Assignment -1

Part -1 Logistic Regression

Logistic Regression is a model used to predict an outcome based on one or more variables or columns in a data frame. Simple Logistic Regression has outcome as '0' or '1' i.e., binary and the independent variables can be categorical. The goal of logistic regression is to create a model that accurately predicts the likelihood of the outcome based on the values of the independent variables.

For this Assignment, we used penguin dataset with 8 columns and removed the null values. After cleaning the data, we have string data columns to categorical which uses numbers like in sex column 1 for "male" and 0 for "female", for species 1 for "Adelie" 2 for "Gentoo" and 3 for "Chipstrap", and for island 1 for "Torgersen", 2 for "Biscoe", 3 for "Dream". For non-categorical columns like bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g we do normalization to transform features to be on a similar scale which improves the performance and training stability of the model. Normalization is done by taking max and min value of each column and using lamba we scale the data between 0 to 1.

We took 'Sex' column as the target variable 'Y'. So 'X' has all columns expect "sex" and "year" as it is not being used. Later data split of 80 to 20 is done. We built a 'LogitRegression' class that has functions like sigmod, cost, gradient_descent, fit and predict which uses the respective functions and the inputs learning rate and number of iterations. Later the model is called, and functions fit and predict for predictions of 'Y' and for the accuracy the model predictions are compared with the test data that is split beforehand.

1) The accuracy of the model we built is 86.56716417910447 % for the model with learning rate 1e-3 and iterations 100000.

```
def acc_model(model_y,Y_test):
  s = 0
  for i in range(Y_test.shape[0]):
    if(model_y[i] == Y_test[i]):
      s = s + 1
  return s/Y_test.shape[0]
print("Accuracy is :", (acc_model(model_y,Y_test))*100)
```

```
Accuracy is : 86.56716417910447
```

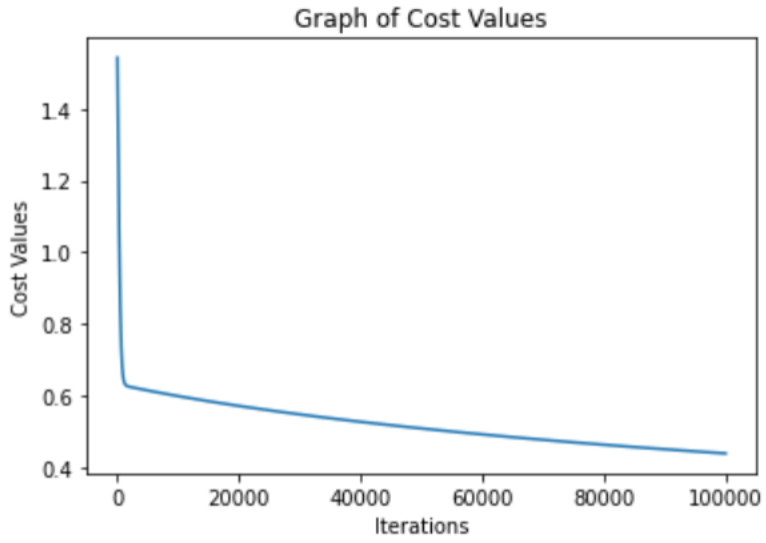
2) Loss values:

```
print("Cost/Loss values of the model are: ",model.loss())
```

```
Cost/Loss values of the model are:  [1.54312388 1.54118274 1.53924292 ... 0.43791948 0.43791837 0.43791726]
```

Graph:

```
import matplotlib.pyplot as plt
plt.plot(model.cost_values)
plt.title('Graph of Cost Values')
plt.xlabel('Iterations')
plt.ylabel('Cost Values')
plt.show()
```



Analysis:

The above plot shows cost values i.e., loss values over iterations during the training of the logistic regression model. The loss value at 0 is 1.543123 which is reduced over number of iterations which comes down to 0.43791 at 100000th iteration. We used matlablib library to use plot () function which plots the cost value, title to set Title graph, xlabel and ylabel to set x and y axis.

3) Hyperparameters are the parameters that are set before to train a model like learning rate and number of iterations used in logistic regression. These parameters have a significant impact on the accuracy of the model.

Example of 3 different learning rates and iterations and their respective accuracy. Here learning rate is same but we changed iterations and the change of iterations is effecting the accuracy and the leading the model with more errors.

