Regression Analysis

Data:[Housing_Price.csv]

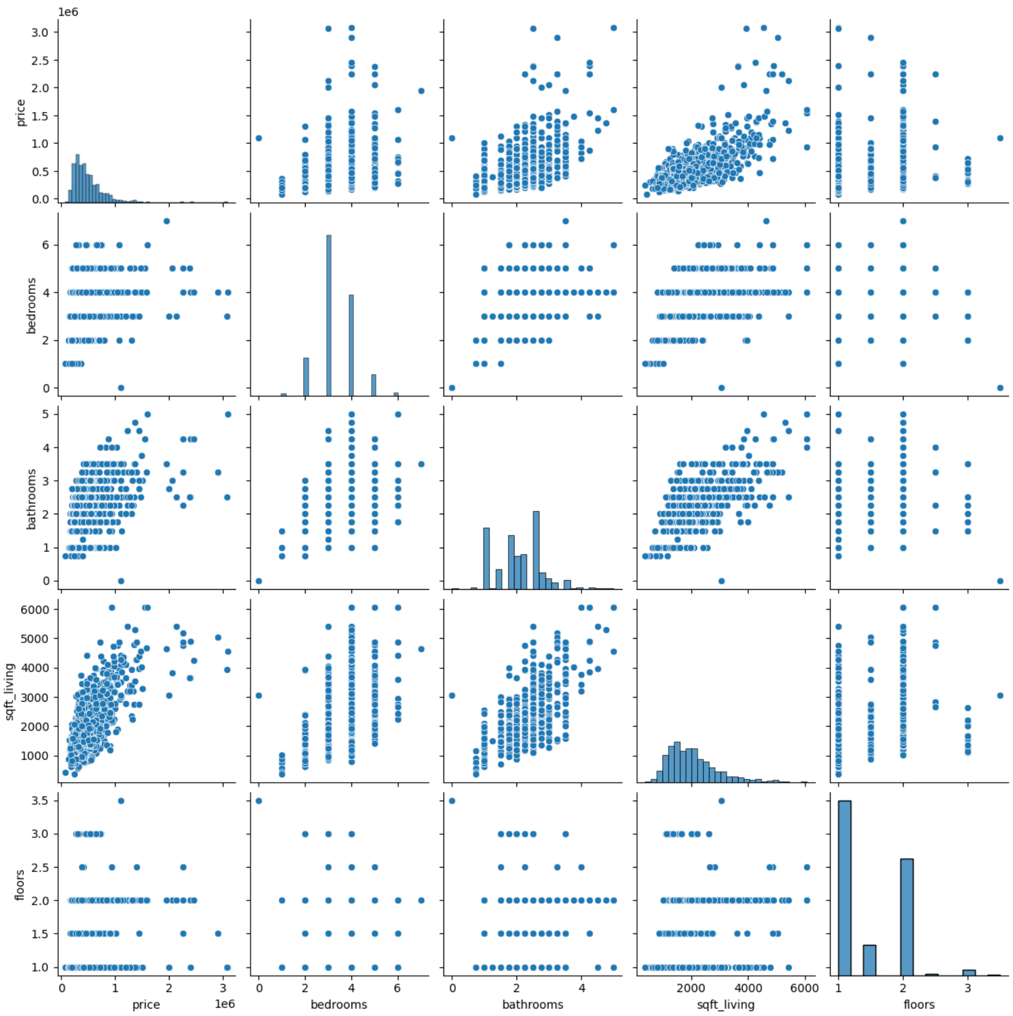
Before we tackle regression analysis, we need to understand correlation.

Correlation:

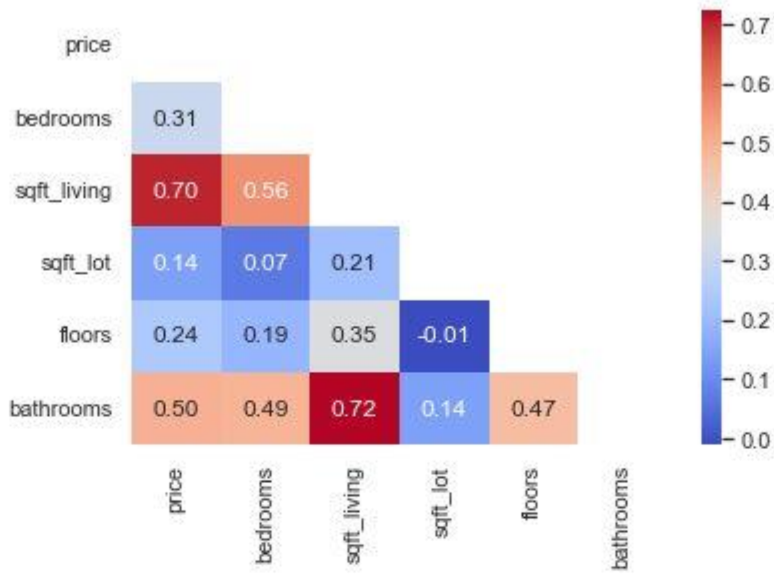
A correlation between variables indicates that as one variable changes in value, the other variable tends to change in a specific direction.

Graph Your Data to Find Correlations

Scatterplots are a great way to check quickly for relationships between pairs of continuous data



Pearson's Correlation Coefficient



Interpret the Pearson's Correlation Coefficient
Pearson's correlation coefficient is a single number that measures both the strength and direction of the linear relationship between two continuous variables. Values can range from -1 to +1.

- Pearson correlation coefficient ranges from -1 to 1, where -1 indicates a perfect negative linear relationship
- 0.2 or -0.2 would indicate a weak positive or negative linear relationship
- -0.8 or 0.8 would indicate a strong negative or positive linear relationship

Pearson's Correlation Measures Linear Relationships:
Pearson's correlation measures only linear relationships. Consequently, if your data contain a curvilinear relationship, the correlation coefficient will not detect

Regression Takes Correlation to the Next Level:
Regression analysis allows us to expand on correlation in other ways.
If the relationship is curved, we can still fit a regression model to the data.

Regression analysis can handle many things:
- Model multiple independent variables
- Include continuous and categorical variables
- Model linear and curvilinear relationships
- Assess interaction terms to determine whether the effect of one independent variable depends on the value of another variable
-

Goals of Regression Analysis:

- To understand the relationships between these variables. How do changes in the independent variables relate to changes in the dependent variable?
- To predict the dependent variable by entering values for the independent variables into the regression equation

Simple versus Multiple Regression:
When you include one independent variable in the model, you are performing simple regression. For more than one independent variable, it is multiple regression. Despite the different names, it’s really the same analysis with the same interpretations and assumption

Simple Linear Regression

Simple Linear Regression Model:

ŷ = b0 + b1x

where:
ŷ: The estimated response value
b0: The intercept of the regression line
b1: The slope of the regression line

Data_file: **[Housing_Price.csv]()**
Code_file: **[Housing_Price]()**
Independent Variable: sqft_living(square feet of living space)
dependent variable: Price (price of House)

Model summary:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.497
Model:	OLS	Adj. R-squared:	0.496
Method:	Least Squares	F-statistic:	1037.
Date:	Tue, 09 May 2023	Prob (F-statistic):	7.78e-159
Time:	09:38:18	Log-Likelihood:	-14543.
No. Observations:	1053	AIC:	2.909e+04
Df Residuals:	1051	BIC:	2.910e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-3.463e+04	1.88e+04	-1.845	0.065	-7.15e+04	2194.144
sqft_living	271.7192	8.438	32.200	0.000	255.161	288.277

Omnibus:	622.151	Durbin-Watson:	2.023
Prob(Omnibus):	0.000	Jarque-Bera (JB):	9616.999
Skew:	2.406	Prob(JB):	0.00
Kurtosis:	17.001	Cond. No.	5.62e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.62e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

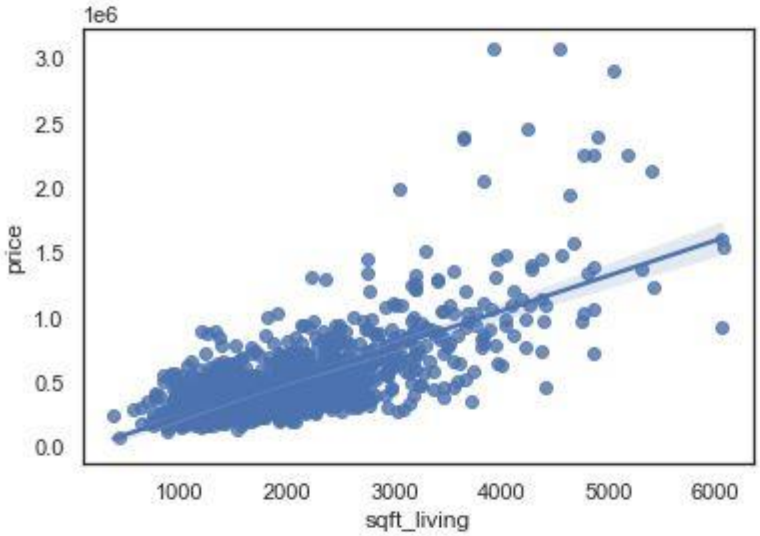
Interpretation:

Price= -34634.8057067+ 271.7192*(sqft_living)

We interpret this to mean that each additional sqft_living is associated with an average increase of 271.7192 in house price

The intercept value tells us that the average actual housing price without considering any sqft_living is -34634.8057067(Negative coefficient for intercept represents that as the sqft_living increases, the house price also increases, but at a decreasing rate. This could be because of house of the price may contains some extra factors such as no. of bedrooms floor bathrooms etc.)

