

Linear Regression

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [2]: df=pd.read_csv(r"C:\Users\91903\Downloads\used_cars_data.csv")
df
```

Out[2]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Se
...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes- Benz E- Class 2009- 2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns



In [3]: `df.head()`

Out[3]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second

In [4]: `df.tail()`

Out[4]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_1
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	1
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

In [5]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   S.No.                 7253 non-null   int64
1   Name                  7253 non-null   object
2   Location              7253 non-null   object
3   Year                  7253 non-null   int64
4   Kilometers_Driven     7253 non-null   int64
5   Fuel_Type             7253 non-null   object
6   Transmission          7253 non-null   object
7   Owner_Type            7253 non-null   object
8   Mileage               7251 non-null   object
9   Engine                7207 non-null   object
10  Power                 7207 non-null   object
11  Seats                 7200 non-null   float64
12  New_Price             1006 non-null   object
13  Price                 6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [6]: df.describe()

Out[6]:

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

```
In [7]: df.isna().any()
```

```
Out[7]: S.No.                False
        Name                False
        Location            False
        Year                False
        Kilometers_Driven   False
        Fuel_Type           False
        Transmission        False
        Owner_Type          False
        Mileage              True
        Engine               True
        Power                True
        Seats                True
        New_Price            True
        Price                True
        dtype: bool
```

```
In [8]: df.isnull().sum()
```

```
Out[8]: S.No.                0
        Name                0
        Location            0
        Year                0
        Kilometers_Driven   0
        Fuel_Type           0
        Transmission        0
        Owner_Type          0
        Mileage              2
        Engine              46
        Power               46
        Seats               53
        New_Price           6247
        Price               1234
        dtype: int64
```

```
In [9]: df.fillna(value=0,inplace=True)
```

```
In [10]: df.isnull().sum()
```

```
Out[10]: S.No.          0
         Name          0
         Location      0
         Year          0
         Kilometers_Driven  0
         Fuel_Type      0
         Transmission   0
         Owner_Type     0
         Mileage        0
         Engine         0
         Power          0
         Seats          0
         New_Price      0
         Price          0
         dtype: int64
```

```
In [11]: df=df[['Year','Price']]

         df.columns=['year','price']
```

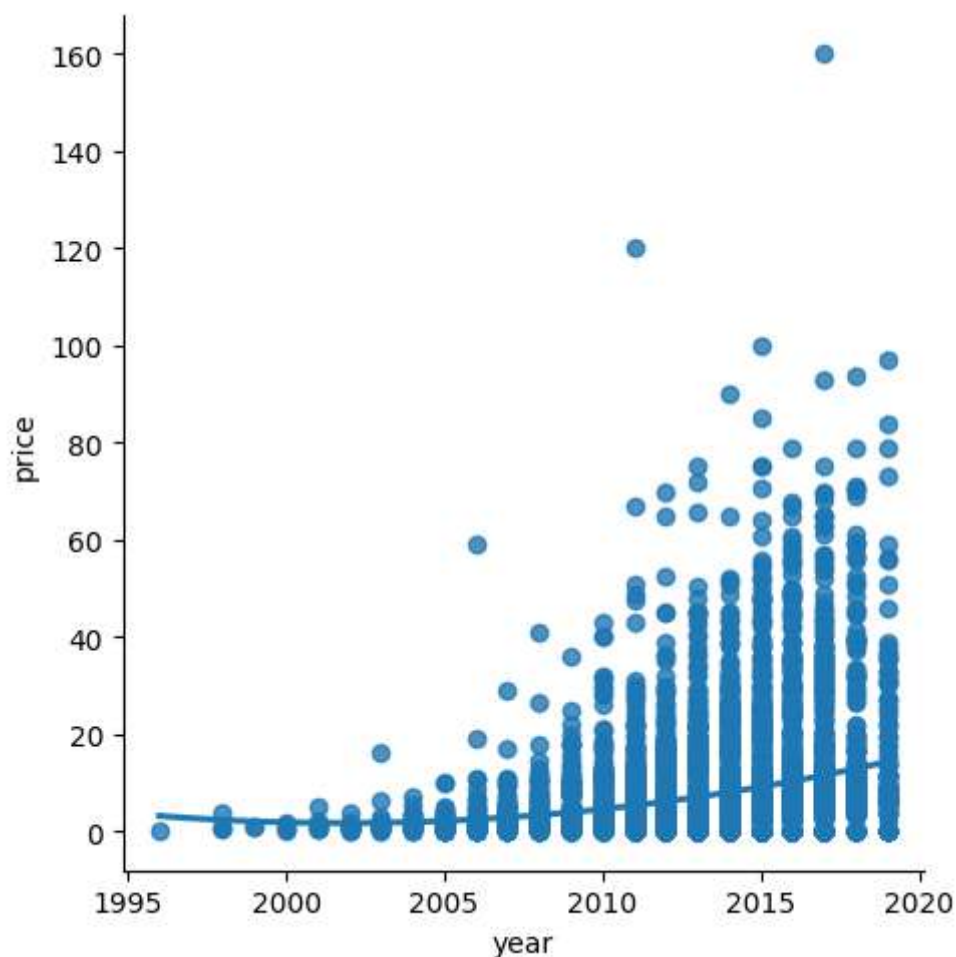
```
In [12]: df.head()
```

