In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline
mpl.style.use('ggplot')
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
%matplotlib inline
mpl.style.use('ggplot')
```

In [2]:
```
car=pd.read_csv('quikr_car.csv')
```

In [3]:
```
car.sample(5)
```

Out[3]:

	name	company	year	Price	kms_driven	fuel_type
131	Chevrolet Beat	Chevrolet	2015	1,50,000	30,000 kms	Petrol
846	Maruti Suzuki Alto 800	Maruti	2016	2,50,000	2,450 kms	Petrol
748	Nissan Micra XL	Nissan	2017	4,30,000	62,500 kms	Diesel
720	Hyundai Eon D Lite Plus	Hyundai	2018	2,80,000	35,000 kms	Petrol
284	Hyundai Santro Xing XO eRLX Euro III	Hyundai	2000	59,000	56,450 kms	Petrol

In [4]:
```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        892 non-null    object
 1   company     892 non-null    object
 2   year        892 non-null    object
 3   Price       892 non-null    object
 4   kms_driven  840 non-null    object
 5   fuel_type   837 non-null    object
dtypes: object(6)
memory usage: 41.9+ KB
```

In [5]:
```
backup=car.copy()
```

Quality

1.names are pretty inconsistent 2.names have company names attached to it 3.some names are spam like 'Maruti Ertiga showroom condition with' and 'Well mentained Tata Sumo' 4.company: many of the names are not of any company like 'Used', 'URJENT', and so on. 5.year has many non-year values 6.year is in object. Change to integer 7.Price has

Ask for Price 8.Price has commas in its prices and is in object 9.kms_driven has object values with kms at last. 10.It has nan values and two rows have 'Petrol' in them 11.fuel_type has nan values

Cleaning Data

. year has many non-year values

In [6]:
```
car=car[car['year'].str.isnumeric()]
```

1.year is in object. Change to integer

In [7]:
```
car['year']=car['year'].astype(int)
```

2.Price has Ask for Price

In [8]:
```
car=car[car['Price']!='Ask For Price']
```

3.Price has commas in its prices and is in object

In [9]:
```
car['Price']=car['Price'].str.replace(',','').astype(int)
```

4.kms_driven has object values with kms at last.

In [11]:
```
car['kms_driven']=car['kms_driven'].str.split().str.get(0).str.replace(',',
```

5.It has nan values and two rows have 'Petrol' in them

In [12]:
```
car=car[car['kms_driven'].str.isnumeric()]
```

In [13]:
```
car['kms_driven']=car['kms_driven'].astype(int)
```

6. fuel_type has nan values

In [14]:
```
car=car[~car['fuel_type'].isna()]
```

In [15]:
```
car.shape
```

Out[15]:
```
(816, 6)
```

name and company had spammed data...but with the previous cleaning, those rows got removed.

Company does not need any cleaning now. Changing car names. Keeping only the first three words

In [16]:
```
car['name']=car['name'].str.split().str.slice(start=0,stop=3).str.join(' ')
```

Resetting the index of the final cleaned data

In [17]:
```
car=car.reset_index(drop=True)
```

Cleaned Data

In [18]:
```
car
```

Out[18]:

	name	company	year	Price	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	80000	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	425000	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	325000	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	575000	36000	Diesel
4	Ford Figo	Ford	2012	175000	41000	Diesel
...	...	...	...	...	...	...
811	Maruti Suzuki Ritz	Maruti	2011	270000	50000	Petrol
812	Tata Indica V2	Tata	2009	110000	30000	Diesel
813	Toyota Corolla Altis	Toyota	2009	300000	132000	Petrol
814	Tata Zest XM	Tata	2018	260000	27000	Diesel
815	Mahindra Quanto C8	Mahindra	2013	390000	40000	Diesel

816 rows × 6 columns

In [20]:
```
car.to_csv('Cleaned_Car_data.csv')
```

In [21]:
```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        816 non-null    object
 1   company     816 non-null    object
 2   year        816 non-null    int32 
 3   Price       816 non-null    int32 
 4   kms_driven  816 non-null    int32 
 5   fuel_type   816 non-null    object
dtypes: int32(3), object(3)
memory usage: 28.8+ KB
```

In [22]:

```
car.describe(include='all')
```

Out[22]:

	name	company	year	Price	kms_driven	fuel_type
count	816	816	816.000000	8.160000e+02	816.000000	816
unique	254	25	NaN	NaN	NaN	3
top	Maruti Suzuki Swift	Maruti	NaN	NaN	NaN	Petrol
freq	51	221	NaN	NaN	NaN	428
mean	NaN	NaN	2012.444853	4.117176e+05	46275.531863	NaN
std	NaN	NaN	4.002992	4.751844e+05	34297.428044	NaN
min	NaN	NaN	1995.000000	3.000000e+04	0.000000	NaN
25%	NaN	NaN	2010.000000	1.750000e+05	27000.000000	NaN
50%	NaN	NaN	2013.000000	2.999990e+05	41000.000000	NaN
75%	NaN	NaN	2015.000000	4.912500e+05	56818.500000	NaN
max	NaN	NaN	2019.000000	8.500003e+06	400000.000000	NaN

In [23]:

```
car=car[car['Price']<6000000]
```

Checking relationship of Company with Price

In [24]:

```
car['company'].unique()
```

Out[24]:

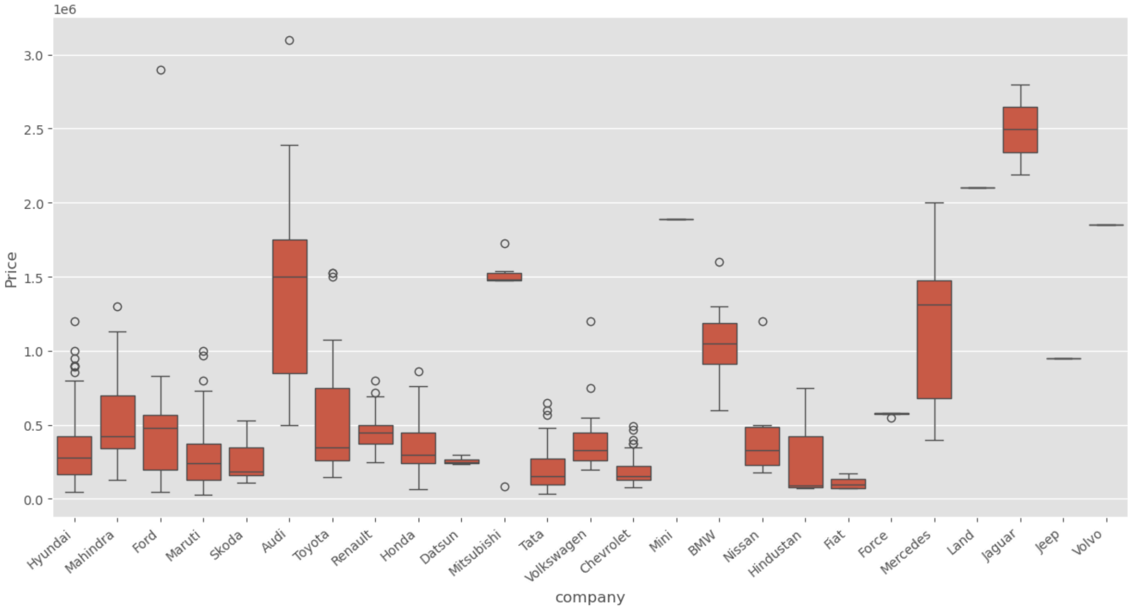
```
array(['Hyundai', 'Mahindra', 'Ford', 'Maruti', 'Skoda', 'Audi', 'Toyota',
       'Renault', 'Honda', 'Datsun', 'Mitsubishi', 'Tata', 'Volkswagen',
       'Chevrolet', 'Mini', 'BMW', 'Nissan', 'Hindustan', 'Fiat', 'Force',
       'Mercedes', 'Land', 'Jaguar', 'Jeep', 'Volvo'], dtype=object)
```

In [25]:

```
import seaborn as sns
```

In [26]:
```
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='company',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

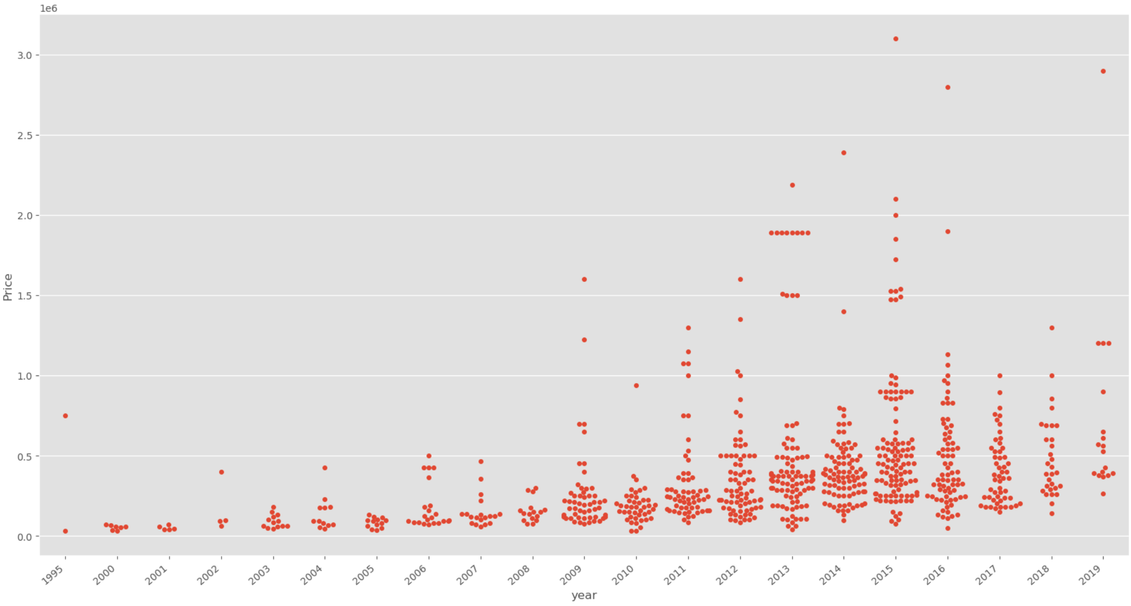
```
C:\Users\ratneshpati tripathi\AppData\Local\Temp\ipykernel_23968\278813051
7.py:3: UserWarning: set_ticklabels() should only be used with a fixed num
ber of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
```



In [27]:

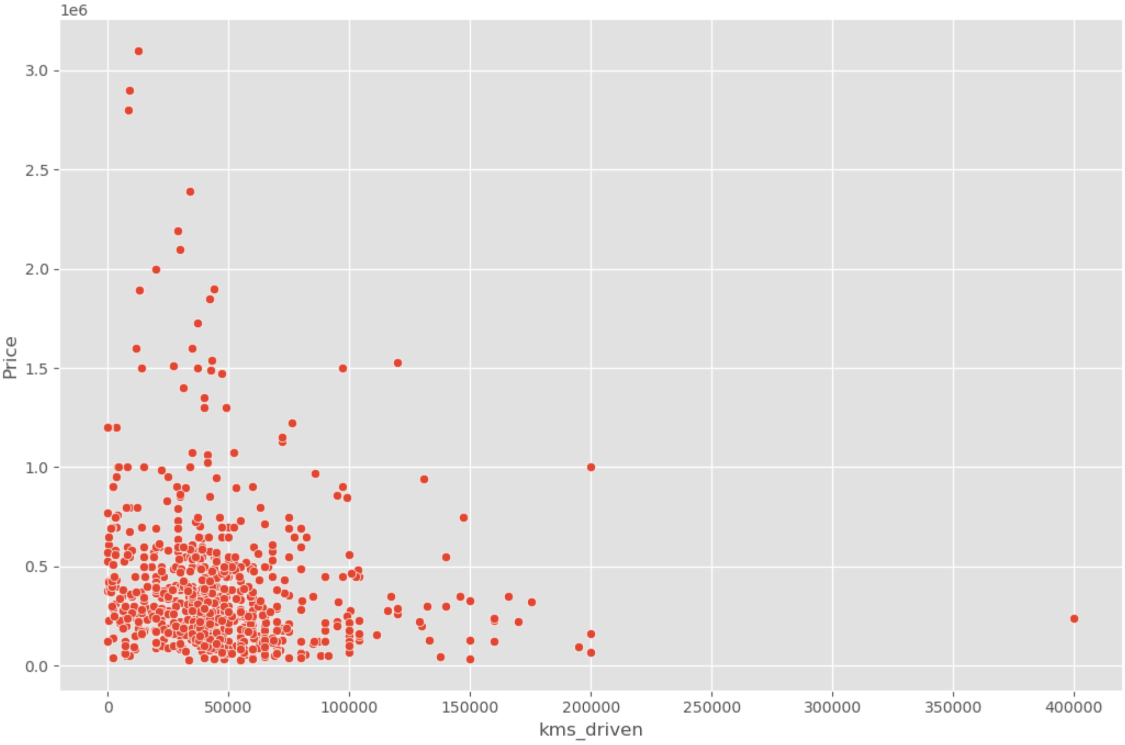
```python
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

```
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 13.6% of the points cannot be placed; you may wa
nt to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 13.0% of the points cannot be placed; you may wa
nt to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 6.8% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 10.6% of the points cannot be placed; you may wa
nt to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 7.7% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\AppData\Local\Temp\ipykernel_23968\254204227
7.py:3: UserWarning: set_ticklabels() should only be used with a fixed num
ber of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 9.3% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 6.8% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 9.6% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\seaborn\categori
cal.py:3399: UserWarning: 5.5% of the points cannot be placed; you may wan
t to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



```
In [28]: sns.relplot(x='kms_driven',y='Price',data=car,height=7,aspect=1.5)
```

```
Out[28]: <seaborn.axisgrid.FacetGrid at 0x2302f05b590>
```