Graphical Representation of Regression Coefficients:



Confidence Intervals for Regression Parameters:

	0	1
const	-71463.755348	2194.143934
sqft_living	255.161224	288.277193

Interpreting P-Values for Continuous Independent Variables:

<u>Hypothesis:</u>

- Null hypothesis: The coefficient for the independent variable equals zero (no relationship).
- Alternative hypothesis: The coefficient for the independent variable does not equal zero

p-value=0.000<0.05

=>The coefficient for the independent variable does not equal zero

<u>coefficient for the independent variable (sqft_living): </u>271.7192

Interpreting P-Values for Regression Model:

	df	sum_sq	mean_sq	F	PR(>F)
df.price	1.0	4.052519e+08	4.052519e+08	1036.866324	7.781661e-159
Residual	1051.0	4.107760e+08	3.908430e+05	NaN	NaN

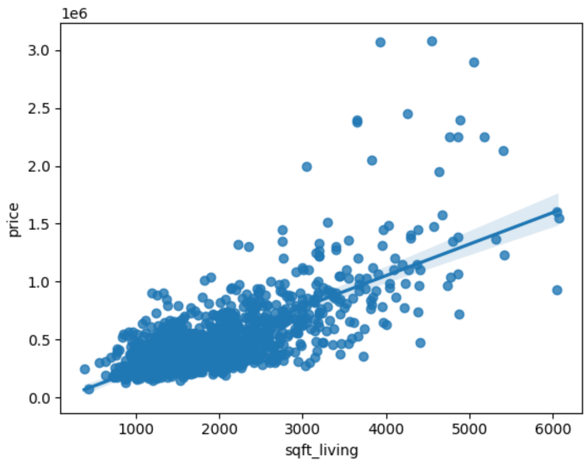
Hypothesis:

- Null hypothesis: The Regression model is not significant
- Alternative hypothesis: The Regression model is significant

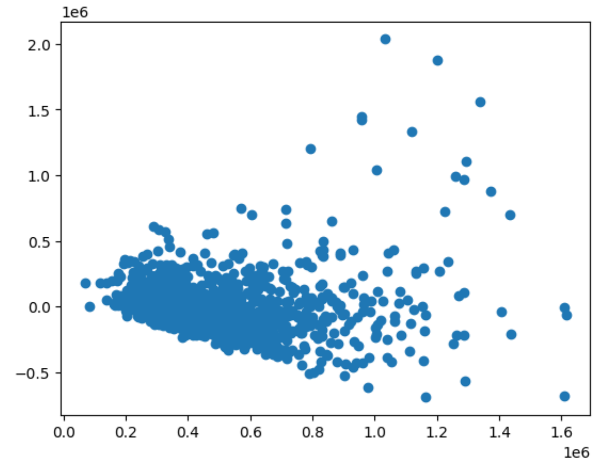
p-value=7.781661e-159<0.05
=>The Regression model is significant

Assumptions:

Linearity:

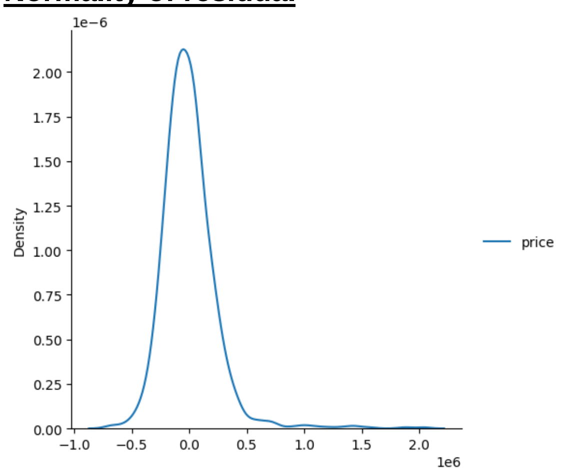


Homogeneity:



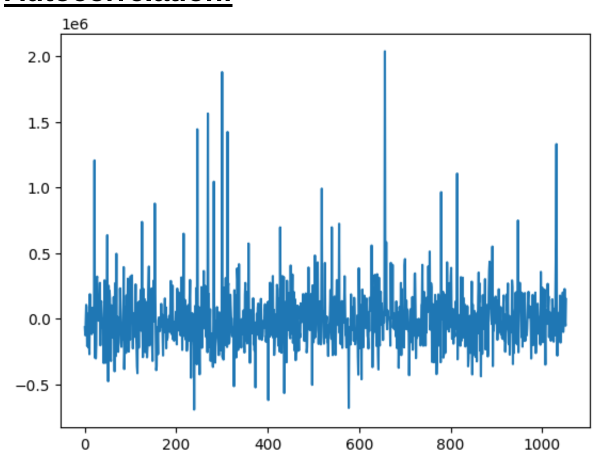
Levene Test:
p value=0.02117<0.05
=> Homogeneity is not present

Normality of residual



Shapiro-Wilk Test:
P-value: 8.53293606211052e-31
=>Residual is not normally distributed

Autocorrelation:



Durbin Watson Test
2.02255627
=>No autocorrelation

Multiple Linear Regression

When we want to understand the relationship between *multiple* predictor variables and a response variable then we can use **multiple linear regression**.
If we have *p* predictor variables, then a multiple linear regression model takes the form:

$$\mathbf{Y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta pXp + \varepsilon}$$

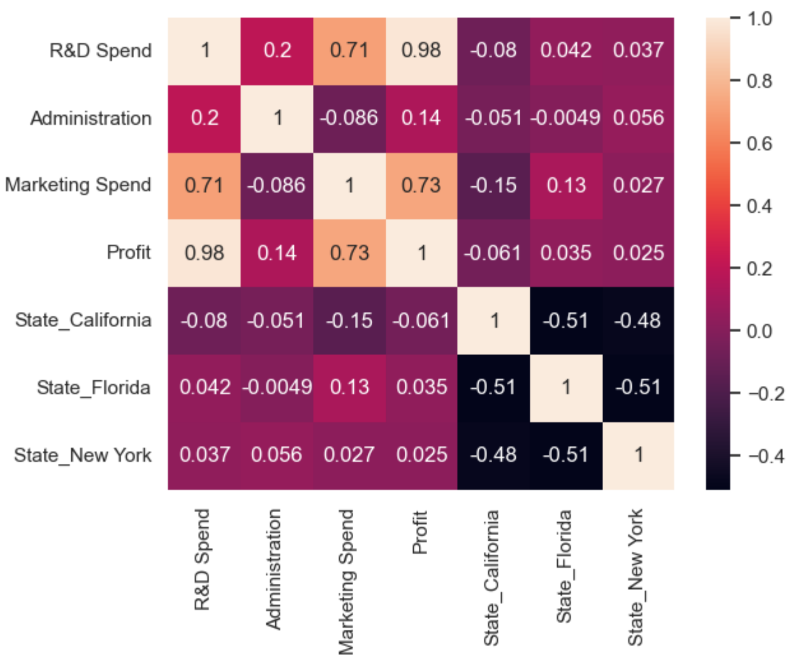
where:

- **Y**: The **response** variable
- **Xj**: The jth **predictor variable**
- **βj**: The average effect on Y of a one unit increase in Xj, holding all other predictors fixed
- **ε**: The error term

The values for β0, β1, B2, … , βp are chosen using **the least square method**,

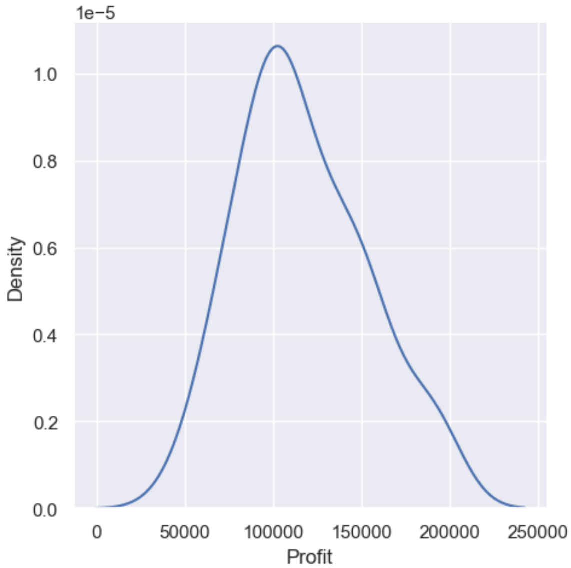
Data_file: [50_Start-up's]

Code_file: [MLR_50_Start-up's]
dependent variable: Profit
Independent variable: State,R&D, Spend,Marketing, Spend,Administration
Correlation_matrix



Highly +ve correlated variables: Profit & R&D_Spend (**0.98**)
So, we drop one of the variables between this

> the dependent variable is not normally distributed.



Applying **Box-cox** to check the which transformation is best for data

Lambda=0.3155

Either we can apply log transformation or square_root transformation.

