INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Used Bike Price Analysis

By

Kiran Kumar

BUSINESS PROBLEM OBJECTIVE :

Problem Statement:

Buyers and sellers often struggle to determine the fair price of used vehicles due to variations in mileage, engine capacity, and ownership history.

Goal:

To analyze key features of used vehicles (like engine capacity, kilometers run, and vehicle type) and identify their relationship with price..

INNOMATICS RESEARCH LABS

Objective of the Project :

- Scrape real-world property data from bike4sale.in.
- Clean and structure data for analysis.
- Analyze Bike prices using Model name, and Enginee CC.
- Find patterns and correlations among features (price, KM run)
- Derive useful insights for buyers and sellers.

INNOMATICS
RESEARCH LABS

Web Scraping Process :

Website: https://www.bikes4sale.in/used/buy/

Tools Used: Python, BeautifulSoup, Requests, Pandas

Steps Followed:

- **Fetched multiple pages using requests.**
  Parsed HTML using BeautifulSoup.
- **Extracted data fields:**

  Vehicle Name, Model, Buy_Year, Price, Engine_CC, No_of_owners, KM runs,etc.
- **Stored in a DataFrame → Exported to CSV**

INNOMATICS RESEARCH LABS

```
No_of_owners=[]
color=[]
KM=[]
for i in range(1,21):
    url=f"https://www.bikes4sale.in/used/buy/?page={i}"
    page=requests.get(url)
    soup=BeautifulSoup(page.content)

    for i in soup.find_all('div',class_="txtBox"):
    #print(i)
        link=re.findall('<a href="(.+)" rel',str(i))
    #print(link[0])
        link="https://www.bikes4sale.in"+str(link[0])
    #print(link)
        pages=requests.get(link)
    #print(pages.text)

        link_soup=BeautifulSoup(pages.content,'html.parser')
        link_bs=link_soup.find('div',attrs={"id":"secondCol"})
        name=re.findall("<h1>Used (\w+\s?)",str(link_bs))

        Name.append(name)

        data=link_bs.find_all("table",class_="b4stable itemDtlsTbl")

        model=re.findall(": (.+)</td>",str(data))

        Model.append(model[0])
        Buy_Year.append(model[1])

        Location.append(model[2])
    #print(Location)


        price=link_soup.find('div',class_="specLnk specLnkEmi serLnks")
        pp=re.findall(": Rs. (\d.+)C",price.text)
        if pp:
            Price.append(pp)
        else:
            Price.append(np.nan)
    #print(Price)




        link_bss=link_soup.find_all("table",class_="table idNewTbl")
        cc=re.findall("<td>(.+cc)",str(link_bss))
        if cc:

            Engine_CC.append(cc[0])
        else:
            Engine_CC.append(np.nan)
    #print(Engine_CC)

        owners = re.findall("<td>(.+) Owner", str(link_bss))
        if len(owners) >= 2:
            No_of_owners.append(owners[1])
        elif len(owners) == 1:
            No_of_owners.append(owners[0])
        else:
            No_of_owners.append(np.nan)

        link_color_table=link_soup.find_all('div',class_="addInfo itemSubDetails")

        kms=re.findall("<td>(.+)kms",str(link_color_table))
```

```
data
```

	Vechicle Name	Model	Buy_Year	Location	Price	Engine_CC	No_of_owners	KM runs
0	['Royal ']	Royal Enfield Thunderbird 350	March 2019 Model	Chennai, Tamil Nadu	['1,20,000']	350 cc	Single	32,000
1	['Royal ']	Royal Enfield Thunderbird TwinSpark 350	2009 Model	Banashankari, Bangalore, Karnataka	['80,000']	NaN	Single	34,700
2	['Bajaj ']	Bajaj Pulsar NS 125	2025 Model	Faridabad, Haryana	NaN	NaN	Single	2,100
3	['Hero ']	Hero Electric Photon	2022 Model	Pharenda, Maharajganj, Uttar Pradesh	['80,000']	NaN	Single	23,000
4	['Ola ']	Ola S1 Pro	2022 Model	Thumkunta, Medchal-Malkajgiri, Telangana	['80,000']	NaN	Single	24,000
...	...	...	...	...	...	...	...	...
820	['Hero ']	Hero CD 100	1999 Model	Vadodara, Gujarat	['10,000']	NaN	Single	40,000
821	['Jawa ']	Jawa forty two	2022 Model	New Delhi, Delhi	['1,10,000']	NaN	Single	10,250
822	['Jawa ']	Jawa 42 Dual Channel ABS AllStar Black	2021 Model	Pune, Maharashtra	['1,65,000']	NaN	Single	6,700
823	['Suzuki ']	Suzuki Gixxer 150	2016 Model	Kozhikode, Kerala	NaN	NaN	No of	1,01,800
824	['Honda ']	Honda CB Hornet 160R	2016 Model	Pune, Maharashtra	['60,000']	NaN	Single	12,000

825 rows × 8 columns

```

```
data.isnull().sum()
```

```
Vechicle Name   0
Model   0
Buy_Year   0
Location   0
Price   97
Engine_CC   508
No_of_owners   0
KM runs   14
```
```

Data Summary

Title: Summary of Collected Data

- Total
- Records:825
- Columns: 8
- Vechile name
- Model Buy
- year Location
- Price
- Engine CC
- KM Run
- No Of Owners

```
[140]: data.describe()
```

[140]:

	Buy_Year
count	810.000000
mean	2019.224691
std	5.068274
min	1972.000000
25%	2016.000000
50%	2020.000000
75%	2023.000000
max	2025.000000

```
[141]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 825 entries, 0 to 824
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Vechicle Name  825 non-null    object
 1   Model          825 non-null    object
 2   Buy_Year       810 non-null    float64
 3   Location       825 non-null    object
 4   Price          728 non-null    object
 5   Engine_CC      806 non-null    object
 6   No_of_owners   825 non-null    object
 7   KM runs        811 non-null    object
dtypes: float64(1), object(7)
```

Insight:

Data includes diverse property types (apartments, villas, independent houses) across multiple Hyderabad locations.