```
Out[12]:
```

	year	price
0	2010	1.75
1	2015	12.50
2	2011	4.50
3	2012	6.00
4	2013	17.74

```
In [13]: sns.lmplot(x='year',y='price',data=df,order=2,ci=None)
```

```
Out[13]: <seaborn.axisgrid.FacetGrid at 0x2e7b80be500>
```



```
In [14]: X=np.array(df['year']).reshape(-1,1)
y=np.array(df['price']).reshape(-1,1)
```

```
In [15]: df.describe()
```

```
Out[15]:
```

	year	price
count	7253.000000	7253.000000
mean	2013.365366	7.866665
std	3.254421	10.796286
min	1996.000000	0.000000
25%	2011.000000	2.290000
50%	2014.000000	4.650000
75%	2016.000000	8.400000
max	2019.000000	160.000000

```
In [16]: df.dropna(inplace=True)
```

C:\Users\91903\AppData\Local\Temp\ipykernel_16648\1379821321.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

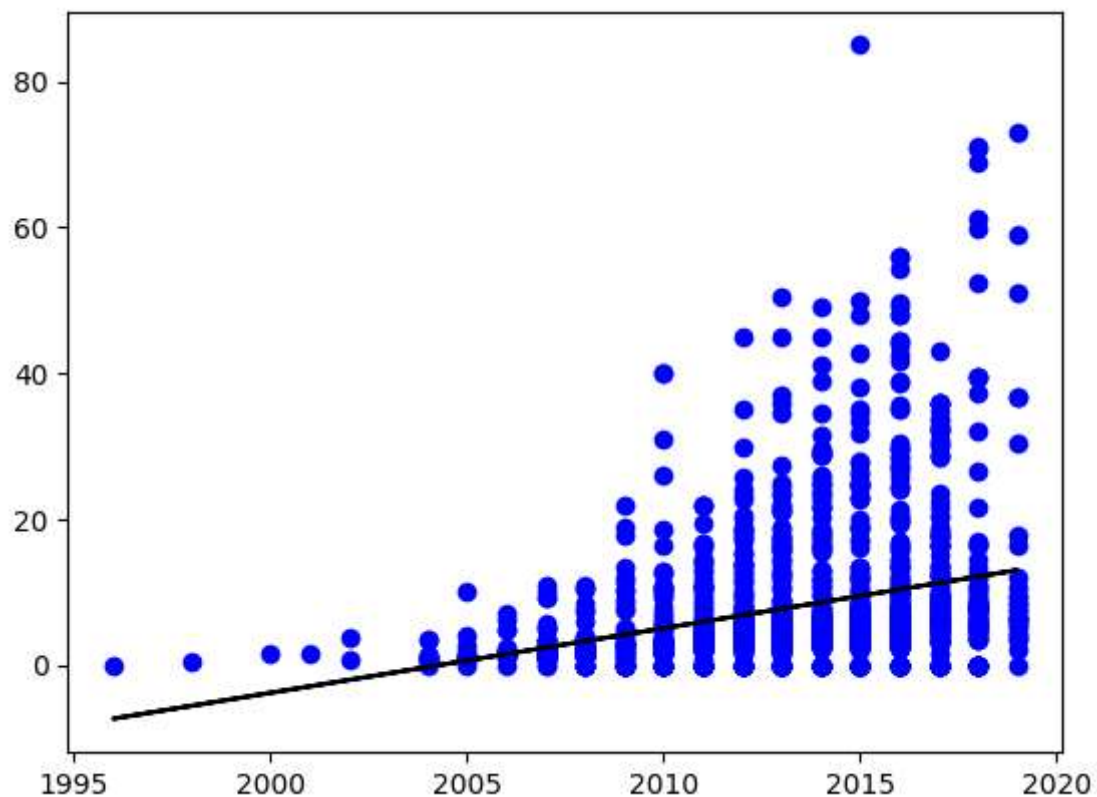
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df.dropna(inplace=True)
```

```
In [17]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
reg=LinearRegression()
reg.fit(X_train,y_train)
print(reg.score(X_test,y_test))
```