Applying log_transformation, MLR_Model

Model:	OLS	Adj. R-squared:	0.478
Dependent Variable:	Profit	AIC:	-0.7526
Date:	2023-05-12 10:23	BIC:	8.3906
No. Observations:	46	Log-Likelihood:	5.3763
Df Model:	4	F-statistic:	11.29
Df Residuals:	41	Prob (F-statistic):	2.87e-06
R-squared:	0.524	Scale:	0.051997

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	8.1405	0.1350	60.2871	0.0000	7.8678	8.4132
Administration	0.0000	0.0000	1.9537	0.0576	-0.0000	0.0000
Marketing Spend	0.0000	0.0000	6.5601	0.0000	0.0000	0.0000
State_California	2.7414	0.0622	44.0630	0.0000	2.6157	2.8670
State_Florida	2.6902	0.0670	40.1462	0.0000	2.5549	2.8255
State_New York	2.7089	0.0679	39.8671	0.0000	2.5716	2.8461

Omnibus:	26.552	Durbin-Watson:	0.891
Prob(Omnibus):	0.000	Jarque-Bera (JB):	66.927
Skew:	-1.462	Prob(JB):	0.000
Kurtosis:	8.135	Condition No.:	3017126942023893385216

MSE: 0.04634

Applying sqrt_transformation, MLR_Model

Model:	OLS	Adj. R-squared:	0.516
Dependent Variable:	Profit	AIC:	466.2763
Date:	2023-05-12 10:30	BIC:	475.4195
No. Observations:	46	Log-Likelihood:	-228.14
Df Model:	4	F-statistic:	12.99
Df Residuals:	41	Prob (F-statistic):	6.42e-07
R-squared:	0.559	Scale:	1334.4

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	156.1016	21.6311	7.2166	0.0000	112.4168	199.7864
Administration	0.0004	0.0002	2.0013	0.0520	-0.0000	0.0008
Marketing Spend	0.0004	0.0000	7.0563	0.0000	0.0003	0.0005
State_California	56.8195	9.9667	5.7010	0.0000	36.6914	76.9476
State_Florida	47.9895	10.7348	4.4705	0.0001	26.3101	69.6689
State_New York	51.2926	10.8849	4.7123	0.0000	29.3101	73.2751

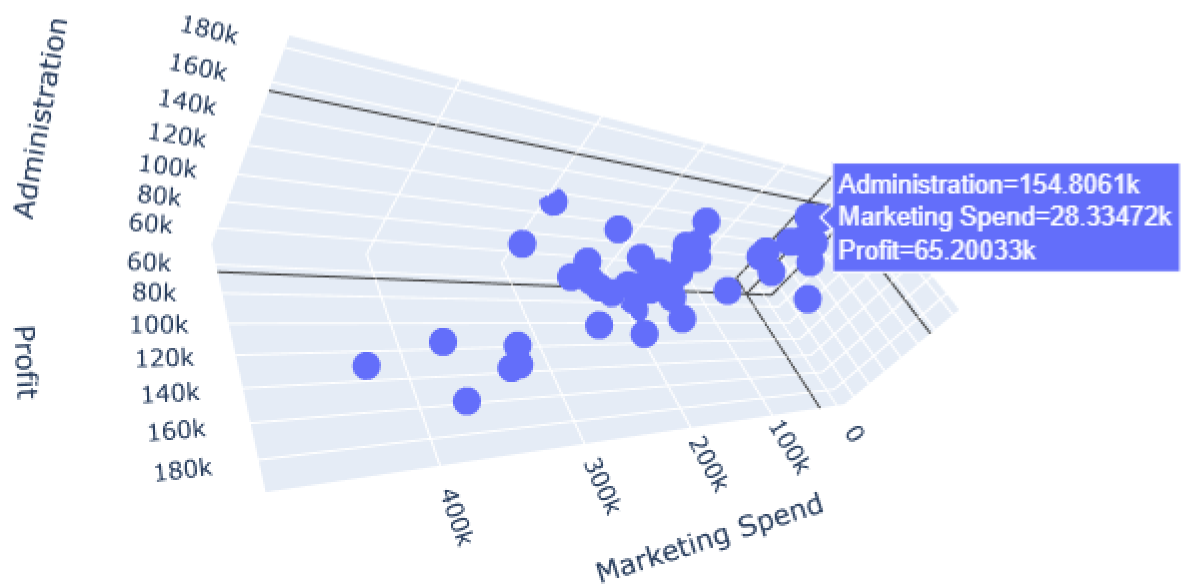
Omnibus:	16.542	Durbin-Watson:	0.946
Prob(Omnibus):	0.000	Jarque-Bera (JB):	27.515
Skew:	-1.015	Prob(JB):	0.000
Kurtosis:	6.199	Condition No.:	3017126942023893385216

MSE: 1189.3623

P_value: 0.000002	**P_value: 0.0000006**

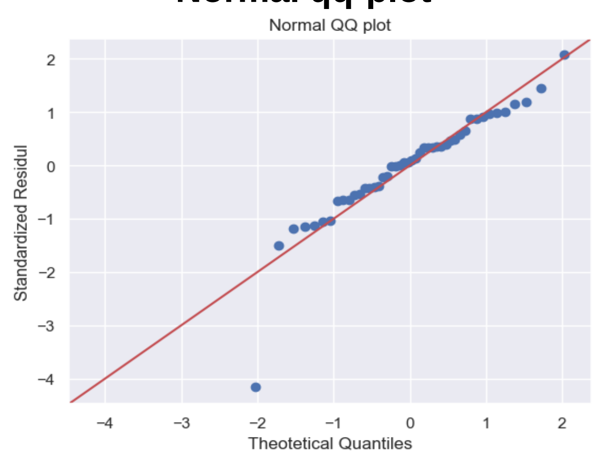
Log transformed model explain around **47% variation** but the sqrt transformed model explain **51% variation**, but still we prefer log transformed model because it gives Less MSE

3D scatter plot



Diagnostic Plots

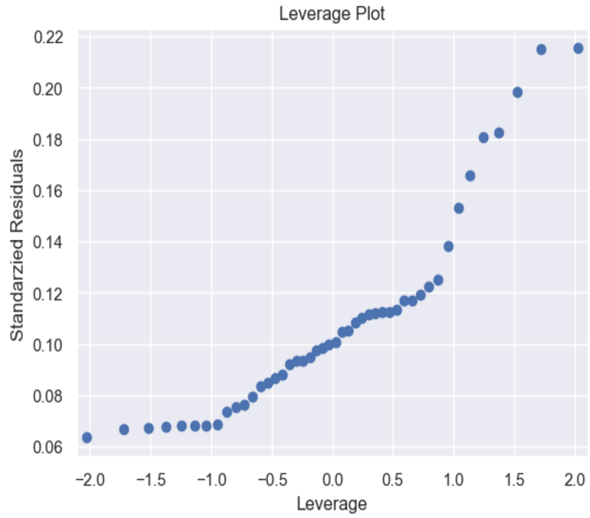
Normal qq-plot



Residual is normally distributed

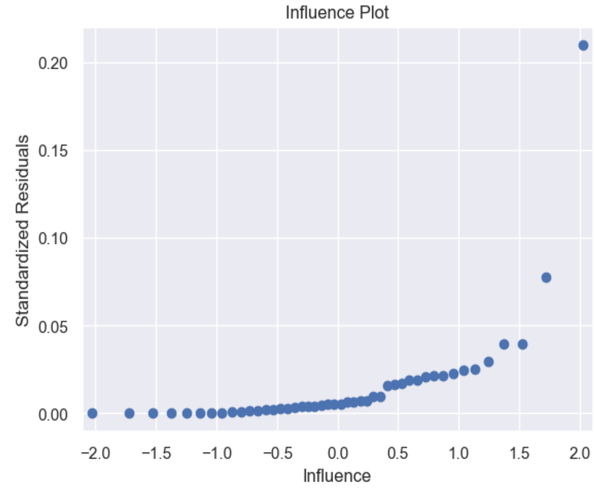
Residual Plots:

Leverage plot



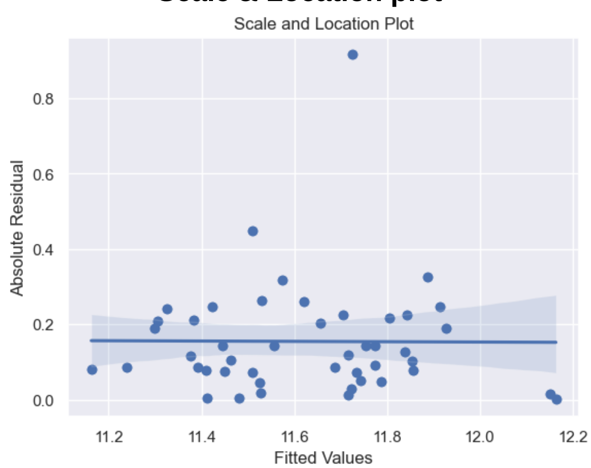
Points with high leverage have extreme predictor values compared to rest of the data

Influence Plot



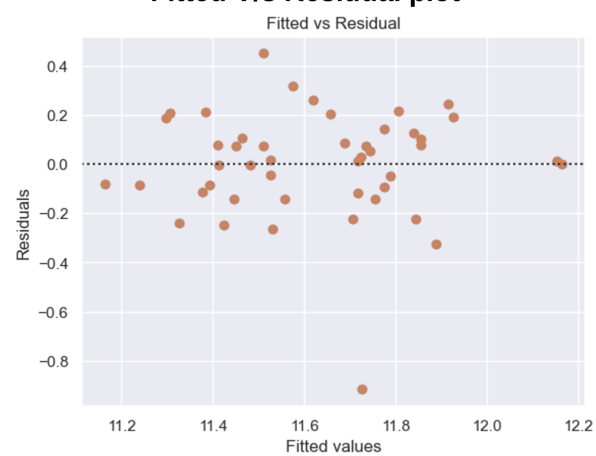
Larger points indicate higher influence suggesting that these data points have a greater impact the estimated regression coefficients

