Analyzing Selling Price of used Cars using Python

Last Updated : 28 Mar, 2025

Analyzing the selling price of used cars is essential for making informed decisions in the automotive market. Using Python, we can efficiently process and visualize data to uncover key factors influencing car prices. This analysis not only aids buyers and sellers but also enables predictive modeling for future price estimation. This article will explore how to analyze the selling price of used cars using Python.

Step 1: Understanding the Dataset

The dataset contains various attributes of used cars, including price, brand, color, horsepower and more. Our goal is to analyze these factors and determine their impact on selling price. To download the file used in this example, click here.

> ***Problem Statement:*** *Our friend Otis wants to sell his car but isn't sure about the price. He wants to maximize profit while ensuring a reasonable deal for buyers. To help Otis we will analyze the dataset and determine the factors affecting car prices.*

Step 2: Converting `.data` File to `.csv`

If the dataset is in `.data` format, follow these steps to convert it to `.csv`:

1. Open MS Excel.
2. Go to **Data** > **From Text**.
3. Select **Comma Delimiter**.
4. Save the file as `.csv`.

Now we can proceed with loading the dataset into Python.

To analyze the data install the following Python libraries using the command below:

pip install pandas numpy matplotlib seaborn scipy

Import the following python libraries: numpy, pandas, matplotlib, seaborn and scipy.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
```

Step 4: Load the Dataset

Now, we load the dataset into a Pandas DataFrame and preview the first few rows. Let's check the first five entries of dataset.

```
df = pd.read_csv('output.csv')

df = df.iloc[: , 1:]

df.head()
```

Output:

	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.60	...	130	mpfi	3.47	2.68	9.00	111	5000	21	27	13495
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500
2	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
3	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450
4	2	?	audi	gas	std	two	sedan	fwd	front	99.8	...	136	mpfi	3.19	3.40	8.5	110	5500	19	25	15250

5 rows × 26 columns

Dataset

Step 5: Assign Column Headers

To make our dataset more readable we assign column headers:

```
headers = ["symboling", "normalized-losses", "make",
           "fuel-type", "aspiration","num-of-doors",
           "body-style","drive-wheels", "engine-location",
```

```
            "horsepower", "peak-rpm","city-mpg","highway-
mpg","price"]

df.columns=headers
df.head()
```

Output:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	compression-ratio
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0
2	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0
3	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0
4	2	?	audi	gas	std	two	sedan	fwd	front	99.8	...	136	mpfi	3.19	3.40	8.5

5 rows × 26 columns

Column Header

Step 6: Check for Missing Values

Missing values can impact our analysis. Let's check if any columns contain missing values.

```
data = df

data.isna().any()

data.isnull().any()
```

Output:

	0
symboling	False
normalized-losses	False
make	False
fuel-type	False
aspiration	False
num-of-doors	False
body-style	False
drive-wheels	False
engine-location	False
wheel-base	False
length	False
width	False
height	False
curb-weight	False
engine-type	False
num-of-cylinders	False
engine-size	False
fuel-system	False

Missing Values

Since fuel consumption is measured differently in different regions, we convert miles per gallon (MPG) to liters per 100 kilometers (L/100km)

```
data['city-mpg'] = 235 / df['city-mpg']
data.rename(columns = {'city_mpg': "city-L / 100km"}, inplace = True)

print(data.columns)

data.dtypes
```

Output:

```
Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration',
       'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',
       'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type',
       'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke',
       'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')
```

	0
symboling	int64
normalized-losses	object
make	object
fuel-type	object
aspiration	object
num-of-doors	object
body-style	object
drive-wheels	object
engine-location	object
wheel-base	float64
length	float64
width	float64
height	float64
curb-weight	int64

MPG

Step 8: Convert Price Column to Integer

The price column should be numerical, but it may contain string values like `?`. We need to clean and convert it:

```
data.price.unique()

data = data[data.price != '?']

data['price'] = data['price'].astype(int)

data.dtypes
```

Output:

```
<ipython-input-6-e3e67a236340>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['price'] = data['price'].astype(int)
```

	0
symboling	int64
normalized-losses	object
make	object
fuel-type	object
aspiration	object
num-of-doors	object
body-style	object
drive-wheels	object
engine-location	object
wheel-base	float64
length	float64
width	float64

Step 9: Normalize Features

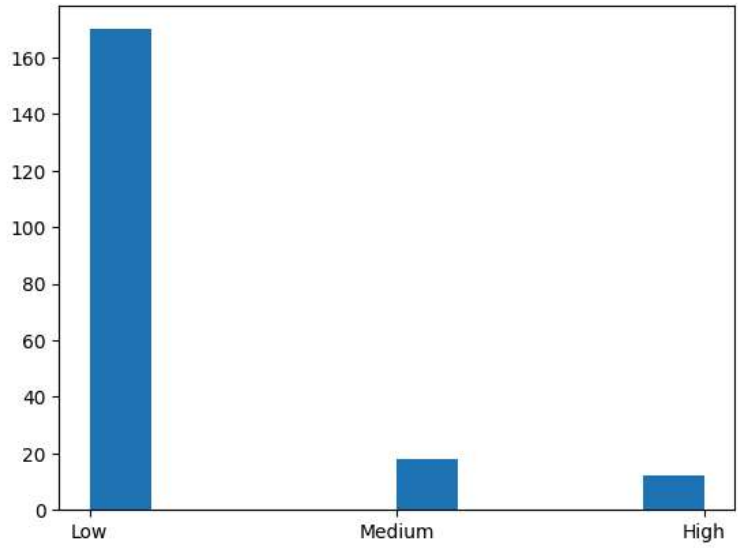
To ensure fair comparisons between different features, we normalize numerical columns. To categorize cars based on their price we divide the price range into three categories: Low, Medium and High.

```
data['length'] = data['length']/data['length'].max()
data['width'] = data['width']/data['width'].max()
data['height'] = data['height']/data['height'].max()

# binning- grouping values
bins = np.linspace(min(data['price']), max(data['price']), 4)
group_names = ['Low', 'Medium', 'High']
data['price-binned'] = pd.cut(data['price'], bins,
                              labels = group_names,
                              include_lowest = True)

print(data['price-binned'])
plt.hist(data['price-binned'])
plt.show()
```

Output:



Normalization Features

Step 10: Convert Categorical Data to Numerical

Machine learning models require numerical data. We convert categorical variables into numerical ones using one-hot encoding:

```
pd.get_dummies(data['fuel-type']).head()

data.describe()
```

Output:

	symboling	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	city-mpg	highway-mpg	price
count	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000
mean	0.830000	98.848000	0.837232	0.915250	0.899523	2555.705000	126.860000	10.170100	9.937914	30.705000	13205.690000
std	1.246557	6.038261	0.059333	0.029207	0.040610	518.594552	41.650501	4.014163	2.539415	6.827227	7966.982558
min	-2.000000	86.600000	0.678039	0.837500	0.799331	1488.000000	61.000000	7.000000	4.795918	16.000000	5118.000000
25%	0.000000	94.500000	0.809937	0.891319	0.869565	2163.000000	97.750000	8.575000	7.833333	25.000000	7775.000000
50%	1.000000	97.000000	0.832292	0.909722	0.904682	2414.000000	119.500000	9.000000	9.791667	30.000000	10270.000000
75%	2.000000	102.400000	0.881788	0.926042	0.928512	2926.250000	142.000000	9.400000	12.368421	34.000000	16500.750000
max	3.000000	120.900000	1.000000	1.000000	1.000000	4066.000000	326.000000	23.000000	18.075923	54.000000	45400.000000

Convert Categorical Data to Numerical

Step 11: Data Visualization

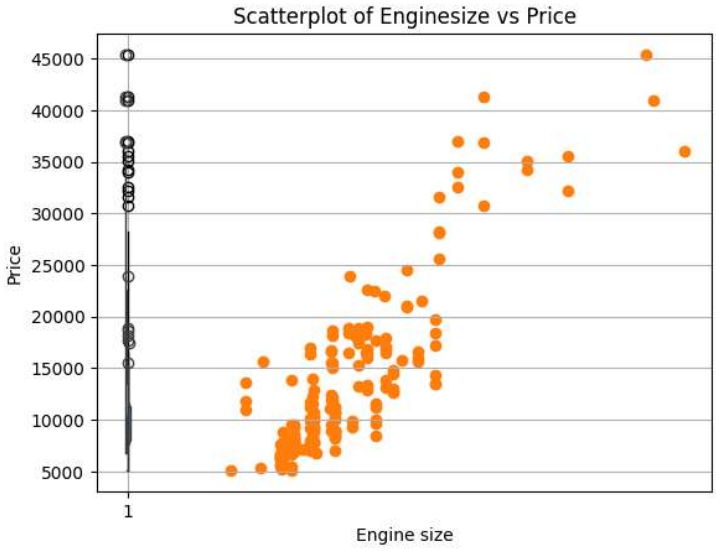
```
plt.boxplot(data['price'])

sns.boxplot(x ='drive-wheels', y ='price', data = data)

plt.scatter(data['engine-size'], data['price'])
```

```
plt.grid()
plt.show()
```

Output:



Engine Size

Step 12: Grouping Data by Drive-Wheels and Body-Style

Grouping data helps identify trends based on key variables:

```
test = data[['drive-wheels', 'body-style', 'price']]
data_grp = test.groupby(['drive-wheels', 'body-style'],
                        as_index = False).mean()

data_grp
```

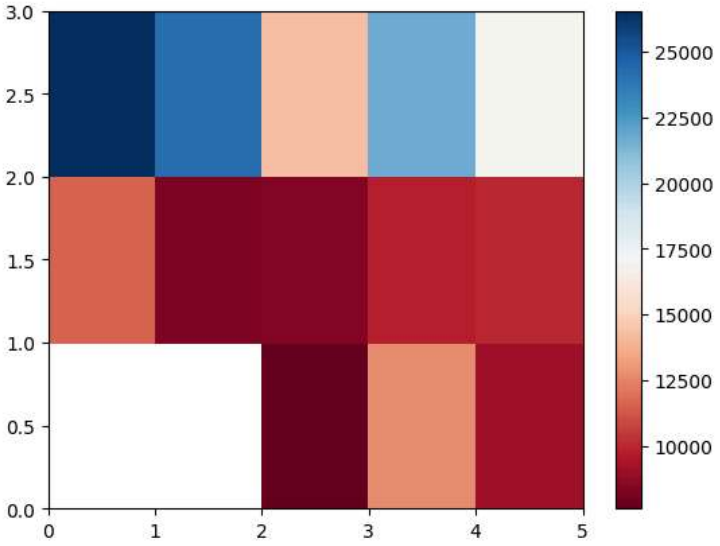
Output:

	drive-wheels	body-style	price
0	4wd	hatchback	7603.000000
1	4wd	sedan	12647.333333
2	4wd	wagon	9095.750000
3	fwd	convertible	11595.000000
4	fwd	hardtop	8249.000000
5	fwd	hatchback	8396.387755
6	fwd	sedan	9811.800000
7	fwd	wagon	9997.333333
8	rwd	convertible	26563.250000
9	rwd	hardtop	24202.714286
10	rwd	hatchback	14337.777778
11	rwd	sedan	21711.833333

```
data_pivot = data_grp.pivot(index = 'drive-wheels',
                            columns = 'body-style')
data_pivot

plt.pcolor(data_pivot, cmap ='RdBu')
plt.colorbar()
plt.show()
```

Output:



Pivot Table

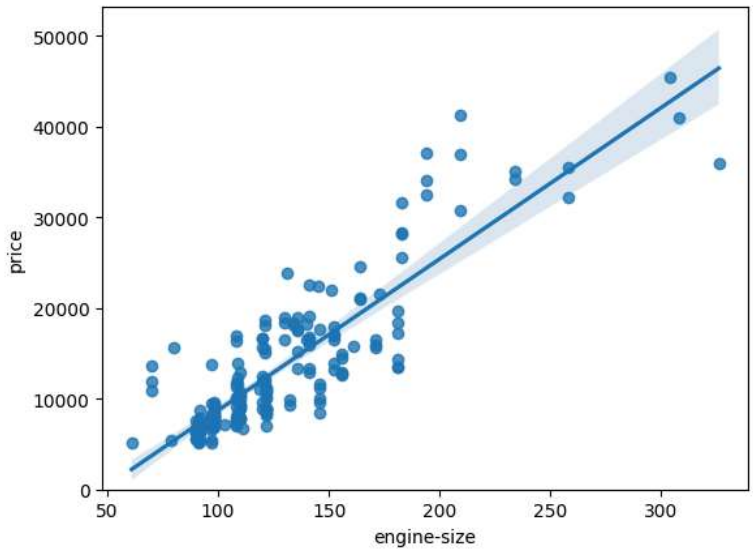
Step 14: Perform ANOVA Test

The Analysis of Variance (ANOVA) test helps determine if different groups have significantly different means.