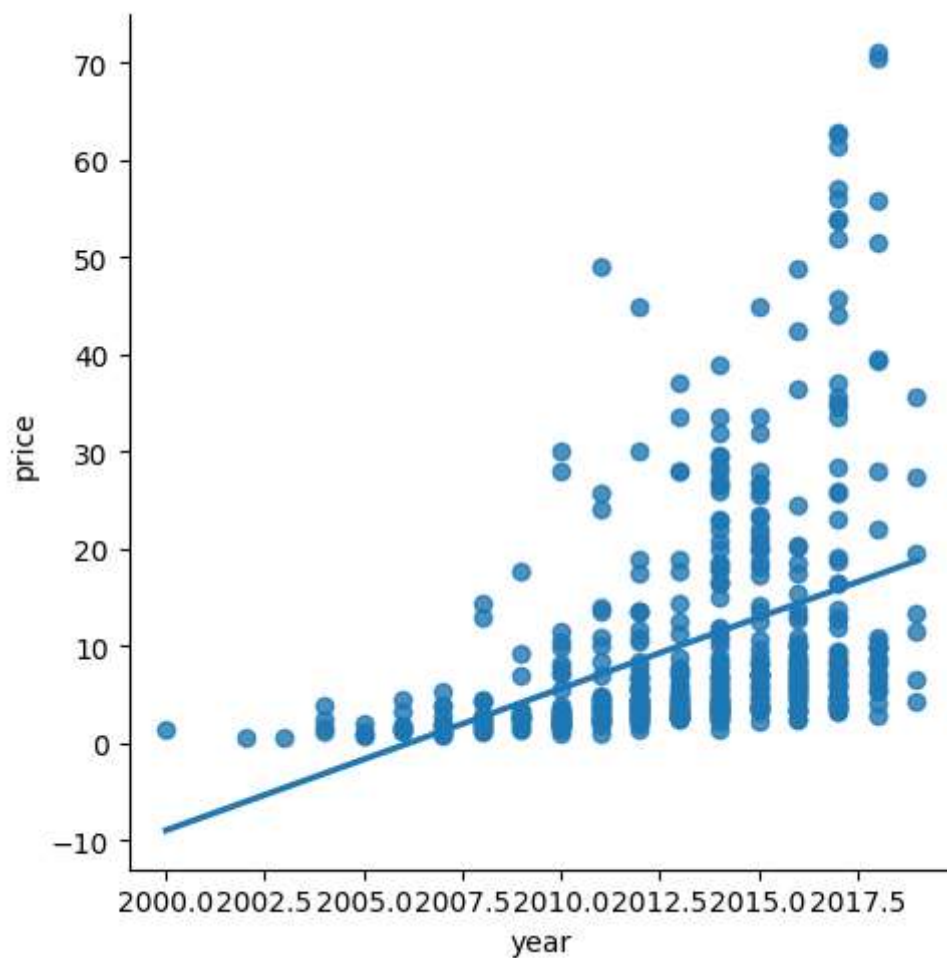
0.06623299550060102

```
In [18]: y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```