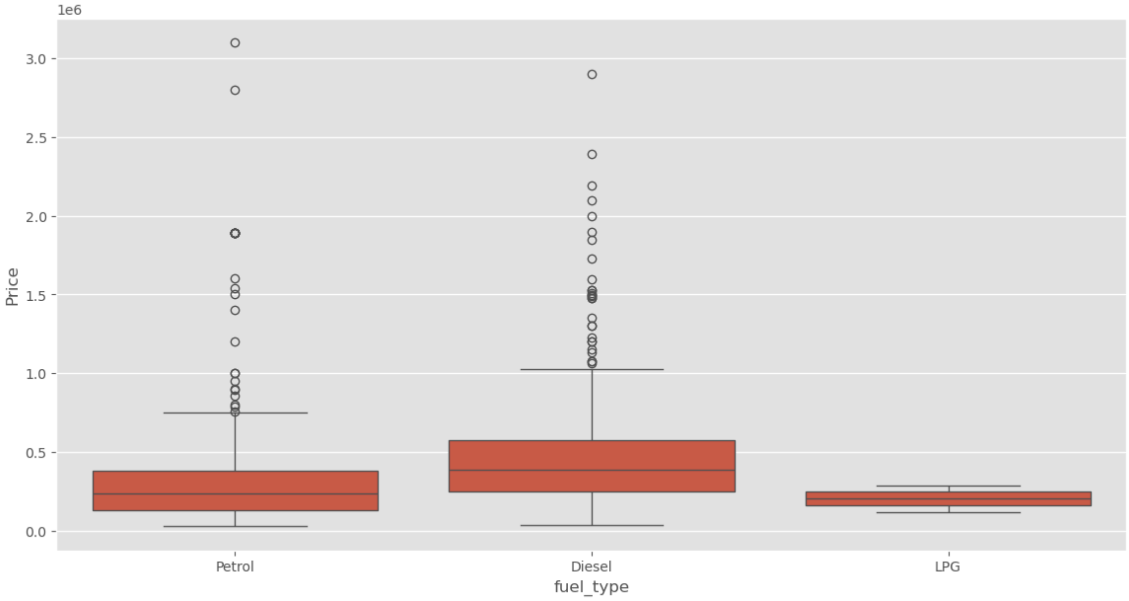


In [29]:

```
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=car)
```

Out[29]:

```
<Axes: xlabel='fuel_type', ylabel='Price'>
```



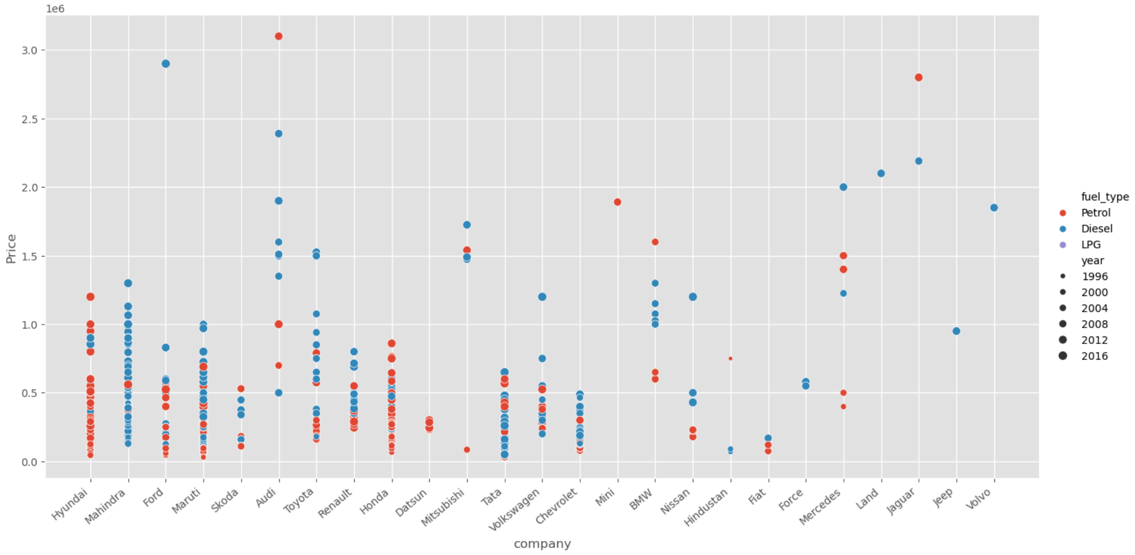
Relationship of Price with FuelType, Year and Company mixed

In [30]:

```
ax=sns.relplot(x='company',y='Price',data=car,hue='fuel_type',size='year',h
ax.set_xticklabels(rotation=40,ha='right')
```