Scale & Location plot



The plot helps to assess 95% CI ,Homoscedasticity,also the graph shows outliers or influential observations and evaluate the adequacy of the model's fit

Fitted V/s Residual plot



The plot helps to assess Linearity, Homoscedasticity,also the graph shows outliers or influential observations and evaluate the adequacy of the model's fit

Conclusion:

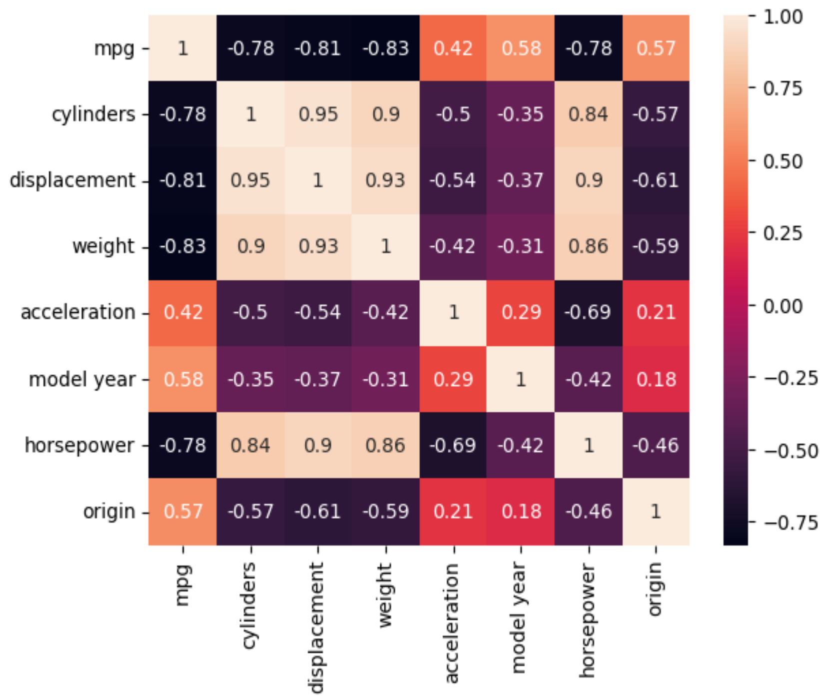
Open point is Highly influenced so, coefficients of estimates may not be appropriate to interpret the results.

So, one option is removing that point & run the model again.

Ridge and Lasso Regression

Data_file: [Car_mpg]
Code_file: [Ridge & Lasso Regression]
dependent variable: mpg
Independent variable: 'cylinders', 'displacement', 'weight', 'acceleration', 'model year','horsepower','origin'

Correlation_matrix



>Mpg is highly correlated with cylinders (-0.78), displacement (-81) and weights (-0.83)
>Cylinder Displacement and Weight are internally highly correlated

Linear Regression model summary:

Model:	OLS	Adj. R-squared:	0.819
Dependent Variable:	mpg	AIC:	350.9121
Date:	2023-05-15 18:38	BIC:	380.3807
No. Observations:	294	Log-Likelihood:	-167.46
Df Model:	7	F-statistic:	190.3
Df Residuals:	286	Prob (F-statistic):	1.11e-103
R-squared:	0.823	Scale:	0.18803

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	0.0032	0.0254	0.1260	0.8998	-0.0468	0.0532
cylinders	-0.0568	0.0829	-0.6850	0.4939	-0.2201	0.1064
displacement	0.2229	0.1201	1.8558	0.0645	-0.0135	0.4593
weight	-0.7791	0.0904	-8.6189	0.0000	-0.9571	-0.6012
acceleration	0.0749	0.0413	1.8119	0.0710	-0.0065	0.1563
model year	0.3817	0.0284	13.4294	0.0000	0.3258	0.4376
horsepower	0.0052	0.0818	0.0634	0.9495	-0.1558	0.1661
origin	0.1244	0.0332	3.7497	0.0002	0.0591	0.1896

Omnibus:	23.151	Durbin-Watson:	1.961
Prob(Omnibus):	0.000	Jarque-Bera (JB):	38.361
Skew:	0.488	Prob(JB):	0.000
Kurtosis:	4.476	Condition No.:	12

- Since the corresponding p-value of some independent variable is greater than 0.55 so that variables are insignificant for the model.
- If we remove these variables from the model, we lose important information from data.
- To tackle this problem, we use ridge regression

Ridge & Lasso Regression:

Both Ridge & Lasso Regression is a variation of linear regression. We use ridge regression to tackle the multicollinearity problem. Due to multicollinearity, we see a very large variance in the least square estimates of the model. So, to reduce this variance a degree of bias is added to the regression estimates.

Ridge regression when you want to handle multicollinearity and retain all predictors, and use Lasso regression when you want to perform feature selection and reduce the number of variables in your model.

Ordinary Least Square (OLS) will create a model by minimizing the value of Sum Square Error (SSE), Whereas the Ridge regression will create a model by minimizing:

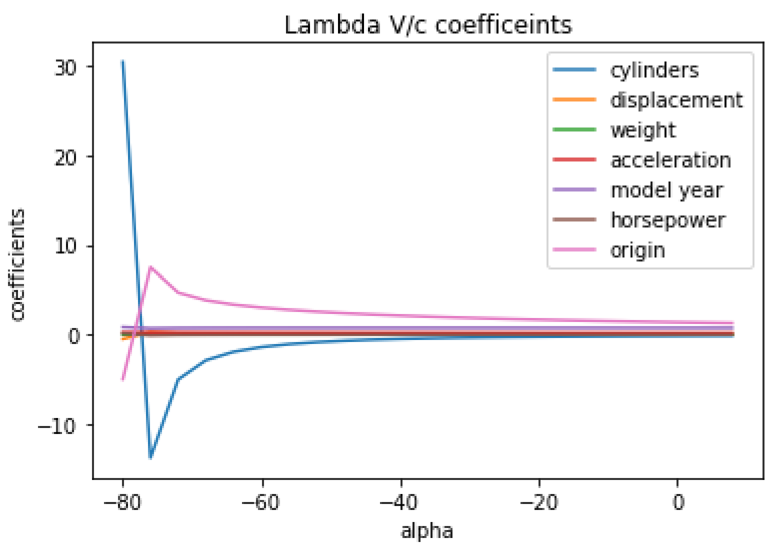
$$SSE + \lambda \sum_{i=1}^{n} (\beta_i)^2$$

SSE: Loss
Lambda: Penalty term
Beta: coefficients of regression model

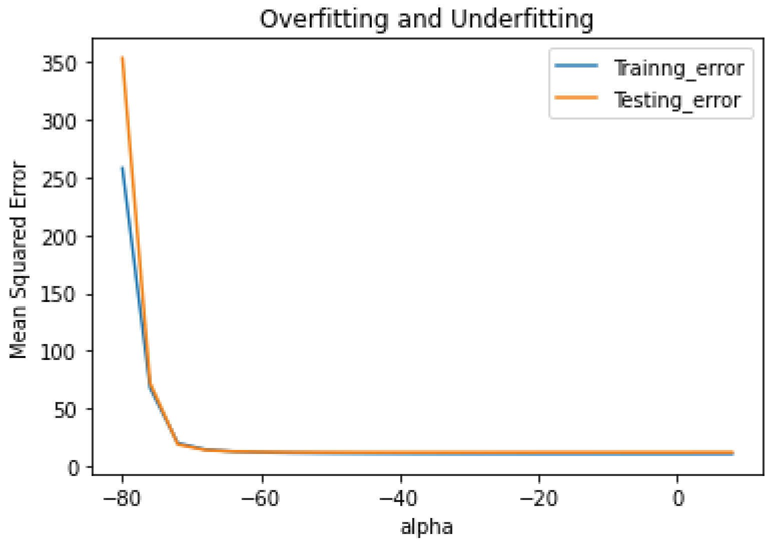
Ridge Model Summary For lambda (-80 to 10, differ by 4)

Ridge_model_summary(lambda=8)
Since lambda=8, provides the less value of MSE, We choose value of lambda as 8

Plot of Ridge coefficients as a function of the regularization:



From the plot as alpha increases the coefficients convert to smaller values of their original. This is the power of ridge regression, making the coefficients smaller to limit the collinearity between predictors.



