	Pimpri Chinchwad Education Trust's **Pimpri Chinchwad College of Engineering** **(PCCoE)** (An Autonomous Institute) Affiliated to Savitribai Phule Pune University(SPPU) ISO 21001:2018 Certified by TUV	
	VSEC MINI PROJECT	

Student Details

Name: Shivanjali Bhosale

Gaurav Biradar

Harshvardhan Borude

Branch : Information Technology

Division : A

PRN: 123B1F009

123B1F011

123B1F013

Course Name : Data Science Laboratory

Course Code : BIT23VS01

Project Definition and Objectives

Indian Used Car Price Prediction

The aim of this project to predict the price of the used cars in indian metro cities by analyzing the car's features such as company, model, variant, fuel type, quality score and many more.

About the Dataset

The "Indian IT Cities Used Car Dataset 2023" is a comprehensive collection of data that offers valuable insights into the used car market across major metro cities in India. This dataset provides a wealth of information on a wide range of used car listings, encompassing details such as car models, variants, pricing, fuel types, dealer locations, warranty information, colors, kilometers driven, body styles, transmission types, ownership history, manufacture dates, model years, dealer names, CNG kit availability, and quality scores.

Data Preparation

Data Preparation

A. Importing Necessary Libraries

```python
#Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Python

B. Loading the Dataset

```python
#Loading the dataset
df = pd.read_csv("C:\\Users\\usern\\Downloads\\usedCars.csv")
df.info()
df.head()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1064 entries, 0 to 1063
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Id                1064 non-null   int64
 1   Company           1064 non-null   object
 2   Model             1064 non-null   object
 3   Variant           1064 non-null   object
 4   FuelType          1063 non-null   object
 5   Colour            1064 non-null   object
 6   Kilometer         1064 non-null   int64
 7   BodyStyle         1064 non-null   object
 8   TransmissionType  350 non-null    object
 9   ManufactureDate   1064 non-null   object
 10  ModelYear         1064 non-null   int64
 11  CngKit            22 non-null     object
 12  Price             1064 non-null   object
 13  Owner             1064 non-null   object
 14  DealerState       1064 non-null   object
 15  DealerName        1064 non-null   object
 16  City              1064 non-null   object
 17  Warranty          1064 non-null   int64
 18  QualityScore      1064 non-null   float64
dtypes: float64(1), int64(4), object(14)
memory usage: 158.1+ KB
```

```
...       Id        Company             Model           Variant FuelType  \
   0  555675  MARUTI SUZUKI  CELERIO(2017-2019)     1.0 ZXI AMT O   PETROL
   1  556383  MARUTI SUZUKI                ALTO               LXI   PETROL
   2  556422        HYUNDAI           GRAND I10    1.2 KAPPA ASTA   PETROL
   3  556771           TATA               NEXON           XT PLUS   PETROL
   4  559619           FORD                FIGO  EXI DURATORQ 1.4   DIESEL

      Colour  Kilometer  BodyStyle TransmissionType ManufactureDate  ModelYear  \
   0  Silver      33197  HATCHBACK              NaN      2018-02-01       2018
   1     Red      10322  HATCHBACK           Manual      2021-03-01       2021
   2    Grey      37889  HATCHBACK           Manual      2015-03-01       2015
   3  A Blue      13106  HATCHBACK              NaN      2020-08-01       2020
   4  Silver     104614  HATCHBACK           Manual      2010-11-01       2010

     CngKit        Price      Owner DealerState                      DealerName  \
   0    NaN  5.75 Lakhs  1st Owner   Karnataka                   Top Gear Cars
   1    NaN  4.35 Lakhs  1st Owner   Karnataka  Renew 4 u Automobiles PVT Ltd
   2    NaN   4.7 Lakhs  1st Owner   Karnataka         Anant Cars Auto Pvt Ltd
   3    NaN   9.9 Lakhs  1st Owner   Karnataka                    Adeep Motors
   4    NaN   2.7 Lakhs  2nd Owner   Karnataka                  Zippy Automart

           City  Warranty  QualityScore
   0  Bangalore         1           7.8
   1  Bangalore         1           8.3
   2  Bangalore         1           7.9
   3  Bangalore         1           8.1
   4  Bangalore         0           7.5
```

Data Preprocessing Part 1

```
df.dtypes
```

Python

```
Company               object
Model                 object
Variant               object
FuelType              object
Colour                object
Kilometer              int64
BodyStyle             object
TransmissionType      object
ManufactureDate       object
ModelYear              int64
CngKit                object
Price                 object
Owner                 object
DealerState           object
DealerName            object
City                  object
Warranty               int64
QualityScore         float64
dtype: object
```

```
def convert_amount(amount_str):
    if "Lakhs" in amount_str:
        return float(amount_str.replace(' Lakhs', '').replace(',', '')) * 100000
    else:
        return float(amount_str.replace(',', ''))

df['Price'] = df['Price'].apply(convert_amount)
```

Python

```
df.isnull().sum()/df.shape[0]*100
```

Python

```
Company             0.000000
Model               0.000000
Variant             0.000000
FuelType            0.093985
Colour              0.000000
Kilometer           0.000000
BodyStyle           0.000000
TransmissionType   67.105263
ManufactureDate     0.000000
ModelYear           0.000000
CngKit             97.932331
Price               0.000000
Owner               0.000000
DealerState         0.000000
DealerName          0.000000
City                0.000000
Warranty            0.000000
QualityScore        0.000000
dtype: float64
```

Here in the dataset, three columns have missing values - FuelType, TransmissionType and CngKit. I will be removing the CngKit column becuase in majority of the cars don't run on CNG and the CNG cars can be easily identified from the FuelType column. So we will replace the null values with 'No' in CngKit column. In case of the TransmissionType, 67% data is missing, so we can't include this column in our analysis. In case of the FuelType, we will drop the rows with null values.

C. Dropping NA Values

```python
df.drop('CngKit', axis=1, inplace=True)
```
Python

```python
df.drop('TransmissionType',axis=1,inplace=True)
```
Python

```python
df['FuelType'].dropna(inplace=True)
```
Python

```python
df.drop('ManufactureDate', axis = 1, inplace=True)
```
Python

```python
df.drop('Variant', axis = 1, inplace=True)
```
Python

```python
df['ModelYear'] = 2024 - df['ModelYear']
df.rename(columns={'ModelYear':'Age'},inplace=True)
```
Python

```python
for i in df.columns:
    print(i,df[i].nunique())
```
Python

```
Company 23
Model 218
FuelType 5
Colour 76
Kilometer 1006
BodyStyle 10
Age 17
Price 362
Owner 4
DealerState 10
DealerName 57
City 11
Warranty 2
QualityScore 43
```

```python
df.describe()
```

Python

	Kilometer	Age	Price	Warranty	QualityScore
count	1064.000000	1064.000000	1.064000e+03	1064.000000	1064.000000
mean	52807.187970	7.135338	8.350536e+05	0.738722	7.770207
std	33840.296979	2.996786	5.726538e+05	0.439538	0.719717
min	101.000000	1.000000	9.500000e+04	0.000000	0.000000
25%	32113.500000	5.000000	4.850000e+05	0.000000	7.500000
50%	49432.000000	7.000000	6.750000e+05	1.000000	7.800000
75%	68828.500000	9.000000	9.850000e+05	1.000000	8.100000
max	640000.000000	21.000000	8.500000e+06	1.000000	9.400000

```python
df.head()
```

Python

```
         Company                 Model FuelType  Colour  Kilometer  BodyStyle  \
0  MARUTI SUZUKI  CELERIO(2017-2019)   PETROL  Silver      33197  HATCHBACK
1  MARUTI SUZUKI                  ALTO   PETROL     Red      10322  HATCHBACK
2        HYUNDAI             GRAND I10   PETROL    Grey      37889  HATCHBACK
3           TATA                 NEXON   PETROL  A Blue      13106  HATCHBACK
4           FORD                  FIGO   DIESEL  Silver     104614  HATCHBACK

   Age     Price      Owner DealerState                      DealerName  \
0    6  575000.0  1st Owner   Karnataka                  Top Gear Cars
1    3  435000.0  1st Owner   Karnataka  Renew 4 u Automobiles PVT Ltd
2    9  470000.0  1st Owner   Karnataka      Anant Cars Auto Pvt Ltd
3    4  990000.0  1st Owner   Karnataka                   Adeep Motors
4   14  270000.0  2nd Owner   Karnataka                 Zippy Automart