Out[30]:

```
<seaborn.axisgrid.FacetGrid at 0x2302fae5af0>
```



Extracting Training Data

In [40]:

```
!pip install ydata-profiling
```

```
Requirement already satisfied: ydata-profiling in c:\users\ratneshpati tri
pathi\anaconda3\lib\site-packages (4.10.0)
Requirement already satisfied: scipy<1.14,>=1.4.1 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (1.13.1)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in c:\users\ratneshpa
ti tripathi\anaconda3\lib\site-packages (from ydata-profiling) (2.2.2)
Requirement already satisfied: matplotlib<3.10,>=3.5 in c:\users\ratneshpa
ti tripathi\anaconda3\lib\site-packages (from ydata-profiling) (3.8.4)
Requirement already satisfied: pydantic>=2 in c:\users\ratneshpati tripath
i\anaconda3\lib\site-packages (from ydata-profiling) (2.5.3)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (6.0.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (3.1.4)
Requirement already satisfied: visions<0.7.7,>=0.7.5 in c:\users\ratneshpa
ti tripathi\anaconda3\lib\site-packages (from visions[type_image_path]<0.
7.7,>=0.7.5->ydata-profiling) (0.7.6)
Requirement already satisfied: numpy<2.2,>=1.16.0 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (1.26.4)
Requirement already satisfied: htmlmin==0.1.12 in c:\users\ratneshpati tri
pathi\anaconda3\lib\site-packages (from ydata-profiling) (0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (0.12.4)
Requirement already satisfied: requests<3,>=2.24.0 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (2.32.2)
Requirement already satisfied: tqdm<5,>=4.48.2 in c:\users\ratneshpati tri
pathi\anaconda3\lib\site-packages (from ydata-profiling) (4.66.4)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in c:\users\ratneshpa
ti tripathi\anaconda3\lib\site-packages (from ydata-profiling) (0.13.2)
Requirement already satisfied: multimethod<2,>=1.4 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from ydata-profiling) (1.12)
Requirement already satisfied: statsmodels<1,>=0.13.2 in c:\users\ratneshp
ati tripathi\anaconda3\lib\site-packages (from ydata-profiling) (0.14.2)
Requirement already satisfied: typeguard<5,>=3 in c:\users\ratneshpati tri
pathi\anaconda3\lib\site-packages (from ydata-profiling) (4.3.0)
Requirement already satisfied: imagehash==4.3.1 in c:\users\ratneshpati tr
ipathi\anaconda3\lib\site-packages (from ydata-profiling) (4.3.1)
Requirement already satisfied: wordcloud>=1.9.3 in c:\users\ratneshpati tr
ipathi\anaconda3\lib\site-packages (from ydata-profiling) (1.9.3)
Requirement already satisfied: dacite>=1.8 in c:\users\ratneshpati tripath
i\anaconda3\lib\site-packages (from ydata-profiling) (1.8.1)
Requirement already satisfied: numba<1,>=0.56.0 in c:\users\ratneshpati tr
ipathi\anaconda3\lib\site-packages (from ydata-profiling) (0.59.1)
Requirement already satisfied: PyWavelets in c:\users\ratneshpati tripathi
\anaconda3\lib\site-packages (from imagehash==4.3.1->ydata-profiling) (1.
5.0)
Requirement already satisfied: pillow in c:\users\ratneshpati tripathi\ana
conda3\lib\site-packages (from imagehash==4.3.1->ydata-profiling) (10.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\ratneshpati tri
pathi\anaconda3\lib\site-packages (from jinja2<3.2,>=2.11.1->ydata-profili
ng) (2.1.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\ratneshpati tr
ipathi\anaconda3\lib\site-packages (from matplotlib<3.10,>=3.5->ydata-prof
iling) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\ratneshpati tripat
hi\anaconda3\lib\site-packages (from matplotlib<3.10,>=3.5->ydata-profilin
g) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\ratneshpati t
ripathi\anaconda3\lib\site-packages (from matplotlib<3.10,>=3.5->ydata-pro
filing) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\ratneshpati t
```

```
ripathi\anaconda3\lib\site-packages (from matplotlib<3.10,>=3.5->ydata-pro
filing) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\ratneshpati tri
pathi\appdata\roaming\python\python312\site-packages (from matplotlib<3.1
0,>=3.5->ydata-profiling) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\ratneshpati tr
ipathi\anaconda3\lib\site-packages (from matplotlib<3.10,>=3.5->ydata-prof
iling) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ratneshpat
i tripathi\appdata\roaming\python\python312\site-packages (from matplotlib
<3.10,>=3.5->ydata-profiling) (2.9.0.post0)
Requirement already satisfied: llvmlite<0.43,>=0.42.0dev0 in c:\users\ratn
eshpati tripathi\anaconda3\lib\site-packages (from numba<1,>=0.56.0->ydata
-profiling) (0.42.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\ratneshpati tripat