Case - 1:

learning_rate=0.001

iterations=100000

Accuracy = 83.58

Case - 2:

learning_rate= 0.001

iterations= 200000

Accuracy = 86.56

Case - 3:

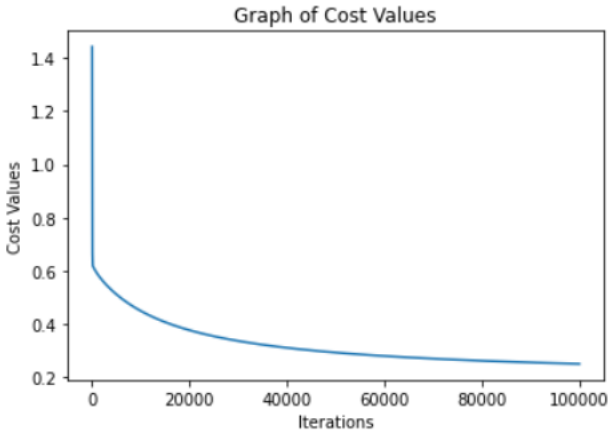
learning_rate= 0.001

iterations=10000

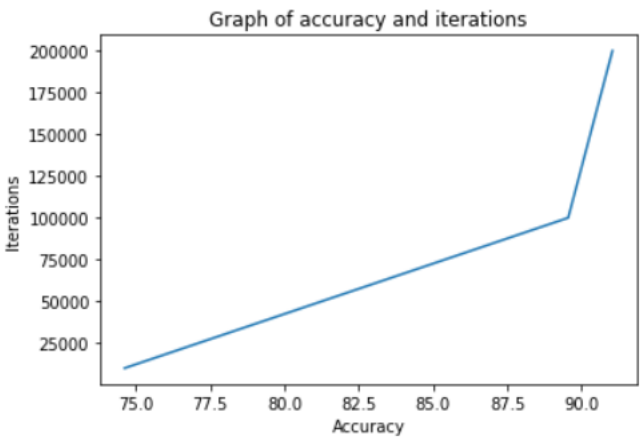
Accuracy = 58.208

Graphs:

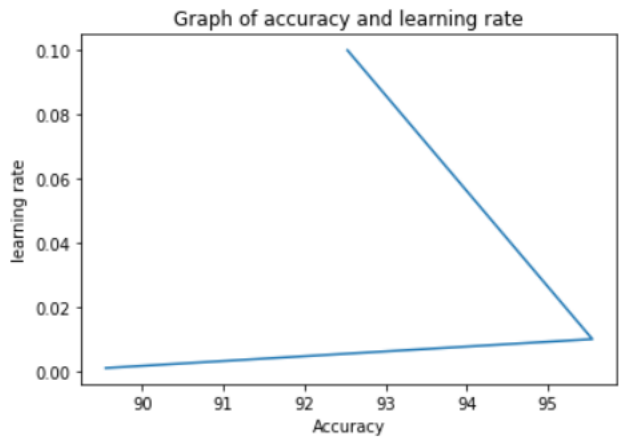
Line graph between Cost Values and iterations



Line graph between Cost values and accuracy



Line graph between Cost Values and Learning rate



4) Advantages of Logistic Regression:

a. It is a simple and easy to understand model.
b. The prediction outputs are easy to understand and interpret.
c. It has good accuracy for simple data.
d. It is fast to classify the unknown records.

Disadvantages of Logistic Regression:

a. It is limited to binary classification and cannot be used for multi-class classification without modifications.
b. It can overfit if the number of features is too large relative to the number of training examples.
c. It has limited complexity, which may make it less suitable for complex problems with non-linear relationships between features and outcomes.
d. Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.

References:

https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

https://c3.ai/glossary/data-science/hyperparameters/

Part – II

The dataset we used for the project is “flight_price_prediction” of 12 columns which are 'Unnamed: 0', 'airline', 'flight', 'source_city', 'departure_time', 'stops', 'arrival_time', 'destination_city', 'class', 'duration', 'days_left', 'price'. It has 300153 records . The data in the flight_price_prediction is of numerical as columns like duration, days_left, price have numerical values and categorical data as columns like airline, flight, source_city, departure_time, stops, arrival_time, destination_city, class has string values.

It has information about the airlines and their prices. The target variavble for the model is price which is a continuous value, and we use linear regression to predict the price based on given data. This data can be used by travel booking agents to check prices and give recommendations to their customers or to book flights as cheap as possible.