Ridge

Price=18.460--0.120*(cylinders)+0.016*(displacement)--0.007*(Weight)+0.186*(acceleration)+0.737*(model_year)-0.005*(horse_power) +1.287*(Origin)

Lasso Regression

Ordinary Least Square (OLS) will create a model by minimizing the value of Sum Square Error (SSE), Whereas the Ridge regression will create a model by minimizing:

$$\mathbf{SSE + \lambda\Sigma|\beta_j|}$$

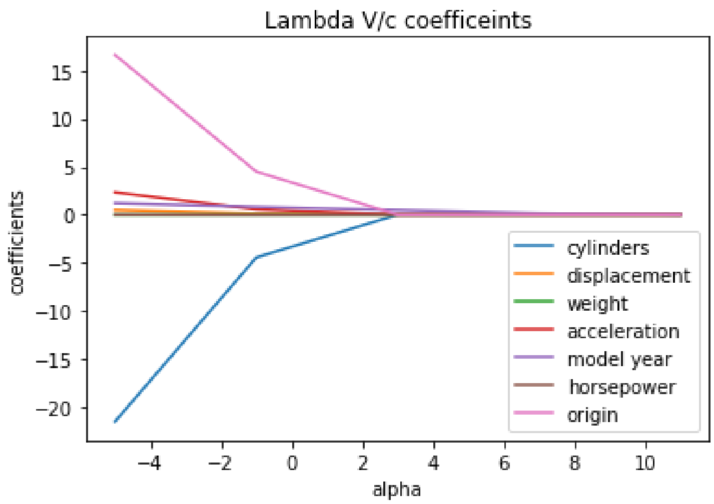
SSE: Loss
Lambda: Penalty term
Beta: coefficients of regression model

Lasso Model Summary For lambda (-5 to differ by 4**)**

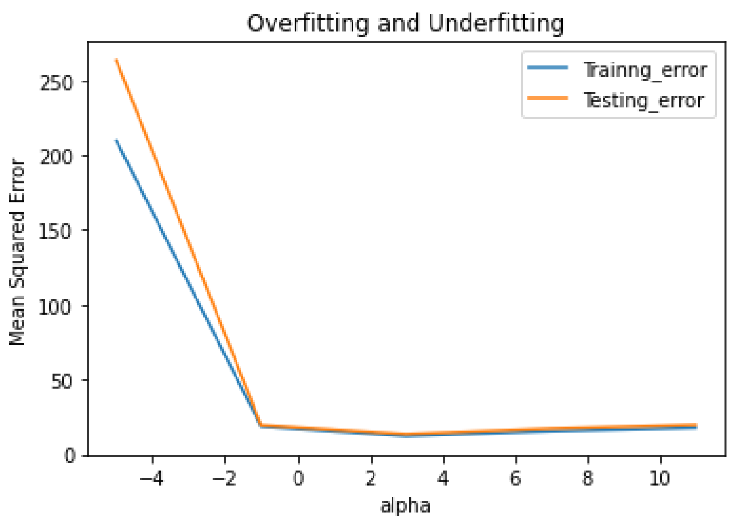
alpha	R_squre	Adj_rsquare	MSE_train	MSE_test	intercept	cylinders	displacement	weight	acceleration	model year	horsepow
-5	0.146670	0.161131	209.555193	263.230698	-76.652114	-21.556485	0.484003	-0.010123	2.330751	1.199917	0.00553
-1	0.685727	0.457232	18.628452	19.385952	-28.344537	-4.452926	0.111946	-0.007316	0.556629	0.819053	-0.01757
3	0.716459	0.567187	12.206487	13.552629	7.034546	-0.000000	-0.000063	-0.006461	0.000000	0.483227	-0.01142
7	0.595878	0.589047	15.530406	17.238123	33.364681	-0.000000	-0.005254	-0.006076	0.000000	0.141081	-0.01636
11	0.521609	0.585844	17.801944	19.614243	44.343931	-0.000000	-0.007042	-0.006203	0.000000	0.000000	-0.01201

Since lambda=3, provides the less value of MSE, we choose value of lambda as 3

Plot of Lasso coefficients as a function of the regularization:



From the plot as alpha increases the **coefficients convert to zero.** This is the power of Lasso regression.



Coefficients	Ridge (8)	Lasso (3)	Linear_regression
cylinders	-0.120909		-0.1453
displacement	0.01644	-0.005	0.0175
weight	-0.007079	-0.006	-0.0071
acceleration	0.186449		0.1895
model year	0.73758	0.141	0.7388
horsepower	-0.005999	-0.016	-0.0067
origin	1.287833		1.3764

Conclusion:

- The cost function for both ridge and lasso regression are similar. However, ridge regression takes the square of the coefficients and lasso takes the magnitude.
- Lasso regression can be used for automatic feature selection, as the geometry of its constrained region allows coefficient values to inert to zero.
- An alpha value of zero in either ridge or lasso model will have results similar to the regression model.
- The larger the alpha value, the more aggressive the penalization.

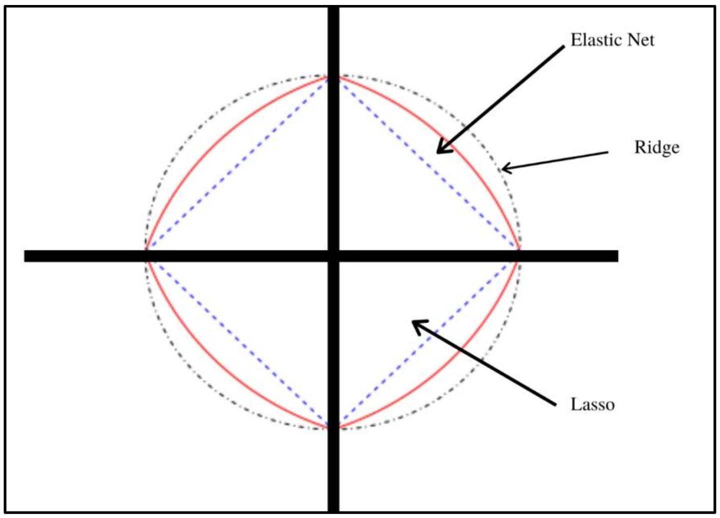
When to use Ridge and lasso Regression:

Ridge regression when you want to handle multicollinearity and retain all predictors, and use Lasso regression when you want to perform feature selection and reduce the number of variables in your model.

Elastic Net Regression

Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve the regularization of statistical models.

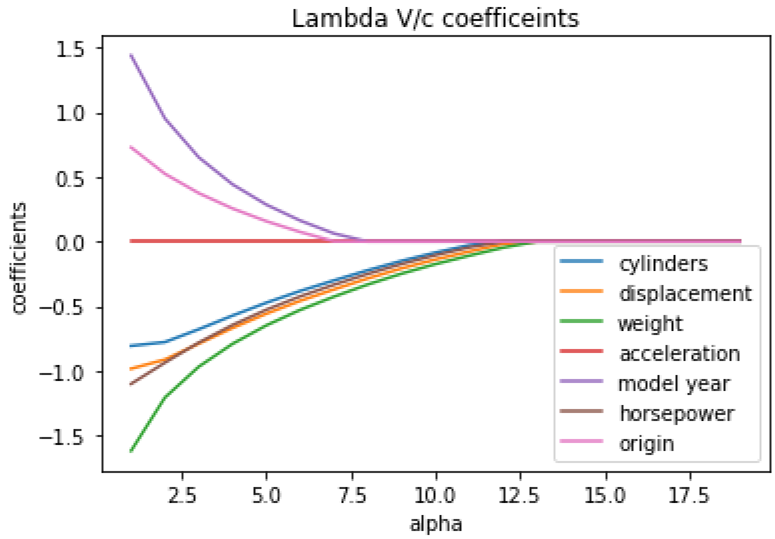
$$
\begin{aligned}
P_{\alpha}(\beta) &= (1-\alpha)2\|\beta\|_{22} + \alpha\|\beta\|_{1} \\
&= {}_{p}\textstyle\sum_{j\,=\,1}((1-\alpha)2\beta_{2j} + \alpha \mid \beta_{j} \mid)
\end{aligned}
$$



The elastic net draws on the best of both worlds – i.e., lasso and ridge regression.In the procedure for finding the elastic net method’s estimator, two stages involve both the lasso and regression techniques. It first finds the ridge regression coefficients and then conducts the second step by using a lasso sort of shrinkage of the coefficients.

Code_file: Elasticnet.ipynb
dependent variable: mpg
Independent variable: 'cylinders', 'displacement', 'weight', 'acceleration', 'model year','horsepower','origin'

Elastic Net Regression Model Summary for (lambda 0 to 10):
Elasticnet_summary_(Car_dataset).csv



From the plot we can see that as the value of alpha increases regression coefficients tends to zero i.e., it reduces dimensions of data and tackle the problem of multicollinearity.
As our data is lower dimensional Result provided by the Elastic Net Regression and Lasso Regression both are insignificant.

Polynomial Regression

Polynomial regression is a type of regression analysis in which the relationship between the two variables is not linear, but rather follows a curved pattern. This can happen for a variety of reasons, such as when the independent variable is a categorical variable or when the dependent variable is a measure of something that changes over time.

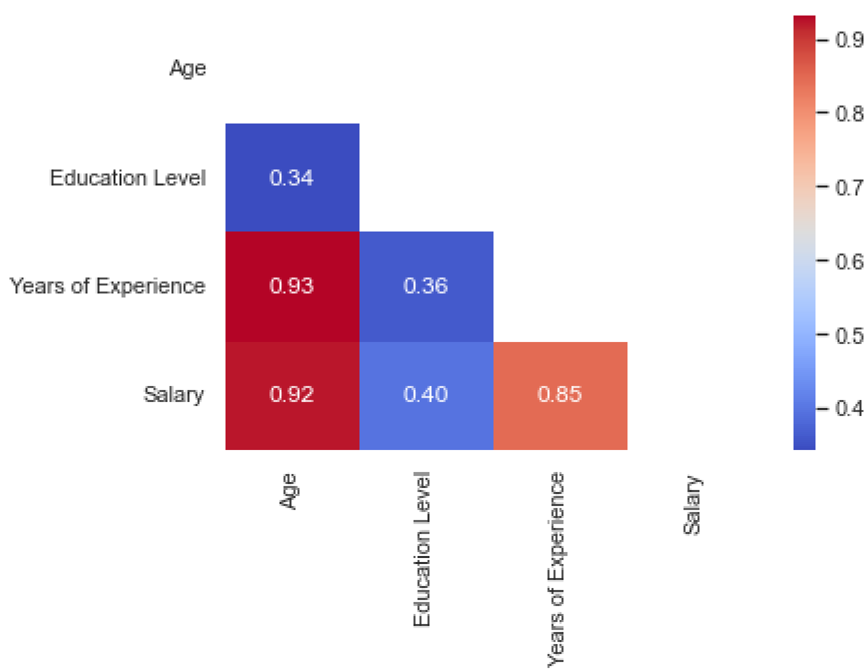
The most common type of polynomial regression is quadratic regression, which uses a quadratic function to fit the data. Quadratic functions are U-shaped curves, and they can be used to fit data that has a curvilinear relationship.