hi\anaconda3\lib\site-packages (from pandas!=1.4.0,<3,>1.1->ydata-profilin
g) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\ratneshpati trip
athi\anaconda3\lib\site-packages (from pandas!=1.4.0,<3,>1.1->ydata-profil
ing) (2023.3)
Requirement already satisfied: joblib>=0.14.1 in c:\users\ratneshpati trip
athi\anaconda3\lib\site-packages (from phik<0.13,>=0.11.1->ydata-profilin
g) (1.4.2)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\ratneshp
ati tripathi\anaconda3\lib\site-packages (from pydantic>=2->ydata-profilin
g) (0.6.0)
Requirement already satisfied: pydantic-core==2.14.6 in c:\users\ratneshpa
ti tripathi\anaconda3\lib\site-packages (from pydantic>=2->ydata-profilin
g) (2.14.6)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\ratnes
hpati tripathi\anaconda3\lib\site-packages (from pydantic>=2->ydata-profil
ing) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\ratnes
hpati tripathi\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydat
a-profiling) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\ratneshpati tripat
hi\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling)
(3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-prof
iling) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ratneshpati
tripathi\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-prof
iling) (2024.8.30)
Requirement already satisfied: patsy>=0.5.6 in c:\users\ratneshpati tripat
hi\anaconda3\lib\site-packages (from statsmodels<1,>=0.13.2->ydata-profili
ng) (0.5.6)
Requirement already satisfied: colorama in c:\users\ratneshpati tripathi\a
ppdata\roaming\python\python312\site-packages (from tqdm<5,>=4.48.2->ydata
-profiling) (0.4.6)
Requirement already satisfied: attrs>=19.3.0 in c:\users\ratneshpati tripa
thi\anaconda3\lib\site-packages (from visions<0.7.7,>=0.7.5->visions[type_
image_path]<0.7.7,>=0.7.5->ydata-profiling) (23.1.0)
Requirement already satisfied: networkx>=2.4 in c:\users\ratneshpati tripa
thi\anaconda3\lib\site-packages (from visions<0.7.7,>=0.7.5->visions[type_
image_path]<0.7.7,>=0.7.5->ydata-profiling) (3.2.1)
Requirement already satisfied: six in c:\users\ratneshpati tripathi\appdat
a\roaming\python\python312\site-packages (from patsy>=0.5.6->statsmodels<
1,>=0.13.2->ydata-profiling) (1.16.0)
```

```
In [43]: from ydata_profiling import ProfileReport
         prof=ProfileReport(car)
         prof.to_file(output_file='car_prof_report.html')
```

```
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\ydata_profiling
\utils\dataframe.py:137: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df.rename(columns={"index": "df_index"}, inplace=True)
```

```
Summarize dataset: 100%                    24/24 [00:04<00:00,  2.74it/s, Completed]
```

```
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\ydata_profiling
\model\pandas\discretize_pandas.py:52: FutureWarning: Setting an item of i
ncompatible dtype is deprecated and will raise in a future error of panda
s. Value '[4 4 7 7 7 7 8 8 6 8 7 8 4 4 9 7 8 7 8 8 8 7 7 7 9 8 5 9 7 8 9 7
7 7 9 7 9
 7 7 6 8 7 7 8 8 9 6 8 9 7 8 9 7 7 8 9 9 7 7 7 9 8 9 7 7 6 8 6 7 8 7 9 6 8
 8 7 7 8 7 7 9 7 9 5 6 5 5 4 7 6 6 6 7 7 7 7 5 7 7 8 7 8 7 8 7 7 7 7 7 7 8
 7 4 7 5 7 7 8 7 6 7 9 6 4 9 7 8 8 6 8 5 7 4 2 6 7 7 7 8 7 8 7 8 7 9 8 7 9
 7 6 9 6 8 6 7 6 7 7 6 6 6 7 7 6 8 7 6 7 8 7 7 6 7 8 7 7 5 8 8 7 5 6 7 4 8
 2 8 9 4 7 4 6 6 7 7 5 4 7 6 6 5 8 7 7 8 7 9 7 7 8 7 9 8 9 6 8 5 4 8 8 8 5
 8 8 9 7 7 8 8 8 8 7 7 7 8 8 5 8 8 8 7 9 7 7 7 7 3 4 8 7 7 7 8 7 4 7 5 7 6
 8 7 8 7 7 8 7 9 3 9 9 7 7 6 6 8 2 8 6 7 5 8 6 8 4 8 7 7 5 7 5 9 3 5 8 7 7
 8 7 7 6 4 7 8 7 5 7 7 8 9 8 7 7 8 8 8 7 7 7 7 6 8 7 7 7 7 7 7 7 8 8 7 7 8
 7 9 7 7 7 8 9 6 6 7 7 6 7 9 8 6 5 9 7 8 7 7 8 7 6 8 7 3 9 6 8 8 5 9 4 9 6
 7 7 0 6 8 7 8 7 3 8 6 2 7 2 4 2 9 9 6 9 6 5 9 2 7 7 8 7 8 7 8 7 7 9 6 9 8
 9 8 2 6 6 7 8 7 8 6 8 8 7 8 9 6 6 7 7 8 7 9 5 6 8 8 8 8 9 8 6 8 9 8 3 9 8
 5 3 6 8 8 8 8 8 5 8 9 9 7 6 7 8 7 9 2 8 3 7 8 9 8 4 8 7 7 4 3 6 7 4 9 3 5
 8 6 9 6 8 7 7 2 6 9 8 8 9 7 7 8 8 8 7 7 4 2 8 8 9 5 6 7 4 4 7 3 5 5 6 8 9
 5 5 2 5 7 9 7 7 4 6 6 5 7 7 8 6 9 7 4 7 9 6 5 7 7 7 9 5 7 3 8 6 8 7 6 7 9
 9 7 8 6 7 8 8 3 8 7 8 5 6 7 6 4 7 7 9 6 5 9 5 6 7 9 7 7 4 7 6 7 9 7 4 2 6
 5 7 9 3 6 6 7 9 7 7 7 7 6 8 8 8 7 8 8 7 3 8 7 7 6 5 7 8 6 7 6 8 8 7 7 6 8
 7 8 7 8 7 6 4 8 9 9 8 9 7 5 3 5 8 7 5 7 7 8 7 8 7 6 9 5 3 6 9 9 4 5 7 8 5
 3 8 7 9 5 8 7 5 3 7 8 8 9 5 5 6 5 8 6 3 8 8 8 7 7 4 7 9 4 5 6 4 5 6 9 4 4
 9 4 8 3 7 7 7 7 7 8 7 4 9 9 4 9 7 0 4 4 2 5 5 9 7 8 4 5 7 8 4 9 6 6 7 3 9
 5 5 8 7 6 7 7 7 9 5 8 7 7 8 4 4 6 9 5 7 5 4 7 7 6 6 2 8 9 8 9 4 4 8 5 9 6
 8 9 9 5 2 6 7 3 4 9 8 7 5 3 3 8 5 9 7 8 8 6 6 4 8 9 8 8 4 4 4 5 5 6 5 5 9
