Car price Analysis

Author : Asish Jose

Date : 04/02/2025

Objective

Introduction

This report presents an analysis of the car price dataset to explore whether specific features can be used to predict car prices. The dataset includes various attributes such as engine type, fuel system, dimensions, horsepower, and fuel efficiency.

Dataset Overview

- **Source:** Kaggle
- **Total Variables:** 27
- **Target Variable:** Price
- **Categorical Features:** 'make', 'num-of-doors', 'body-style', 'drive-wheels', 'engine-location', 'engine-type', 'fuel-system', 'aspiration-std', 'aspiration-turbo', etc.
- **Numerical Features:** 'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-size', 'bore', 'stroke', 'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg', 'highway-L/100km', 'price'.

Data Cleaning

Handling Missing Values

- Identified missing values in **normalized-losses, num-of-doors, bore, stroke, horsepower, peak-rpm, and price** columns.
- Replaced missing **numerical values** with the **mean** of the respective column.
- Replaced missing **categorical values** with the **mode** of the column.

```
[11]: data.isnull().sum()

[11]: symboling              0
      normalized-losses     41
      make                   0
      fuel-type              0
      aspiration             0
      num-of-doors           2
      body-style             0
      drive-wheels           0
      engine-location        0
      wheel-base             0
      length                 0
      width                  0
      height                 0
      curb-weight            0
      engine-type            0
      num-of-cylinders       0
      engine-size            0
      fuel-system            0
      bore                   4
      stroke                 4
      compression-ratio      0
      horsepower             2
      peak-rpm               2
      city-mpg               0
      highway-mpg            0
      price                  4
      dtype: int64
```

```
avg_bore = data['bore'].astype('float').mean()
data['bore'].replace(np.NaN, avg_bore, inplace = True)

avg_stroke = data['stroke'].astype('float').mean()
data['stroke'].replace(np.NaN, avg_stroke, inplace = True)

avg_horsepower = data['horsepower'].astype('float').mean()
data['horsepower'].replace(np.NaN, avg_horsepower, inplace = True)

avg_peakrpm = data['peak-rpm'].astype('float').mean()
data['peak-rpm'].replace(np.NaN, avg_peakrpm, inplace = True)

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

replacing null values using mean values

```
data['num-of-doors'].value_counts()
```

```
four    114
two      89
Name: num-of-doors, dtype: int64
```

```
data['num-of-doors'].value_counts().idxmax()
```

```
'four'
```

```
data['num-of-doors'].replace(np.NaN, 'four', inplace = True)
data.head()
```

replacing null values in categorical values such as num of doors using mode

Drop all values(rows) with no price

```
data.dropna(subset = ['price'], axis = 0, inplace = True)
data.shape
```

```
(201, 26)
```

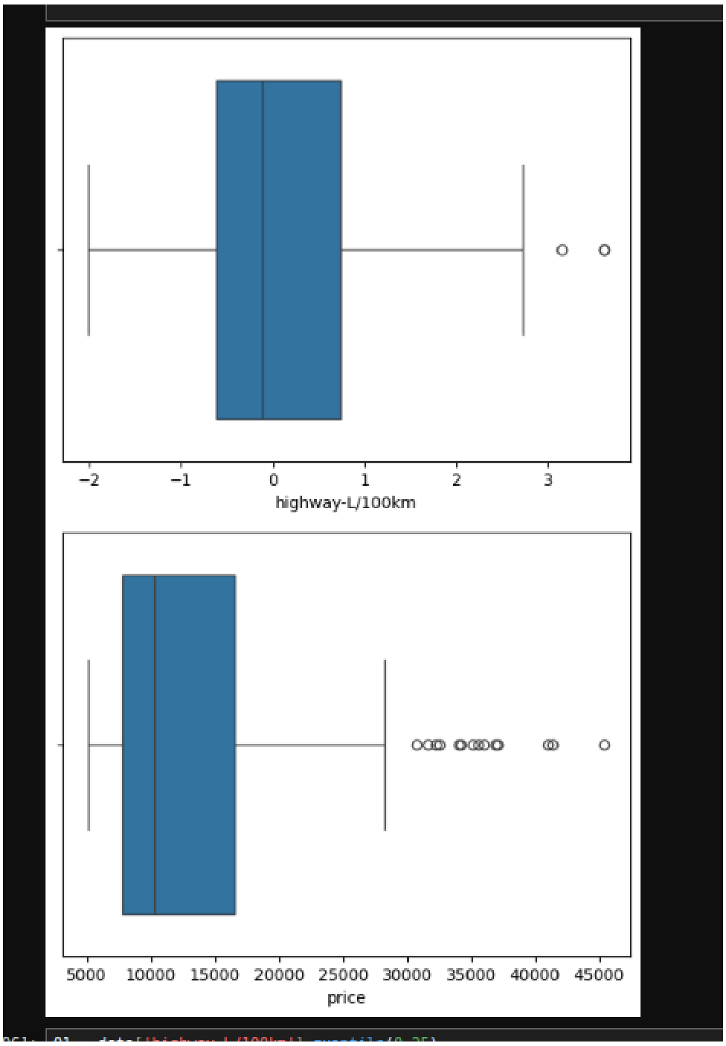
```
data['price'].isnull().sum()
```