Dataset:

We loaded the dataset into two variable flight_data and graph_data were we used the flight_data for the model and graph_data for visualization of dataset.

	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955
...	...	...	...	...	...	...	...	...	...	...	...	...
300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099
300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585
300152	300152	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585

300153 rows × 12 columns

The shape of the dataset is:

```
flight_data.shape
```

```
(300153, 12)
```

Description of the dataset:

```
flight_data.describe
```

```
<bound method NDFrame.describe of        Unnamed: 0   airline   flight source_city departure_time stops  \
0               0  SpiceJet  SG-8709       Delhi        Evening  zero
1               1  SpiceJet  SG-8157       Delhi  Early_Morning  zero
2               2   AirAsia   I5-764       Delhi  Early_Morning  zero
3               3   Vistara   UK-995       Delhi        Morning  zero
4               4   Vistara   UK-963       Delhi        Morning  zero
...           ...       ...      ...         ...            ...   ...
300148     300148   Vistara   UK-822     Chennai        Morning   one
300149     300149   Vistara   UK-826     Chennai      Afternoon   one
300150     300150   Vistara   UK-832     Chennai  Early_Morning   one
300151     300151   Vistara   UK-828     Chennai  Early_Morning   one
300152     300152   Vistara   UK-822     Chennai        Morning   one

        arrival_time destination_city     class  duration  days_left  price
0              Night           Mumbai   Economy      2.17          1   5953
1            Morning           Mumbai   Economy      2.33          1   5953
2      Early_Morning           Mumbai   Economy      2.17          1   5956
3          Afternoon           Mumbai   Economy      2.25          1   5955
4            Morning           Mumbai   Economy      2.33          1   5955
...              ...              ...       ...       ...        ...    ...
300148       Evening        Hyderabad  Business     10.08         49  69265
300149         Night        Hyderabad  Business     10.42         49  77105
300150         Night        Hyderabad  Business     13.83         49  79099
300151       Evening        Hyderabad  Business     10.00         49  81585
300152       Evening        Hyderabad  Business     10.08         49  81585

[300153 rows x 12 columns]>
```

As Unnamed column has id we dropped that column along with flight and the description of numerical columns is:

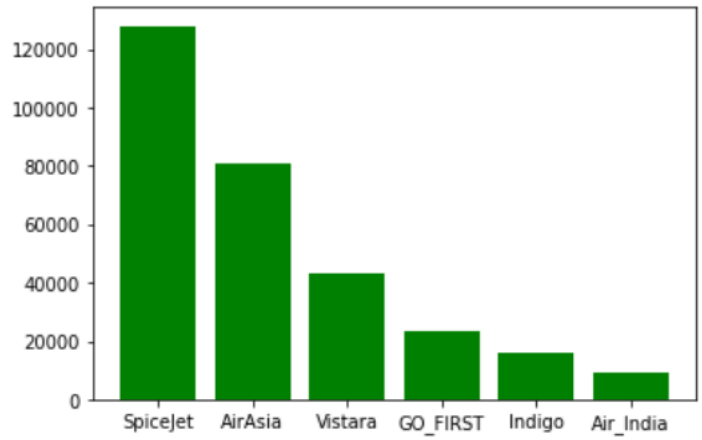
```
graph_data = graph_data.drop(['Unnamed: 0','flight'], axis = 1)
graph_data.describe()
```

	duration	days_left	price
count	300153.000000	300153.000000	300153.000000
mean	12.221021	26.004751	20889.660523
std	7.191997	13.561004	22697.767366
min	0.830000	1.000000	1105.000000
25%	6.830000	15.000000	4783.000000
50%	11.250000	26.000000	7425.000000
75%	16.170000	38.000000	42521.000000
max	49.830000	49.000000	123071.000000

Visualization of the dataset are:

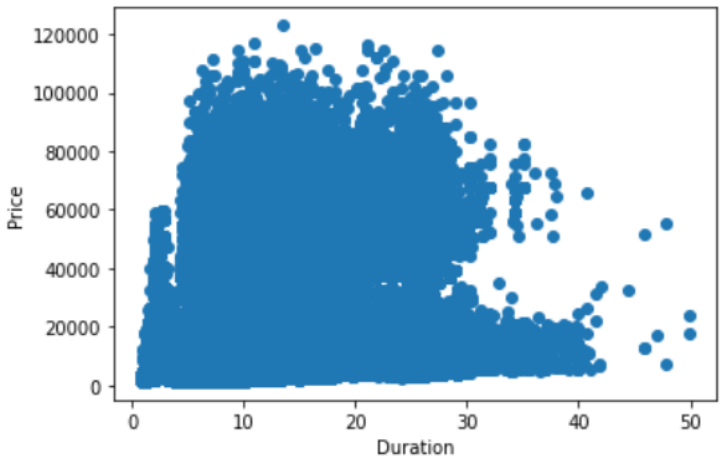
1) The following is a bar graph of unique airlines with respect to count of the fights i.e., no.of times the specific airlines were booked.

```
<BarContainer object of 6 artists>
```



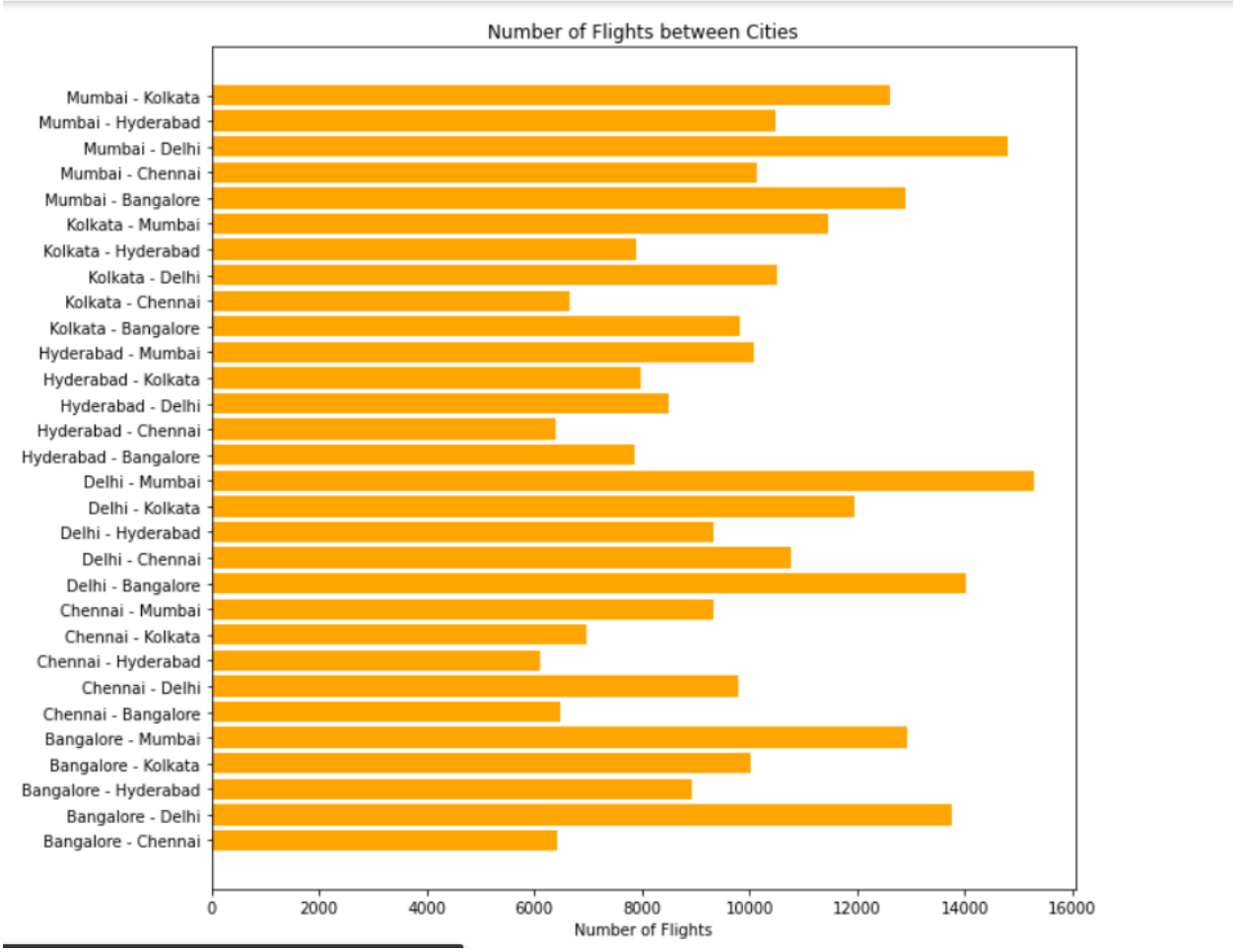
From this graph we can see that spiceJet has highest bookings and Air_India has lowest bookings.