```
data_annova = data[['make', 'price']]
grouped_annova = data_annova.groupby(['make'])
annova_results_l = sp.stats.f_oneway(
                            grouped_annova.get_group('honda')
['price'],
                            grouped_annova.get_group('subaru')
['price']
                                    )
print(annova_results_l)

sns.regplot(x ='engine-size', y ='price', data = data)
plt.ylim(0, )
```

Output:



ANOVA Test

This step-by-step analysis helps in understanding the key factors influencing the selling price of used cars. Proper data cleaning, visualization and statistical tests ensure that our findings are accurate and insightful.

Next Article

Box Office Revenue Prediction Using Linear Regression in ML

Similar Reads

100+ Machine Learning Projects with Source Code [2025]

This article provides over 100 Machine Learning projects and ideas to provide hands-on experience for both beginners and professionals. Whether you're a student enhancing your resume or a professional...

5 min read

Classification Projects

Regression Projects

IPL Score Prediction using Deep Learning

In todayâ€™s world of cricket every run and decision can turn the game around. Using Deep Learning to predict IPL scores during live matches is becoming a game changer. This article shows how advanced...

7 min read

Dogecoin Price Prediction with Machine Learning

Dogecoin is a cryptocurrency, like Ethereum or Bitcoin â€" despite the fact that it's totally different than both of these famous coins. Dogecoin was initially made to some extent as a joke for crypto devotees an...

4 min read

Zillow Home Value (Zestimate) Prediction in ML

In this article, we will try to implement a house price index calculator which revolutionized the whole real estate industry in the US. This will be a regression task in which we have been provided with logarithm...

6 min read

Calories Burnt Prediction using Machine Learning

In this article, we will learn how to develop a machine learning model using Python which can predict the number of calories a person has burnt during a workout based on some biological measures.Importing...

5 min read

Vehicle Count Prediction From Sensor Data

Sensors at road junctions collect vehicle count data at different times which helps transport managers make informed decisions. In this article we will predict vehicle count based on this sensor data using...

3 min read

Analyzing Selling Price of used Cars using Python

Analyzing the selling price of used cars is essential for making informed decisions in the automotive market. Using Python, we can efficiently process and visualize data to uncover key factors influencing ca...

4 min read

Box Office Revenue Prediction Using Linear Regression in ML

The objective of this project is to develop a machine learning model using Linear Regression to accurately predict the box office revenue of movies based on various available features. The model will be trained ...

9 min read

House Price Prediction using Machine Learning in Python

House price prediction is a problem in the real estate industry to make informed decisions. By using machine learning algorithms we can predict the price of a house based on various features such as...

6 min read

Linear Regression using Boston Housing Dataset - ML

Boston Housing Data: This dataset was taken from the StatLib library and is maintained by Carnegie Mellon University. This dataset concerns the housing prices in the housing city of Boston. The dataset...

3 min read

Stock Price Prediction Project using TensorFlow

Stock price prediction is a challenging task in the field of finance with applications ranging from personal investment strategies to algorithmic trading. In this article we will explore how to build a stock price...

You must have heard some advertisements regarding medical insurance that promises to help financially in case of any medical emergency. One who purchases this type of insurance has to pay premiums...

7 min read

Inventory Demand Forecasting using Machine Learning - Python

Vendors selling everyday items need to keep their stock updated so that customers donâ€™t leave empty-handed. Maintaining the right stock levels helps avoid shortages that disappoint customers and...

6 min read

Ola Bike Ride Request Forecast using ML

From telling rickshaw-wala where to go, to tell him where to come we have grown up. Yes, we are talking about online cab and bike facility providers like OLA and Uber. If you had used this app some times then...

8 min read

Waiter's Tip Prediction using Machine Learning

If you have recently visited a restaurant for a family dinner or lunch and you have tipped the waiter for his generous behavior then this project might excite you. As in this article, we will try to predict what amou...

7 min read

Predict Fuel Efficiency Using Tensorflow in Python

Predicting fuel efficiency is a important task in automotive design and environmental sustainability. In this article we will build a fuel efficiency prediction model using TensorFlow one of the most popular deep...

5 min read

Microsoft Stock Price Prediction with Machine Learning

In this article, we will implement Microsoft Stock Price Prediction with a Machine Learning technique. We will use TensorFlow, an Open-Source Python Machine Learning Framework developed by Google....

5 min read

Share Price Forecasting Using Facebook Prophet

Time series forecast can be used in a wide variety of applications such as Budget Forecasting, Stock Market Analysis, etc. But as useful it is also challenging to forecast the correct projections, Thus can't be...

6 min read

Implementation of Movie Recommender System - Python

Recommender Systems provide personalized suggestions for items that are most relevant to each user by predicting preferences according to user's past choices. They are used in various areas like movies, musi...

4 min read

How can Tensorflow be used with abalone dataset to build a sequential model?

In this article, we will learn how to build a sequential model using TensorFlow in Python to predict the age of an abalone. We may wonder what is an abalone. Answer to this question is that it is a kind of...

7 min read

Natural Language Processing Projects

Clustering Projects

Recommender System Project