 7]' has dtype incompatible with int32, please explicitly cast to a compat
ible dtype first.
  discretized_df.loc[:, column] = self._discretize_column(
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\ydata_profiling
\model\pandas\discretize_pandas.py:52: FutureWarning: Setting an item of i
ncompatible dtype is deprecated and will raise in a future error of panda
s. Value '[0 1 0 1 0 0 2 0 0 0 1 0 0 1 3 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0
0 2 2 1 1
 1 1 0 0 1 0 1 1 0 0 4 0 1 0 0 0 0 1 0 0 0 0 0 2 0 1 0 0 0 0 1 1 6 0 0 1
 1 1 6 2 2 1 1 0 3 0 1 0 0 0 1 0 1 0 1 0 1 0 5 2 0 1 1 1 1 1 1 0 1 0 0 0 0
 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 2 0 1 0 0 0 1 2 2 1 0 3 1 1 0 1 0 1 1
 1 1 0 0 0 0 0 0 0 1 0 0 0 0 2 0 1 1 0 0 2 2 0 0 0 1 0 1 0 0 1 1 0 0 6 0 2
 0 1 2 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 1 2 4 2
 6 6 1 1 4 0 0 0 1 0 0 1 1 1 2 1 2 1 6 2 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0
 0 6 1 1 1 1 1 0 0 1 2 1 1 0 0 0 0 1 0 1 0 0 0 1 0 2 6 6 0 6 0 1 0 0 2 1 1
 1 0 0 0 0 1 2 1 0 1 1 1 2 1 2 1 2 1 2 0 2 1 1 0 1 0 0 1 1 4 1 1 1 1 1 1 1
 1 1 1 0 0 1 0 0 0 1 0 3 0 2 3 0 0 1 1 3 0 0 1 1 0 0 0 0 1 0 0 1 0 1 1 0 0
 1 2 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 2 0 2 1 0 1 0 3 0
 3 0 0 2 0 4 0 1 0 0 1 0 1 0 2 4 1 4 0 2 0 1 0 0 1 1 1 1 2 1 0 2 2 1 0 0 4
 0 0 0 0 2 2 0 1 0 1 1 1 0 0 0 4 1 2 0 1 0 1 2 2 0 0 4 0 0 0 0 0 0 0 1 0 0
 0 0 1 0 9 0 1 0 0 1 0 0 1 1 0 1 0 3 2 1 0 0 0 1 0 0 0 1 0 0 5 0 0 0 0 1 2
 0 0 0 0 0 1 1 1 0 3 0 0 0 1 1 0 4 0 0 1 0 0 0 1 1 1 0 2 0 1 0 1 2 1 0 0 0
 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0
 0 1 1 0 2 0 0 0 0 0 0 1 0 1 3 1 1 1 1 0 0 2 0 1 0 0 0 1 1 1 0 0 0 7 7 3 1
 3 1 1 0 1 0 0 6 1 1 2 2 0 0 0 0 4 2 1 3 4 5 2 5 0 3 0 3 0 0 0 1 0 0 1 1 0
 0 1 1 2 0 0 1 0 0 0 3 9 0 0 0 0 0 2 1 0 0 1 2 4 0 0 0 3 0 0 1 0 0 0 1 0 0
 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 1 0 1
 0 0 0 0 1 0 0 0 1 1 2 0 1 1 0 0 0 0 0 1 0 0 2 2 1 0 0 1 1 0 9 0 0 1 0 1 2
 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 2 1 0 1 0 0 0 0 0 0 0
 1]' has dtype incompatible with int32, please explicitly cast to a compat
ible dtype first.
  discretized_df.loc[:, column] = self._discretize_column(
C:\Users\ratneshpati tripathi\anaconda3\Lib\site-packages\ydata_profiling
\model\pandas\discretize_pandas.py:52: FutureWarning: Setting an item of i
ncompatible dtype is deprecated and will raise in a future error of panda
```

```
s. Value '[1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 0 1 0 0
0 2 0 1 0
 0 0 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 3 0 3 0 1 1 0 1 0 1 0 0 0 0
 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1
 1 1 0 1 0 0 2 1 0 0 0 1 1 0 1 0 0 0 0 1 1 0 4 1 1 2 1 0 0 0 1 0 1 0 0 1 0
 1 1 0 1 0 0 0 1 0 1 1 0 1 0 1 0 1 0 1 0 2 1 0 1 1 0 1 1 0 4 0 0 1 0 0 0 0
 1 1 0 1 1 1 1 0 0 0 1 1 1 1 0 2 0 0 0 0 0 0 0 2 0 1 0 0 0 0 0 1 1 1 0 1 1
 0 0 0 0 0 0 4 4 1 0 1 1 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1 0 1 1 1 1 0 1 0 0
 0 0 1 0 1 0 0 0 1 0 0 1 1 1 1 4 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 3 1 0 1 1
 1 0 0 1 4 1 0 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 9 1 0 0 1 0 2 0 1 2 0 1 1 0 1 0 4 1 0 0 1 1
 0 0 1 0 0 1 0 0 1 2 1 1 2 1 1 1 0 0 1 1 0 0 0 0 1 0 1 2 0 0 1 0 0 1 0 0 3
 0 0 1 3 2 0 0 0 1 0 0 1 0 3 0 1 1 0 1 2 2 0 1 1 0 0 0 0 0 0 1 2 0 0 1 0 1
 2 2 0 2 2 2 3 1 1 0 2 1 2 1 1 2 1 0 2 1 1 1 1 0 1 1 2 0 1 1 1 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 2 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 3 0 0 1 0 0
 1 0 2 0 1 0 0 2 3 1 0 1 0 1 1 1 0 0 1 3 0 0 0 0 1 0 0 0 0 0 2 0 0 1 1 0 0
 2 1 2 0 4 2 2 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 4 0 0 1 1 3 1 1 1 1 1 0 1
 1 0 0 1 3 1 2 0 1 1 1 1 1 0 1 0 2 0 0 1 1 1 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0
 1 0 1 0 0 1 1 1 0 2 0 1 2 1 1 1 1 0 1 4 0 1 0 0 0 0 0 1 1 2 1 0 1 1 0 1 1
 2 0 0 0 1 2 0 2 1 0 2 0 0 0 1 1 1 0 1 1 2 1 1 2 0 1 2 0 1 1 1 1 0 0 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 0 0 0 2 1 1 1 1 0 1 1 1 1 1 1 0 0 0 2 1 0 1 0
 2 2 3 0 1 0 2 0 0 0 1 1 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 0 0 0 0 1 1 0 1 0 1
 0 0 1 1 0 1 0 1 0 0 2 1 0 2 3 2 1 2 1 2 3 0 0 2 3 0 2 0 1 0 1 1 1 1 0 3 0
 0]' has dtype incompatible with int32, please explicitly cast to a compat
ible dtype first.
  discretized_df.loc[:, column] = self._discretize_column(
```

```
Generate report structure: 100%                    1/1 [00:07<00:00,  7.08s/it]

Render HTML: 100%                                  1/1 [00:01<00:00,  1.95s/it]

Export report to file: 100%                        1/1 [00:00<00:00, 29.50it/s]
```