```
In [19]: df500=df[:][:500]  
sns.lmplot(x='year',y='price',data=df500,order=1,ci=None)
```

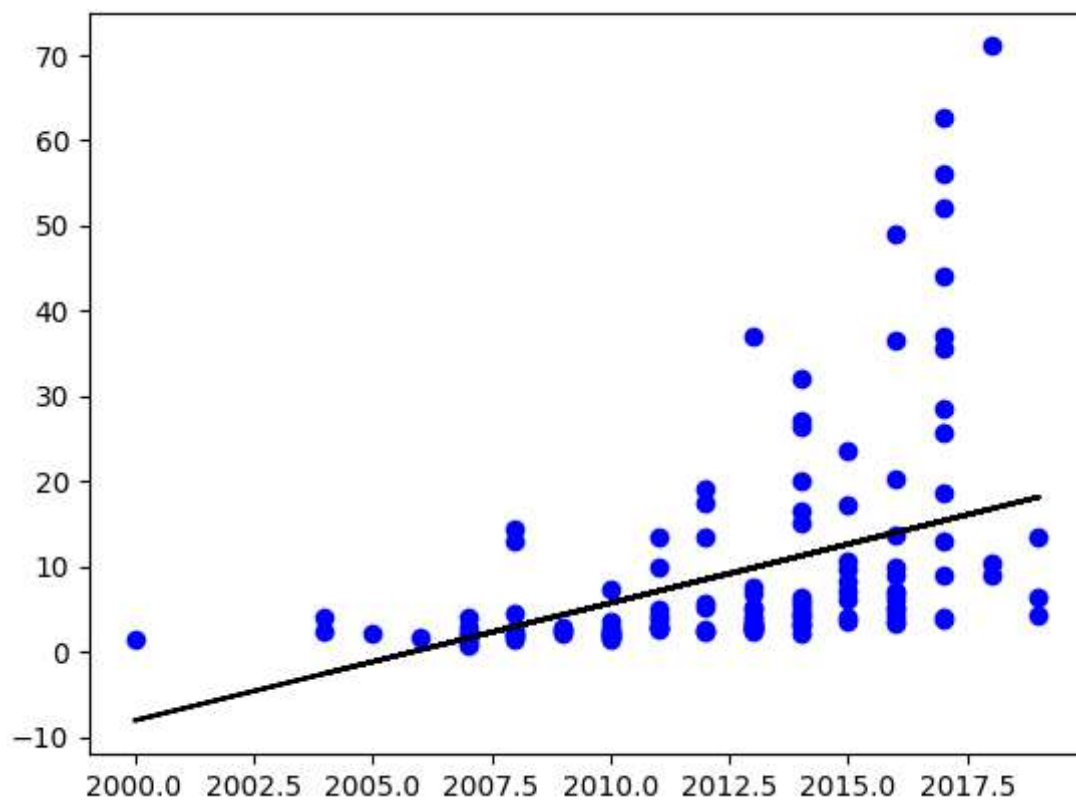
Out[19]: <seaborn.axisgrid.FacetGrid at 0x2e7a5efff40>