It is important to note that polynomial regression can also be used to fit data that does not have a curvilinear relationship. This can lead to overfitting, to avoid overfitting,
Use a technique called cross-validation. Cross-validation involves dividing the data into two sets: a training set and a test set. The training set is used to fit the model, and the test set is used to evaluate the model.

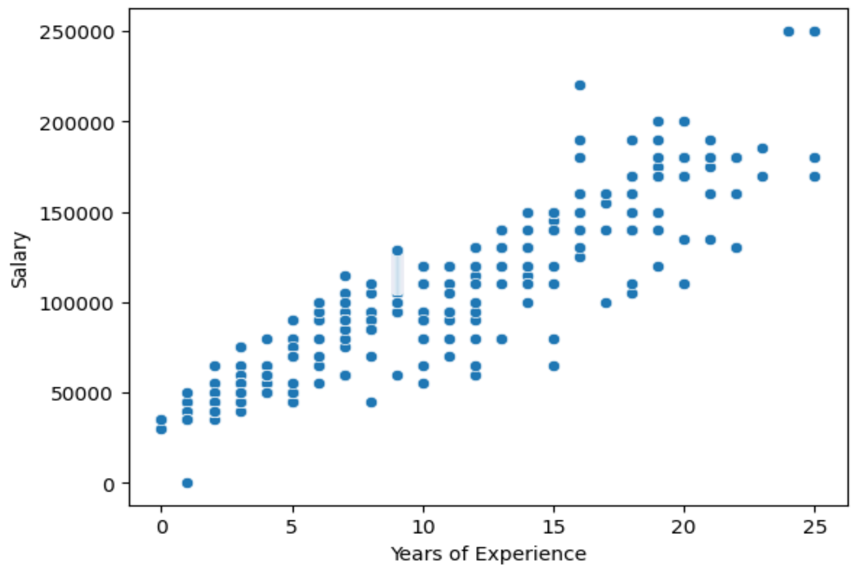
A polynomial regression model takes the following form:

$$\mathbf{Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_h X^h + \varepsilon}$$

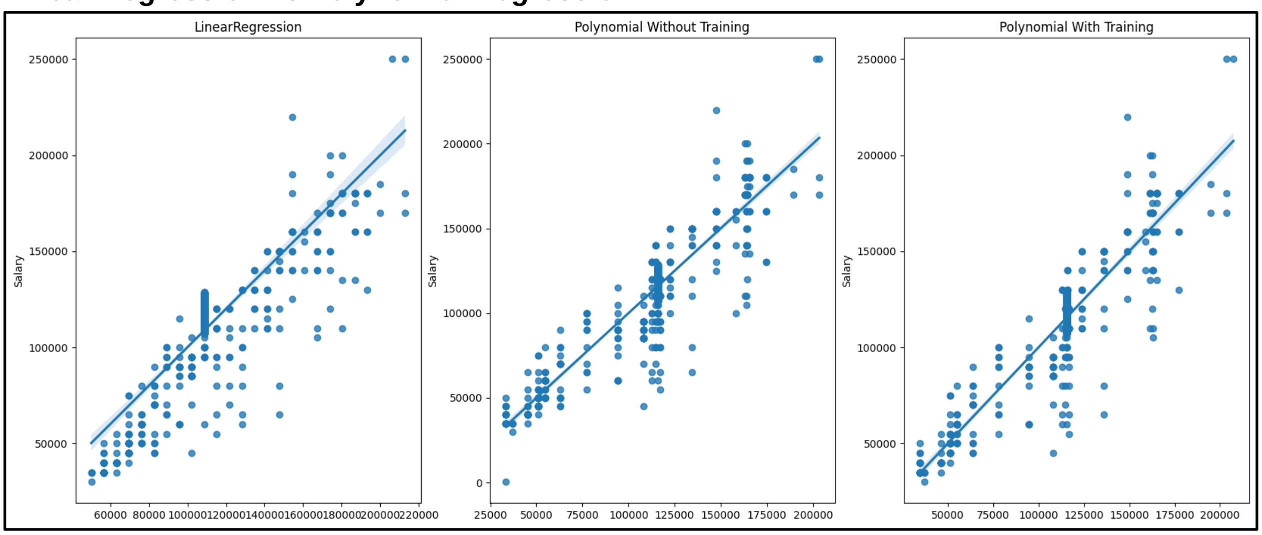
Data_file:[**Salary_Data**]
Code_file: [**Polynimial_Regression**]
dependent variable: Salary
Correlation_matrix:



Relationship Between Experience and Salary:



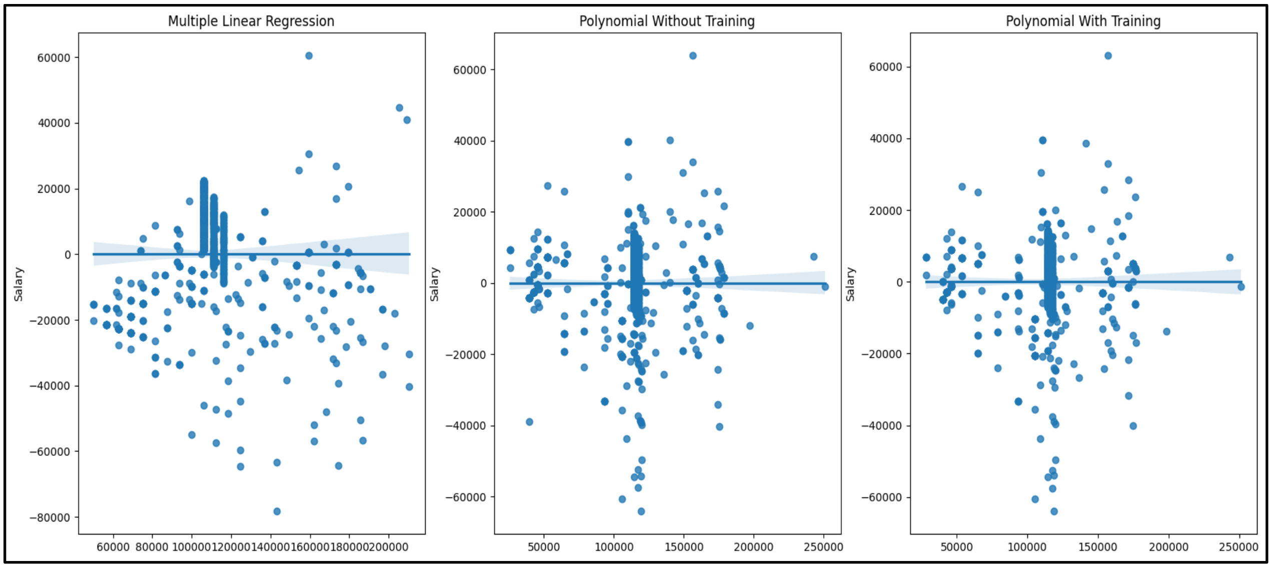
Linear Regression vs Polynomial Regression:



Linear Regression	**Polynomial Regression (without cross validation)**	**Polynomial Regression (cross validation)**
R2_square 0.554477323639196 **RMSE:** 16724.798898195742	**R2_square** 0.79928840264464 **RMSE:** 12746.931970497773	**R2_square** 0.7818955573682126 **RMSE:** 12187.365606307107

By comparing these three models we prefer Polynomial Regression (Cross Validation) because it gives less Root mean squared error and Overall variation explained by model is around 78%.

Multiple Linear Regression vs 3D Polynomial Regression:



Multiple Linear Regression	3D Polynomial Regression (Without cross validation)	3D Polynomial Regression (Cross validation)
R2_square 0.5712684588529795 **RMSE:** 16754.304051575935	**R2_square** 0.8471069797151853 **RMSE:** 11359.956846052663	**R2_square** 0.830127740215452 **RMSE:** 10992.296400943856

By comparing these three models we prefer 3D-Polynomial Regression(Cross Validation) because it gives less Root mean squared error and Overall variation explained by model is around 83% .

