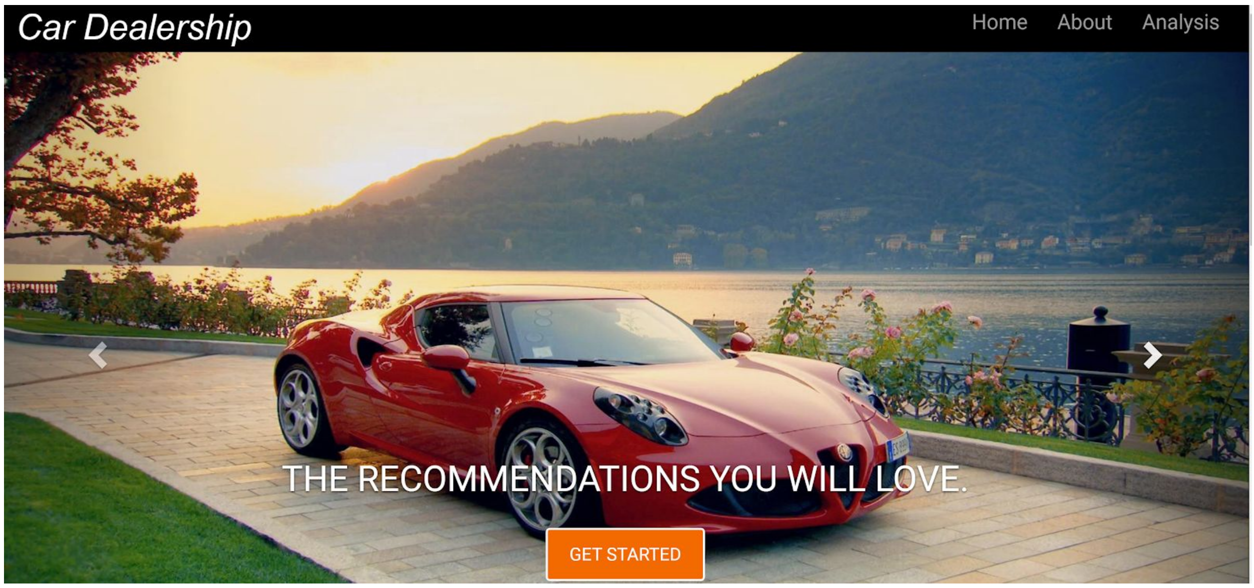
Algorithmic Digital Marketing Final Project - Car Dealership

Who we are?

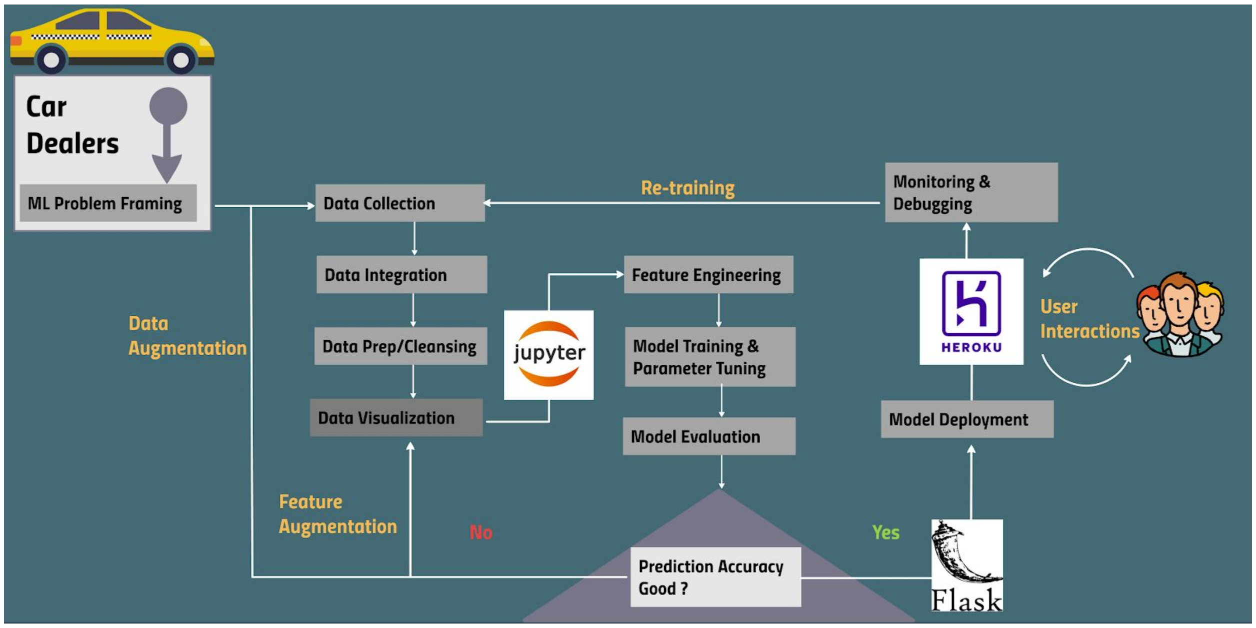
The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes.But due to the increased price of new cars, there is a need for a car price **prediction** system to effectively determine the worthiness of the car using a variety of features. It is important to know their **actual market value** while buying, **similarity** with other models and for us to know the right **segment** of the customer.



We provide:

- THE PRICES YOU WILL LOVE
- THE RECOMMENDATIONS YOU WILL LOVE
- THE TRUST YOUR FAMILY DESERVES

Process Outline



Technologies used :

Language - Python

Platform - Jupyter Notebook

Web Application - Flask

Cloud Service - Heroku

Data Visualization/Analysis - Power BI

Data Sources

We basically have two datasets: Car sales and Customer Data

Car Sales -

https://github.com/KumarAnand11/INFO7374/data/Input/Car_sales.csv

Field	**Business Term**
Manufacturer	Car Manufacturer
Model	Car model
Sales_in_thousands	Total Car sales
__year_resale_value	Year resale value of car
Vehicle_type	Passenger type (7-8 seater) or a 5 seater car
Price_in_thousands	Price of the car
Engine_size	volume of fuel and air that can be pushed through a car's cylinders
Horsepower	Measurement of power in reference to the output of engines or motors
Wheelbase	Distance between the centers of the front and rear wheels
Width	Width of the car
Length	Length of the car
Curb_weight	The total mass of the car
Fuel_capacity	Fuel capacity of the car
Fuel_efficiency	the ability of the car to extract energy from the fuel
Power_perf_factor	Performance of Car
Path	Path of the Images

Customer data - This dataset consists of the following features -

https://github.com/KumarAnand11/INFO7374/data/Input/OnlineRetail.csv

Field	**Business Term**
InvoiceNo	Invoice number for the Purchase
Quantity	Quantity purchased
InvoiceDate	Invoice date of the car purchased
CustomerID	Unique Customer ID
Country	Country of the Customer
Unit_Price	Unit Price of the Car
Model	Car Model

Considering this dataset of the Customers and Car Sales dataset, we have tweaked the columns accordingly and generated a few new columns such as and combined that with them together and made it into a single dataset for our model building system.

