Project: Analyzing the Impact of Car Features on Price and Profitability

Project Description:

The automotive industry has been rapidly evolving with a growing focus on fuel efficiency, environmental sustainability, and technological innovation. With increasing competition among manufacturers and a changing consumer landscape, it has become more important than ever to understand the factors that drive consumer demand for cars. This project aimed to analyze the relationship between a car's features, market category, and pricing, and identify which features and categories are most popular among consumers and most profitable for the manufacturer. The goal was to develop a pricing strategy that balances consumer demand with profitability and identifies which product features to focus on in future product development efforts.

Approach:

Data:

The dataset used in this project contained information on over 11,000 car models and their specifications, including details on the car's make, model, year, fuel type, engine power, transmission, wheels, number of doors, market category, size, style, estimated miles per gallon, popularity, and manufacturer's suggested retail price (MSRP). The dataset was cleaned and prepared before analysis to ensure accurate and reliable results.

Data Cleaning: Python and its libraries, such as Pandas, NumPy, Matplotlib, Seaborn, and YData Profiling, were used for the data cleaning process inside Jupyter Notebook.

Steps Performed to clean the data are:

Steps performed:

step1: importing necessary libraries

step2: loading csv file

step3: analyzing columns

step4: cleaning the DataFrame

1)droping duplicate rows

2)check dtypes

3)check null values

4)fill null values

step5: Save file after data cleaning

Detailed Process:

Step 1:

1.Importing necessary libaries

```
]: import pandas as pd
   import numpy as np
   import seaborn as sns
   import matplotlib.pyplot as plt
   %matplotlib inline
```

Step 2:

2.Loading data to dataframe

```
df=pd.read_csv("Car_data.csv",index_col=0)
df.head()
```

Step 3:

3.Analying Dataframe by profile_report using y_data profiling

```
import ydata_profiling as yp
```

```
pf=yp.ProfileReport(df,title="Car_data_Analysis_Report")
```

```
pf.to_notebook_iframe()
```

This will generate complete profile report with analysis of all column to proceed further.

Step 4:

1)

```
: df.drop_duplicates(inplace=True)
```

2)

4.2 check dtypes

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11199 entries, BMW to Lincoln
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Model              11199 non-null  object 
 1   Year               11199 non-null  int64  
 2   Engine Fuel Type   11196 non-null  object 
 3   Engine HP          11130 non-null  float64
 4   Engine Cylinders   11169 non-null  float64
 5   Transmission Type  11199 non-null  object 
 6   Driven_Wheels      11199 non-null  object 
 7   Number of Doors    11193 non-null  float64
 8   Market Category    7823 non-null   object 
 9   Vehicle Size       11199 non-null  object 
 10  Vehicle Style      11199 non-null  object 
 11  highway MPG        11199 non-null  int64  
 12  city mpg           11199 non-null  int64  
 13  Popularity         11199 non-null  int64  
 14  MSRP               11199 non-null  int64  
dtypes: float64(3), int64(5), object(7)
memory usage: 1.4+ MB
```

Here all datatypes are correct .so we don’t need to change any.

3) Below is the null value percentage shown in each column.

4.3 check null value %

```
round(df.isna().sum().sort_values(ascending=False)/len(df)*100,2)
```

```
Market Category      30.15
Engine HP             0.62
Engine Cylinders      0.27
Number of Doors       0.05
Engine Fuel Type      0.03
Model                 0.00
Year                  0.00
Transmission Type     0.00
Driven_Wheels         0.00
Vehicle Size          0.00
Vehicle Style         0.00
highway MPG           0.00
city mpg              0.00
Popularity            0.00
MSRP                  0.00
dtype: float64
```

There are only 5 columns that have null values. We dig inside those columns to fill those Na values present.

4)

Column:Market Category

```
df["Market Category"].nunique()
```

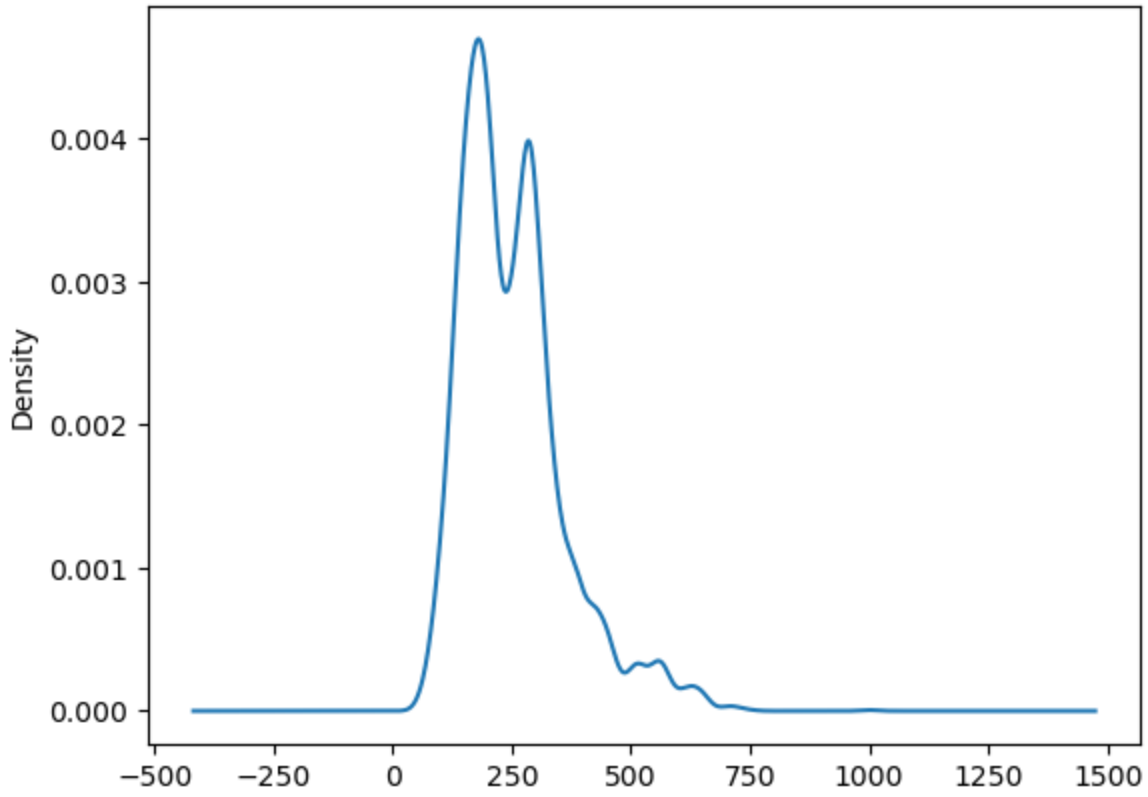
```
71
```

```
df["Market Category"].replace(np.NAN,"Other",inplace=True)
```

```
df["Market Category"].value_counts().plot(kind="bar",figsize=(15,5))
```

column: Engine HP

```
: df['Engine HP'].plot.density()
```

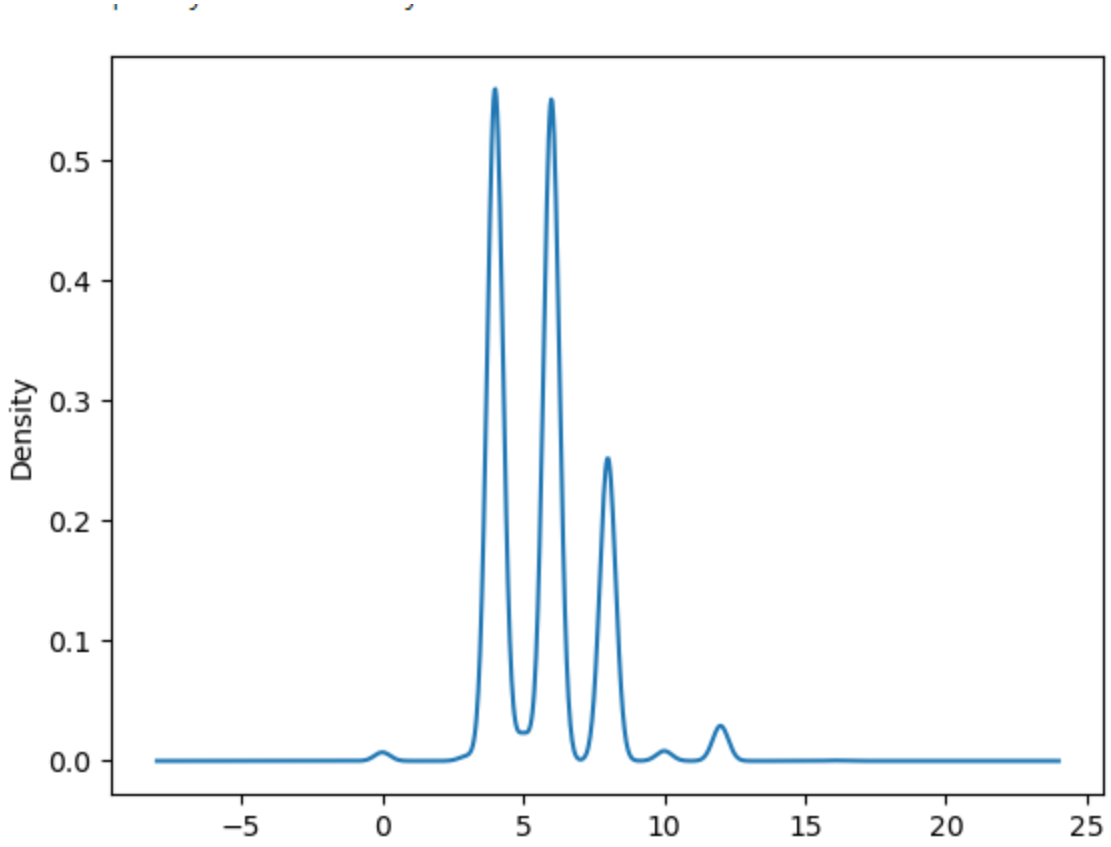


With that density chart .I figure that most of the data is centered. So, it Is better to use median to fill Na values

```
df["Engine HP"].fillna(df["Engine HP"].median(),inplace=True)
```

column:Engine Cylinders

```
df["Engine Cylinders"].plot.density()
```



Engine Cylinders must be a natural number. Using Mean and Median does not make a sense here. Mode is best way to go .

```
df["Engine Cylinders"].fillna(df["Engine Cylinders"].mode().iloc[0],inplace=True)
```

```
df["Engine Cylinders"].mode().iloc[0]
```

```
4.0
```

column: Number of Doors

Similar to Engine Cylinders here also Na values are fill using mode.

```
df["Number of Doors"].fillna(df["Number of Doors"].mode().iloc[0],inplace=True)
```

```
df["Number of Doors"].mode().iloc[0]
```

```
4.0
```

column: Engine Fuel Type

Here we just need to replace Na values to ‘Other’. That’s it.

```
df["Engine Fuel Type"].replace(np.NAN,"Other",inplace=True)
```

```
df["Engine Fuel Type"].value_counts()
```

```
regular unleaded                                 6658
premium unleaded (required)                      1956
premium unleaded (recommended)                   1392
flex-fuel (unleaded/E85)                          887
diesel                                            150
electric                                           66
flex-fuel (premium unleaded required/E85)          53
flex-fuel (premium unleaded recommended/E85)       26
flex-fuel (unleaded/natural gas)                    6
Other                                               3
natural gas                                         2
Name: Engine Fuel Type, dtype: int64
```

Finally,

```
round(df.isna().sum().sort_values(ascending=False)/len(df)*100,2)
```

```
Model                0.0
Year                 0.0
Engine Fuel Type     0.0
Engine HP            0.0
Engine Cylinders     0.0
Transmission Type    0.0
Driven_Wheels        0.0
Number of Doors      0.0
Market Category      0.0
Vehicle Size         0.0
Vehicle Style        0.0
highway MPG          0.0
city mpg             0.0
Popularity           0.0
MSRP                 0.0
dtype: float64
```

There is no null values remains .Thus , df is perfectly cleaned now.

Step 5:

5.Saving File after cleaning

```
df.to_csv("Cleaned_Car_data.csv",index=False)
```

Dashboard is build using this cleaned_Car_data.csv.

Building Dashboard:

First we load the Cleaned_Car_data.csv file in Tableau
Then we create following Worksheets:
Task1: For Model Building I created a parameter to filter the model for each feature and then create a calculated field Car_Features.

CASE [Features/Pricing]
WHEN 'City Mpg' THEN [City Mpg]
WHEN 'Engine Cylinders' THEN [Engine Cylinders]
WHEN 'Engine HP' THEN [Engine HP]
WHEN 'highway MPG' THEN [highway MPG]
WHEN 'Msrp' THEN [Msrp]
WHEN 'Number of Doors' THEN[Number of Doors]
WHEN 'Popularity' THEN [Popularity]
END
Task2 Pivot Table: For this another Calculated field is created named Market Category Segment to Segment Each category .
Task 2 Bar Chart: Same Market Category Segment is used to create Bar Chart.
Task 3 :We use same Parameter Features/Pricing to show how different features affect the price.
Task 4:Here we Created a calculated field named Profitabilty
As Profitabilty= [Msrp]*[Popularity] and use this to build the Regression Model and filter the Model using feature(model).

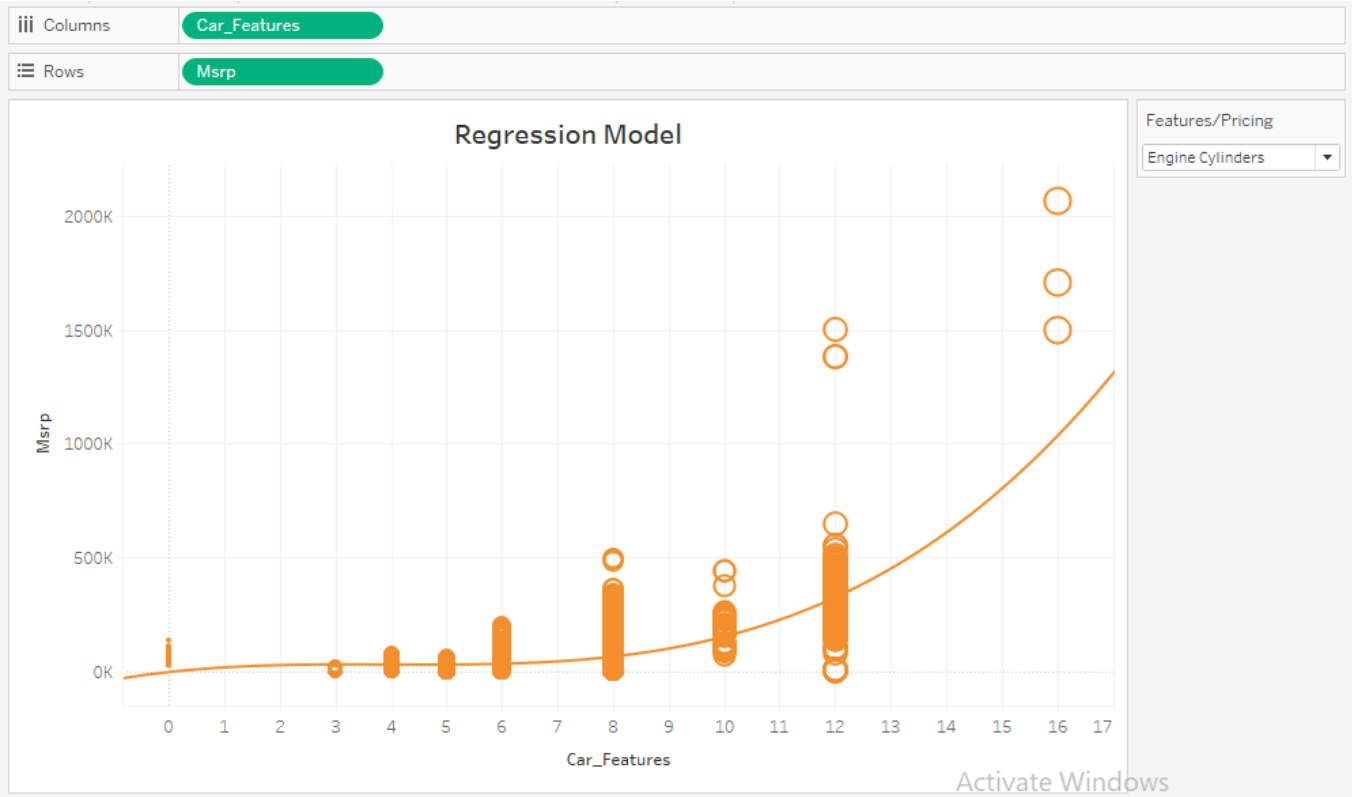
Task 5:Year is changed to date type measure. Then we use Parameter Feature/Pricing to show the trend of car_features and price with line chart.
After that all the worksheets are combined to Build Dashboard.

Tech-Stack Used: The tech-stack used in this project includes Jupyter Notebook and Tableau. Python and its libraries, such as Pandas, NumPy, Matplotlib, Seaborn, and YData Profiling, were used for the data cleaning process inside Jupyter Notebook. The cleaned data was then used to build the dashboard using Tableau Desktop Professional Edition.

Insight:

Insight Required: What is the relationship between a car's features and its price?

- **Task 1:**



Trend Lines Model

A polynomial trend model of degree 3 is computed for Msrp given Car_Features. The model may be significant at p <= 0.05.

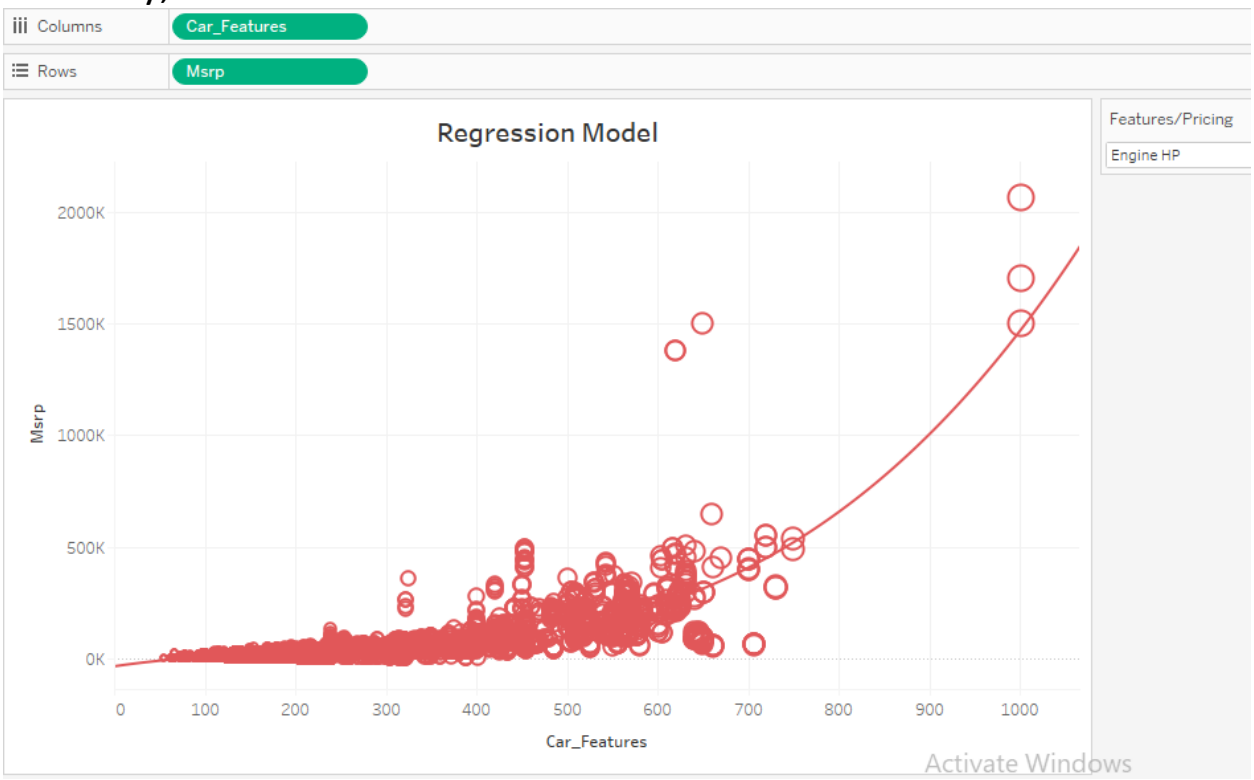
Model formula: Features/Pricing*(Car_Features^3 + Car_Features^2 + Car_Features + intercept)

For Engine Cylinders:

Msrp = 583.318*Car_Features^3 + -6953.02*Car_Features^2 + 26788*Car_Features + -5440.43

R-Squared:0.56

Similarily,



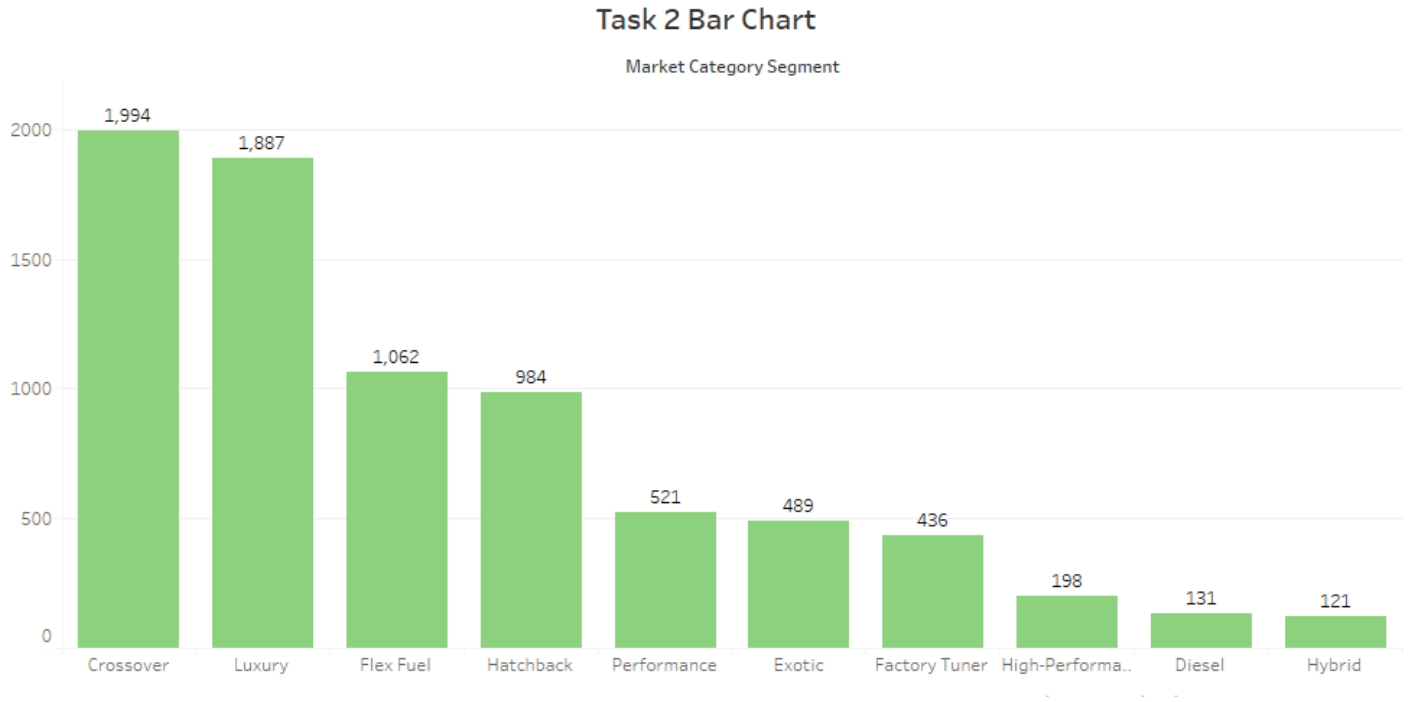
We car filter the model using Feature/Pricing and can obtain relationship b/w all features w.r.t Msrp. Most of the features shows positive relationship with the Mrsp.

Insight Required: Which market categories are most popular among different groups of consumers?

- **Task 2:** Use a pivot table to segment the market based on consumer preferences and demographics, and display the results in a bar chart that shows the distribution of market categories among different segments.

Pivot Table Market Segment

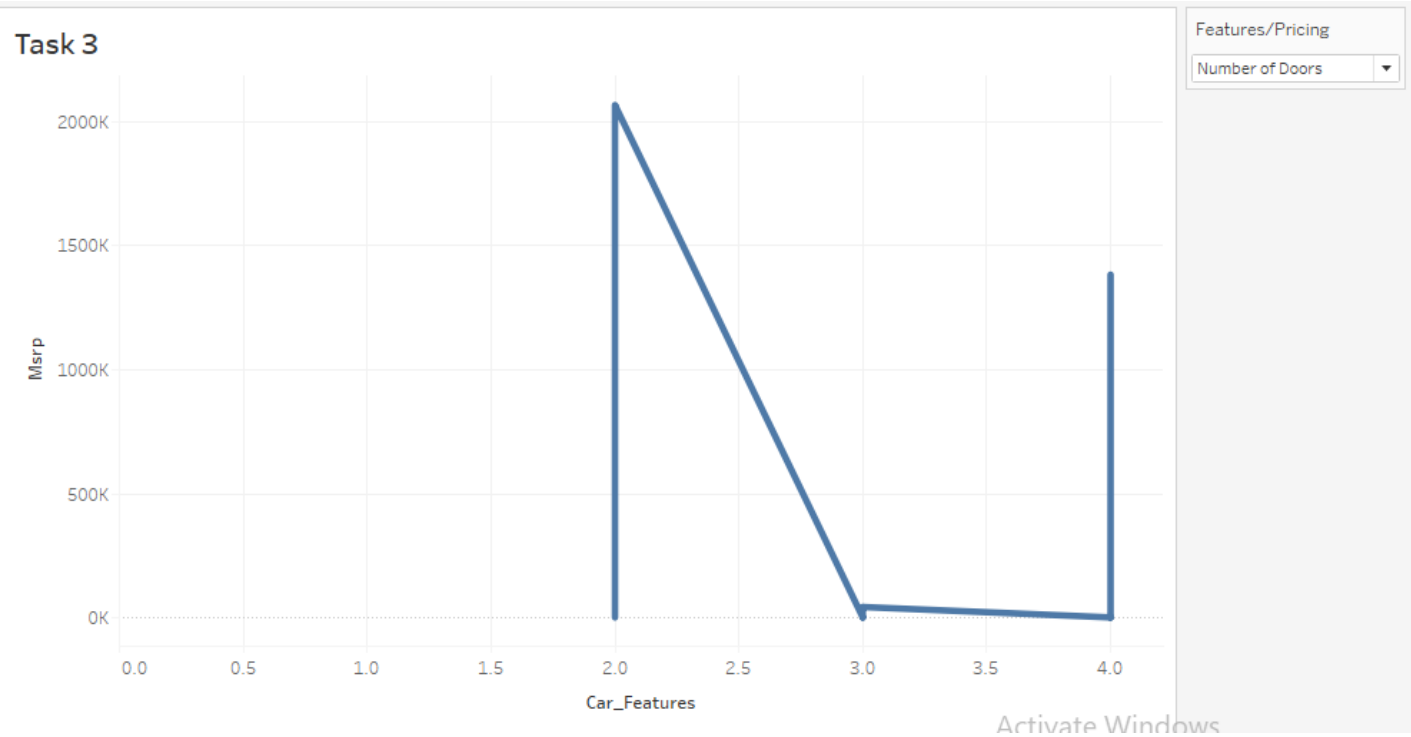
Market Category S..	
Crossover	1,994
Diesel	131
Exotic	489
Factory Tuner	436
Flex Fuel	1,062
Hatchback	984
High-Performance	198
Hybrid	121
Luxury	1,887
Other	3,376
Performance	521



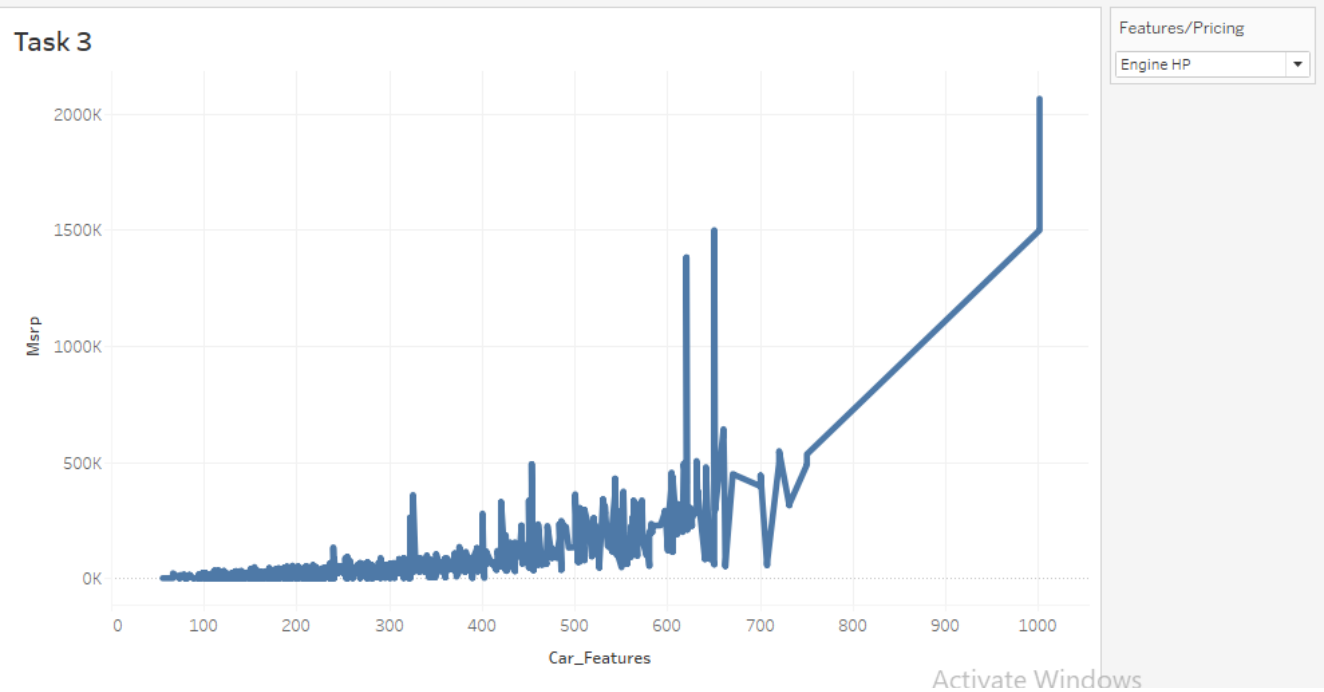
With this Bar chart we conclude that Crossover and Luxury are covering most of the market segment.

Insight Required: How do different car features affect consumer demand and profitability?

- **Task 3:** Use a sensitivity analysis to analyze the impact of different features on sales and profitability



Clearly , we can see that cars with only 2 doors are the most expensive .



Here, also as the Engine HP increase the estimated price also increase.

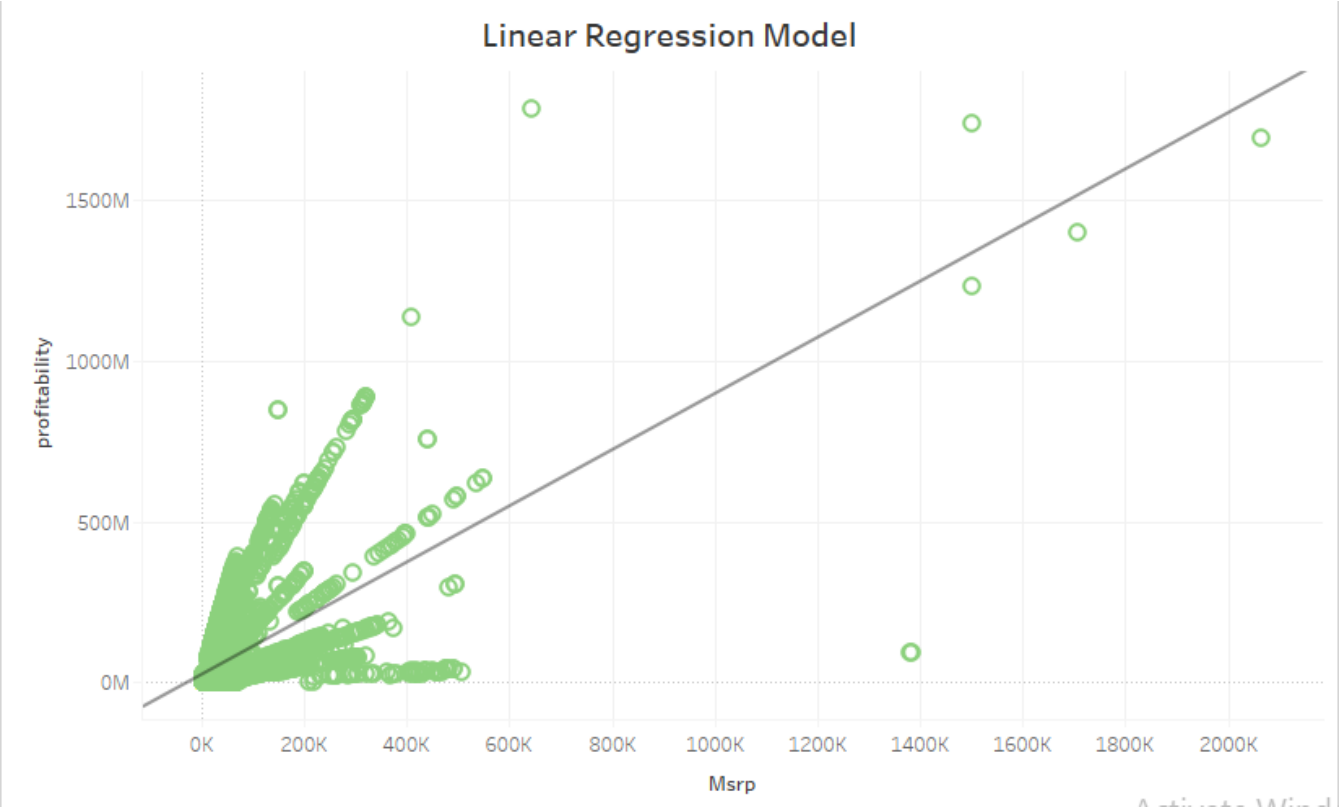
More insights can be drawn by filtering the chart using Features/Pricing

Insight Required: What is the optimal pricing strategy for different car models based on consumer demand and manufacturing costs?

- **Task 4:** Develop an optimization model that maximizes profitability while meeting consumer demand, and display the results in a line chart that shows the relationship between price, demand, and profitability.

Model Building:- Develop a model that relates price to demand and profitability: To do this, we can use a simple linear model that relates the price of a car to its demand and profitability:

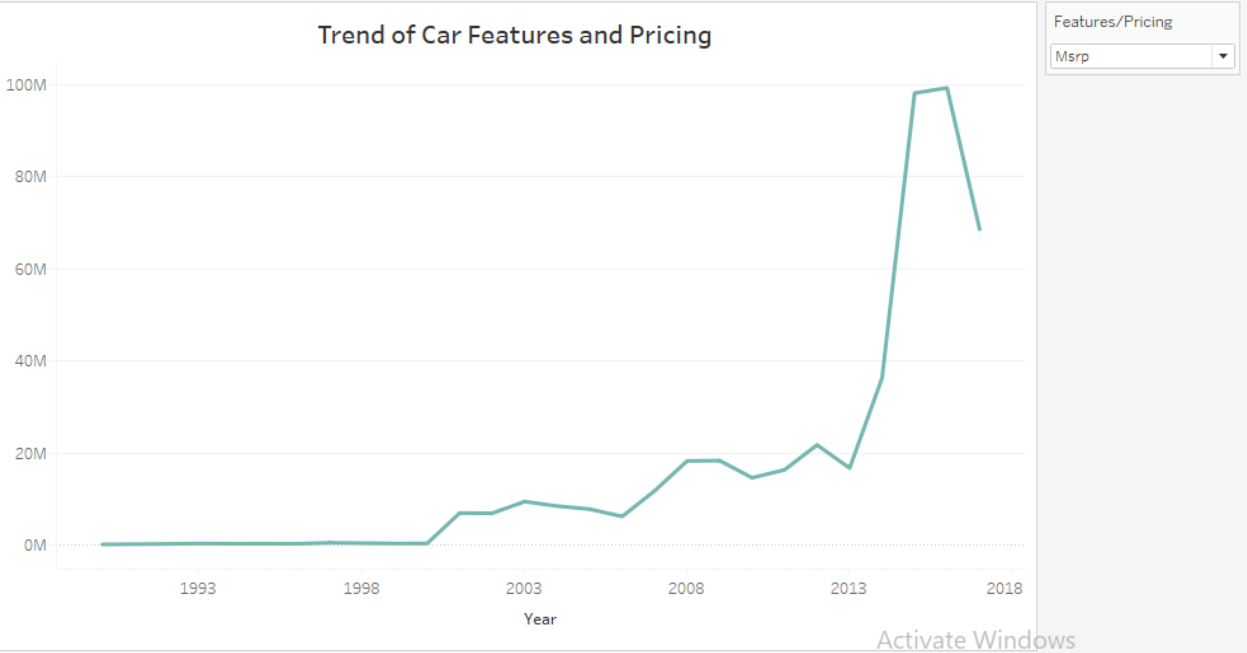
Profitability = Price * Popularity(Demand)



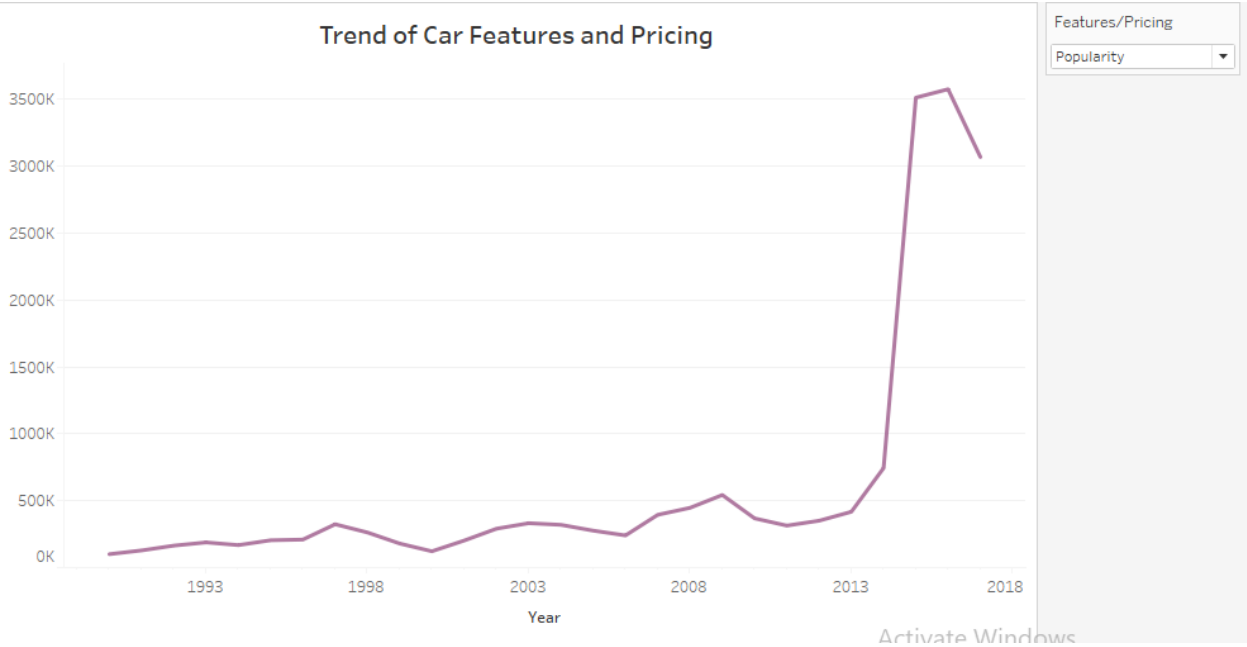
With this model we can say to get maximum profit. We can increase price along with demand or Popularity. With High Demand High Price Profit will be maximum.

Insight Required: How has the automotive market evolved over time in terms of pricing and features?

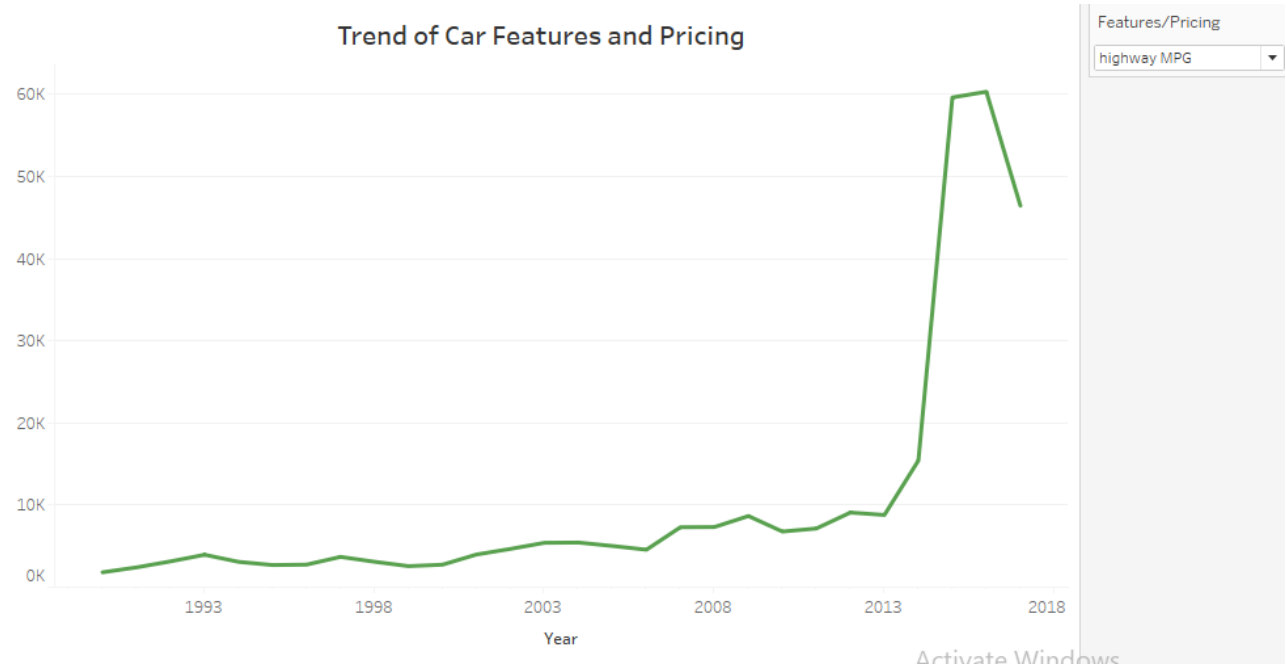
- **Task 5:** Use a time series analysis to analyze trends in car features and pricing over time, and display the results in a line chart that shows the average price and features of cars sold over time.

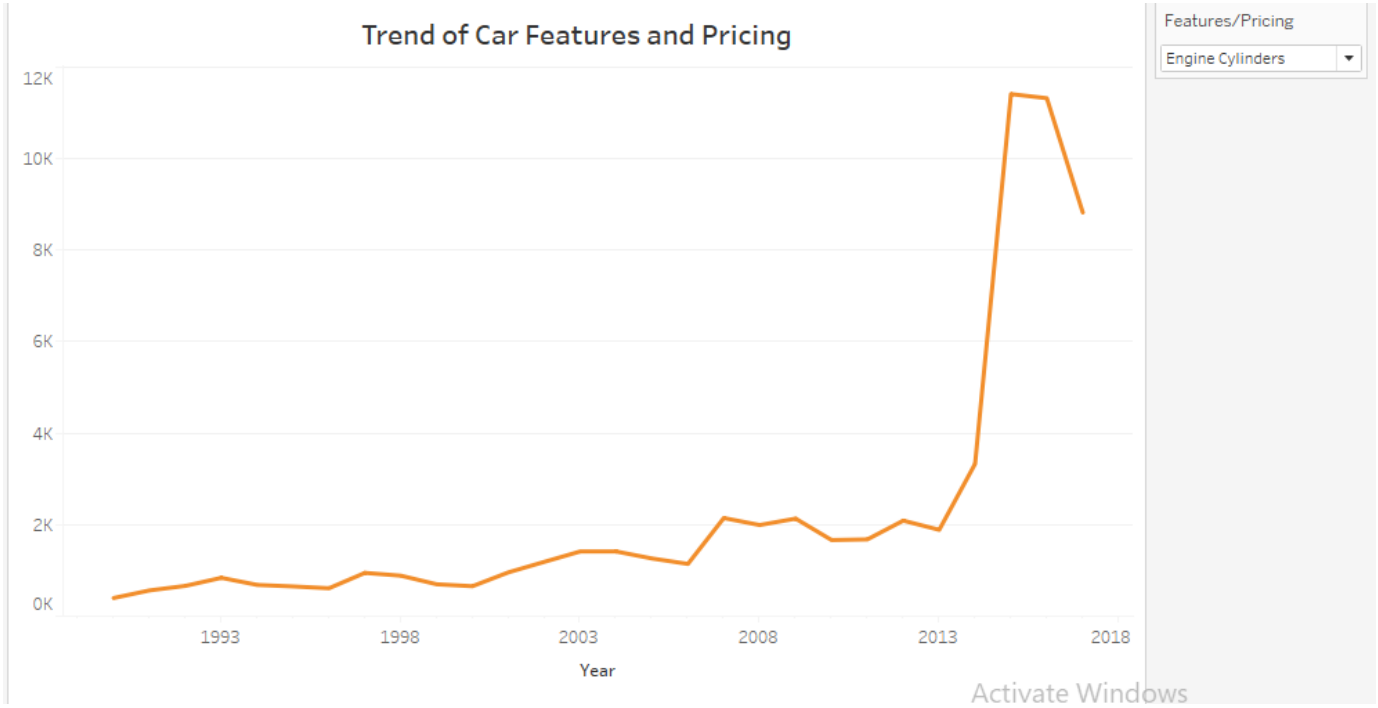


The trend of Msrp is increasing till 2016 but unfortunately , it is decreasing after that .



Trend of popularity is also same may be that is the reason of decreasing trend of price after 2016.





Same thing happens with all the Features they all show same type of trend.

Result:

This project aimed to analyze the impact of car features on price and profitability and develop a pricing strategy that balances consumer demand with profitability. The results showed that car features, market category, and manufacturing costs all play important roles in determining the optimal pricing strategy. The project provided valuable insights to a car manufacturer and helped them optimize their pricing and product development decisions to maximize profitability while meeting consumer demand.

DashBoard:

Final dashboard look like this:-