        City  Warranty  QualityScore
0  Bangalore         1           7.8
1  Bangalore         1           8.3
2  Bangalore         1           7.9
3  Bangalore         1           8.1
4  Bangalore         0           7.5
```

Exploratory Data Analysis (EDA)

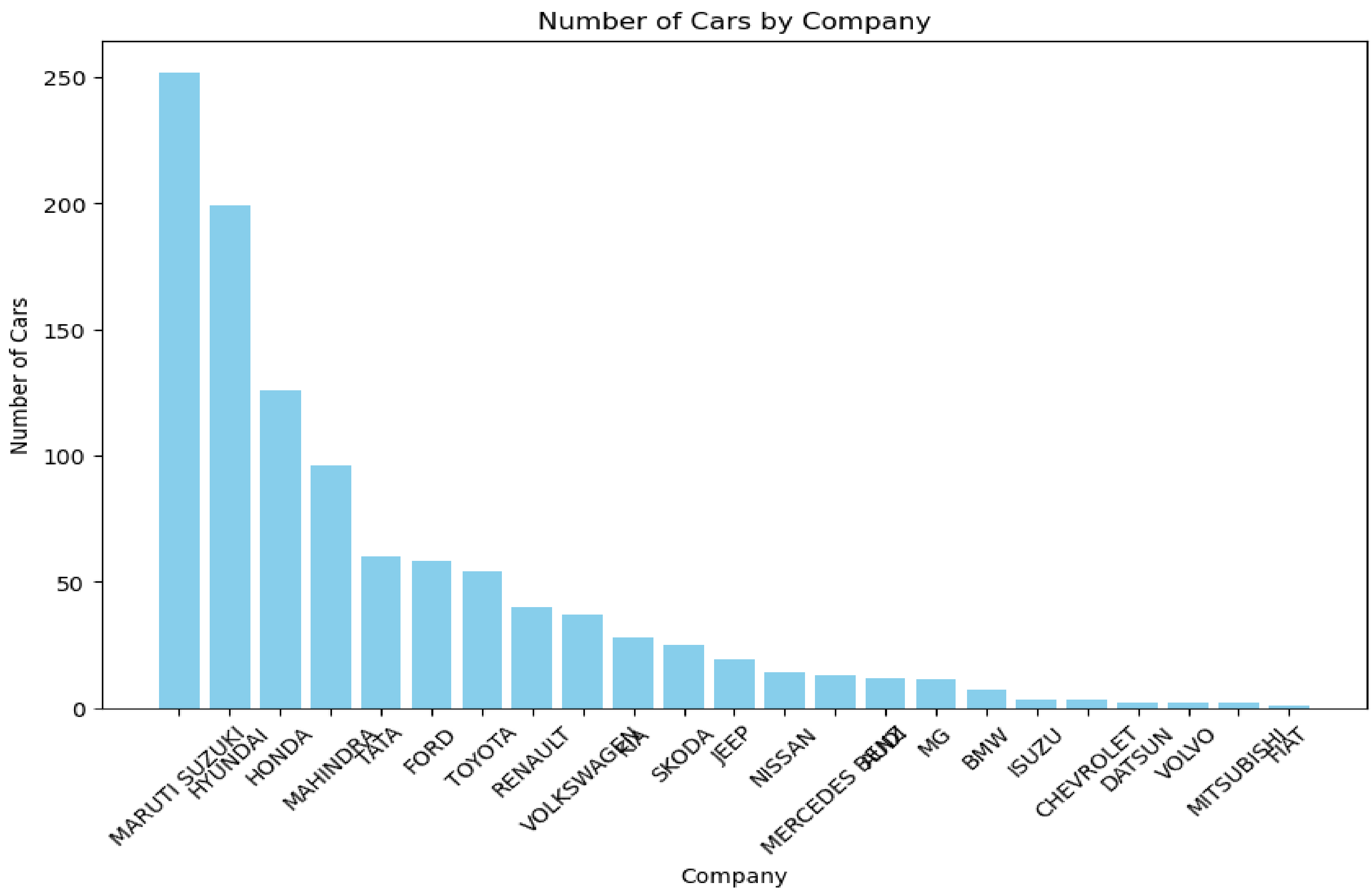
The goal of EDA is to understand the dataset's structure, identify key relationships between features and the target variable (price), and detect any anomalies or patterns. This process includes univariate, bivariate, and multivariate analysis using descriptive statistics and visualizations.

Visualization of the data is a good way to understand the data. In this section, I will plot the distribution of each variable to get an overview about their counts and distributions.

Car Company

```python
company_counts = df['Company'].value_counts()

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(company_counts.index, company_counts.values, color='skyblue')
plt.xlabel('Company')
plt.ylabel('Number of Cars')
plt.title('Number of Cars by Company')
plt.xticks(rotation=45)  # Rotate x labels if they overlap
plt.show()
```

Python

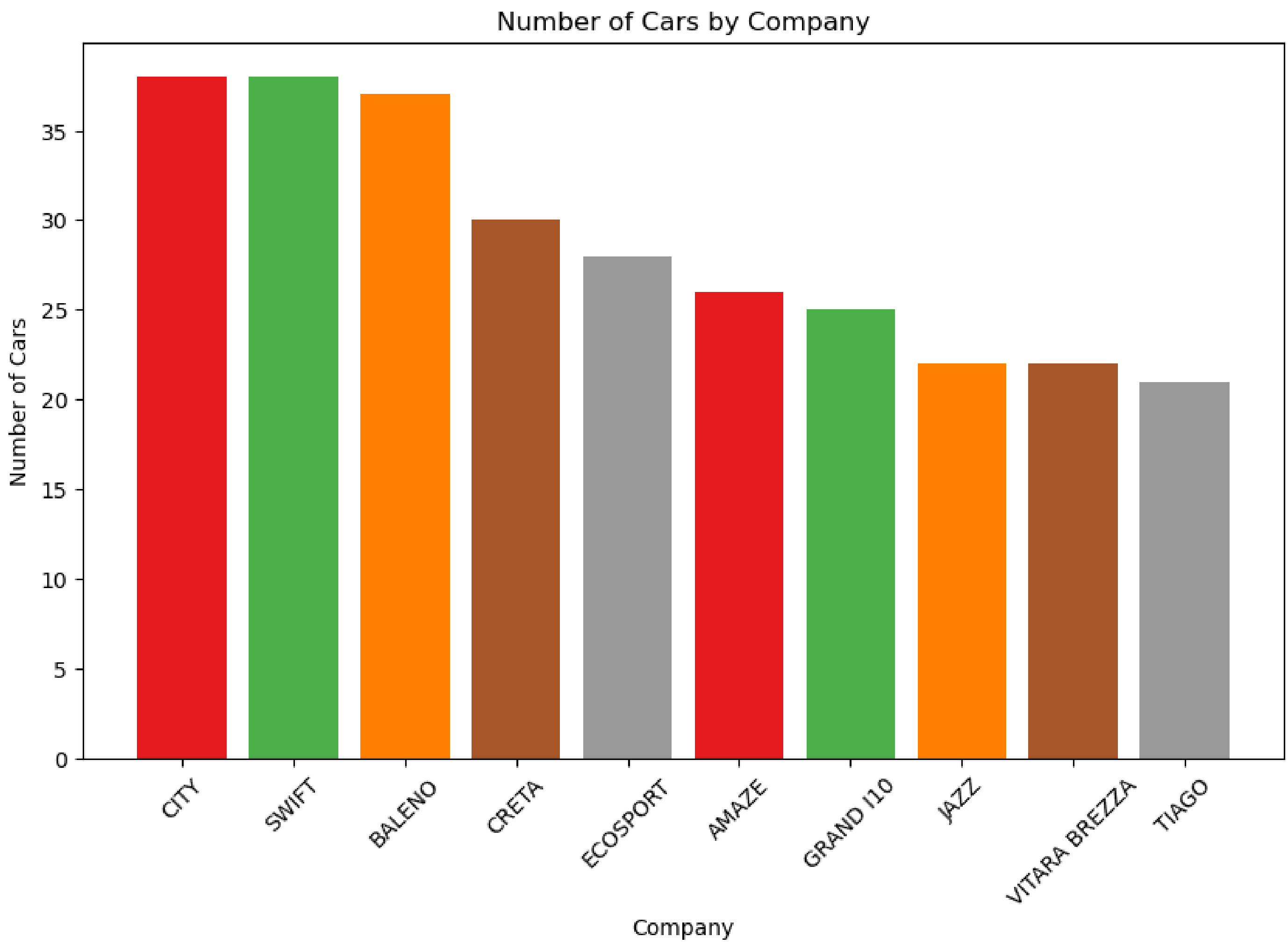


From this graph, we get know about the distribution of cars in the dataset from different companies.There are total 23 companies in the dataset, out which Maruti Suzuki, Hyundai, Honda, Mahindra and Tata are the top five companies who used cars are for sale. Therefore, we can assume that these company's car are more durable and have a good resale value

Top 10 Car Models

```python
import matplotlib.pyplot as plt

# Count the number of cars by company
company_counts = df['Model'].value_counts().iloc[:10]
colors = cm.Set1(np.linspace(0, 1, len(fuel_counts)))
# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(company_counts.index, company_counts.values, color=colors)
plt.xlabel('Company')
plt.ylabel('Number of Cars')
plt.title('Number of Cars by Company')
plt.xticks(rotation=45)  # Rotate x labels if they overlap
plt.show()
```

Python

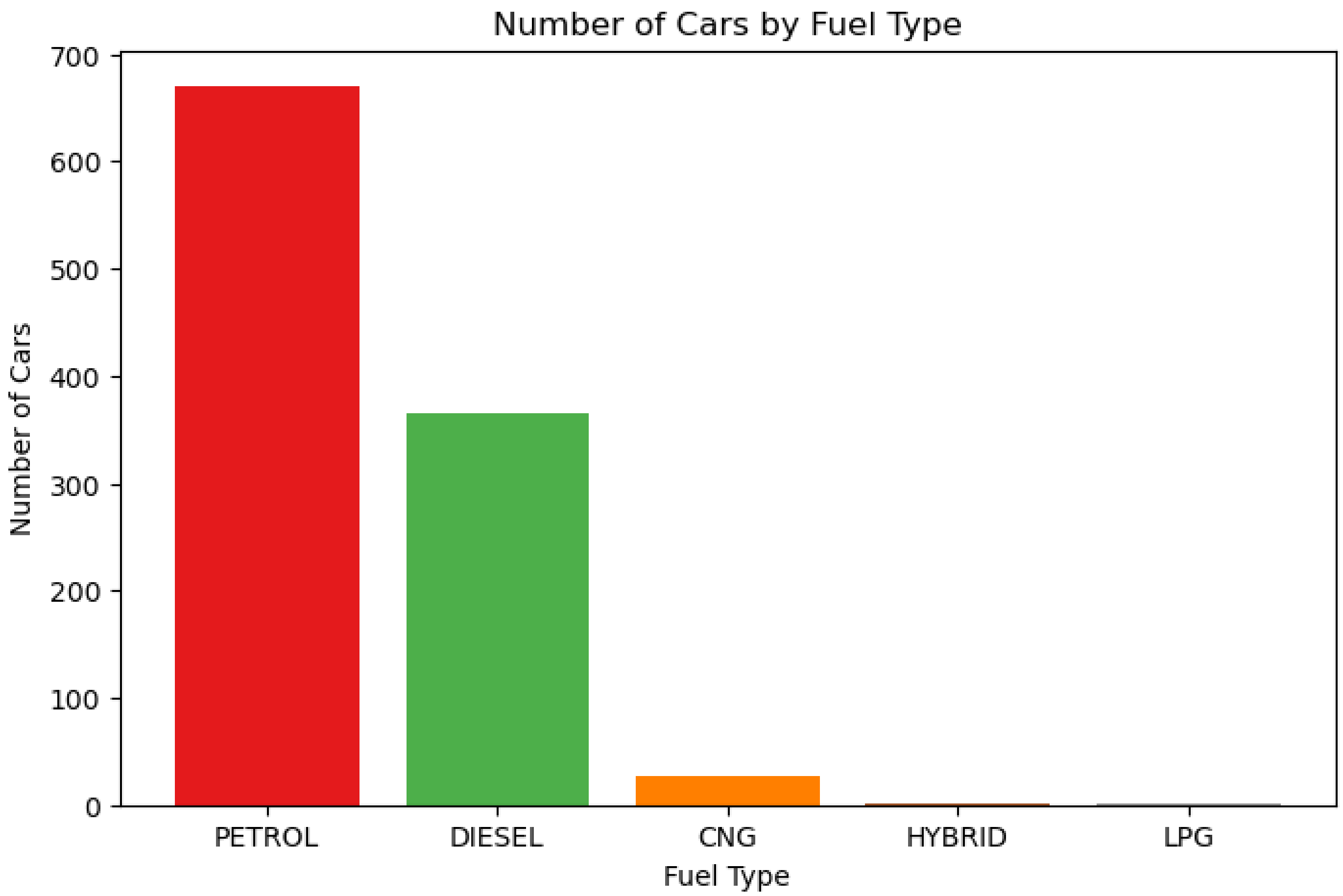


Honda City and Swift are the top two car models in the dataset, followed by Baleno, Creata and EcoSport. Therefore, we can assume that these car models are more durable and have a good resale value. Moreover, this graph also shows that Honda City and Swift are more in demand in the used car market

Car Fuel Type