```
In [20]: df500.fillna(method='ffill',inplace=True)
X=np.array(df500['year']).reshape(-1,1)
y=np.array(df500['price']).reshape(-1,1)
df500.dropna(inplace=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
reg=LinearRegression()
reg.fit(X_train,y_train)
print("Regression:",reg.score(X_test,y_test))
y_pred=reg.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show
```

Regression: 0.19969768759053663

Out[20]: <function matplotlib.pyplot.show(close=None, block=None)>

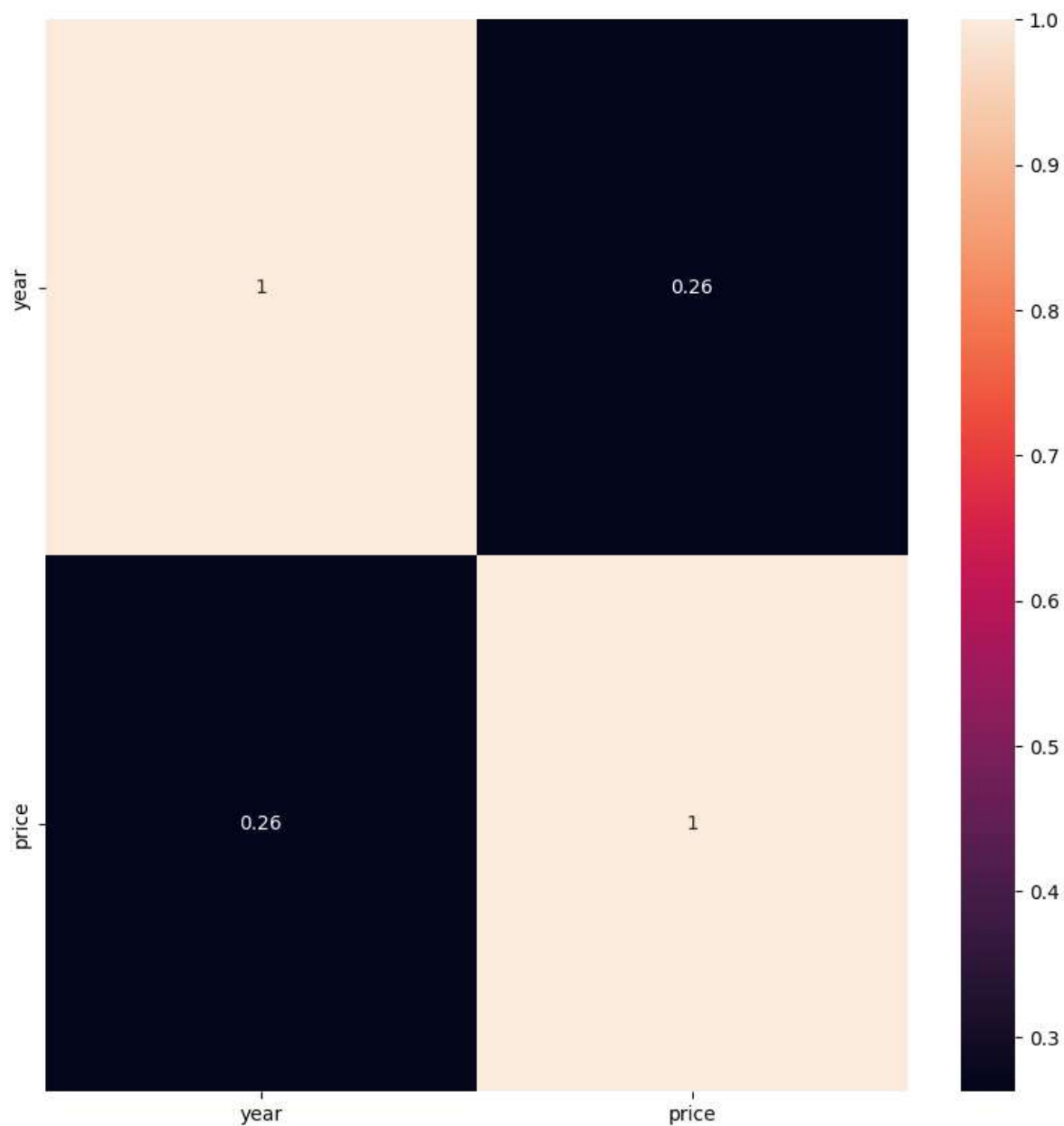


```
In [21]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
r2=r2_score(y_test,y_pred)
print("R2 score: ",r2)
```

R2 score: 0.19969768759053663

```
In [22]: plt.figure(figsize = (10, 10))  
sns.heatmap(df.corr(), annot = True)
```

Out[22]: <Axes: >



Logistic Regression

```
In [23]: import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
import warnings
warnings.simplefilter(action='ignore')
```