Final Data-set: https://github.com/KumarAnand11/INFO7374/data/Output/Main.csv

Personas

1. **Customer or Sales Representatives**

   To check the Customer Category (Gold, Silver, Bronze & Platinum) they belong to and also the price of their desired cars based on the feature inputs.

2. **Car Seller (Dealer)**

   To recommend car models closely similar to the model features as input.

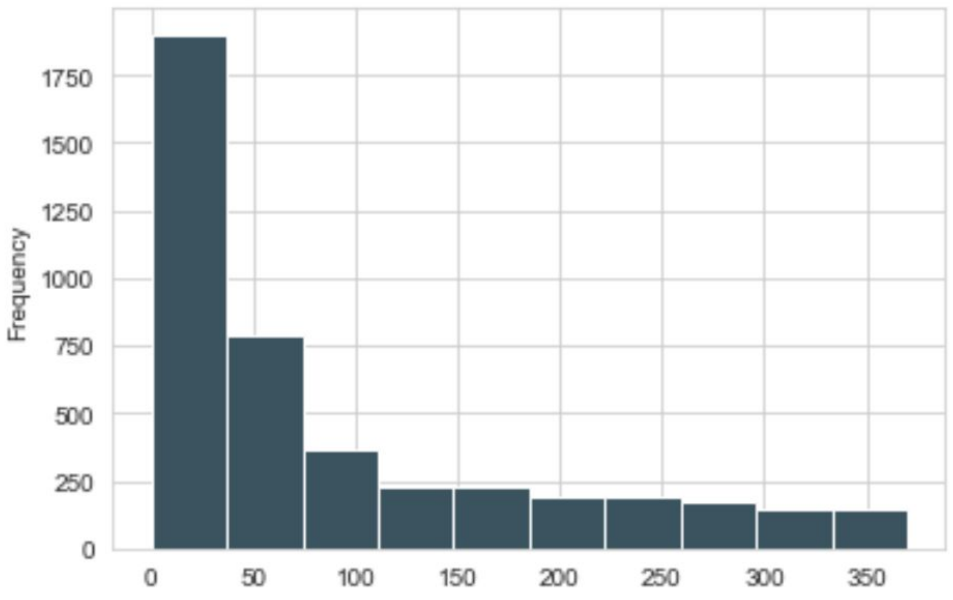
Customer Segmentation

Machine Learning Model Used: *KMeans*

Recency: *To calculate recency, we need to find out the most recent purchase date of each customer and see how many days they are inactive for.*

```
User_Details['Recency'].plot.hist( alpha=1,bins=10,)
```

```
: <matplotlib.axes._subplots.AxesSubplot at 0x17da48a85c8>
```



```
User_Details.groupby('RecencyCluster')['RecencyCluster'].describe()
```

RecencyCluster	count	mean	std	min	25%	50%	75%	max
0	513.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	620.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0
2	955.0	2.0	0.0	2.0	2.0	2.0	2.0	2.0
3	2269.0	3.0	0.0	3.0	3.0	3.0	3.0	3.0

Frequency: *To create frequency clusters, we need to find total number orders for each customer.*

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(User_Details[['Frequency']])
User_Details['FrequencyCluster'] = kmeans.predict(User_Details[['Frequency']])

User_Details = order_cluster('FrequencyCluster', 'Frequency',User_Details,True)

User_Details.groupby('FrequencyCluster')['Frequency'].describe()
```

	count	mean	std	min	25%	50%	75%	max
FrequencyCluster								
0	3859.0	49.628401	44.777803	1.0	15.00	33.0	72.50	188.0
1	470.0	326.961702	131.182122	189.0	227.25	284.5	392.00	803.0
2	24.0	1327.708333	489.773193	872.0	980.75	1114.5	1538.25	2681.0
3	4.0	5750.500000	1356.713308	4543.0	4981.75	5396.5	6165.25	7666.0

Monetary Value: *Let's see how our customer database looks like when we cluster them based on revenue. We will calculate revenue for each customer.*

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(User_Details[['Revenue']])
User_Details['RevenueCluster'] = kmeans.predict(User_Details[['Revenue']])

User_Details = order_cluster('RevenueCluster', 'Revenue',User_Details,True)

User_Details.groupby('RevenueCluster')['Revenue'].describe()
```

	count	mean	std	min	25%	50%	75%	max
RevenueCluster								
0	3975.0	1.204349e+04	11781.230115	13.20	3.367750e+03	7779.100	1.689160e+04	51300.98
1	369.0	9.063492e+04	43513.483799	51630.11	5.961853e+04	75666.150	1.052573e+05	309951.61
2	11.0	5.386230e+05	151919.449858	378996.20	4.434713e+05	521081.580	5.491859e+05	908855.05
3	2.0	1.419319e+06	320089.555103	1192981.65	1.306150e+06	1419319.145	1.532488e+06	1645656.64

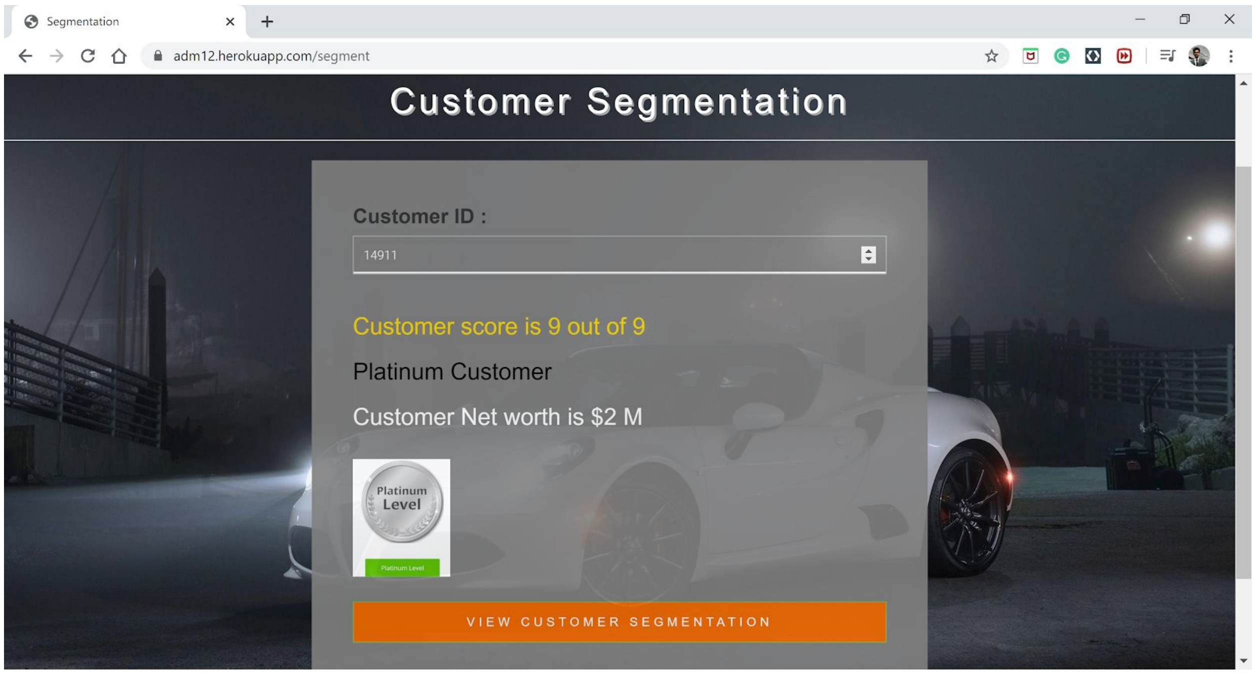
```
User_Details['OverallScore'] = User_Details['RecencyCluster'] + User_Details['FrequencyCluster'] + User_Details['RevenueClust
```

```
User_Details['Revenue'] = User_Details['Revenue'].apply(lambda x: millify(x))
```

```
User_Details
```