INNOMATICS RESEARCH LABS

DATA CLEANING :

- Removed duplicates & irrelevant rows.
- Handled missing values in Price, Buy_Year and Enginee CC columns.
- Standardized price units .
- Remove the extra Characters(Symbols []) from the columns to get the Original data.
- Converted text columns to proper format.

Insights

Data was cleaned and standardized for consistent analysis

INNOMATICS RESEARCH LABS

```
data['Price'] = data['Price'].astype(str).str.replace(r"[\[\]']", '', regex=True)
data['Vechicle Name'] = data['Vechicle Name'].astype(str).str.replace(r"[\[\]']", '', regex=True)
```

```
data[['City', 'State']] = data['Location'].str.split(',', n=1, expand=True)
data["State"]=data["State"].str.strip()
data["City"]=data["City"].str.strip()
```

```
data[["Village","State"]]=data["State"].str.split(',',n=1,expand=True)
data["State"]=data["State"].str.strip()
```

```
#data.drop("Location",axis=1,inplace=True)
data.loc[data['State'].isna(), 'State'] = data.loc[data['State'].isna(), 'Village']
data.drop("Village",axis=1,inplace=True)
```

```
data.loc[data['State'].isna(), 'State'] = data.loc[data['State'].isna(), 'City']
```

```
data
```

	Vechicle Name	Model	Buy_Year	Location	Price	Engine_CC	No_of_owners	KM runs	City	State
0	Royal	Royal Enfield Thunderbird 350	2019	Chennai, Tamil Nadu	1,20,000	340	Single	32,000	Chennai	Tamil Nadu
1	Royal	Royal Enfield Thunderbird TwinSpark 350	2009	Banashankari, Bangalore, Karnataka	80,000	340	Single	34,700	Banashankari	Karnataka
2	Bajaj	Bajaj Pulsar NS 125	2025	Faridabad, Haryana	nan	125 cc	Single	2,100	Faridabad	Haryana
3	Hero	Hero Electric Photon	2022	Pharenda, Maharajganj, Uttar Pradesh	80,000	105 cc	Single	23,000	Pharenda	Uttar Pradesh
4	Ola	Ola S1 Pro	2022	Thumkunta, Medchal-Malkajgiri, Telangana	80,000	Not Applicable	Single	24,000	Thumkunta	Telangana