```python
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import numpy as np

# Count the number of cars by fuel type
fuel_counts = df['FuelType'].value_counts()
# Define a color palette
colors = cm.Set1(np.linspace(0, 1, len(fuel_counts)))

# Plotting the bar chart
plt.figure(figsize=(8, 5))
plt.bar(fuel_counts.index, fuel_counts.values, color=colors)
plt.xlabel('Fuel Type')
plt.ylabel('Number of Cars')
plt.title('Number of Cars by Fuel Type')
plt.show()
```

Python

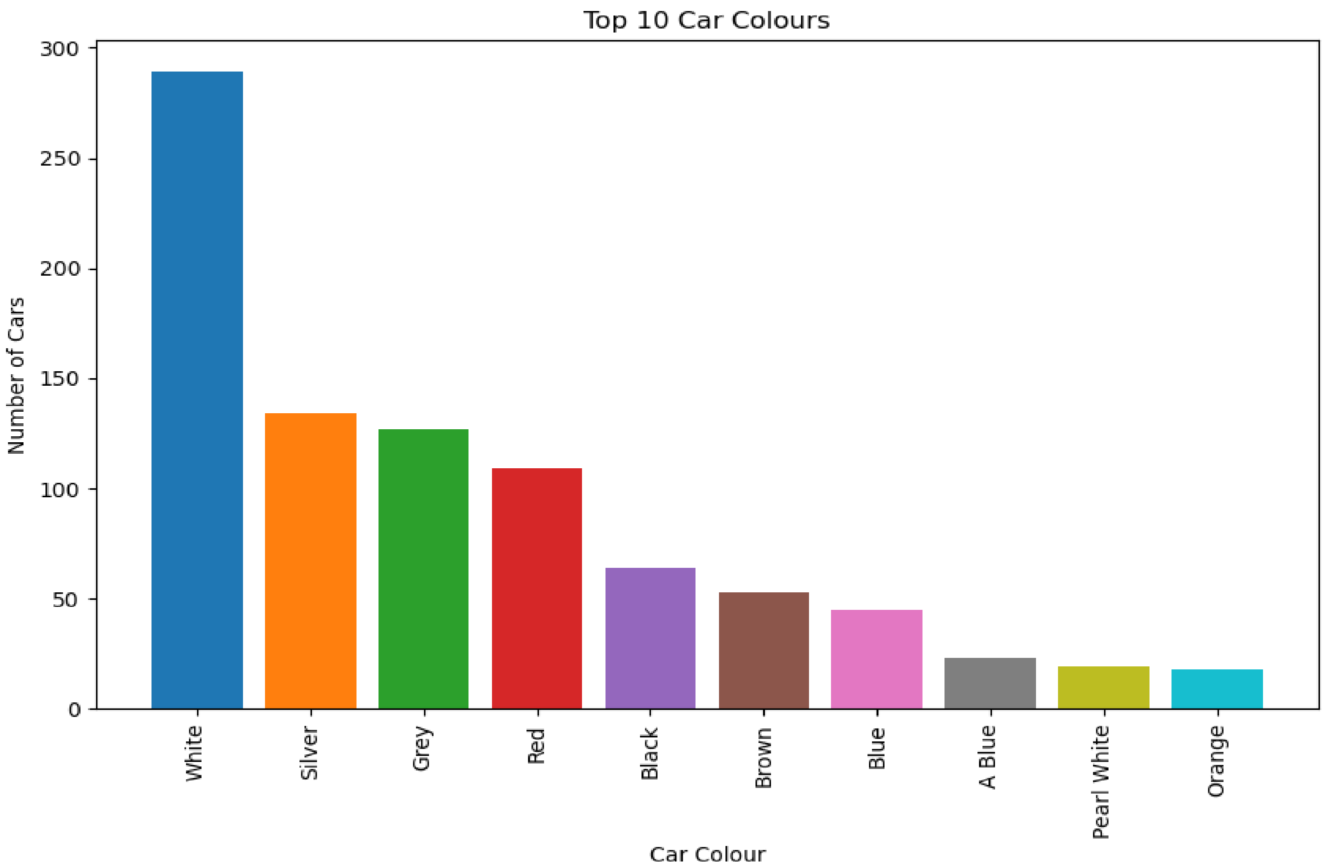


Majority of cars for resale have a petrol engine which is more than 650 cars, followed by 350 cars with diesel engine. Very few of the cars have CNG engine and negligible number of cars are hybrid or on LPG. Therefore, we can assume that petrol and diesel cars are more in demand in the used car market.

Top 10 Colors for Cars

```python
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import numpy as np

# Get the top 10 car colors
top_colors = df['Colour'].value_counts().iloc[:10]

# Define color palette using a colormap (e.g., 'tab10' colormap for 10 colors)
colors = cm.tab10(np.linspace(0, 1, len(top_colors)))

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(top_colors.index, top_colors.values, color=colors)
plt.xlabel('Car Colour')
plt.ylabel('Number of Cars')
plt.title('Top 10 Car Colours')
plt.xticks(rotation=90)  # Rotate x labels for better readability
plt.show()
```

Python

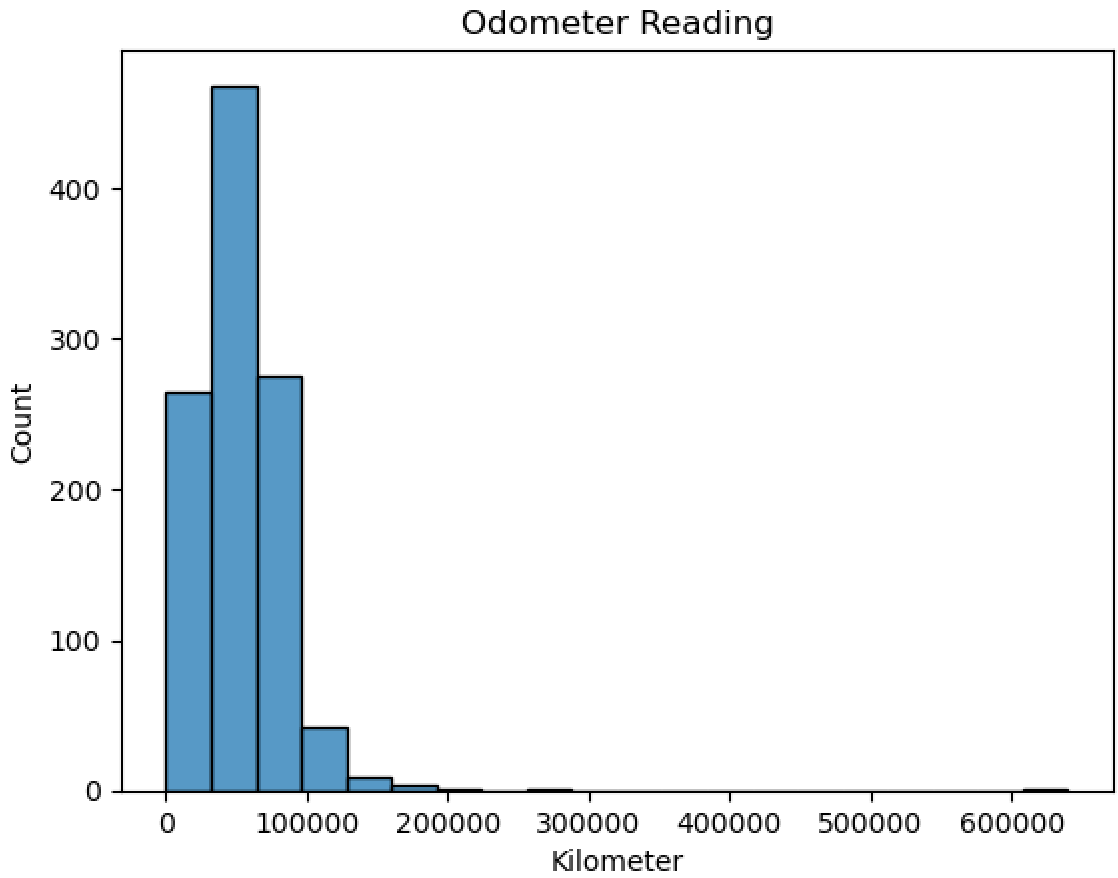


Although color of car has no impact on the cars performance, but still it plays a major role in the car demand. From the graph, we can see that white color is the most preferred color for the used cars, followed by silver, grey, red and black. Therefore, we can assume that white, silver, grey, red and black color cars are more in demand in the used car market will have a good resale value.

Odometre Reading