Gradient Descent

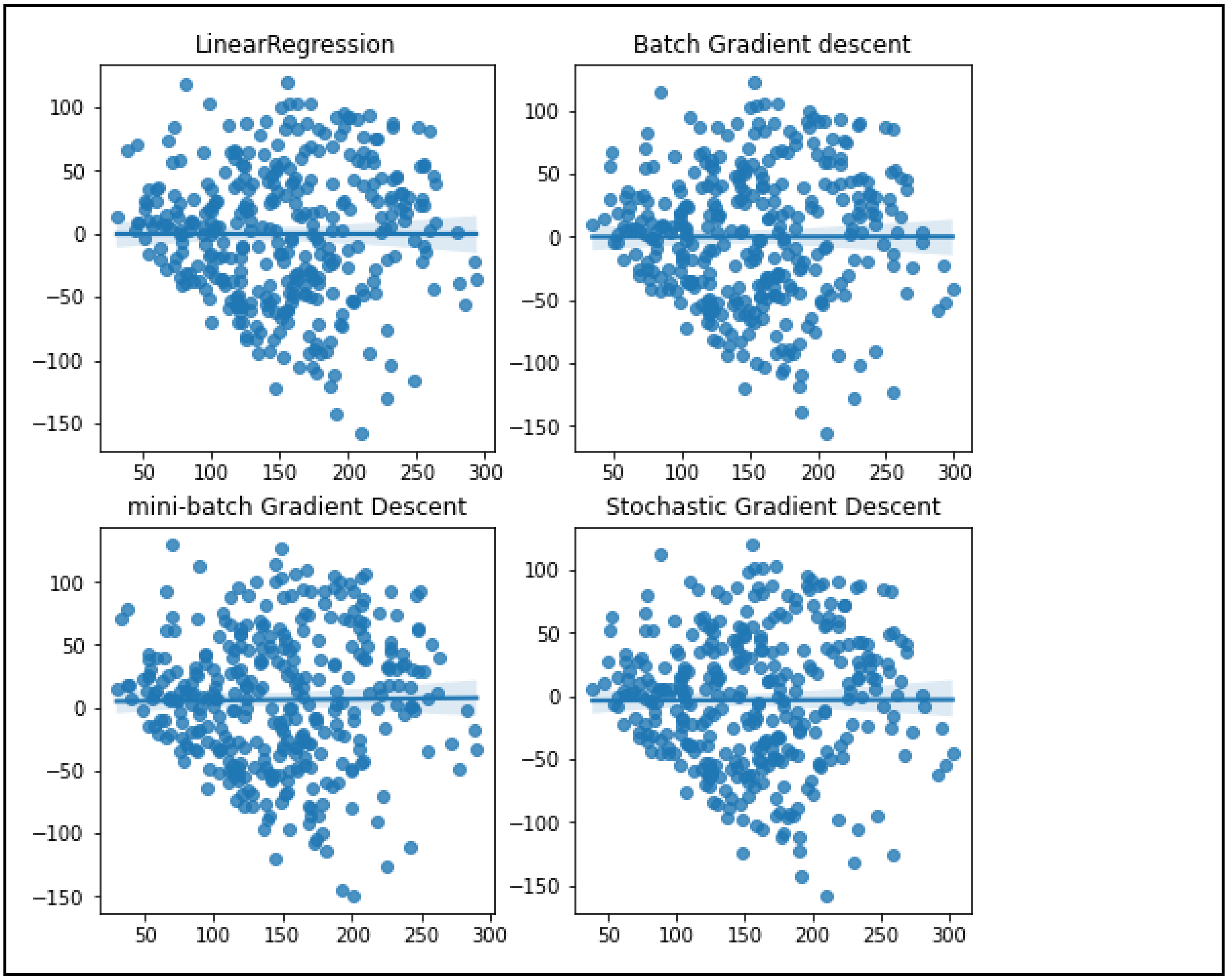
Gradient descent is an iterative optimization algorithm used in machine learning to find the minimum of a function. The idea is to start at a point and then repeatedly move in the direction of the negative gradient of the function until you reach a minimum.

There are three main types of gradient descent:

- **Batch gradient** descent uses the entire training dataset to calculate the gradient at each step. This is the most computationally **expensive** type of gradient descent, but it can be the **most accurate**.
- **Stochastic gradient descent** uses a single training example to calculate the gradient at each step. This is the **least computationally expensive** type of gradient descent, but it can be less accurate than batch gradient descent.
- **Mini-batch gradient descent** uses a small subset of the training dataset to calculate the gradient at each step. This is a compromise between batch gradient descent and stochastic gradient descent, and it is often the **most efficient and accurate** type of gradient descent

Type	Pros	Cons
Batch gradient descent	* Most accurate * Can handle large datasets	* Most computationally expensive
Stochastic gradient descent	* Least computationally expensive * Can handle small datasets	* Can be less accurate than batch gradient descent
Mini-batch gradient descent	* Compromise between batch gradient descent and stochastic gradient descent * Efficient and accurate	* Not as accurate as batch gradient descent

Data : Sklearn.dibetes
Code_File:Gradient_Descent



	Linear Regression	Batch Gradient Descent	Stochastic Gradient Descent	Mini-Batch Gradient Descent
R_square	0.4384	0.4297	0.4315	0.4266
MSE	2992.557	3038.991	3029.210	3055.170

	Linear Regression	Batch Gradient Descent	Stochastic Gradient Descent	Mini-Batch Gradient Descent
Intercept	151.430	151.363	154.750	153.318
coefficient_1	-30.621	-30.937	-30.469	4.946
coefficient_2	-272.254	-269.906	-273.743	-188.776
coefficient_3	528.844	534.483	533.367	464.025
coefficient_4	327.702	324.969	330.653	279.084
coefficient_5	-581.014	-129.413	-96.295	-28.6785
coefficient_6	332.962	-25.795	-52.575	-85.702
coefficient_7	-27.976	-226.911	-236.116	-220.861
coefficient_8	139.284	88.443	88.639	144.337
coefficient_9	665.075	493.822	479.981	370.584
coefficient_10	61.905	63.845	58.306	133.322

The choice of which type of gradient descent to use depends on the specific problem being solved. For **large datasets, mini-batch gradient** descent is often the best choice. For **small datasets, batch gradient descent** may be more accurate. And for very **small datasets, stochastic gradient descent** may be the best option.

Logistic Regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. The independent variables can be either categorical or continuous variables. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

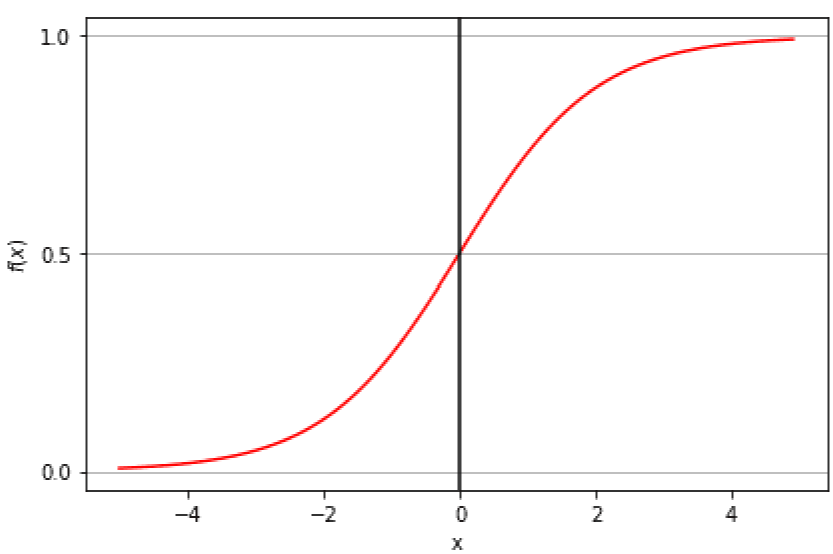
Properties of Logistic Regression:

- The dependent variable in logistic regression follows Bernoulli Distribution.
- Estimation is done through maximum likelihood.
- No R Square for Model fitness

Sigmoid Function

$$P = e^{\beta_0} / (1 + e^{\beta_0})$$

The sigmoid function, also called logistic function gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0.



Types of Logistic Regression:

- **Binary Logistic Regression**: The target variable has only two possible outcomes
- **Multinomial Logistic Regression**: The target variable has three or more nominal
- **Ordinal Logistic Regression**: the target variable has three or more ordinal categories

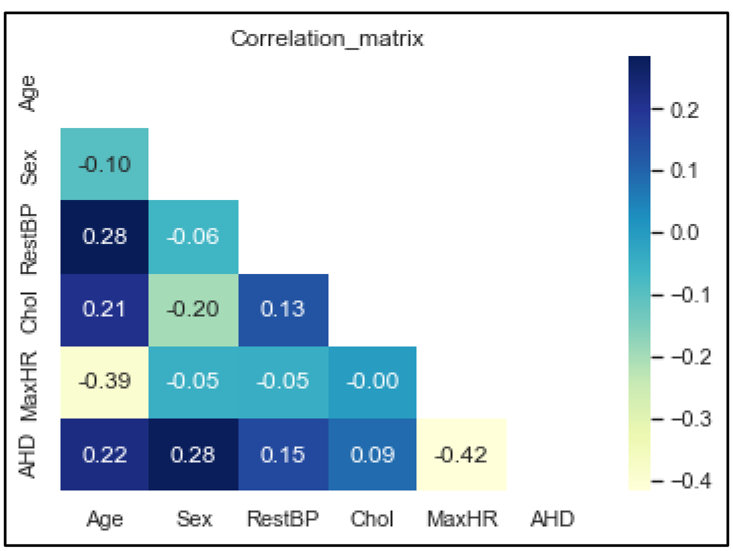
Advantages

- Its efficient and straightforward nature,
- it doesn't require high computation power, is easy to implement, easily interpretable, and used widely by data analysts and scientists.
- Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations.