```
0
```

droping rows with null values in price column

Identifying and Dealing with Outliers

- Used **boxplots** to detect outliers in price and highway-L/100km



identified outliers using boxplots

```
Q1 = data['highway-L/100km'].quantile(0.25)
Q3 = data['highway-L/100km'].quantile(0.75)
IQR = Q3 - Q1

# Remove outliers
data = data[(data['highway-L/100km'] >= Q1 - 1.5*IQR) & (data['highway-L/100km'] <= Q3 + 1.5*IQR)]
```

```
Q1 = data['price'].quantile(0.25)
Q3 = data['price'].quantile(0.75)
IQR = Q3 - Q1

# Remove outliers
data = data[(data['price'] >= Q1 - 1.5*IQR) & (data['price'] <= Q3 + 1.5*IQR)]
```

Data Transformation and Normalization

- Converted **city-mpg** and **highway-mpg** to **L/100km** for better interpretability.

- Standardized numerical features to ensure better model performance and prevent bias due to varying scales.

```
data['highway-mpg'] = 235.215/data['highway-mpg']
```

```
data['highway-L/100km']
```

```
0       8.711667
1       8.711667
2       9.046731
3       7.840500
4      10.691591
         ...
200     8.400536
201     9.408600
202    10.226739
203     8.711667
204     9.408600
Name: highway-L/100km, Length: 201, dtype: float64
```

```
data.drop('highway-L/100km', axis = 1, inplace=True)
```

```
data.rename(columns = {'highway-mpg': 'highway-L/100km'}, inplace = True)
data.head()
```

```
52]: data['length'] = data['length']/data['length'].max()
     data['width'] = data['width']/data['width'].max()
     data.head()
```

```
[54]: data[['length','width']]
```

[54]:

	length	width
0	0.811148	0.890278
1	0.811148	0.890278
2	0.822681	0.909722
3	0.848630	0.919444
4	0.848630	0.922222
...	...	...
200	0.907256	0.956944
201	0.907256	0.955556
202	0.907256	0.956944
203	0.907256	0.956944
204	0.907256	0.956944

201 rows × 2 columns

normalized values of lenght and width

This ensures that the dataset is clean, consistent, and ready for further analysis or model training.

Exploratory Data Analysis (EDA)

Summary Statistics

- Generated descriptive statistics for numerical columns to understand data distribution.

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

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore
count	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000	187.000000
mean	0.844920	121.925134	98.310160	0.831156	0.911735	0.899388	2483.262032	118.652406	3.307428
std	1.236764	33.034761	5.308094	0.054418	0.025559	0.040715	443.307984	26.892299	0.261905
min	-2.000000	65.000000	86.600000	0.678039	0.837500	0.816054	1488.000000	61.000000	2.540000
25%	0.000000	95.000000	94.500000	0.799135	0.888889	0.869565	2134.000000	97.000000	3.150000
50%	1.000000	122.000000	96.500000	0.829409	0.908333	0.904682	2395.000000	110.000000	3.310000
75%	2.000000	146.500000	100.800000	0.857520	0.923611	0.928094	2823.500000	136.000000	3.540000
max	3.000000	256.000000	114.200000	0.955790	0.991667	1.000000	3750.000000	183.000000	3.940000