```
In [24]: da=pd.read_csv(r"C:\Users\91903\Downloads\used_cars_data.csv")
da
```

Out[24]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Se
...
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	
7252	7252	Mercedes- Benz E- Class 2009- 2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

7253 rows × 14 columns



In [25]: `da.head()`

Out[25]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second

In [26]: `da.tail()`

Out[26]:

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_1
7248	7248	Volkswagen Vento Diesel Trendline	Hyderabad	2011	89411	Diesel	Manual	
7249	7249	Volkswagen Polo GT TSI	Mumbai	2015	59000	Petrol	Automatic	
7250	7250	Nissan Micra Diesel XV	Kolkata	2012	28000	Diesel	Manual	
7251	7251	Volkswagen Polo GT TSI	Pune	2013	52262	Petrol	Automatic	1
7252	7252	Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...	Kochi	2014	72443	Diesel	Automatic	

In [27]: `da.describe()`

Out[27]:

	S.No.	Year	Kilometers_Driven	Seats	Price
count	7253.000000	7253.000000	7.253000e+03	7200.000000	6019.000000
mean	3626.000000	2013.365366	5.869906e+04	5.279722	9.479468
std	2093.905084	3.254421	8.442772e+04	0.811660	11.187917
min	0.000000	1996.000000	1.710000e+02	0.000000	0.440000
25%	1813.000000	2011.000000	3.400000e+04	5.000000	3.500000
50%	3626.000000	2014.000000	5.341600e+04	5.000000	5.640000
75%	5439.000000	2016.000000	7.300000e+04	5.000000	9.950000
max	7252.000000	2019.000000	6.500000e+06	10.000000	160.000000

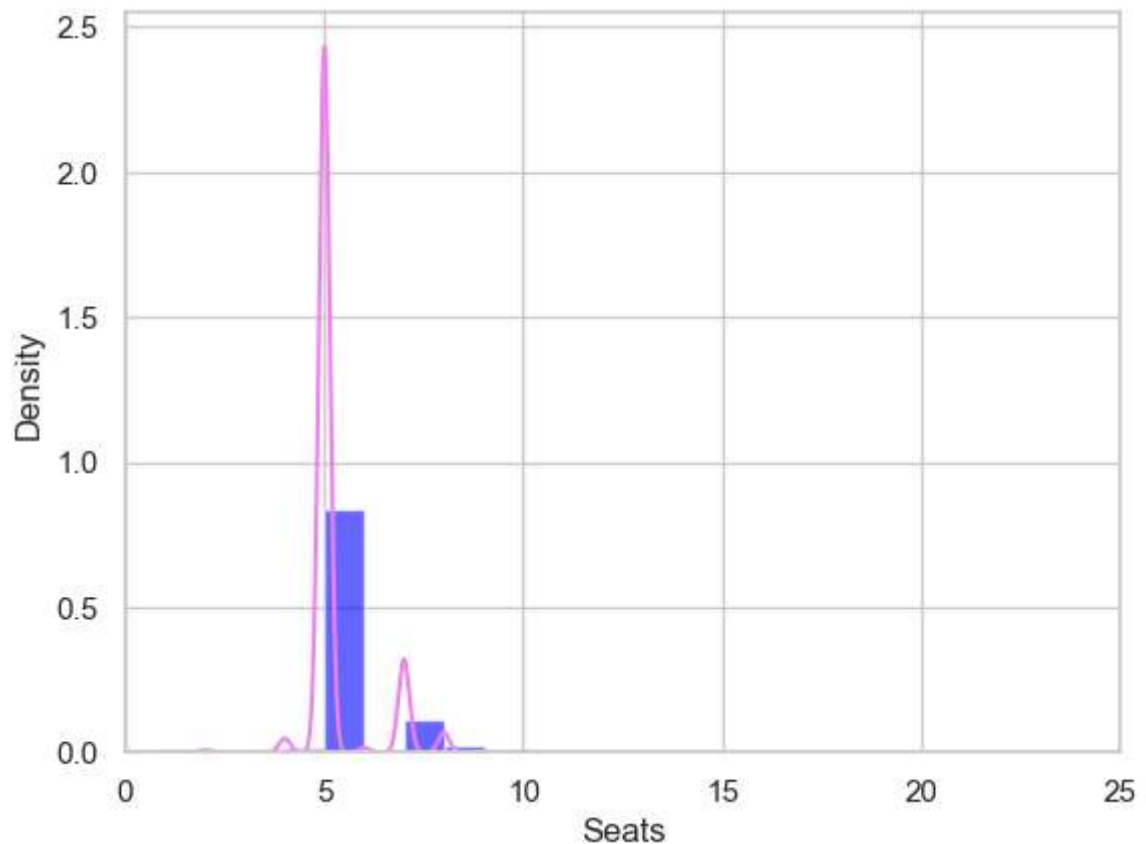
In [28]: `da.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   S.No.                  7253 non-null  int64
1   Name                   7253 non-null  object
2   Location               7253 non-null  object
3   Year                   7253 non-null  int64
4   Kilometers_Driven      7253 non-null  int64
5   Fuel_Type              7253 non-null  object
6   Transmission           7253 non-null  object
7   Owner_Type             7253 non-null  object
8   Mileage                7251 non-null  object
9   Engine                 7207 non-null  object
10  Power                  7207 non-null  object
11  Seats                  7200 non-null  float64
12  New_Price              1006 non-null  object
13  Price                  6019 non-null  float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [29]: `da.shape`

Out[29]: (7253, 14)

```
In [30]: ax=da['Seats'].hist(bins=10,density=True,stacked=True,color='blue',alpha=0.6)
da['Seats'].plot(kind='density',color='violet')
ax.set(xlabel='Seats')
plt.xlim(-0,25)
plt.show()
```



```
In [31]: print(da["Seats"].mean(skipna=True))
print(da["Seats"].median(skipna=True))
```

```
5.279722222222222
5.0
```

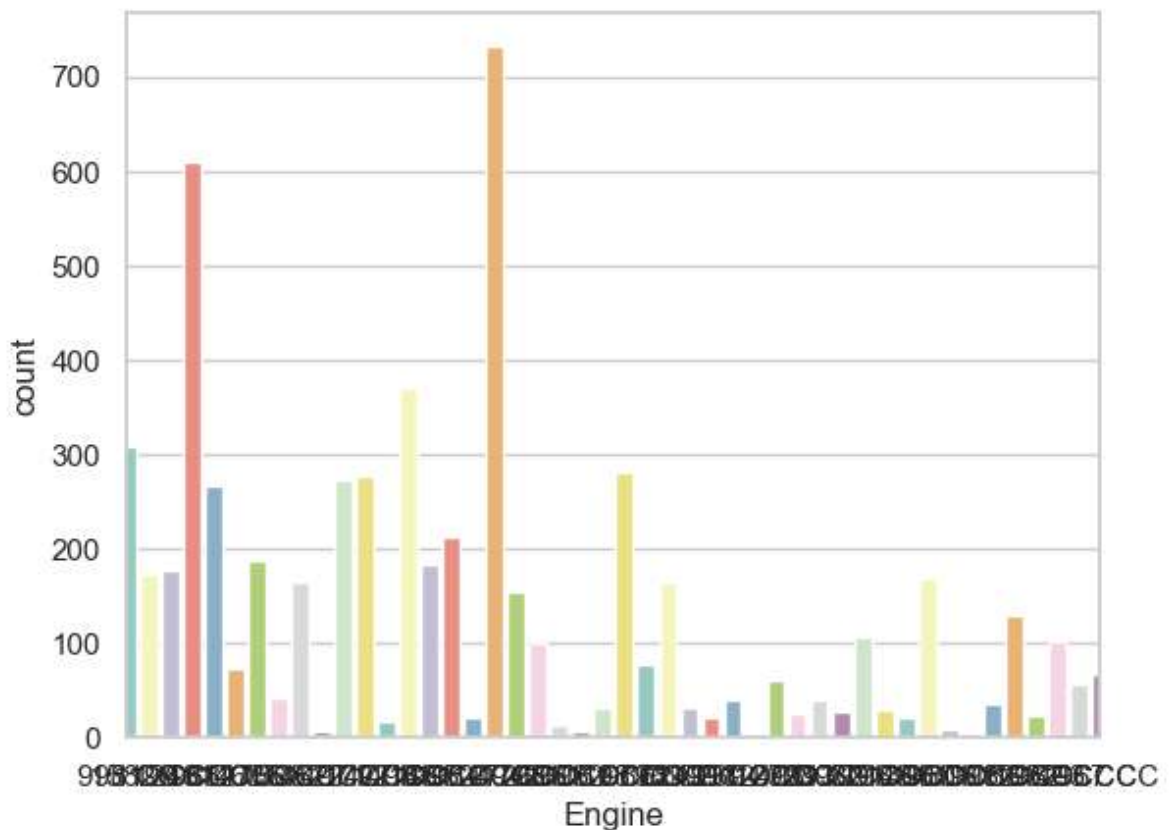
```
In [32]: print(da["New_Price"].isnull().sum()/da.shape[0])
print(da["Price"].isnull().sum()/da.shape[0])
print(da["Mileage"].isnull().sum()/da.shape[0])
print(da["Engine"].isnull().sum()/da.shape[0])
print(da["Power"].isnull().sum()/da.shape[0])
```