In [44]:

```
X=car[['name','company','year','kms_driven','fuel_type']]
y=car['Price']
```

In [ ]:

```

```

In [45]:

```
X
```

Out[45]:

	name	company	year	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	36000	Diesel
4	Ford Figo	Ford	2012	41000	Diesel
...	...	...	...	...	...
811	Maruti Suzuki Ritz	Maruti	2011	50000	Petrol
812	Tata Indica V2	Tata	2009	30000	Diesel
813	Toyota Corolla Altis	Toyota	2009	132000	Petrol
814	Tata Zest XM	Tata	2018	27000	Diesel
815	Mahindra Quanto C8	Mahindra	2013	40000	Diesel

815 rows × 5 columns

In [46]:

```
y
```

Out[46]:

```
0       80000
1      425000
2      325000
3      575000
4      175000
        ...  
811    270000
812    110000
813    300000
814    260000
815    390000
Name: Price, Length: 815, dtype: int32
```

Applying Train Test Split

In [47]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

from sklearn.linear_model import LinearRegression

In [48]:

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

Creating an OneHotEncoder object to contain all the possible categories

In [49]:
```
ohe=OneHotEncoder()
ohe.fit(X[['name','company','fuel_type']])
```

Out[49]:
```
▾   OneHotEncoder ⓘ ?
                     (https://scikit-
                     learn.org/1.4/modules/generated/sklearn.preprocessing.OneHotEncod
OneHotEncoder()
```

Creating a column transformer to transform categorical columns

In [50]:
```
column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categori
                                    remainder='passthrough')
```

Linear Regression Model

In [53]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Making a pipeline

In [54]:
```
pipe=make_pipeline(column_trans,lr)
```

Fitting the model

In [55]:
```
pipe.fit(X_train,y_train)
```

Out[55]:
```
▸                    Pipeline                     ⓘ ?
                                                   (https://scikit-
                                                   learn.org/1.4/modules/generated/sk
   ▸ columntransformer: ColumnTransformer ?
                                                 (https://scikit-
                                                 learn.org/1.4/modules/generated/sklea
   ▸    onehotencoder         ▸    remainder

      ▸ OneHotEncoder ?          ▸ passthrough
                      (https://scikit-
                      learn.org/1.4/modules/generated/sklearn.preprocessing.OneHotEn

             ▸ LinearRegression ?
                                (https://scikit-
                                learn.org/1.4/modules/generated/sklearn.linear_mod
```

In [56]:
```
y_pred=pipe.predict(X_test)
```

Checking R2 Score

In [57]:
```
r2_score(y_test,y_pred)
```

Out[57]:
```
0.6734716235317342
```

Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score

In [58]:
```
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

In [59]:
```
np.argmax(scores)
```

Out[59]:
```
302
```

In [60]:
```
scores[np.argmax(scores)]
```

Out[60]:
```
0.8991138463319752
```

to check the prediction

In [61]:
```
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suz
```

Out[61]:
```
array([430287.74002343])
```

In [62]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_sta
lr=LinearRegression()
pipe=make_pipeline(column_trans,lr)
pipe.fit(X_train,y_train)
y_pred=pipe.predict(X_test)
r2_score(y_test,y_pred)
```

Out[62]:
```
0.8991138463319752
```

In [65]:

```
import pickle
import ipywidgets as widgets
from IPython.display import display
pickle.dump(pipe,open('LinearRegressionModel.pkl','wb'))
```

In [67]:

```
# Load the trained model (replace 'linearreg_carprice.pkl' with your actual
# pipe = pickle.load(open('linearreg_carprice.pkl', 'rb'))

# Assuming the data is loaded into car_data as shown earlier
# car_data = pd.read_csv('cleaned_car.csv')  # Replace with actual data loa

# Unique companies and car names
car_data = pd.read_csv('cleaned_car.csv')
unique_companies = car_data['company'].unique()

# Widgets
company_widget = widgets.Dropdown(
    options=unique_companies,
    description='Company:'
)

# The car name widget starts empty and will update based on company selecti
name_widget = widgets.Dropdown(
    options=[],  # Initially empty
    description='Car Name:'
)

year_widget = widgets.IntSlider(
    value=2014,
    min=1990,
    max=2025,
    step=1,
    description='Year:'
)

kms_driven_widget = widgets.IntSlider(
    value=30000,
    min=0,
    max=500000,
    step=1000,
    description='Kms Driven:'
)

fuel_type_widget = widgets.Dropdown(
    options=['Petrol', 'Diesel', 'CNG', 'Electric'],
    value='Petrol',
    description='Fuel Type:'
)

# Prediction button
predict_btn = widgets.Button(
    description='Predict'
)

# Output widget for displaying the prediction result
prediction_out = widgets.Output()

# Function to update car model names based on selected company
def update_car_names(change):
    selected_company = change['new']
    models = car_data[car_data['company'] == selected_company]['name'].uniq
    name_widget.options = models

# Link company dropdown to update car names
company_widget.observe(update_car_names, names='value')
```

```
# Define the function that will gather input and make predictions
def make_prediction(btn):
    # Gather input values from the widgets
    data = pd.DataFrame(
        columns=['name', 'company', 'year', 'kms_driven', 'fuel_type'],
        data=np.array([
            name_widget.value,
            company_widget.value,
            year_widget.value,
            kms_driven_widget.value,
            fuel_type_widget.value
        ]).reshape(1, 5)
    )

    # Perform the prediction using the pipeline model
    prediction = pipe.predict(data)
    #prediction = [500000]  # Dummy prediction for illustration purposes

    # Display the result in the output widget
    with prediction_out:
        prediction_out.clear_output()
        print(f"Predicted Price: {prediction[0]:,.2f} INR")

# Link the prediction function to the button click event
predict_btn.on_click(make_prediction)

# Display the widgets and the prediction output
display(company_widget, name_widget, year_widget, kms_driven_widget, fuel_t
```

Company: Hyundai

Car Name:

Year: 2014

Kms Driven: 30000

Fuel Type: Petrol

Predict

In [ ]: