the value between them it means no autocorrelation is detected and Our `model1` Durbin-Watson statistic is 1.98, it means there is no significant autocorrelation detected and this author also check the Variance Inflation Factor (VIF) of the features as a general rule if the VIF number is more then 5 then the features consider as multicollinear. In our model1 take all the features that have the VIF values less than 5 and it varies between 1 to 4.

VIII. Evaluation

Model evaluation is a crucial stage in the development of any machine learning model. The most frequently computed matrices are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Residual Standard Error (RSE) [1], which are all utilized to assess the model's performance in multi regression analysis.

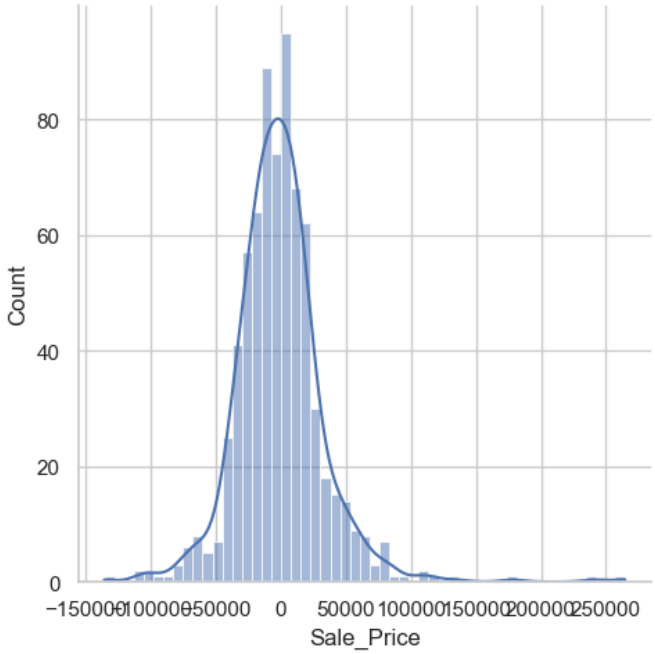

Fig. 12. Residuals Distributed

1. Residual Standard Error (RSE)

The RSE is used to measure the spread of the residuals in a model. The formula for calculated RSE is:

$$RSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}}$$

Where:

n number of observations,
y_i actual value,
$\hat{y}_i$ predicted value.

Our model gives the RSE as 34834.49, and this value represents the standard deviation of the errors or residuals. A lower RSE means the model is good.

2. Mean Absolute Error (MAE)

The MAE is used to assess the performance of the model, and it is calculated as the average of the absolute difference between the y_{test} and y_{pred}. The mean absolute error calculated is 25090.64. The formula for MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where:

n number of observations,
y_i actual value,
$\hat{y}_i$ predicted value.

3. Root Mean Squared Error (RMSE)

The RMSE calculated for our model is 34786.34, which means that the squared difference between the actual and predicted values is around 34786.34. The lower RMSE indicates the good performance of the model. RMSE is a commonly used metric for finding the performance of the multi-linear regression model. The formula for RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Where:

n number of observations,

y_i actual value,

$\hat{y}_i$ predicted value.

IX. Final Summary

According to `model1`, the variables represent 77% of the variation in the model, with a R^2 of 0.772 and an adjusted R^2 of 0.770. For each variable in the model, the P-value (sig) is less than 0.05, and the Durbin-Watson is 1.98, which is nearly equal to 2. Every variable use to create `model1` has a VIF that is less than five. The matrices have a Residual Standard Error (RSE) of 34834.49, Mean Absolute Error (MAE) of 25090.64, and Root Mean Squared Error (RMSE) of 34786.34.

References

[1] Analytics Vidhya. "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R-squared: Which Metric is Better?" Medium, [Dec 8, 2020]. [Online]. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e