2) The following is a scatter plot which plots different prices of the airlines for different duration timing of the flight.



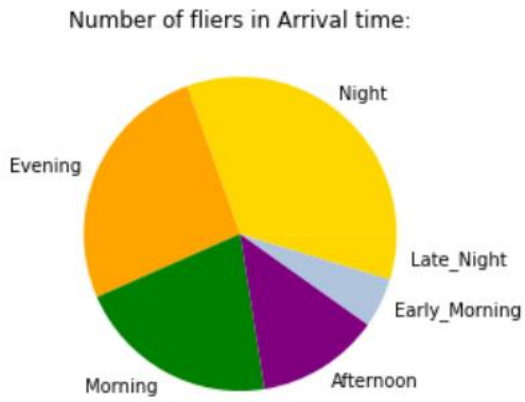
From this graph we can see that price of the flight is high for duration between 10 to 20, and surprisingly the price of flights of less duration and high duration is similar.

3) For the following graph we combination columns of ‘source_city’, ‘destination_city’ and taking its count with respect to each airline and plotted a histogram representing the source_city & destination_city and no.of flighs between those cities.



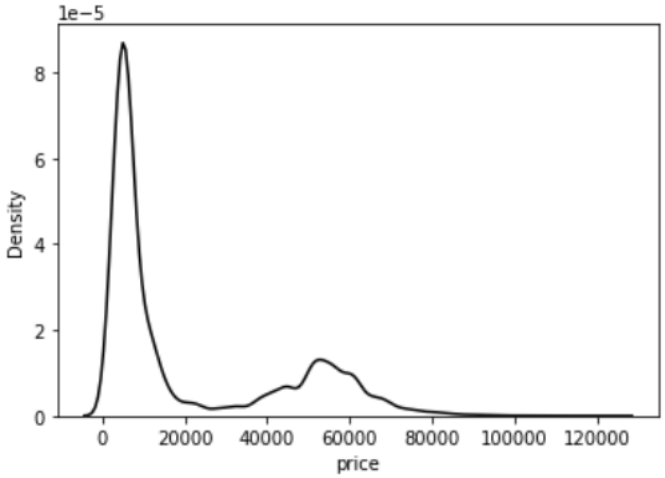
Here x-axis has Number of total flights and y-axis has source and destination combination in total. Flights from Mumbai to delhi and delhi to Mumbai have highest no.of flights flying between them.

4) The following is a pie chart displaying number of flights flying between different arrival times.



The number of evening flights are high compared to other arrival timings.

5) The kdeplot is displayed for the 'price' column to identify the count of flights with similar price range. Here we used seaborn package to use kdeplot and as we used only one column price it is displayed on x-axis and density on y



Here, flights with price range 0-20000 are high in the given dataset.

Linear Regression:

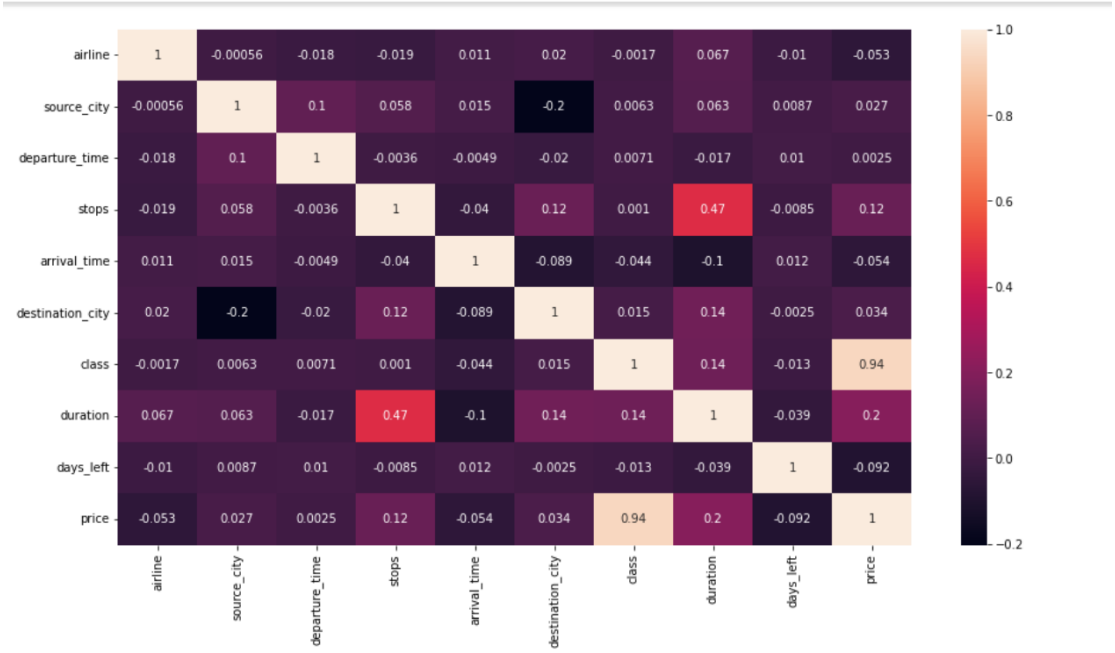
It is a learning model used to find the best fit linear line between the independent and dependent variables. For the dataset we considered, to predict price so it is taken as target variable. For the dataset, we pre processed the data by clearing the null values from the dataset and also removing unnamed column and flight which has name. After cleaning the data, categorical columns like name, city, time, stop, class and covered to numerical data like we did in the first part. Converted the duration of flight into seconds using seld created function samayam(). Later normalized the columns duration and days_left using min, max technique referencing from part 1.

The dataset after normalizing:

	airline	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	0	0	0	4	1	0	0.025987	0.0	5953
1	0	0	1	0	2	1	0	0.031387	0.0	5953
2	1	0	1	0	1	1	0	0.025987	0.0	5956
3	2	0	2	0	3	1	0	0.028687	0.0	5955
4	2	0	2	0	2	1	0	0.031387	0.0	5955
...	...	...	...	...	...	...	...	...	...	...
300148	2	5	2	1	0	4	1	0.184948	1.0	69265
300149	2	5	3	1	4	4	1	0.196423	1.0	77105
300150	2	5	1	1	4	4	1	0.271009	1.0	79099
300151	2	5	1	1	0	4	1	0.182248	1.0	81585
300152	2	5	2	1	0	4	1	0.184948	1.0	81585

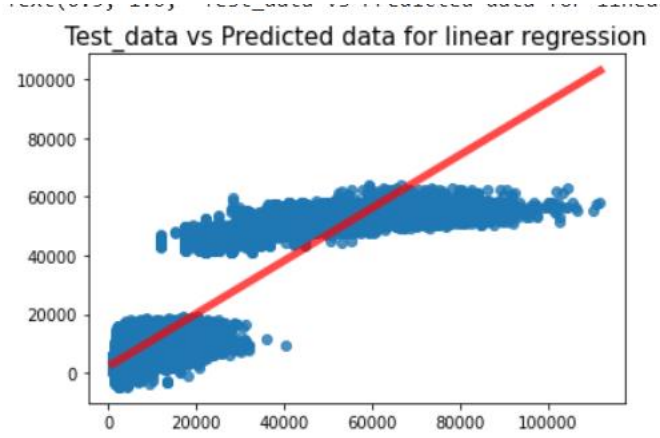
300153 rows × 10 columns

Heatmap of the dataset is :