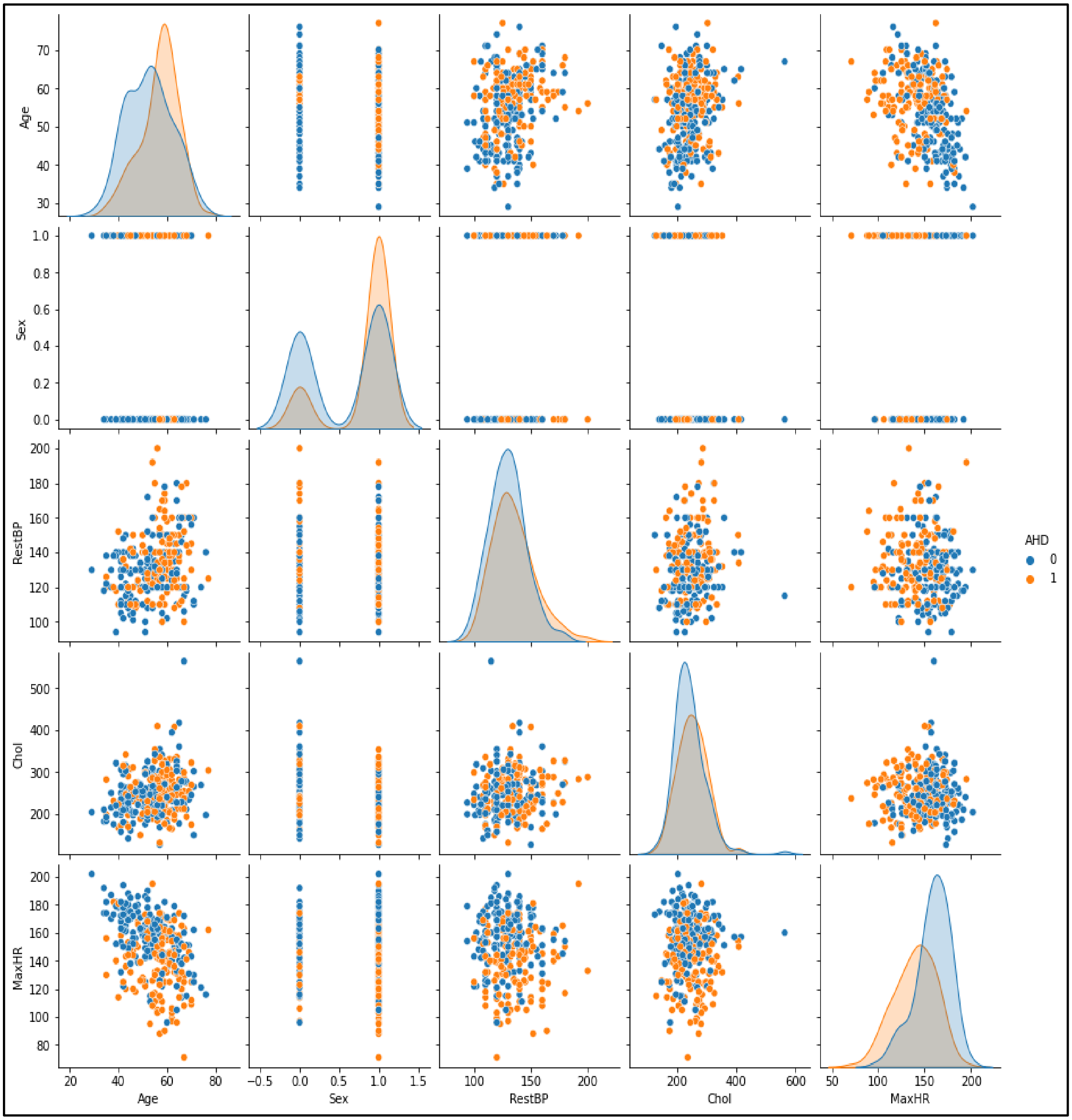
Disadvantages

- Logistic regression is not able to handle a large number of categorical features/variables.
- It is vulnerable to overfitting.
- Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features.
- Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

Data_file:[Acute_Heart_disease.csv]
Code_file: [Logistic_Regression(Heart_data)]
dependent variable: ADH(Acute Heart Disease)
Independent variables:'Age','Sex','RestBP','Chol','MaxHR'



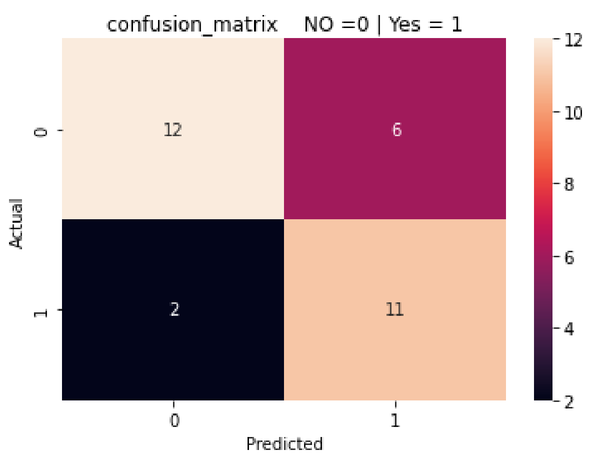
Pairplot (heart disease; No=0 | Yes=1)



Unscaled Data	Scaled Data

Unscaled Data

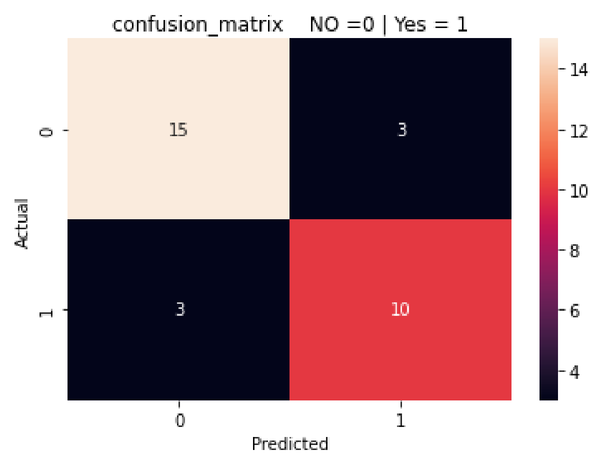
	precision	recall	f1-score	support
0	0.86	0.67	0.75	18
1	0.65	0.85	0.73	13
accuracy			0.74	31
macro avg	0.75	0.76	0.74	31
weighted avg	0.77	0.74	0.74	31



- Patients actual not having heart diseases but model predicted heart diseases: 6 (FP)-Type-I error
- Patients actual having heart diseases but model predicted they don't have heart diseases: 2 (FN)-Type-II error
- So, in this case Type-II error is more dangerous

Scaled Data

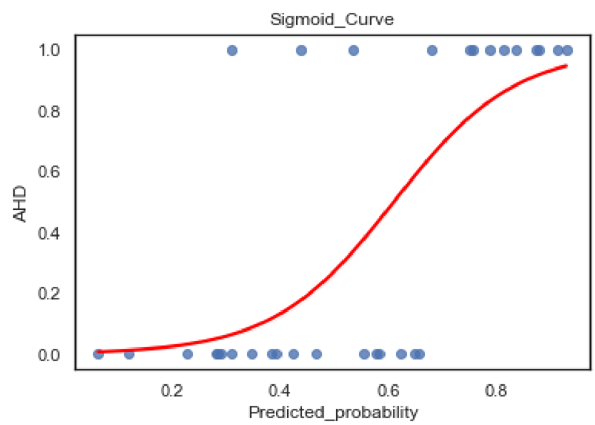
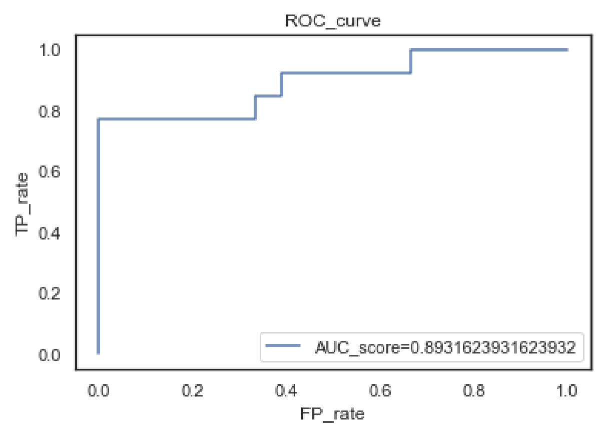
	precision	recall	f1-score	support
0	0.83	0.83	0.83	18
1	0.77	0.77	0.77	13
accuracy			0.81	31
macro avg	0.80	0.80	0.80	31
weighted avg	0.81	0.81	0.81	31



- Patients actual not having heart diseases but model predicted heart diseases: 3 (FP)-Type-I error
- Patients actual having heart diseases but model predicted they don't have heart diseases: 3 (FN)-Type-II error
- So, in this case Type-II error is more dangerous

Type-II error is more dangerous in this case, so unscaled data provides less Type-II error than scaled data.

Plots of Logistic Regression:

Sigmoid Curve	**ROC Curve**
	
And if the outcome of the sigmoid function is more than 0.5 then we classify that label as class 1 or positive class and if it is less than 0.5 then we can classify it to negative class or label as class 0.	It shows the performance of a classification model at all classification thresholds. The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. AUC = 0.89, This implies that the model can be 89% correctly predict the AHD (Acute Heart Disease)