```python
sns.histplot(x = 'Kilometer', data = df, bins = 20).set_title('Odometer Reading')
```

Python

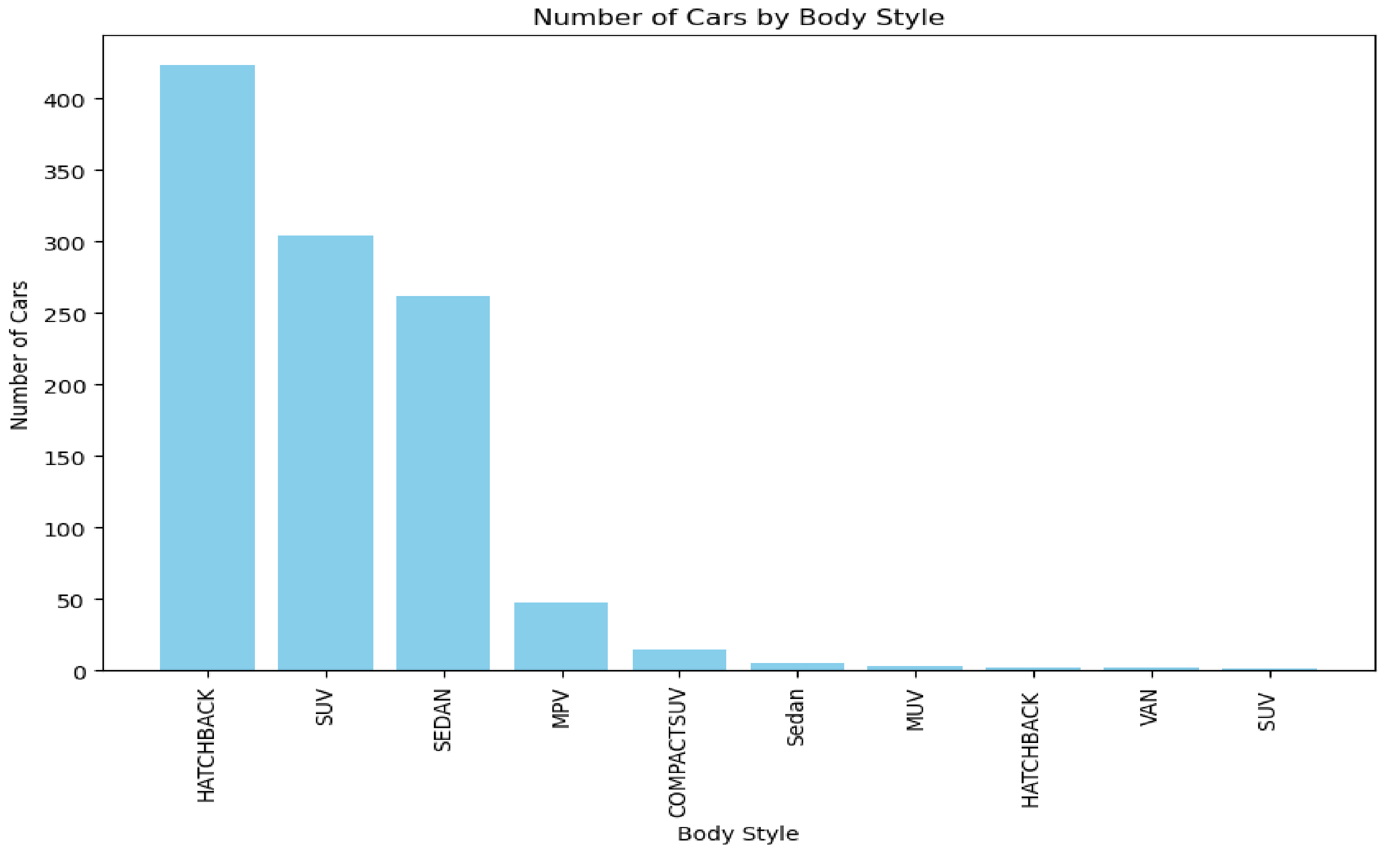


This graph shows the distribution of the odometer readings of the cars in the dataset. From the graph, we can see that most of the cars have odometer reading less than 100000 km. To be more particular majority of cars are driven for 30000 km to 50000 km. Thefore, we can assume that cars with odometer reading less than 100000 km are more in demand in the used car market will have a good resale value

Body Style

```python
import matplotlib.pyplot as plt

# Count the number of cars by BodyStyle
body_style_counts = df['BodyStyle'].value_counts()

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(body_style_counts.index, body_style_counts.values, color='skyblue')
plt.xlabel('Body Style')
plt.ylabel('Number of Cars')
plt.title('Number of Cars by Body Style')
plt.xticks(rotation=90)  # Rotate x labels
plt.show()
```

Python

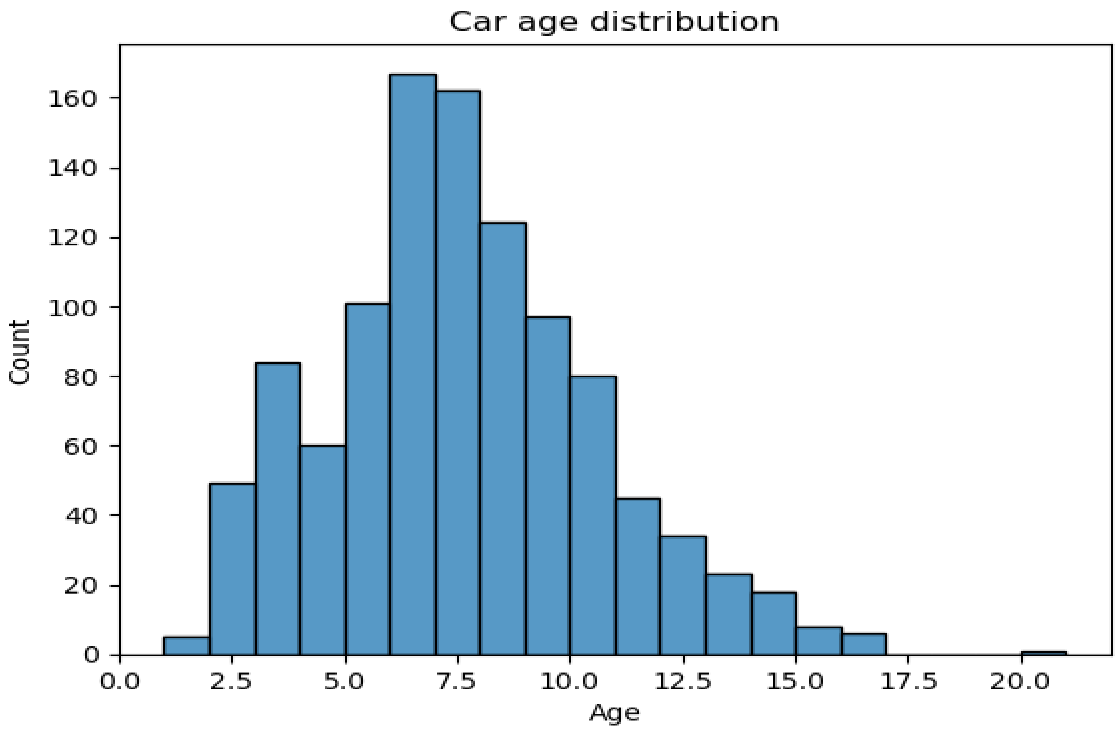


According to this graph, most of the cars have HatchBack, SUV and Sedan body style, which tells us about the market demand of these body styles. Therefore, we can assume that cars with HatchBack, SUV and Sedan body style are more in demand in the used car market will have a good resale value

Car Age Distribution

```
sns.histplot(x = 'Age', data = df, bins = 20).set_title('Car age distribution')
```

Python

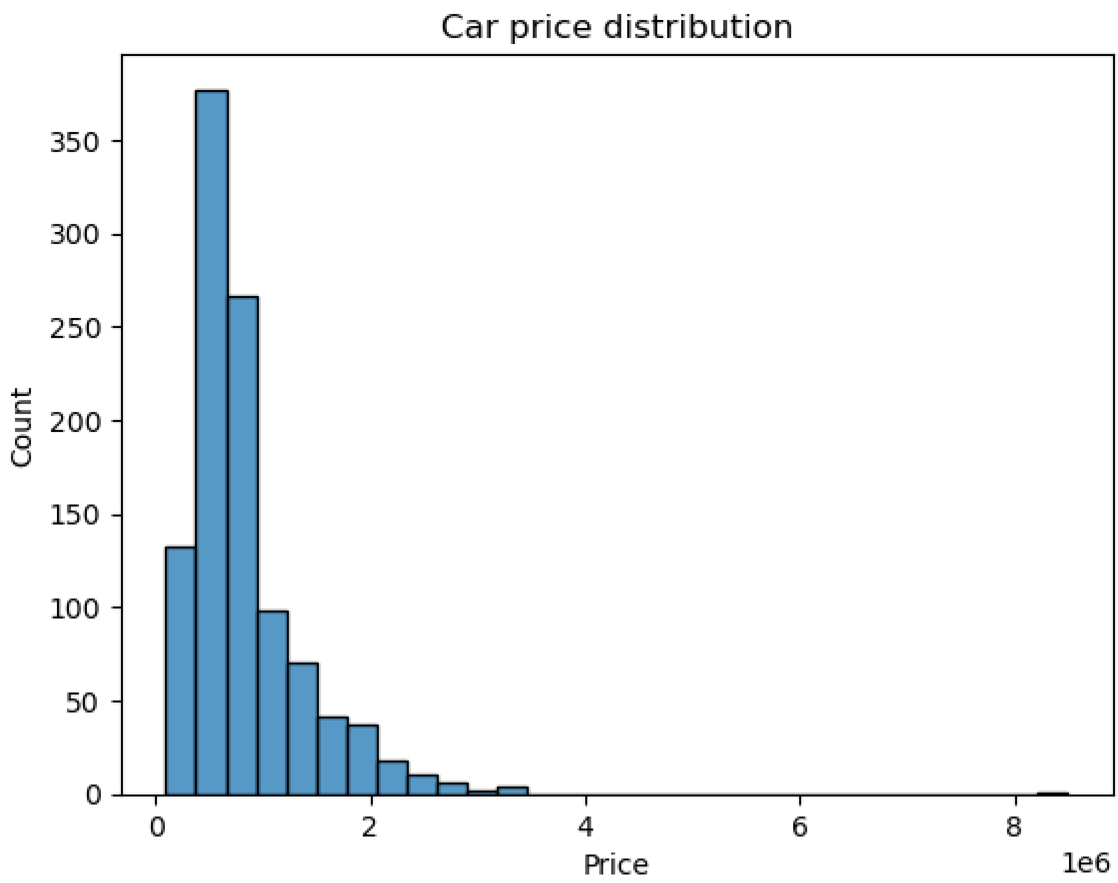


Age of the car plays an important role in deciding its resale value. Here, in the dataset cars that age between 5 to 7 years are more in number. Moreover majority of the cars age more than 5 years, which affect their resale value. However, there are still significant number of cars with age less than 5 years, thereofore, I assume they would have higher resale value. In addition to that, we can see than one car has age near 20 years which could be an outlier

Price Distribution