```
[216]: data.describe(include = ['object'])
```

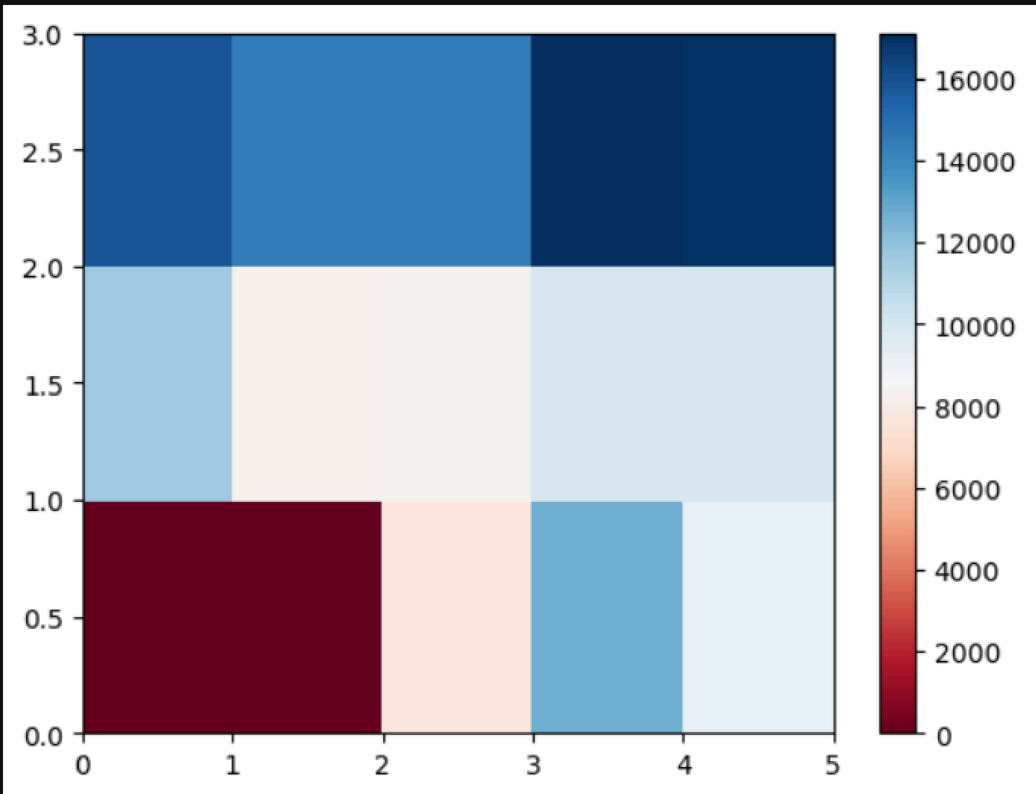
	make	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system
count	187	187	187	187	187	187	187	187
unique	21	2	5	3	1	6	5	8
top	toyota	four	sedan	fwd	front	ohc	four	mpfi
freq	32	108	85	118	187	141	157	79

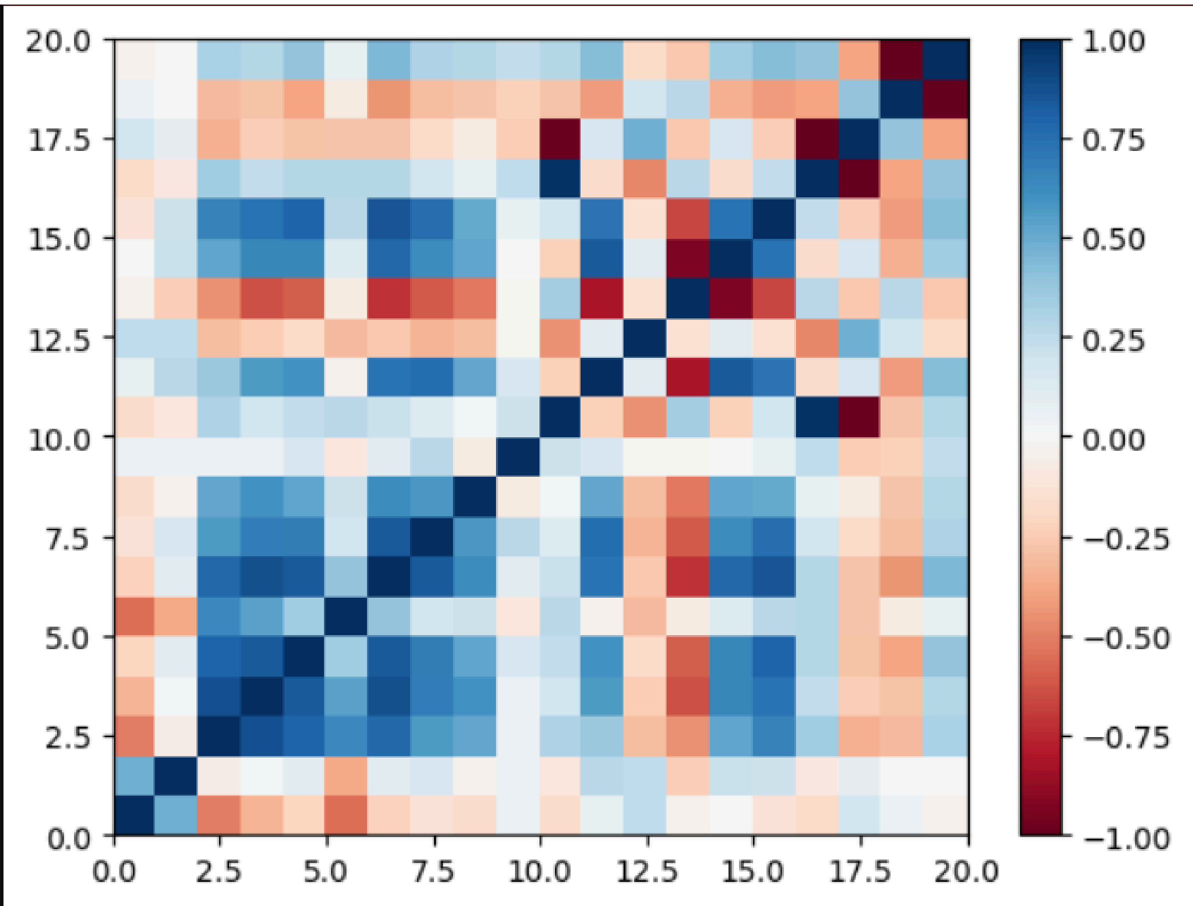
find statistical values for both numerical and categorical features in dataset separately

Data Visualization

- Plotted histograms, box plots, and scatter plots to visualize relationships among variables.
- Used correlation heatmaps to examine feature relationships with price.

[240]:
```
plt.pcolor(grouped_pivot, cmap='RdBu')
plt.colorbar()
plt.show()
```





correlation

Feature Selection

- Identified strong correlations between engine-size, curb-weight, horsepower, and price.
- Used variance inflation factor (VIF) to remove multicollinearity issues.

Statistical Analysis

Hypothesis Testing

- Applied t-tests and ANOVA to determine whether categorical variables like drive-wheels and body-style significantly impact car prices.
- Used regression analysis to confirm relationships between selected numerical features and price.

Findings and Conclusion

- Engine-size, horsepower, and curb-weight are strong predictors of car price.
- Vehicles with turbo aspiration tend to have higher prices than standard-aspiration models.
- We now have a better idea of what our data looks like and which variables are important to take into account when predicting the car price. We have narrowed it down to the following variables:
- Continuous numerical variables:
- Length
- Width
- Curb-weight
- Engine-size
- Horsepower
- City-mpg
- Highway-mpg
- Wheel-base
- Bore
- Categorical variables:
- Drive-wheels
- As we now move into building machine learning models to automate our analysis, feeding the model with variables that meaningfully affect our target variable will improve our model's prediction performance.