825 rows × 10 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 825 entries, 0 to 824
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Vechicle Name  825 non-null    object
 1   Model          825 non-null    object
 2   Buy_Year       825 non-null    int64 
 3   Price          825 non-null    int64 
 4   Engine_CC      825 non-null    object
 5   No_of_owners   825 non-null    object
 6   KM runs        825 non-null    int64 
 7   City           825 non-null    object
 8   State          825 non-null    object
 9   Vehicle_Type   825 non-null    object
dtypes: int64(3), object(7)
```

Univariate Analysis (Numerical) :

- **Variables:**

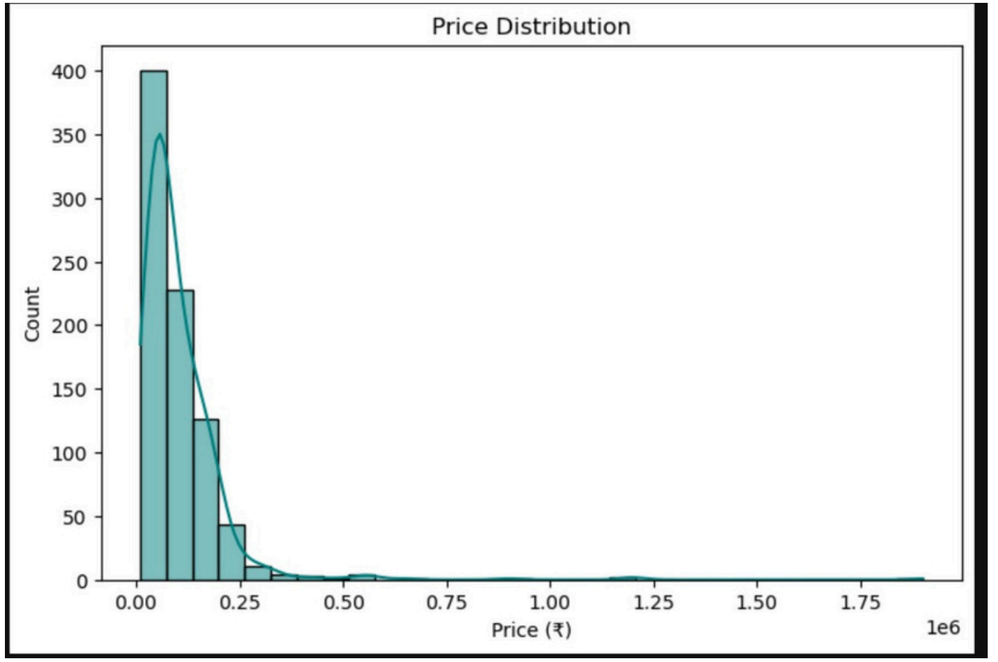
  Price,kms

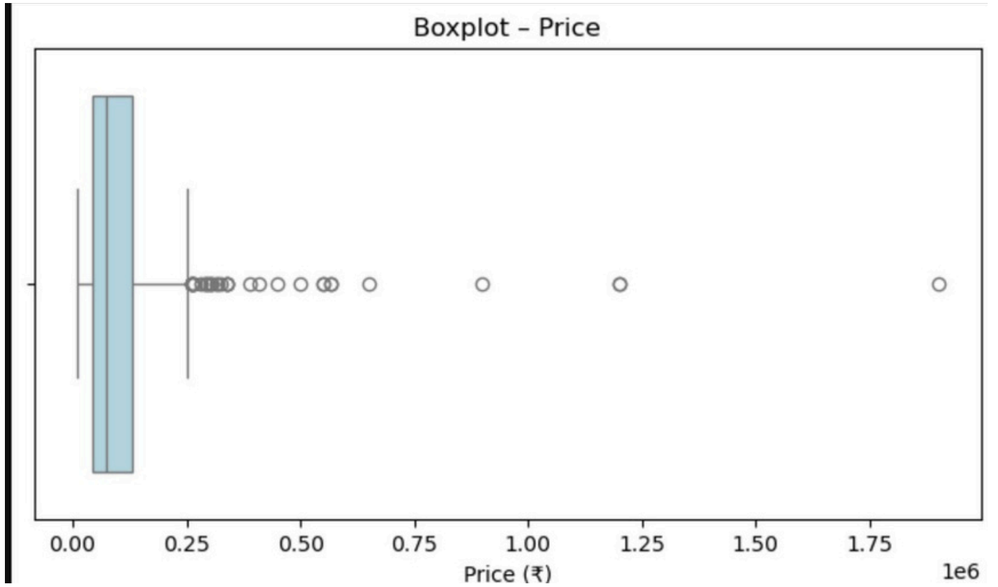
- **Plots Used :**