```
sns.histplot(x = 'Price',data =df, bins = 30).set_title('Car price distribution')
```

Python

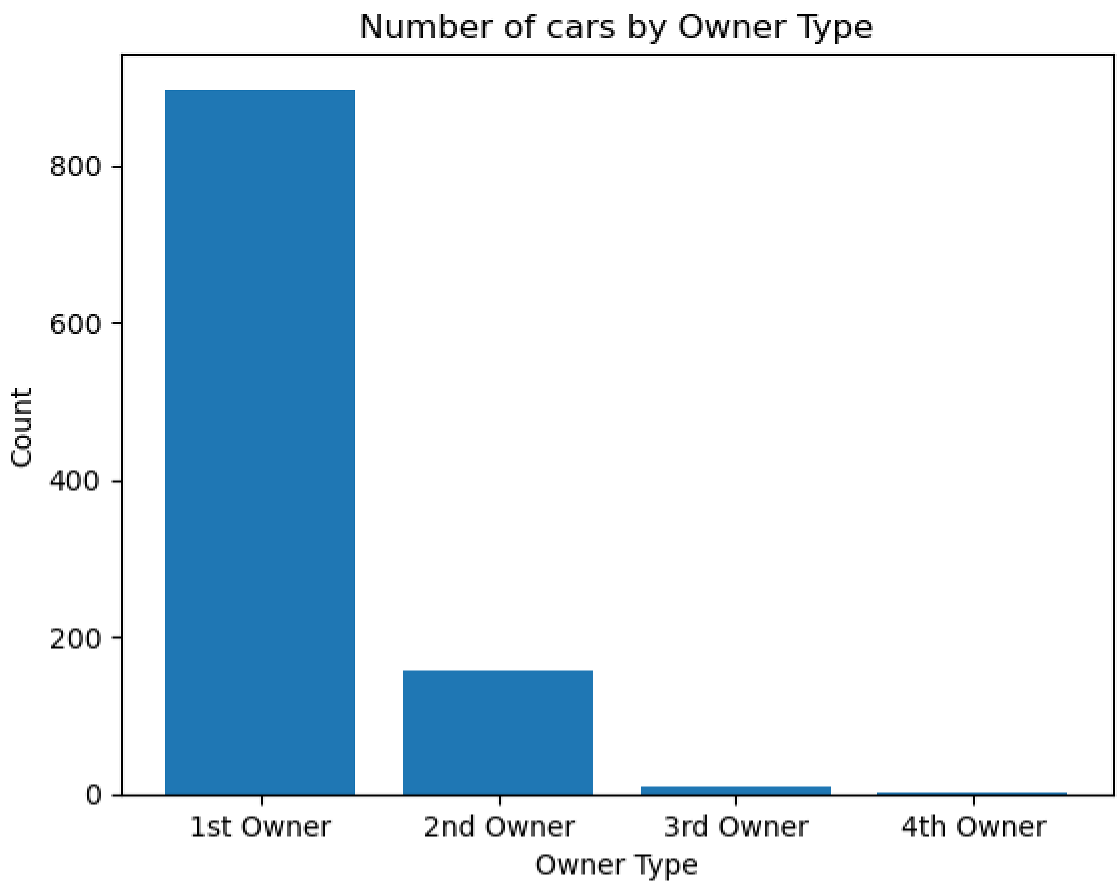


This graph help us to know about the distribution of the car prices in the dataset. In the dataset, most of the cars have price is between 3 to 9 lakhs, with maximum cars between 3 to 6 lakhs. Therefore, we can assume that cars with price between 3 to 9 lakhs are more in demand in the used car market. Moreover there are some cars with resale price more than 20 lakhs, which could be possible for luxury cars or it could be an outlier

Car Owner Type

```
import matplotlib.pyplot as plt

# Calculate the count of each 'Owner' category
owner_counts = df['Owner'].value_counts()

# Create a bar chart
plt.bar(owner_counts.index, owner_counts.values)

# Set title and labels
plt.title('Number of cars by Owner Type')
plt.xlabel('Owner Type')
plt.ylabel('Count')

# Show the plot
plt.show()
```

Python

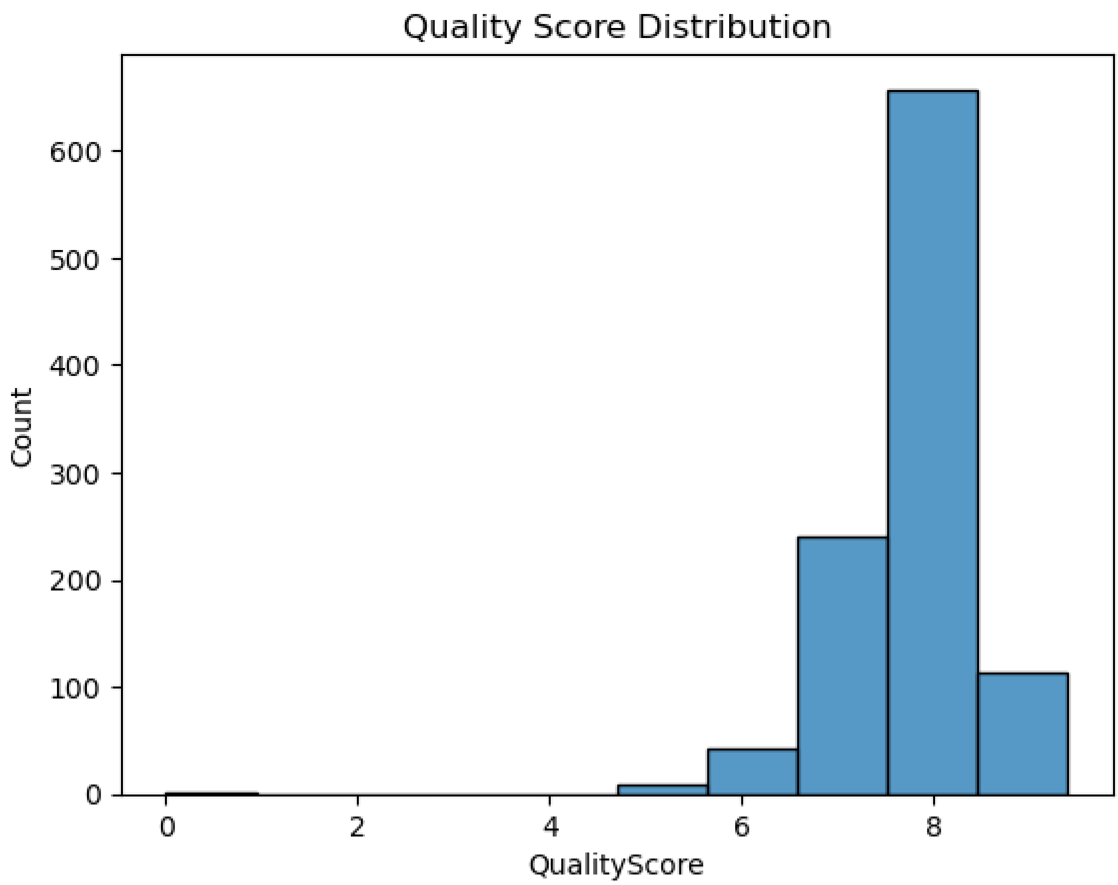


The car owner type has a huge impact on its resale value. Majority of the cars that are been sold are 1at Owner cars followed by 2nd Owner cars which are significantly less in number as compared to 1st Owner. Moreover, the 3rd and 4th owner cars are very less in number. Therefore, we can assume that 1st Owner cars are more preferred in the used car market and have a good resale value

Quality Score Distribution

```
sns.histplot(x = 'QualityScore', data = df, bins = 10).set_title('Quality
Score Distribution')
```

Python



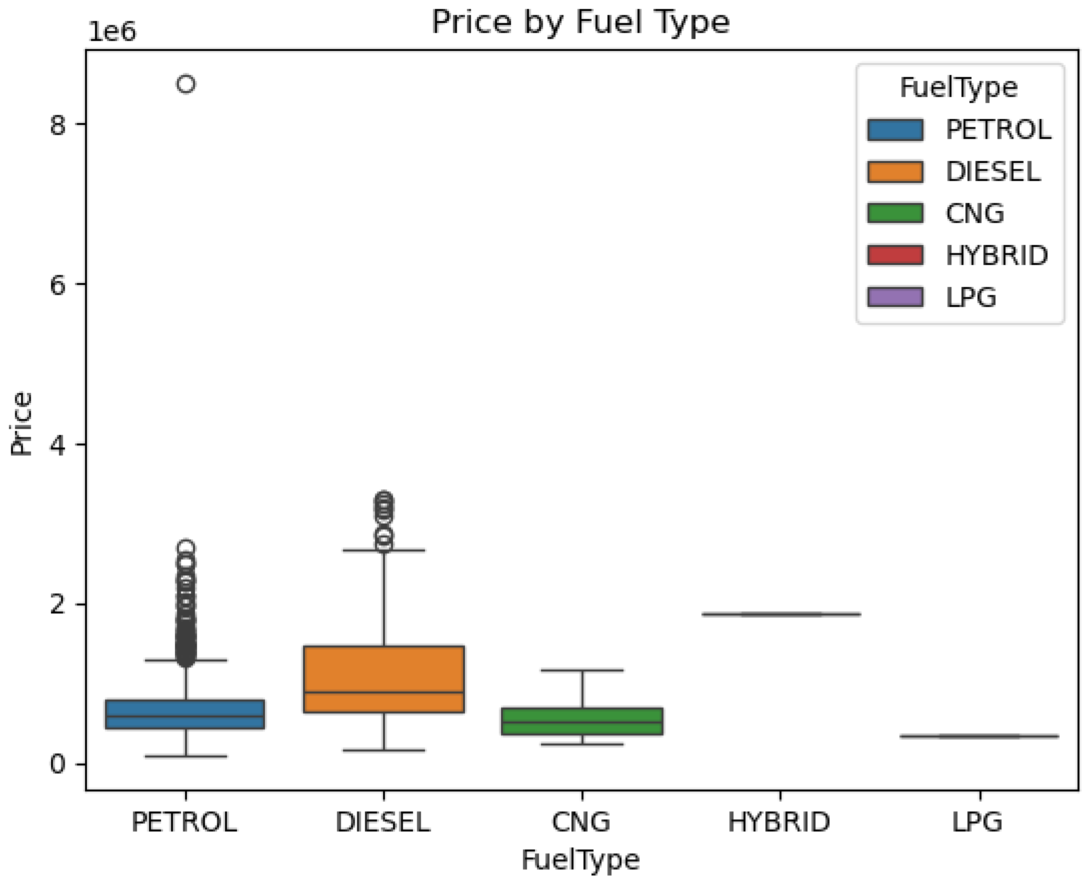
Quality score is an important feature which has a huge impact on the car sales and its preference by the customers. Cars with higher quality scores tend to have a much higher resale value and are more preferred by the customers. In the dataset, most of the cars have a decent quality score between 7-8, which highlights that the cars are thoroughly checked before being sold in the used car market. However, there are some cars with quality score less than 5, which could be due to the fact that they are not in good condition or they are very old.

Till now, I have visualized the distribution of the data and got a better understanding of the data. Now, I will be looking at the relationship between the Car Price aans the independent variables.

Car Fuel Type

```python
sns.boxplot(x = 'FuelType', y = 'Price', data = df,  hue = 'FuelType').set_title(
```

Python

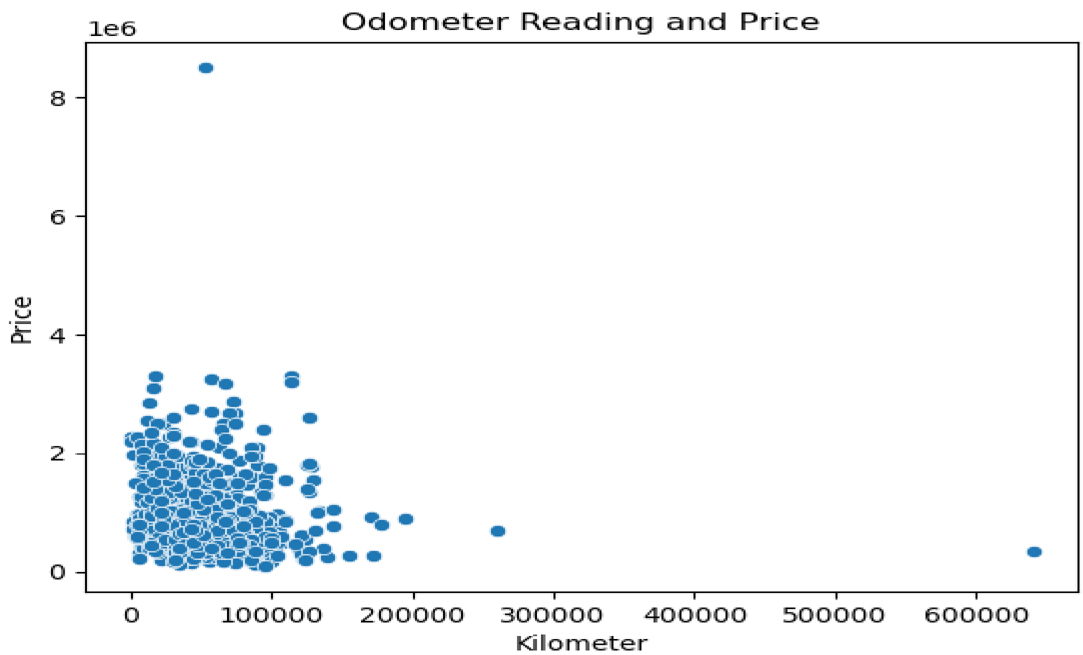


The above plots visualizes the relationship between the car fuel type and its resale value. In the boxplot we can see than cars with diesel fuel type have higher resale value than petrol and CNG and LPG.

Odometer Reading and Price

```python
sns.scatterplot(x = 'Kilometer', y = 'Price', data = df).set_title('Odometer Reading and Price')
```

Python



In the scatter plot we can see than the data is concentrated near the origin, which means that most of the cars have odometer reading less than 100000 km. In addition to that the cars with less odometer reading shows higher resale value and as the odometer reading increases the resale value decreases. Therefore, my hypothesis was correct that cars with odometer reading less than 100000 km are more in demand in the used car market will have a good resale value.

Quality Score and Price

```python
sns.scatterplot(x = 'QualityScore', y = 'Price', data = df).set_title('Quality Score and Price')
```

We can see a very high concentration near the quality score 7 and above having much higher price than the cars with quality score less than 7. Therefore, we can assume that cars with quality score 7 and above are more preferred in the used car market and have a good resale value

After conducting thorough exploratory data analysis (EDA), we have developed a solid understanding of the car price prediction dataset, including key insights about the features and their relationships with car prices. We identified potential predictors of car prices, such as the make, model, year of manufacture, mileage, engine type, and other relevant characteristics. We examined correlations between these features and the target

variable (price), and evaluated the distribution of prices across different car categories, ages, and conditions.

Data Preprocessing Part 2

Dropping column car model beacause, it has too many unique values and it will increase the dimensionality of the dataset

Label Encoding

```python
df.drop('Model', axis = 1, inplace = True)
```

Python

```python
cols = df.select_dtypes(include=['object']).columns

from sklearn.preprocessing import LabelEncoder
#Label encoding object
le = LabelEncoder()

#label encoding for object type columns
for i in cols:
    le.fit(df[i])
    df[i] = le.transform(df[i])
    print(i, df[i].unique())
```

Python

```
Company [12  7 19  5 13 21 11  6 17 16  9  4 20 10  1  3 18 14  0  8 22 15  2]
FuelType [4 1 0 2 5 3]
Colour [61 56 34  0  9 11 66 47 49 38 14 71 72 30 74 52 39 28 60  7 54 62 40 13
 20 70 63 12 24 23 35 26 29 15 31  1 68  4  8 73 22 44 57 65 42 50 32 64
 19 43 46 33 16 27 53 25 10 69 51 17  6 48 59 58  5  3 18 45 67 36 21 55
  2 37 75 41]
BodyStyle [1 5 3 6 2 9 4 0 8 7]
Owner [0 1 2 3]
DealerState [2 4 0 1 8 7 3 6 9 5]
DealerName [52 38  4  1 56 29  0 34 47 51 11 21  9 10 43 33  7 16  5 12 42 17 27 50
 45  6 20 36 23 41 32 31 18  2 48 15 54 40 55 13 49 25 35 46 24 14 44 19
 39 28 26  3 53 30  8 22 37]
City [ 0 10  2  3  9  4  5  8  1  7  6]
```

Outlier Removal

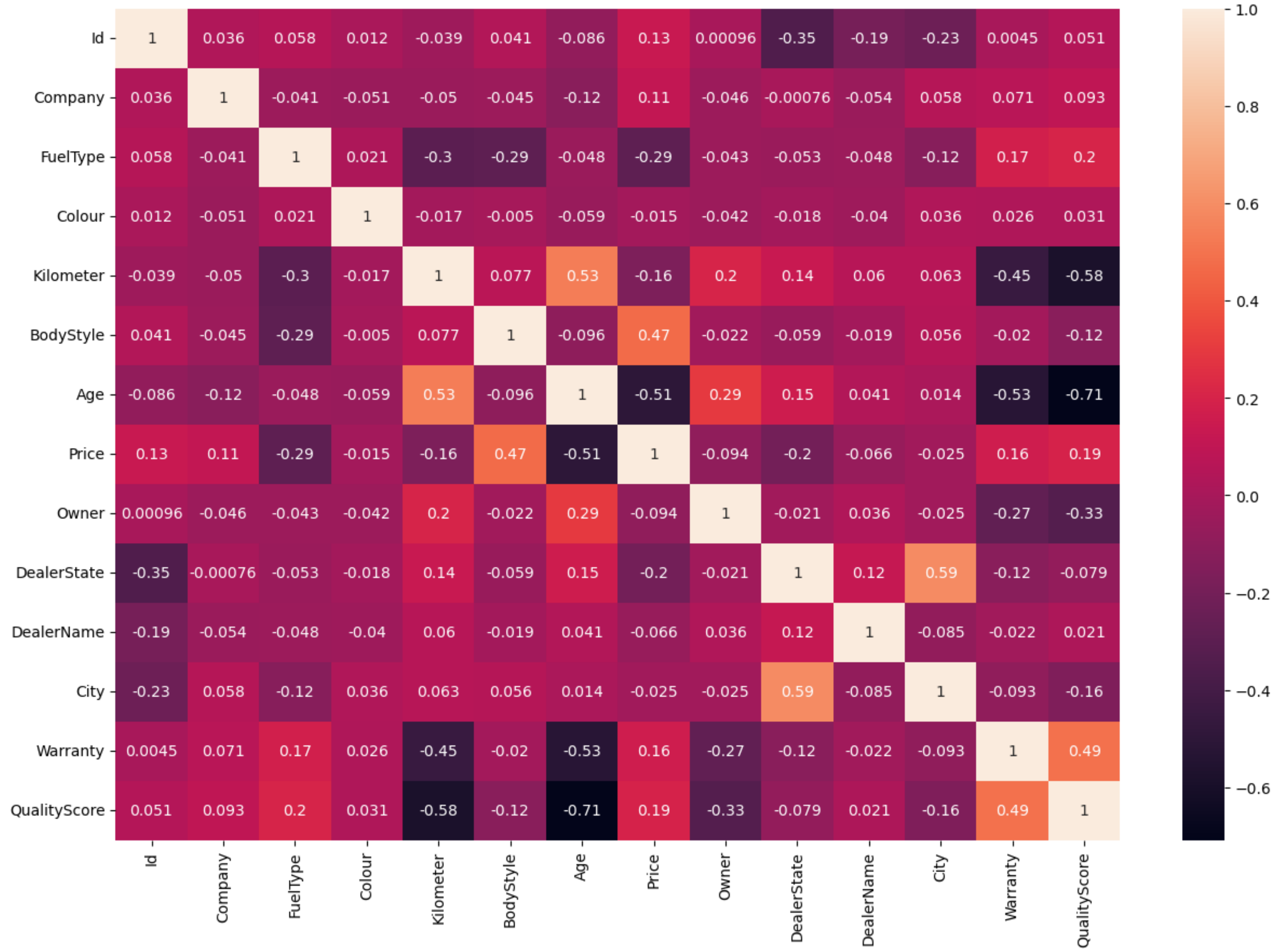
```python
import numpy as np
import pandas as pd
from scipy.stats import zscore

