

# Load the dataset

The screenshot displays a Jupyter Notebook interface with three visible code cells. The first cell, titled "Load data", uses `pd.read_csv` to load a dataset and shows the first five rows of a DataFrame. The second cell, titled "Perform EDA", checks the shape of the data. The third cell, titled "Check unique values of some features", prints the unique values for 'Fuel\_Type', 'Seller\_Type', 'Transmission', and 'Owner'. The notebook is running on a local host, and the system tray at the bottom shows the time as 3:28 PM on 11/7/2022.

### Load data

```
In [2]: df = pd.read_csv('car_data.csv')
df.head()
```

Out[2]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sv4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	claz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

### Perform EDA

#### Check the shape of data

```
In [3]: df.shape
```

Out[3]: (301, 9)

#### Check unique values of some features

```
In [4]: print(df['Fuel_Type'].unique())
print(df['Seller_Type'].unique())
print(df['Transmission'].unique())
print(df['Owner'].unique())
```

['Petrol' 'Diesel' 'CNG']  
['Dealer' 'Individual']  
['Manual' 'Automatic']  
[0 1 3]

#### Check that there is null values or not

```
In [5]: df.isnull().sum()
```

Out[5]: Car\_Name 0  
Year 0  
Selling\_Price 0  
Present\_Price 0  
Kms\_Driven 0  
Fuel\_Type 0  
Seller\_Type 0  
Transmission 0  
Owner 0  
dtype: int64

Home Page - Select or create a notebook

car\_price\_prediction - Jupyter Notebook

localhost:8888/notebooks/car\_price\_prediction.ipynb

jupyter car\_price\_prediction Last Checkpoint: 19 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

describe data

In [6]: df.describe()

Out[6]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

Drop Car\_Name

In [7]: df1 = df.drop(columns='Car\_Name')  
df1.shape

Out[7]: (301, 8)

Find today's date for check that how much car is old

In [8]: x = datetime.datetime.now()  
x.year

Out[8]: 2022

Make new feature of No. \_of\_Years from existing feature

In [9]: df1['No.\_of\_Years'] = x.year - df1.Year  
df1.head()

Out[9]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	No._of_Years
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0	8
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0	9
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0	5
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0	11
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0	8

Now drop Year feature

The screenshot shows a Jupyter Notebook running in a web browser. The browser tabs include 'Home Page - Select or create a new...', 'car\_price\_prediction - Jupyter No...', and 'car\_price\_prediction - Jupyter No...'. The address bar shows 'localhost:8888/notebooks/car\_price\_prediction.ipynb'. The Jupyter interface has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and markdown. The notebook content shows two code cells and their outputs.

**Code Cell 10:**

```
In [10]: df1.drop(columns = 'Year' , axis=1 , inplace=True)
```

**Code Cell 11:**

```
In [11]: df1.head()
```

**Output 11:**

	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	No_of_Years
0	3.35	5.59	27000	Petrol	Dealer	Manual	0	8
1	4.75	9.54	43000	Diesel	Dealer	Manual	0	9
2	7.25	9.85	6900	Petrol	Dealer	Manual	0	5
3	2.85	4.15	5200	Petrol	Dealer	Manual	0	11
4	4.60	6.87	42450	Diesel	Dealer	Manual	0	8

**Code Cell 12:**

```
In [12]: df1 = pd.get_dummies(df1 , drop_first=True)
df1.head()
```

**Output 12:**

	Selling_Price	Present_Price	Kms_Driven	Owner	No_of_Years	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	5.59	27000	0	8	0	1	0	1
1	4.75	9.54	43000	0	9	1	0	0	1
2	7.25	9.85	6900	0	5	0	1	0	1
3	2.85	4.15	5200	0	11	0	1	0	1