  Histogram
  Boxplot

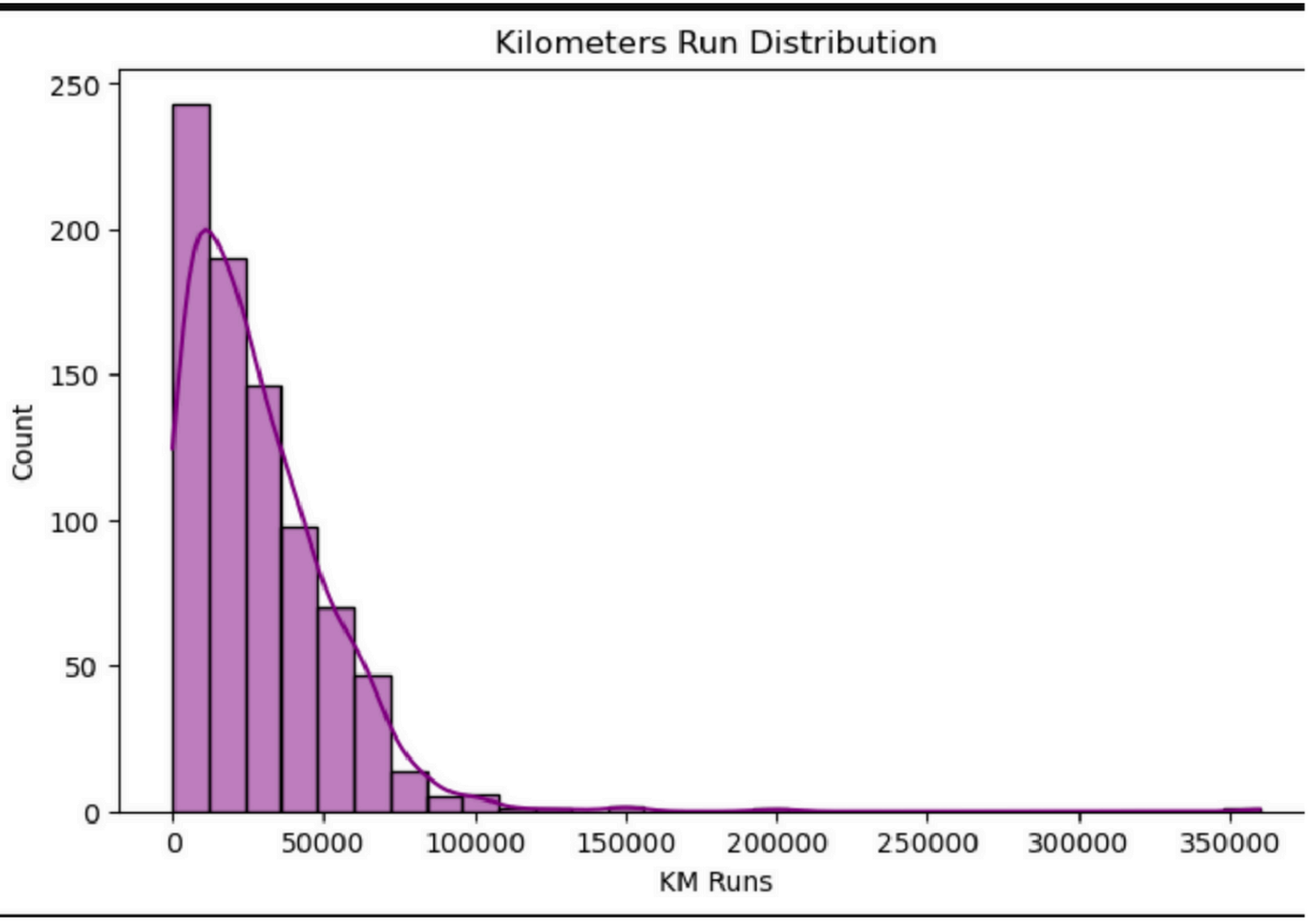
- **Insights:**

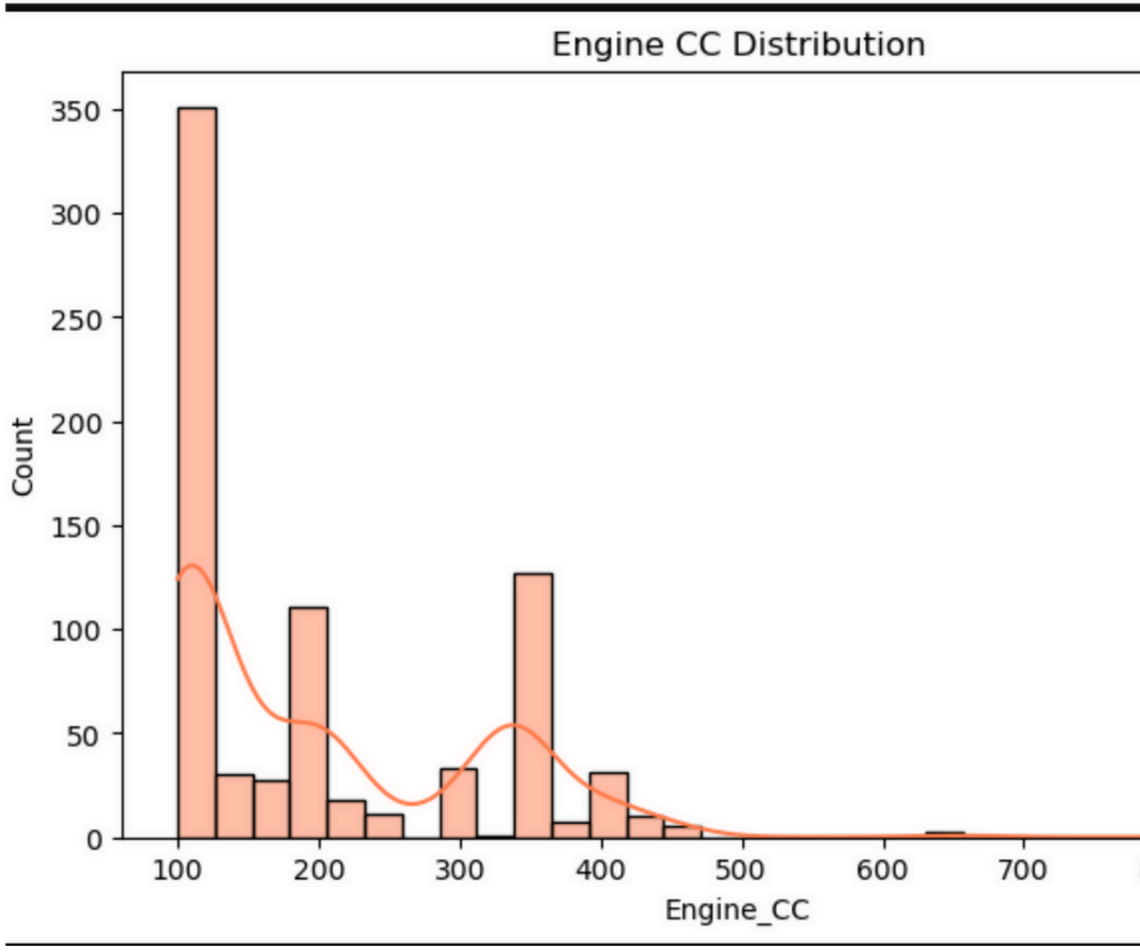
Vehicle Price range from ₹10,000 to ₹2,00,000. Engine Capacities range from 100 to 400cc. Vehicle have run between 5 to 3,60,000km





INNOMATICS RESEARCH LABS


Kilometers Run Distribution
Count
KM Runs
0
50
100
150
200
250
0
50000
100000
150000
200000
250000
300000
350000


Engine CC Distribution
Count
Engine_CC
0
50
100
150
200
250
300
350
100
200
300
400
500
600
700

Univariate Analysis (Categorical) :

- **Variables: Vehicle_Type,No of Owneers , State.**

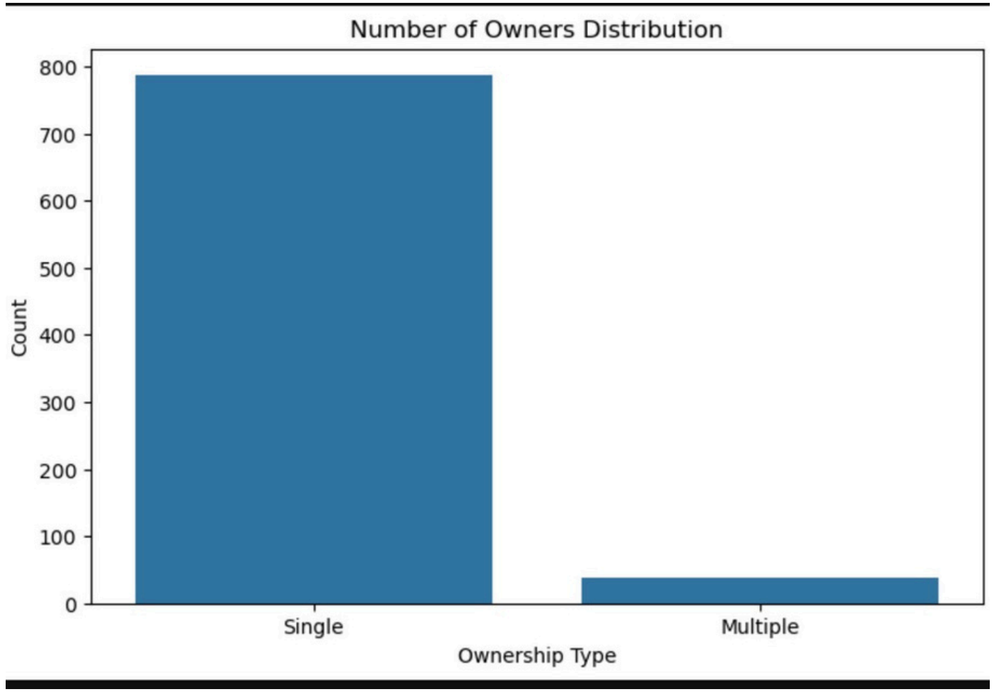
- **Plots Used:**

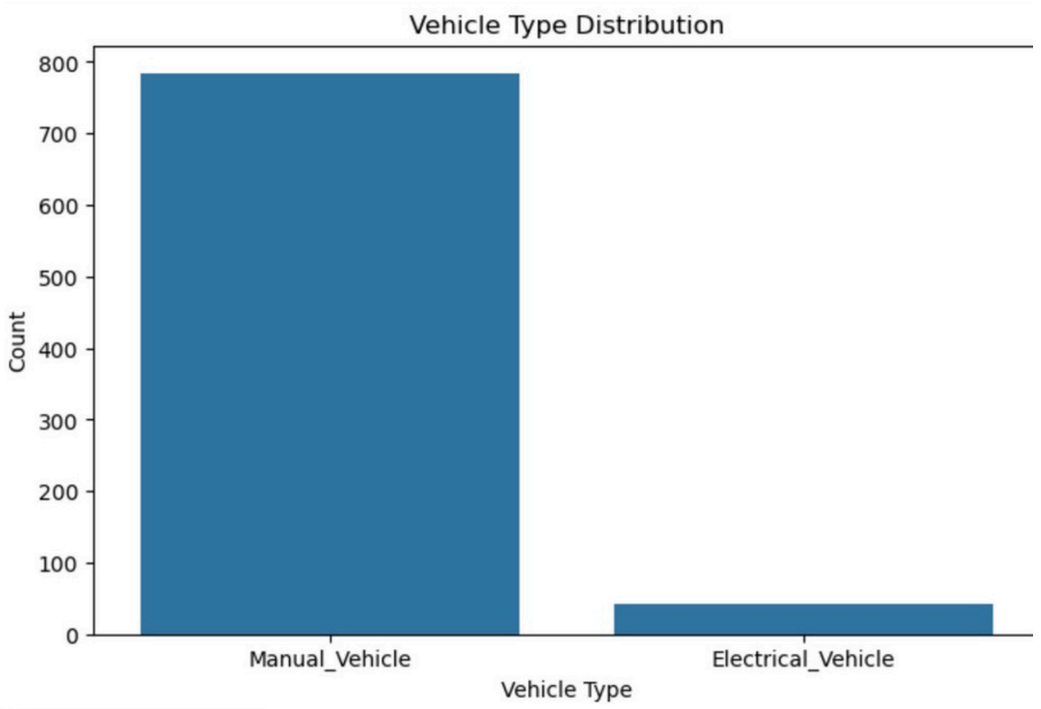
Count Plot
Pie Chart

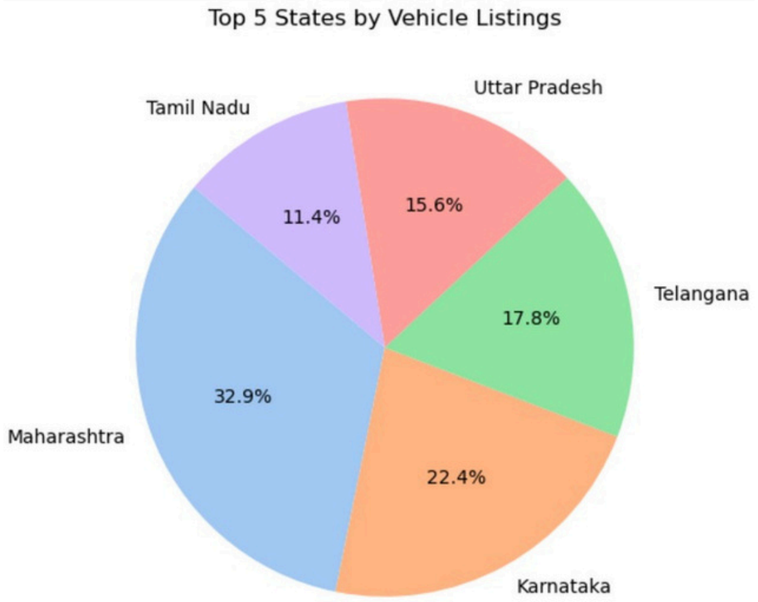
- **Insights:**

Manual_Vehicle models dominated the datase.
Most Vehicles from Single Owners indicating high resale value.

Highest listings come from Maharashtra,Karnataka, and Telangana. .


Number of Owners Distribution
Count
800
700
600
500
400
300
200
100
0
Single
Multiple
Ownership Type


Vehicle Type Distribution
Count
800
700
600
500
400
300
200
100
0
Manual_Vehicle
Electrical_Vehicle
Vehicle Type


Top 5 States by Vehicle Listings
Tamil Nadu
11.4%
Uttar Pradesh
15.6%
Telangana
17.8%
Maharashtra
32.9%
Karnataka
22.4%

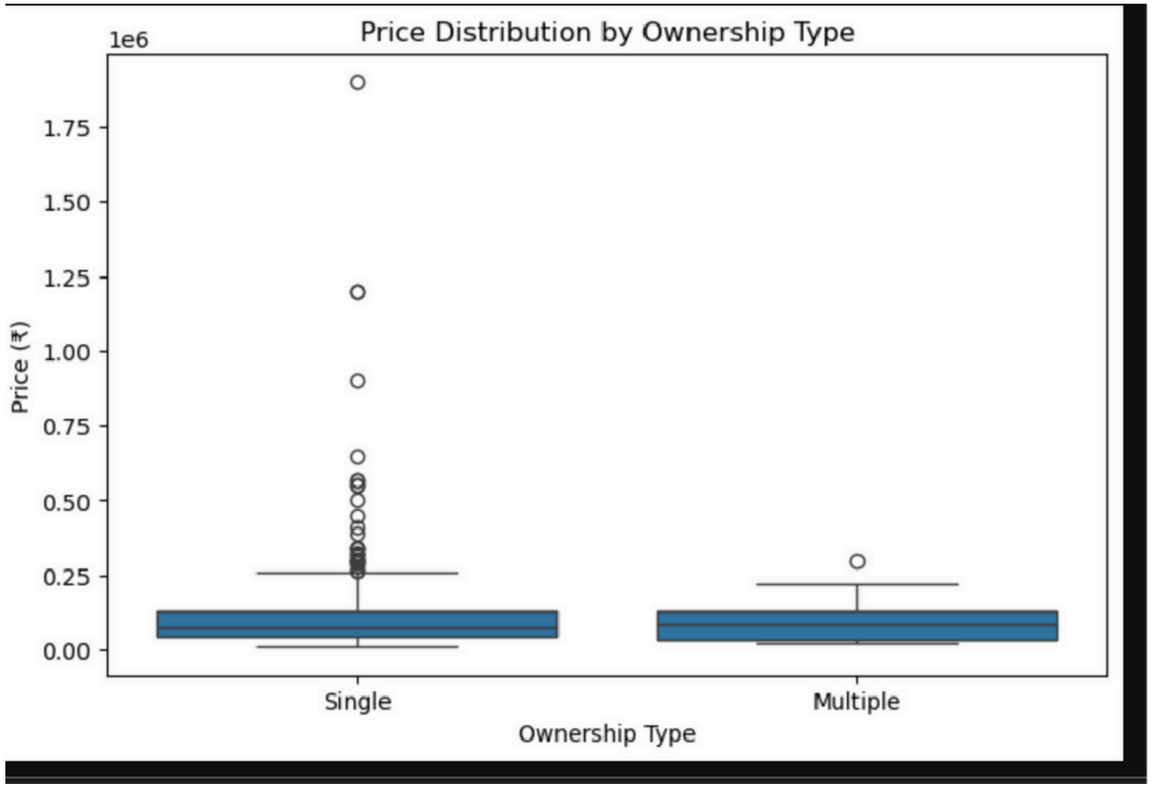
Bivariate Analysis (Continuous vs Categorical) :

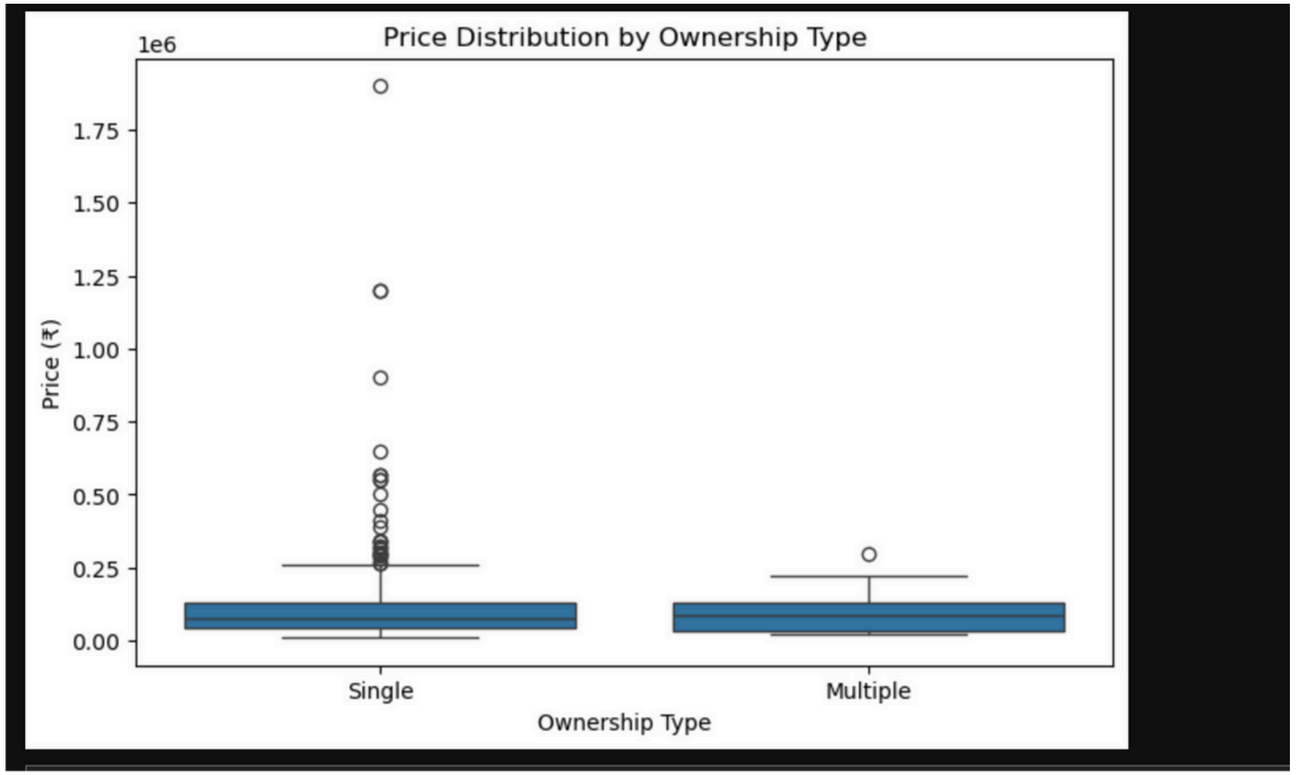
- **Variables: Vehicle_Type vs Price, No_of_owners vs Price**
- **Plots Used:**

  Boxplot

- **Insights:**

  - **The highest average price is for Manual_Vehicle models (₹104,227).**
  - **Single-owner vehicles have the highest average resale price (₹103,464).**

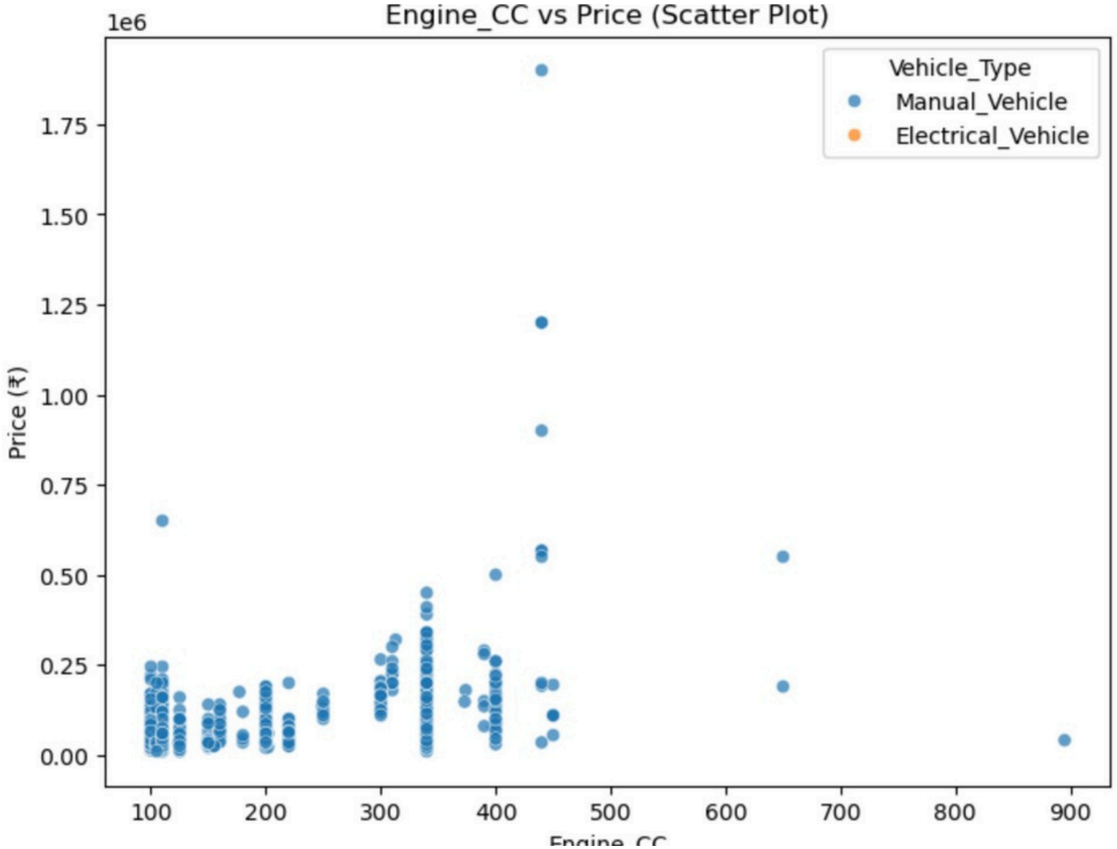

Price Distribution by Ownership Type
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Price (₹)
Single
Multiple
Ownership Type


Price Distribution by Ownership Type
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Price (₹)
Single
Multiple
Ownership Type

Bivariate Analysis (Continuous vs Continuous) :

- **Variables: Engine_CC vs Price**
- **Plots Used:**

  Scatter Plot



- **Insights:**

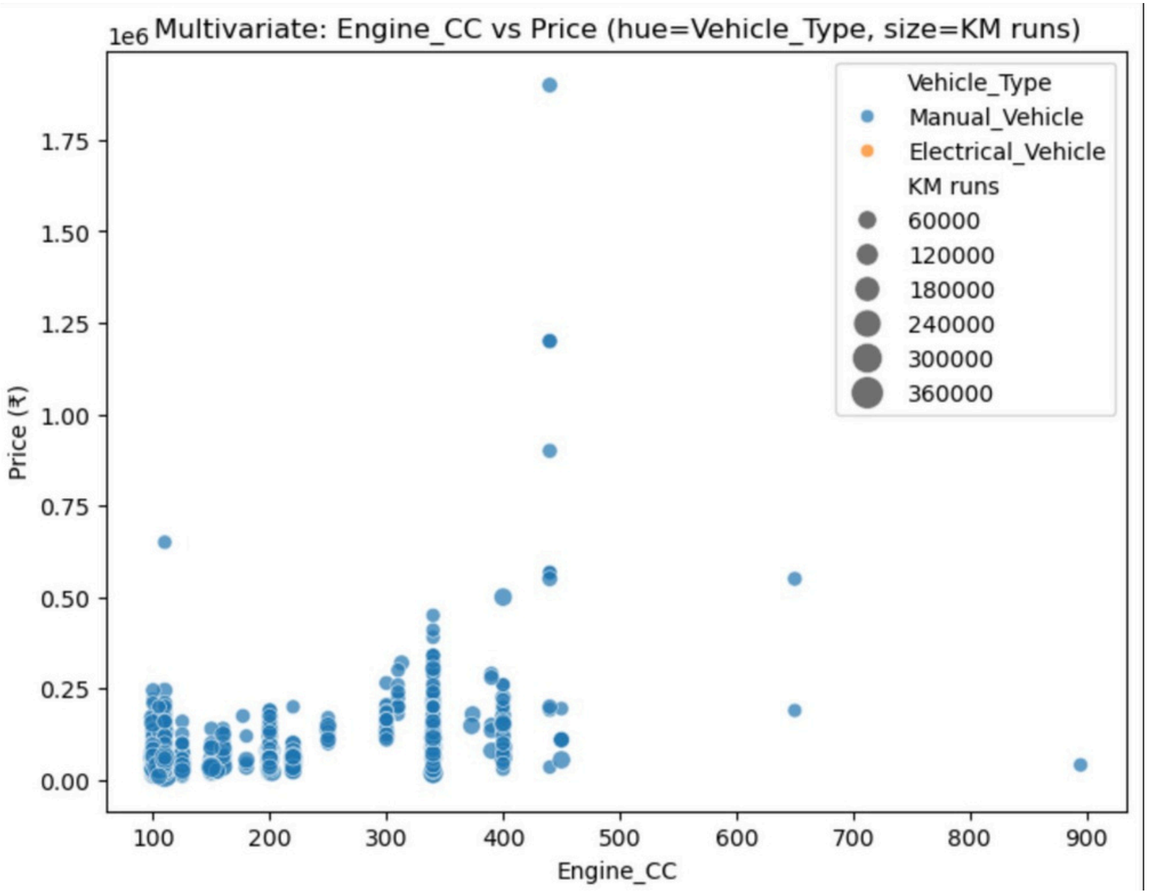
There is a positive correlation (r = 0.31) between Engine_CC and Price — higher engine capacity generally leads to higher price.

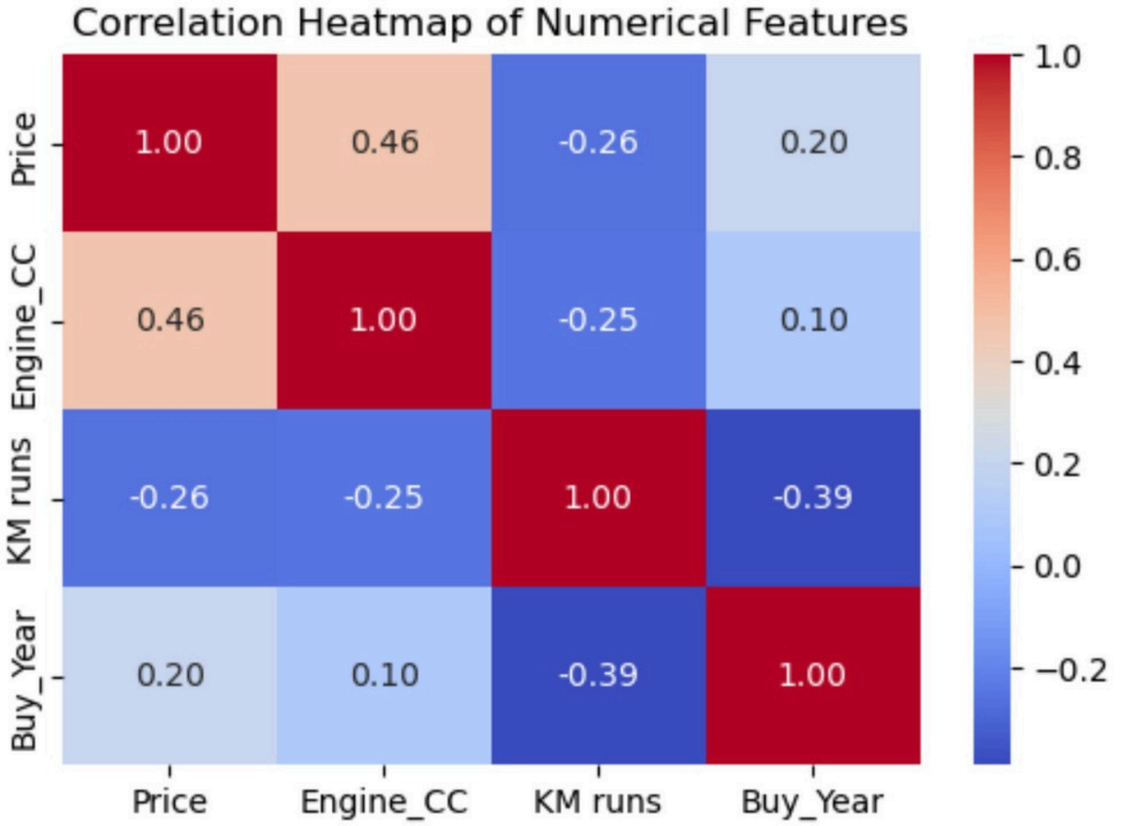
INNOMATICS RESEARCH LABS

Multivariate Analysis :

- **Variables: Price, Engine_CC, KM runs, Buy_Year**
- **Plot Used:**

  Heatmap
- **Insights:**
  - **The feature most strongly correlated with Price is Engine_CC.**
  - **Vehicles with higher engine capacity, fewer kilometers, and recent buy years tend to have higher resale prices.**


Multivariate: Engine_CC vs Price (hue=Vehicle_Type, size=KM runs)
1e6
Price (₹)
Engine_CC
Vehicle_Type
Manual_Vehicle
Electrical_Vehicle
KM runs
60000
120000
180000
240000
300000
360000


Correlation Heatmap of Numerical Features
Price
Engine_CC
KM runs
Buy_Year
1.00
0.46
-0.26
0.20
0.46
1.00
-0.25
0.10
-0.26
-0.25
1.00
-0.39
0.20
0.10
-0.39
1.00

Challenges & Learnings :

Challenges Faced:

- Inconsistent HTML structure during scraping.
- Handling missing or unstructured text.
- While removing the extra symbols.
- Cleaning and aligning data formats.

Learnings:

Improved skills in web scraping, data cleaning, and visualization.

Gained deeper understanding of EDA workflow and insights generation.

INNOMATICS RESEARCH LABS

Conclusion :

- Manual, single-owner vehicles with higher engine capacity and fewer kilometers run retain the best resale value in the used vehicle market

Overall Insight:

Used vehicle prices are mainly driven by engine capacity, mileage, ownership type, and model year, with newer, low-mileage, high-CC vehicles commanding higher resale values.

INNOMATICS RESEARCH LABS

THANK YOU

INNOMATICS RESEARCH LABS