# Select columns with numerical data (int64 or float64)
cols = df.select_dtypes(include=['int64', 'float64']).columns

# Calculate Z-scores for each numerical column
z_scores = np.abs(zscore(df[cols]))

# Define the threshold for outliers (e.g., 3)
threshold = 3

# Filter out the rows where any Z-score is greater than the threshold
df_cleaned = df[(z_scores < threshold).all(axis=1)]
print(df_cleaned)

# Optionally, print the number of rows removed (if any)
print(f"Number of rows before removing outliers: {df.shape[0]}")
print(f"Number of rows after removing outliers: {df_cleaned.shape[0]}")

# The cleaned DataFrame is now in df_cleaned
```

Python

```
      Company  FuelType  Colour  Kilometer  BodyStyle  Age     Price  Owner  \
0          12         4      61      33197          1    6  575000.0      0
1          12         4      56      10322          1    3  435000.0      0
2           7         4      34      37889          1    9  470000.0      0
3          19         4       0      13106          1    4  990000.0      0
4           5         1      61     104614          1   14  270000.0      1
...       ...       ...     ...        ...        ...  ...       ...    ...
1059        7         4      71      42918          1    4  715000.0      0
1060        7         4      71      78910          5    5  500000.0      0
1061       11         1      71      76000          6   11  575000.0      0
1062       12         1      61      80120          1    6  771000.0      0
1063        6         1      68      77500          5   10  499000.0      1

      DealerState  DealerName  City  Warranty  QualityScore
0               2          52     0         1           7.8
1               2          38     0         1           8.3
2               2           4     0         1           7.9
3               2           1     0         1           8.1
4               2          56     0         0           7.5
...           ...         ...   ...       ...           ...
1059            5          22     6         1           8.3
1060            5          37     6         0           7.8
1061            5          37     6         0           6.8
1062            5          37     6         0           7.4
1063            5          37     6         0           6.8

[1034 rows x 13 columns]
Number of rows before removing outliers: 1064
Number of rows after removing outliers: 1034
```

Correlation Matrix Heatmap

```
plt.figure(figsize=(15,10))
sns.heatmap(df_cleaned.corr(), annot=True)
```



Modelling and Machine Learning

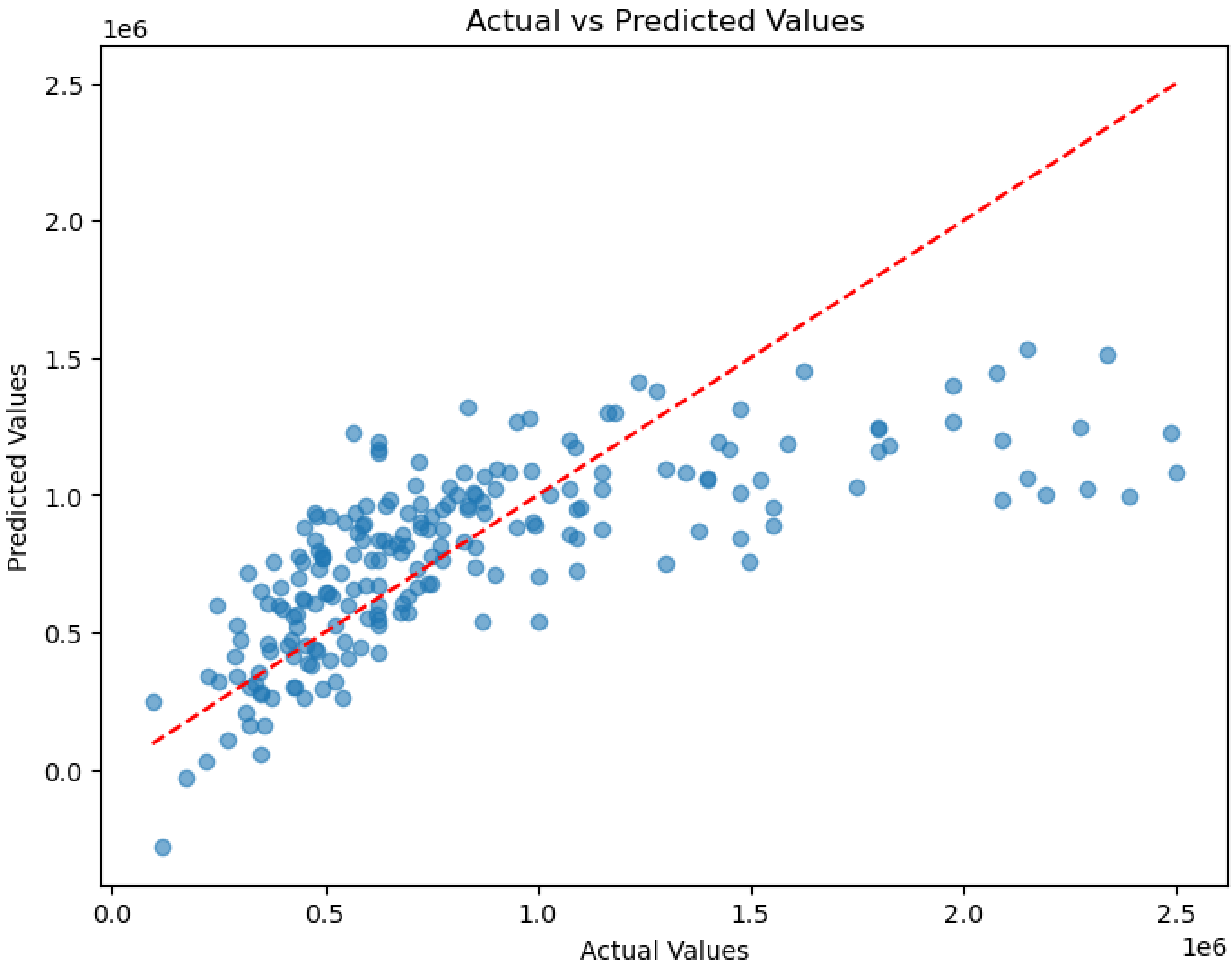
The goal of this stage is to build, evaluate, and compare various machine learning models to predict car Prices based on their age,City,Warranty and other features.This process involves selecting features, splitting the data into training and testing sets, building and training models, tuning hyperparameters, and evaluating each model to identify the best-performing one.

Train Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_cleaned.drop('Price',axis=1),
                                                    df_cleaned['Price'], test_size=0.2,
                                                    random_state=42)
```

```
X = df_cleaned.drop(columns=['Price'])
y = df_cleaned['Price']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Scaling the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Linear Regression model
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
# Predictions
y_pred = lin_reg.predict(X_test)
# Model evaluation
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
# Print the evaluation metrics
print(f'Mean Absolute Error: {mae:.2f}')
print(f'Mean Squared Error: {mse:.2f}')
print(f'R-squared: {r2:.2f}')
# Plotting Actual vs Predicted values
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.6)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Actual vs Predicted Values")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red',
         linestyle='--')  # Line for perfect predictions
plt.show()
```