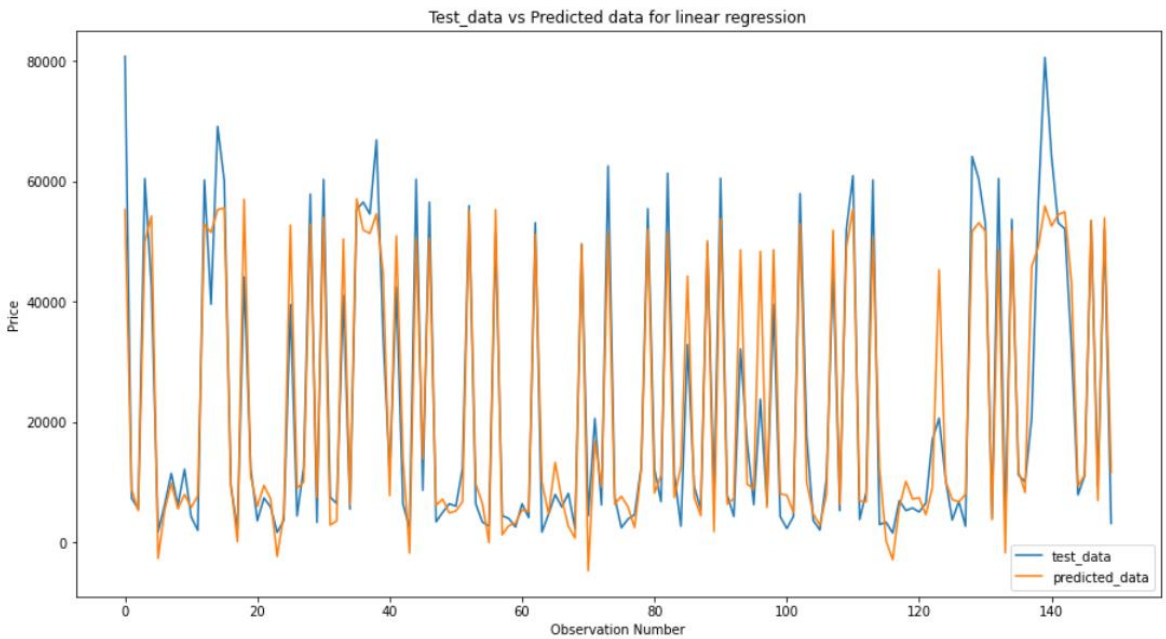


Later data split of 80 to 20 is done. We built a 'linear_regression' class that has functions like init, predict, cost and fit which uses the respective functions and the inputs learning rate and number of iterations. Later the model is called, and functions fit and predict for predictions of 'FY_test'.

The following graph is plotted between test values and model prediction values :



The line graph for 150 records for same model is:



R-squared error for linear regression model is:

```
R-squared: 0.9036887175723854
R-squared percentage: 90.36887175723855
```

Cost values for the model are:

```
model2.cost_values
```

```
[5992444885289.426]
```

Linear regression:

Advantages:

1. The relationship between the many predictor variables and the predicted variable can be expressed using the simplest equation, which is the linear regression model.
2. Linear regression models quickly because they do not call for complex calculations and makes quick use of enormous quantities of information.
3. One of the primary contributing factors to linear regression's appeal is its capability to quantify the relative contribution of one or even more response variable to the projected value when the predictors are independent to one another.

Disadvantages:

1. The linear regression model is just too simple to compensate for the complexity of the real world.
2. It is generally assumed by linear regression that predictor (independent) and predicted (dependent) variables are linearly linked, though this may not always be the case.
3. This assumption, which is rarely true, holds that the predictor variables are not correlated. Thus, it is crucial to eliminate correlation between independent variables, as the system depends on the assumption that independent variables are unrelated. When there is a high level of multicollinearity, two strongly correlated features will influence each other's weight and produce an unreliable model.

The benefits/drawbacks of using OLS estimate for computing the weights are:

Benefits:

1.OLS is a simple and straightforward procedure that is simple to implement using standard statistical software programs. OLS is also computationally efficient. It may also be utilized with big datasets and is computationally efficient.

2.OLS provides unbiased weight estimates: OLS estimates the weights in a method that renders them unbiased and consistent, which implies that as the sample size grows, the estimates will converge to the real values.

Drawbacks:

1. OLS makes the assumption that the dependent and independent variables have a linear relationship, that's not necessarily true. OLS may provide inaccurate estimates when the relationship is nonlinear.
2. OLS assumes that the error term's variance is constant at all of the levels for the independent variables. OLS may provide inaccurate estimates whenever the variance is not constant.

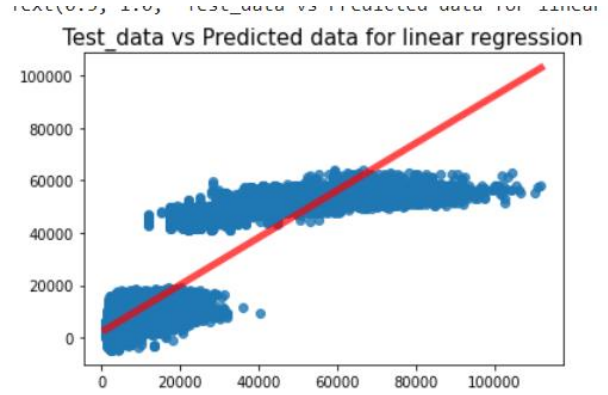
Part – III

Ridge Regression:

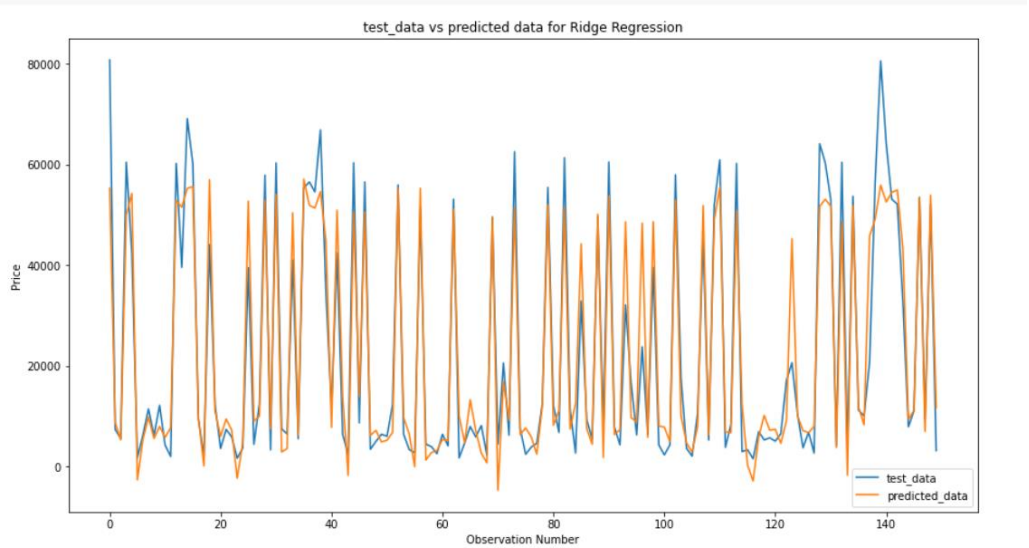
It is a learning model used to analyze any data that suffers from multicollinearity.

For this part we used the same data set which was pre-processed but wrote a Ridge_regression class with init, cost, predict, fit functions. In ridge regression we add an penalty parameter to make bias and variance proportional to each other and it decreases the difference between actual and predicted values.

The following graph is plotted between test values and model prediction values:



The line graph for 150 records for same model is:



R-squared error for linear regression model is:

```
R-squared: 0.9036886712140545
R-squared percentage: 90.36886712140544
```

Ridge regression:

Advantages:

1. It protects against the model getting overfit.
2. Only a small amount of bias occurs, allowing the estimates to be quite accurate approximations of the actual population values.
3. If there is a huge amount of multivariate data with a greater number of predictors (p) than observations, it works well (n).
4. Reduced model complexity.

Disadvantages:

1. It contains every predictor in the finished model.
2. Feature selection cannot be done by it.
3. Coefficients are decreased until they reach zero.
4. Bias is exchanged for variance.
5.

Differences between Linear regression and Ridge Regression:

Linear Regression	Ridge Regression
Linear Regression tries and establishes a relationship between a dependent variable and independent variable by the usage of the best straight fit line	Ridge Regression can be known as a model which estimates the coefficients of various regression models in which the independent variables are highly correlated.
In Linear Regression, model is not punished for any choice of the weights taken. So, if the model assumes that one feature has more importance than the other then more weight can be given to that particular feature.	In this model, weights generally tend to be absolute values which are smaller because model can be punished for the sum squared values of the wights.
Linear regression tries to minimize the sum of squared errors between actual and precedented values of dependent variables.	Ridge regression tries to minimize the sum of squared error plus the penalty term (lambda) between actual and precedented values of the variables.

L2 regularization tends to reduce the coefficients evenly which tends to be more helpful when we are dealing with data which has collinear/codependent features. We used L2 regularization to decrease the chance of over fitting which the performance of regression model is increased.

References:

https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/

https://medium.com/@satyavishnumolakala/linear-regression-pros-cons-62085314aef0

https://www.engati.com/glossary/ridge-regression

https://www.ib-net.org/benchmarking-methodologies/performance-benchmarking/statistical-techniques/

https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/

https://www.mygreatlearning.com/blog/what-is-ridge-regression/#what-is-ridge-regression