	CustomerID	Recency	RecencyCluster	Frequency	FrequencyCluster	Revenue	RevenueCluster	OverallScore
0	17850	299	0	312	1	$48 K	0	1
1	15808	303	0	210	1	$38 K	0	1
2	13448	13	3	199	1	$44 K	0	4
3	14388	5	3	191	1	$50 K	0	4
4	14449	15	3	258	1	$21 K	0	4
...	...	...	...	...	...	...	...	...
4352	13089	1	3	1855	2	$717 K	2	7
4353	13081	8	3	1059	2	$443 K	2	7
4354	14156	6	3	1420	2	$521 K	2	7
4355	14911	1	3	5665	3	$2 M	3	9
4356	14646	13	3	1999	2	$1 M	3	8

4357 rows × 8 columns



Customer Segmentation

Customer ID : 14911

Customer score is 9 out of 9

Platinum Customer

Customer Net worth is $2 M

Pricing Strategy

Machine Learning Model Used: Linear Regression

Linear Regression Equation:

Unit_Price = 0.7606*Year_Resale_Value + 0.2079*Horsepower + 0.2731*Curb_weight + (-0.0822)*Fuel_capacity + 0.0872*Fuel_efficiency + (-0.0357)*Width + 0.0047*Length

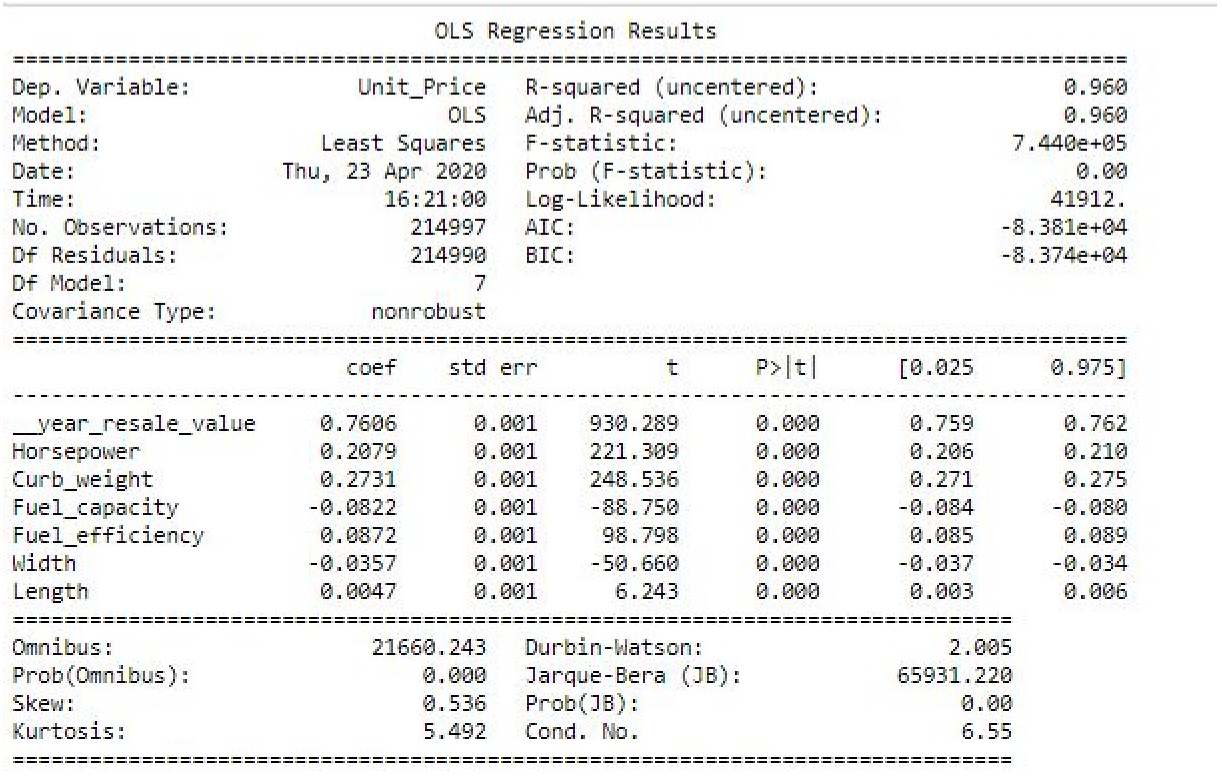
Using this model we will predict the prices of the car based on certain input parameters.

Our dataset consists of total 405217 rows and 24 columns:

We clean the dataset and retrieve only the columns which are highly correlated with our target variable "Unit_Price". After doing so we use only the important features to predict our target using Linear Regression.

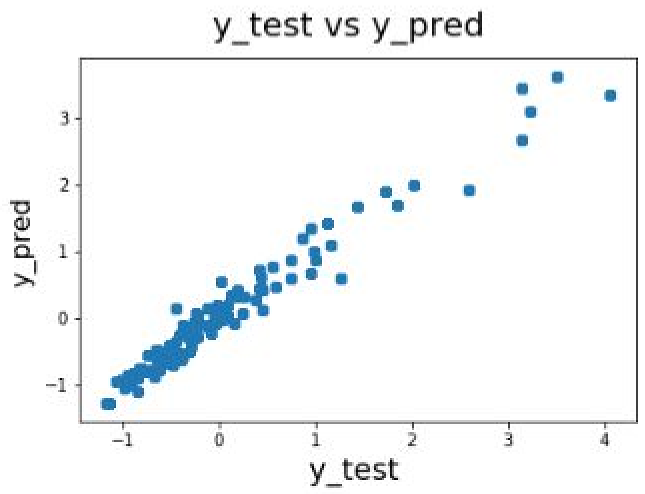
We have used Adjusted R-Square as our metric to evaluate and choose the best model for our Linear Regression. The higher the Adj. R-Square the better the model is.

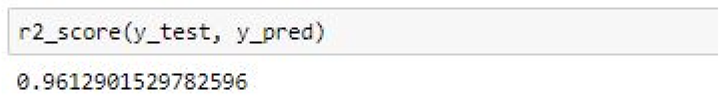
Eg.



```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:             Unit_Price   R-squared (uncentered):                   0.960
Model:                            OLS   Adj. R-squared (uncentered):              0.960
Method:                 Least Squares   F-statistic:                          7.440e+05
Date:                Thu, 23 Apr 2020   Prob (F-statistic):                        0.00
Time:                        16:21:00   Log-Likelihood:                          41912.
No. Observations:              214997   AIC:                                 -8.381e+04
Df Residuals:                  214990   BIC:                                 -8.374e+04
Df Model:                           7
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
__year_resale_value     0.7606      0.001    930.289      0.000       0.759       0.762
Horsepower              0.2079      0.001    221.309      0.000       0.206       0.210
Curb_weight             0.2731      0.001    248.536      0.000       0.271       0.275
Fuel_capacity          -0.0822      0.001    -88.750      0.000      -0.084      -0.080
Fuel_efficiency         0.0872      0.001     98.798      0.000       0.085       0.089
Width                  -0.0357      0.001    -50.660      0.000      -0.037      -0.034
Length                  0.0047      0.001      6.243      0.000       0.003       0.006
==============================================================================
Omnibus:                    21660.243   Durbin-Watson:                   2.005
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            65931.220
Skew:                           0.536   Prob(JB):                         0.00
Kurtosis:                       5.492   Cond. No.                         6.55
==============================================================================
```

For evaluation purposes, we test this model against the test data to evaluate the model's Adj. R-Square value on test data.





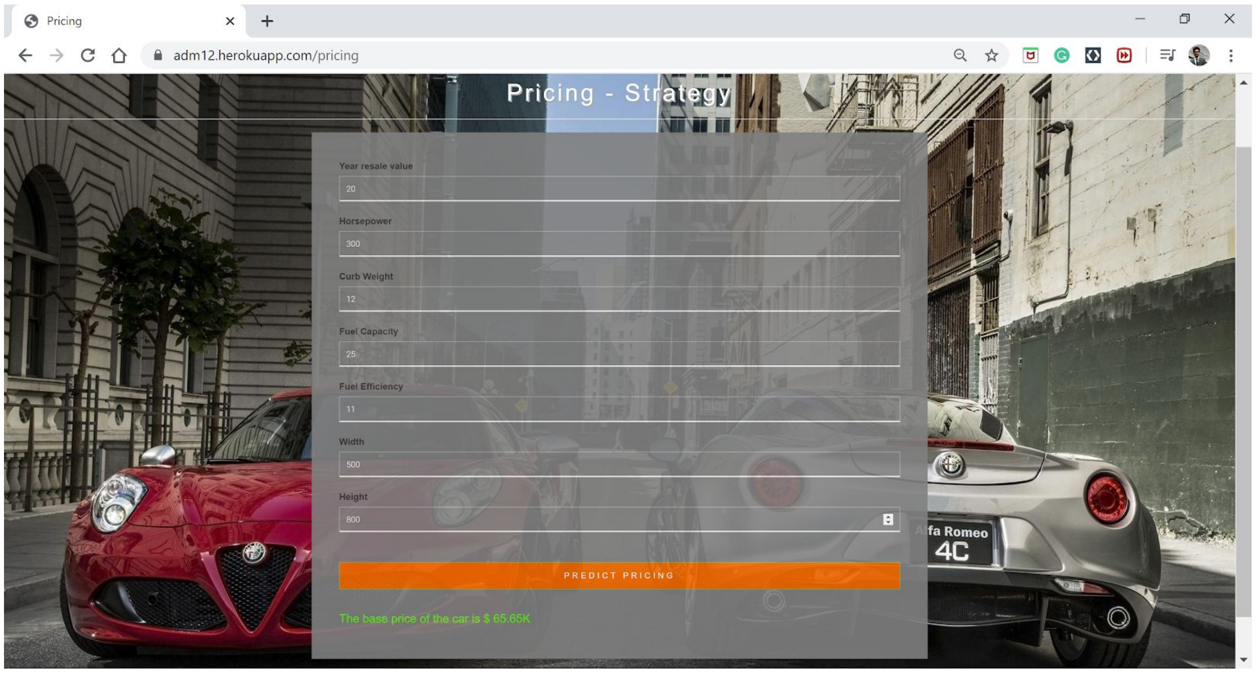
```
r2_score(y_test, y_pred)
```

```
0.9612901529782596
```

The model performed well on both the training data and test data, hence we can deploy it. It doesn't underfit or overfit the data.

After we get our price based on their desired features, we have designed our Pricing Strategy to give our customers the best price by grouping them into different categories (**Gold, Silver, Bronze, Platinum**). Based on the customer Segmentation we are offering them discounts accordingly. For our New customers on their first purchase the base price doesn't change, and for the Silver category of customers there will be a **discount of 10%** and for our **Gold customers - 20%** and the most loyal customers - Platinum category we are offering them a 30% discount on their purchase to retain those customers in business.

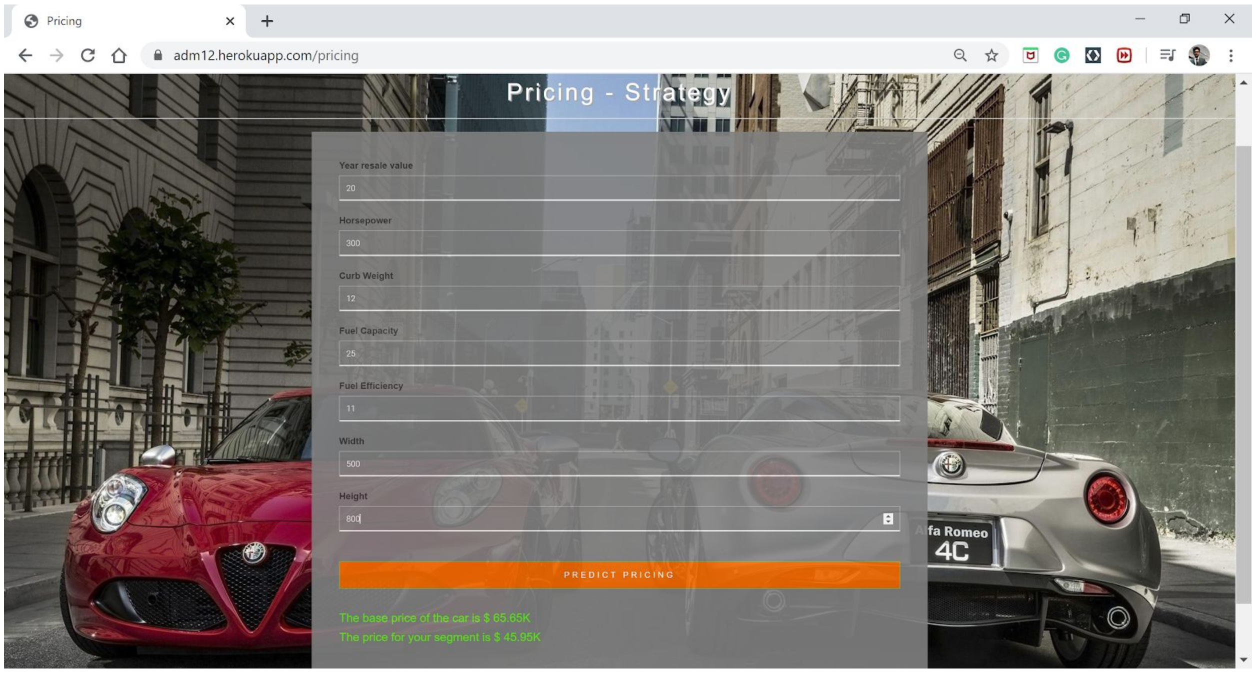
New User (Bronze):



Year resale value: 20
Horsepower: 300
Curb Weight: 12
Fuel Capacity: 25
Fuel Efficiency: 11
Width: 500
Height: 800

The base price of the car is $ 65.65K

Existing User (Gold):



Year resale value: 20
Horsepower: 300
Curb Weight: 12
Fuel Capacity: 25
Fuel Efficiency: 11
Width: 500
Height: 800

The base price of the car is $ 65.65K
The price for your segment is $ 45.95K

Recommendation

For building the recommendation model we will be using a **Content based**

Recommendation model to understand the similarities between different models of either the **same or different** manufacturer.

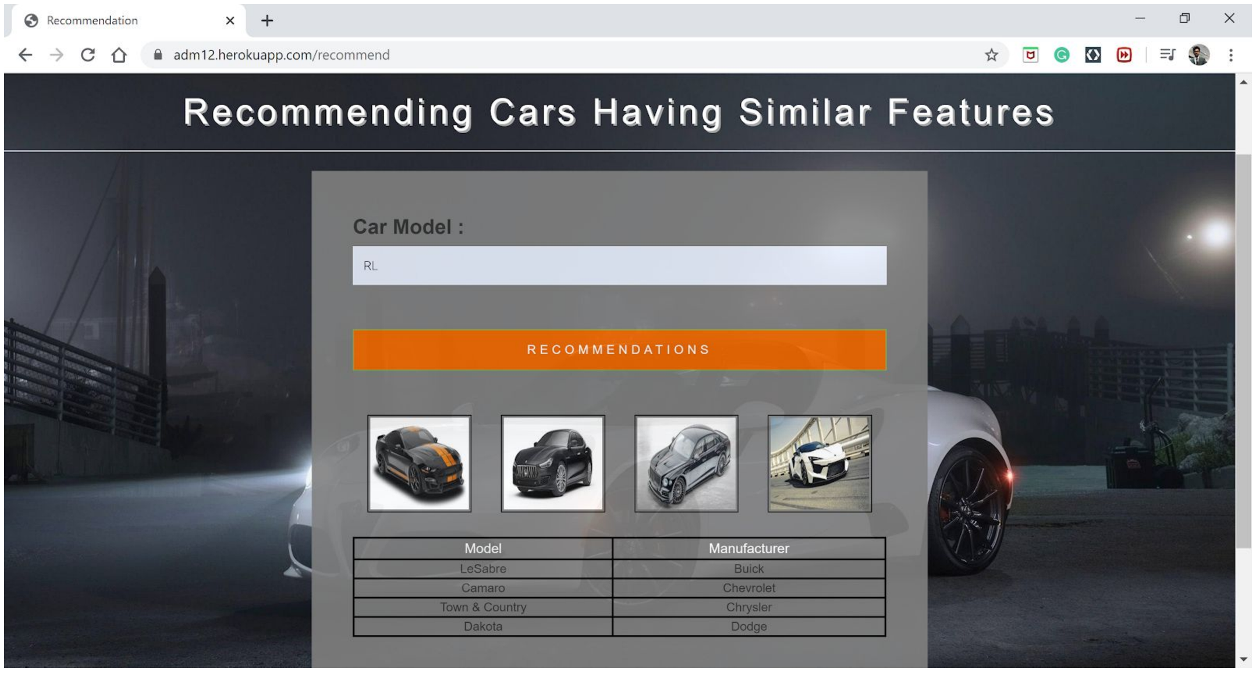
```
car_sales[['Manufacturer','Model','features']]
```

	Manufacturer	Model	features
0	Acura	Integra	gas std two hatchback forward front cylinder-f...
1	Acura	TL	gas std four hatchback backward front cylinder...
2	Acura	CL	gas turbo two sedan forward back cylinder-six ...
3	Acura	RL	diesel std four hatchback forward front cylind...
4	Audi	A4	diesel turbo four sedan forward back cylinder-...
...	...	...	...
149	Volvo	V40	gas turbo two sedan forward back cylinder-six ...
150	Volvo	S70	diesel std four hatchback forward front cylind...
151	Volvo	V70	diesel turbo four sedan forward back cylinder-...
152	Volvo	C70	diesel std four hatchback forward front cylind...
153	Volvo	S80	diesel turbo four sedan forward back cylinder-...

```python
# Function that takes in car model as input and outputs most similar car models
def get_recommendations(Model, cosine_sim=cosine_sim):
    # Get the index of the car models that matches the model
    idx = indices[Model]

    # Get the pairwsie similarity scores of all car models with that car model
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Sort the car models based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Get the scores of the 10 most similar car models
    sim_scores = sim_scores[1:11]

    # Get the car model indices
    car_indices = [i[0] for i in sim_scores]

    # Return the top 10 most similar car models
    return car_sales[['Model','Manufacturer','Path']].iloc[car_indices]
```

```python
get_recommendations('RL')
```

	Model	Manufacturer	Path
13	LeSabre	Buick	14.jpg
23	Camaro	Chevrolet	6.jpg
33	Town & Country	Chrysler	16.jpg
43	Dakota	Dodge	8.jpg
53	Windstar	Ford	18.jpg
63	Elantra	Hyundai	10.jpg
73	LS400	Lexus	2.jpg
83	3000GT	Mitsubishi	12.jpg
93	E-Class	Mercedes-B	4.jpg

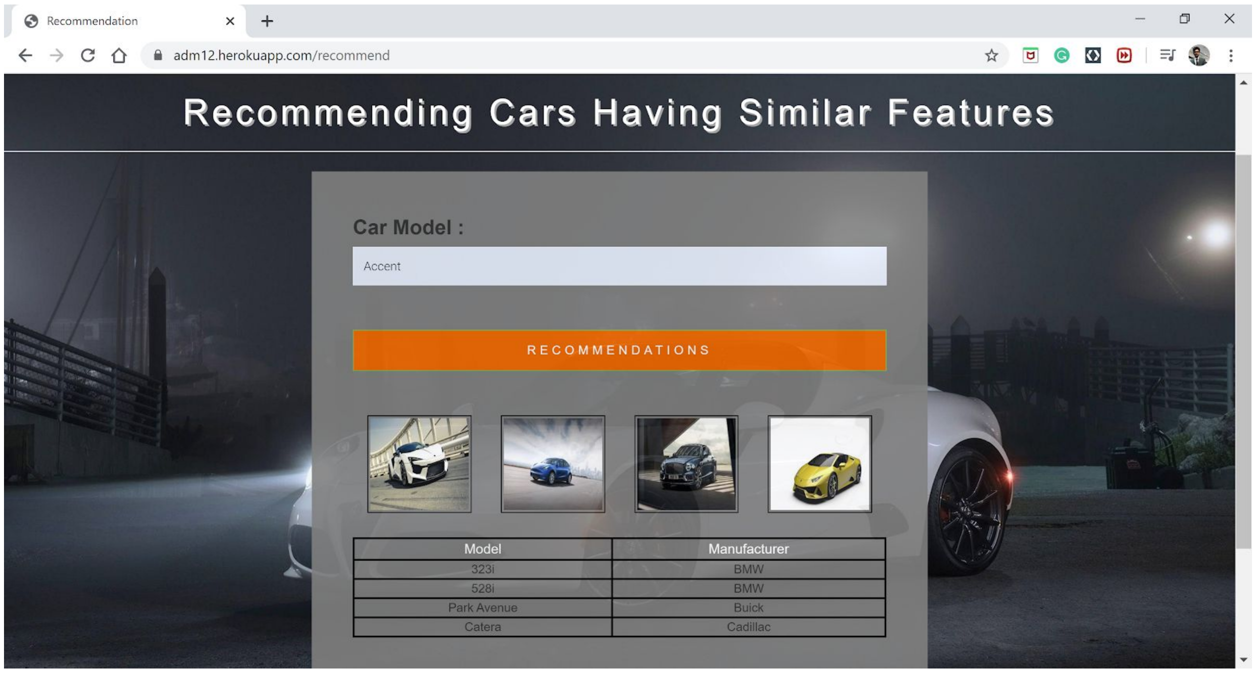


Recommending Cars Having Similar Features

Car Model :

RL

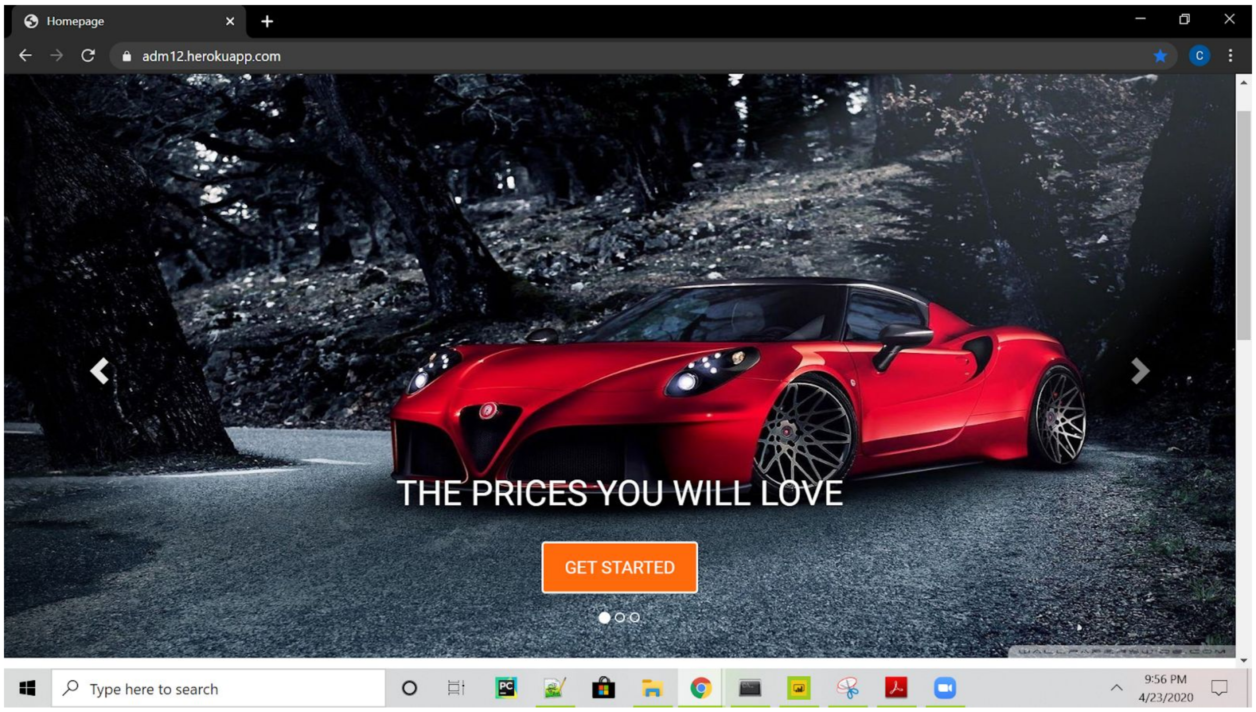
Model	Manufacturer
LeSabre	Buick
Camaro	Chevrolet
Town & Country	Chrysler
Dakota	Dodge


Recommendation
adm12.herokuapp.com/recommend
Recommending Cars Having Similar Features
Car Model :
Accent
RECOMMENDATIONS

Model	Manufacturer
323i	BMW
528i	BMW
Park Avenue	Buick
Catera	Cadillac

Flask App & Deployment

We have built a web application using Flask, pushing that into Git through which we are deploying it into Heroku, which is a platform as a service (PaaS) which can be used to run applications fully in the cloud.



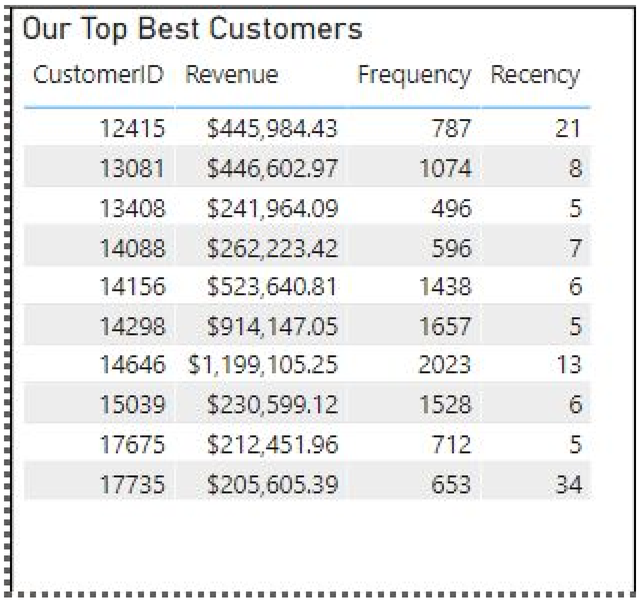
The Flask app consists of the templates folder that has both the html files and css files required to build the front end of the application.

The file structure in the Github consists of **car_dealers.py** - which is the main file of our entire package required to run the app, **requirements.txt** consists of all the necessary required libraries and dependencies to run the application and **Procfile** consists of the command to be executed to launch the app. Using these above mentioned files we are able to successfully deploy the app on **Heroku**.

Analysis

Power BI analysis -

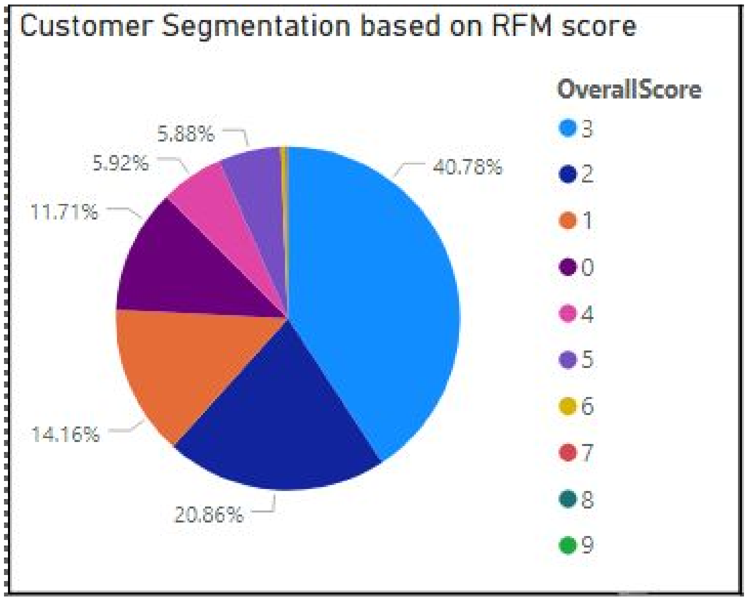
Based on the Recency, Frequency and Monetary values we are predicting the Top best Customers of our Car Dealership business.



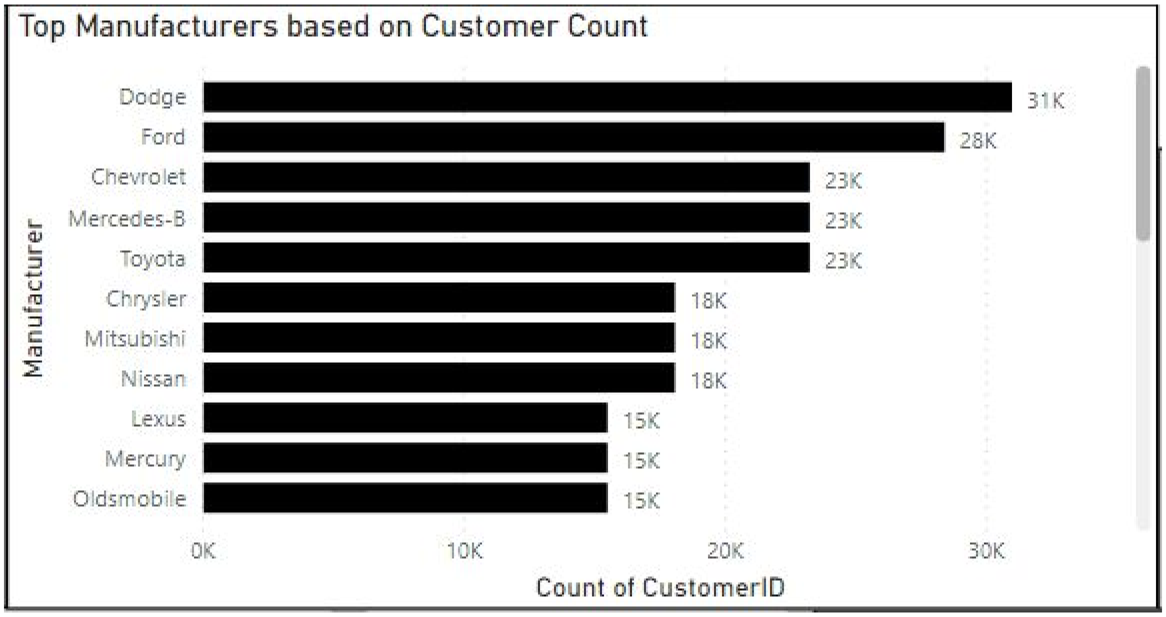
Our Top Best Customers

CustomerID	Revenue	Frequency	Recency
12415	$445,984.43	787	21
13081	$446,602.97	1074	8
13408	$241,964.09	496	5
14088	$262,223.42	596	7
14156	$523,640.81	1438	6
14298	$914,147.05	1657	5
14646	$1,199,105.25	2023	13
15039	$230,599.12	1528	6
17675	$212,451.96	712	5
17735	$205,605.39	653	34

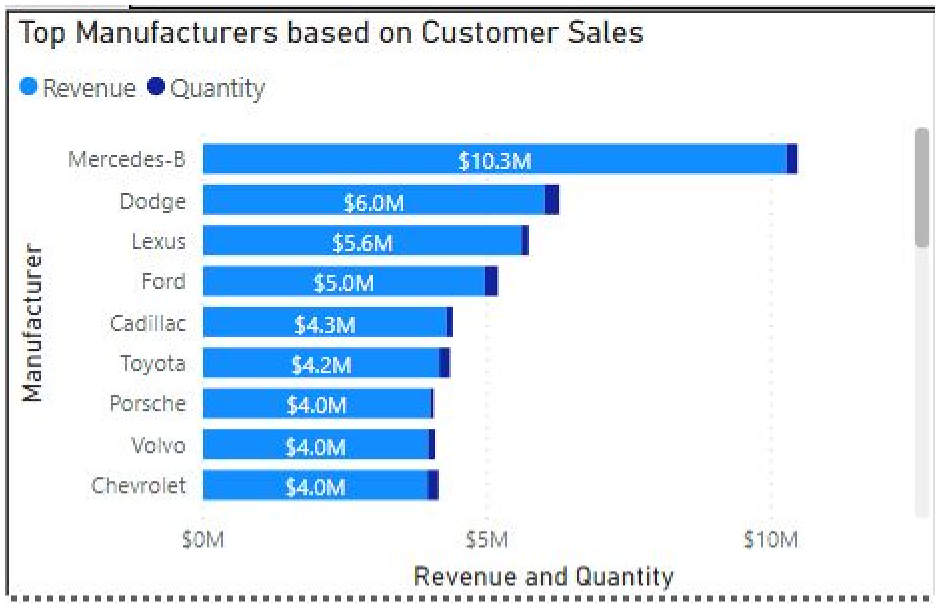
Customers have been grouped into Bronze, Silver, Gold and Platinum categories based on the RFM scores. All the new customers fall under Bronze category and will not be provided with any kind of deals as such, all the customers with RFM score 0-3 will fall under Silver category and will be given a discount of 10% on the base price of the car. The customers with scores from 4-6 will fall under Gold Category and will be benefited with 20% discount and then comes the Platinum customers with the highest discount of 30% on the base price of the car. Here from the below plot we see that about 85% of the customers fall under Silver Category, 13% under Gold category and the remaining 2% under Platinum category.



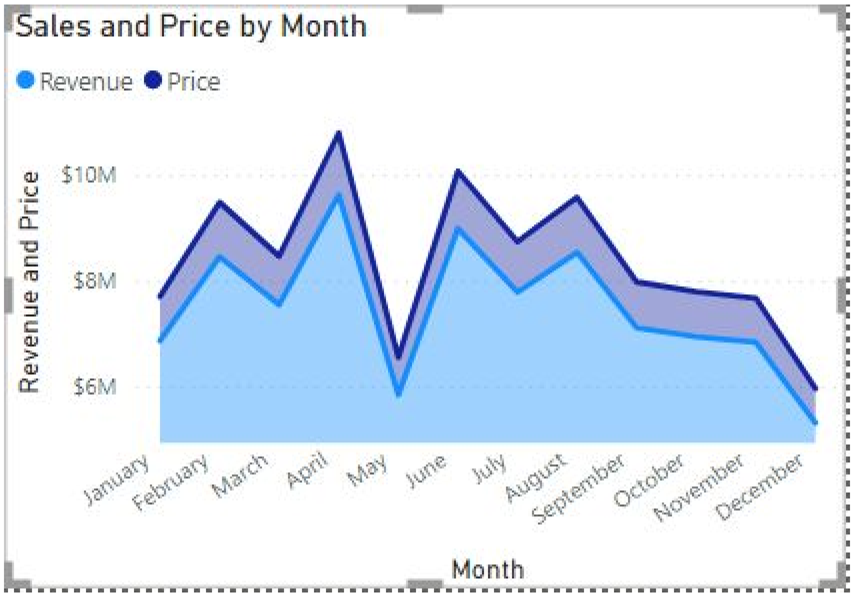
Top manufacturers of Car Dealership business based on the Number of Customers they have acquired from the best prices and the desirable features of the car as per the customers. Dodge has the most number of customers than any of the manufacturers - 31000, followed by Ford with 28000 customers and Chevrolet, Mercedes Benz and Toyota with 23000 customers across the world for the year 2011.



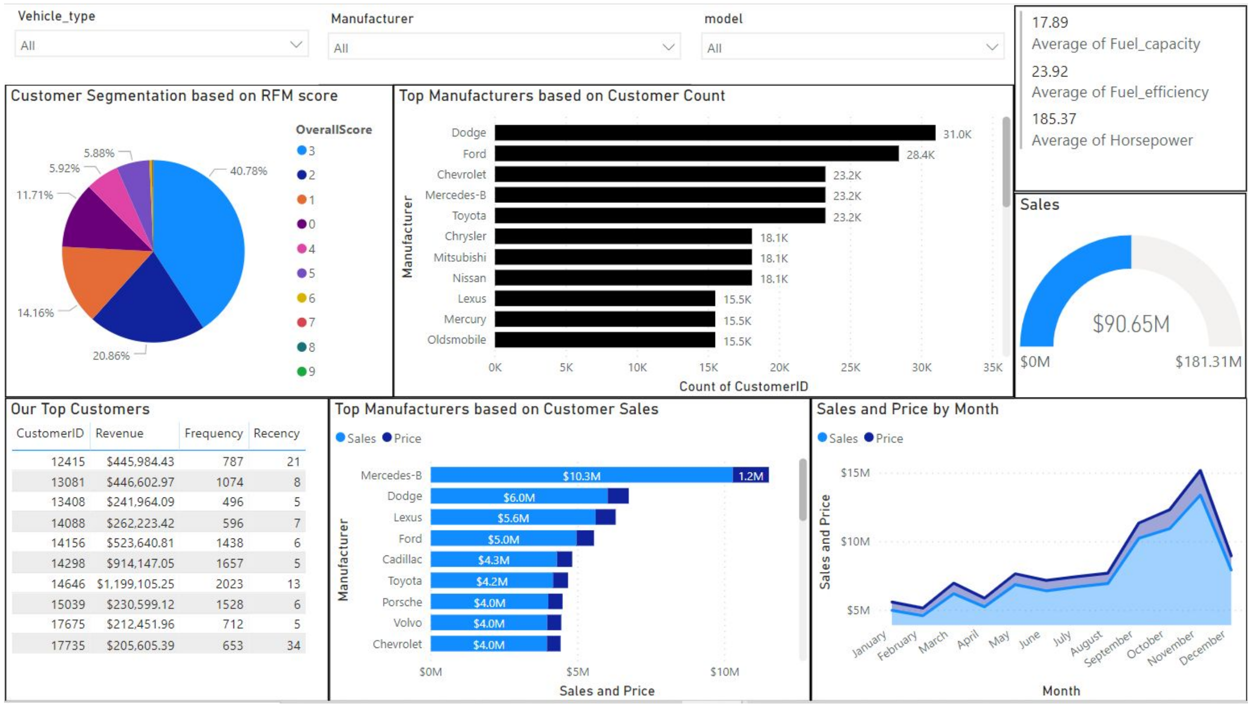
Top Manufacturers based on Customer sales (Revenue), Mercedes-Benz made a revenue of 10.3M after selling 194330 cars followed by Dodge - which has sold a total of 257714 cars that made a total revenue of 6M meaning the price of the Dodge cars is comparatively cheaper with desirable features, with a total customer count of 31,000 customers on the whole. Hence, Dodge is the most sold and best Manufacturer according to the Car Dealership Business.



Plotting Revenue by Month for the year 2011.



Dashboard -



Our Top Customers

CustomerID	Revenue	Frequency	Recency
12415	$445,984.43	787	21
13081	$446,602.97	1074	8
13408	$241,964.09	496	5
14088	$262,223.42	596	7
14156	$523,640.81	1438	6
14298	$914,147.05	1657	5
14646	$1,199,105.25	2023	13
15039	$230,599.12	1528	6
17675	$212,451.96	712	5
17735	$205,605.39	653	34

References

- https://towardsdatascience.com/customer-segmentation-with-machine-learning-a0ac8c3d4d84

- https://www.kaggle.com/gagandeep16/car-sales
- https://www.youtube.com/watch?v=UbCWoMf80PY
- https://www.kaggle.com/hellbuoy/car-price-prediction
- https://realpython.com/flask-by-example-part-1-project-setup/
- https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world
- https://www.nathanwyand.com/2019/09/25/building-a-food-price-comparison-app-with-flask-leaflet-and-sqlalchemy/
- https://www.youtube.com/watch?v=acfKAcqaT2w
- https://towardsdatascience.com/quickly-build-and-deploy-an-application-with-streamlit-988ca08c7e83
- https://towardsdatascience.com/data-driven-growth-with-python-part-2-customer-segmentation-5c019d150444
- https://www.analyticsvidhya.com/blog/2017/09/machine-learning-models-as-apis-using-flask/
- https://towardsdatascience.com/predicting-used-car-prices-with-machine-learning-techniques-8a9d8313952