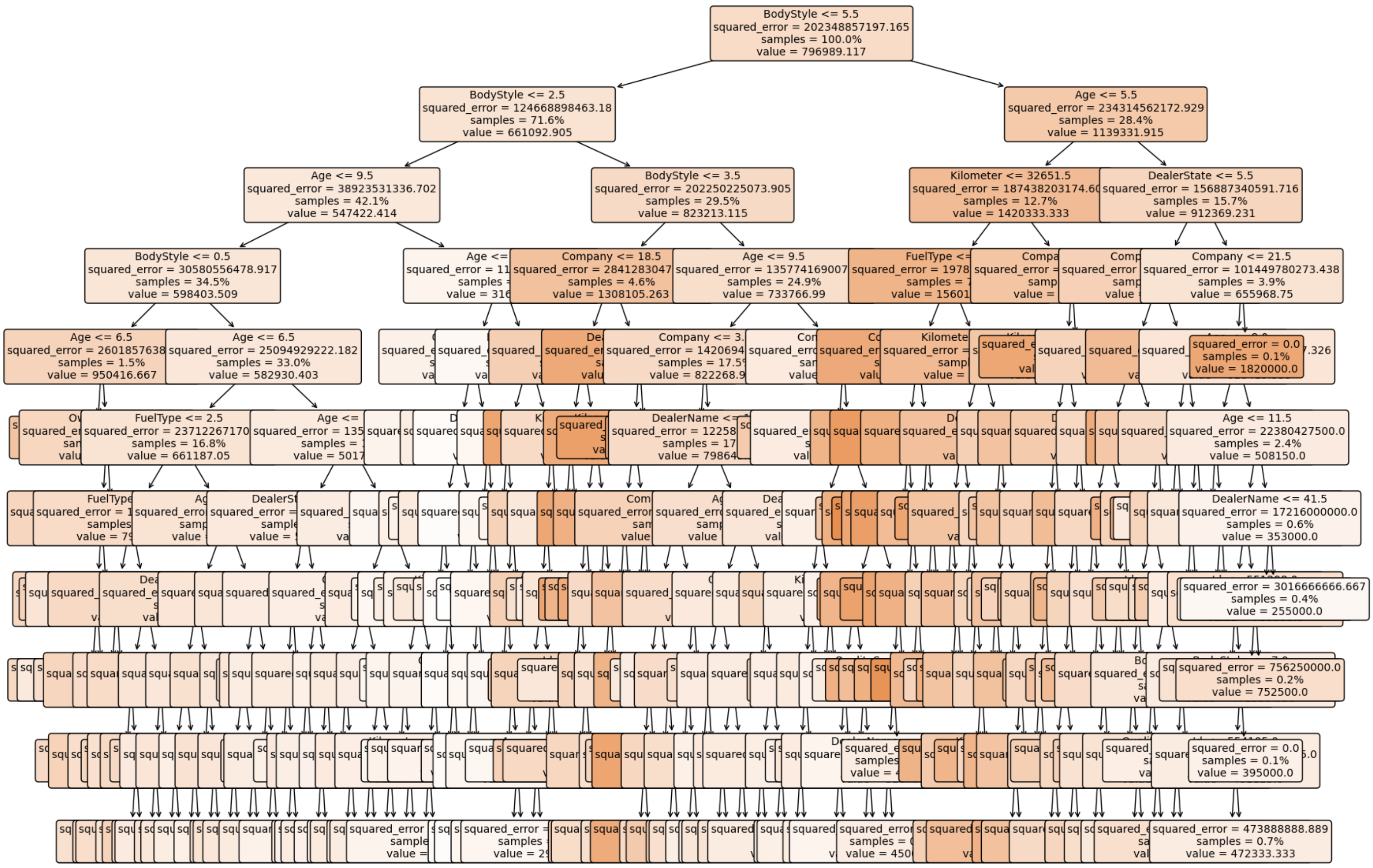
```
Mean Absolute Error: 267041.38
Mean Squared Error: 139734551886.73
R-squared: 0.48
```



```
dt_regressor = DecisionTreeRegressor(max_depth=10, min_samples_split=5, min_samples_leaf=1,
                                     random_state=42)

dt_regressor.fit(X_train, y_train)

y_pred = dt_regressor.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'Mean Absolute Error: {mae:.2f}')
print(f'R-squared: {r2:.2f}')

plt.figure(figsize=(20, 15))
plot_tree(dt_regressor,
          filled=True,
          feature_names=X.columns,
          rounded=True,
          proportion=True,
          fontsize=10)
plt.title("Full Decision Tree Visualization")
plt.show()
```

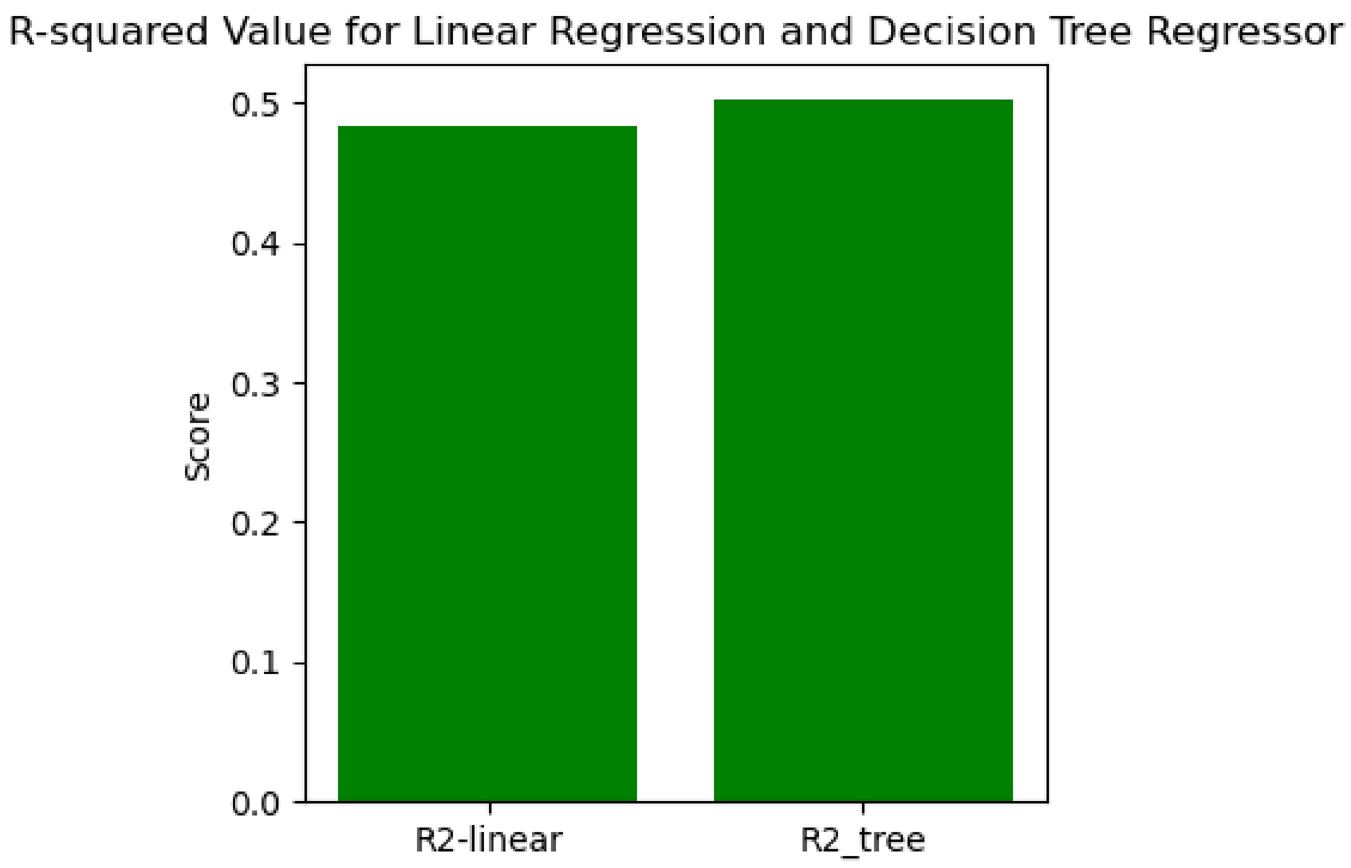
```
Mean Squared Error: 134735733123.07
Mean Absolute Error: 235237.65
R-squared: 0.50
```

Full Decision Tree Visualization



BAR PLOT

```
import matplotlib.pyplot as plt
plt.figure(figsize=(4, 4))
plt.bar(['R2-linear','R2_tree'], [r2_linear,r2_tree], color='green')
plt.ylabel('Score')
plt.title('R-squared Value for Linear Regression and Decision Tree Regressor')
plt.show()
```



CONCLUSION

Demand and Price Insights

1. **Demand for Budget Cars:** Demand is significantly higher for lower-priced used cars, suggesting that many customers are drawn to budget options over luxury brands in the used car market. Luxury brands such as MG, Mercedes-Benz, BMW, Volvo, and KIA command high prices, while brands like Maruti Suzuki, Hyundai, Honda, Mahindra, and Tata are in greater demand due to their affordability. This trend suggests that many customers may prefer to purchase new luxury cars instead of used ones.

2. **Fuel Type and Pricing:** Analyzing price distributions by fuel type revealed that cars powered by diesel tend to be priced slightly higher than petrol cars. This insight was further confirmed by z-score analysis, where diesel cars had z-scores indicating higher relative prices compared to petrol, likely due to their longer durability and fuel efficiency.

3. **Color Trends:** The data analysis showed that common colors (like white, grey, silver, and black) have higher demand, whereas unique colors (like burgundy, riviera red, dark blue, and black magic) often have higher prices. This suggests that exotic colors add perceived value in the used car market.

4. **Mileage and Odometer Reading**: Odometer readings are also a major factor in determining price, as cars with lower readings (under 10,000 km) tend to have higher prices. In terms of z-scores, cars with low mileage show positive z-scores in price, indicating they are more expensive than the average car, likely because of their relatively new condition.

5. **Body Style and Preferences:** Customers prefer body styles like Hatchback, SUV, and Sedan, which also have relatively higher resale values. Meanwhile, body styles like MPV, SUV, and Sedan are among the most expensive options in the market.

6. **Age and Resale Value:** Car age inversely affects resale value: as car age increases, resale value decreases. Cars less than five years old typically command higher prices. This trend was further verified through linear regression analysis, which showed a negative coefficient for car age, reinforcing that older cars tend to have lower prices.

7. **Location and Dealer Influence:** Car prices vary significantly by location, with Delhi, Maharashtra, and Rajasthan having some of the highest prices. Similarly, dealers like Car Estate, Star Auto India, and Car Choice list cars at higher prices, possibly indicating their premium positioning in the market.

8. **Ownership and Warranty:** Cars with a first-owner status generally have higher demand and price, as they offer a sense of assurance about the car’s condition.

Warranty availability also contributes to a higher price due to added customer confidence.

9. **Quality Score:** A car's quality score is positively correlated with price, as expected. Higher-quality cars command higher prices in the market.

Handling Outliers

Data Cleaning with Z-score Method: Before modeling, I used the **Z-score method** to identify and remove outliers in continuous features such as price, odometer reading, and quality score. This process helped eliminate extreme values that could skew the predictions, resulting in a cleaner, more robust dataset.

Machine Learning Model Analysis

To predict car prices, We used linear regression, decision tree regressor models. Each model had its strengths and provided unique insights:

1. **Linear Regression:** This model helped quantify relationships between price and features, especially for numeric variables like car age and mileage. It showed how age negatively impacts price, with a high coefficient indicating the significant effect of newer models on resale value.

2. **Decision Tree Regressor:** This model captured non-linear relationships and interactions between features, such as brand and location, that influence price. Decision trees highlighted key price determinants like car age, body style, and brand. However, it lacked robustness due to its tendency to overfit data, capturing only localized patterns.

Using these techniques allowed for a well-rounded understanding of factors influencing car prices and helped build predictive models with improved accuracy and generalization for real-world applications.