```
0.8612987729215497
0.1701364952433476
0.0002757479663587481
0.006342203226251206
0.006342203226251206
```



```
In [33]: print(da['Engine'].value_counts())
sns.countplot(x='Engine',data=da,palette='Set3')
plt.xlim(-0,45)
plt.show()
```

```
Engine
1197 CC    732
1248 CC    610
1498 CC    370
998 CC     309
1198 CC    281
...
1489 CC     1
1422 CC     1
2706 CC     1
1978 CC     1
1389 CC     1
Name: count, Length: 150, dtype: int64
```



```
In [34]: data=data.copy()
data['Seats'].fillna(da['Seats'].median(skipna=True),inplace=True)
data.drop('New_Price',axis=1,inplace=True)
data['Price'].fillna(da['Price'].median(skipna=True),inplace=True)
data['Mileage'].fillna(da['Mileage'].value_counts().idxmax(),inplace=True)
data.drop('Engine',axis=1,inplace=True)
data.drop('Power',axis=1,inplace=True)
```

In [35]: `data.isnull().sum()`

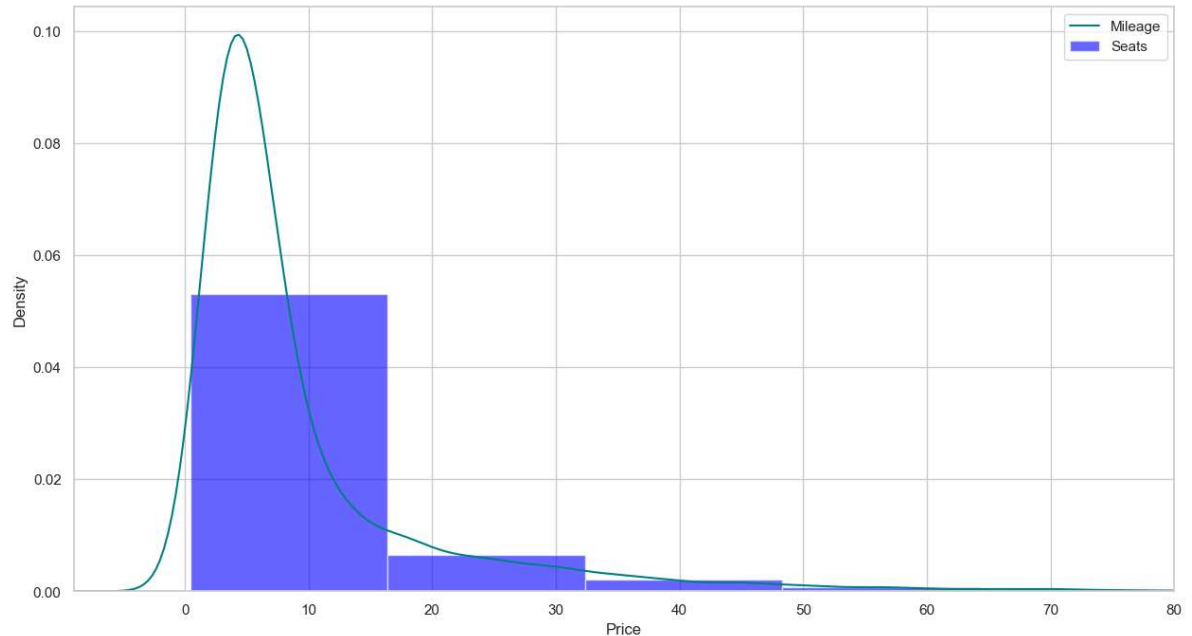
```
Out[35]: S.No.          0
         Name          0
         Location      0
         Year          0
         Kilometers_Driven  0
         Fuel_Type      0
         Transmission   0
         Owner_Type     0
         Mileage        0
         Seats          0
         Price          0
         dtype: int64
```

In [36]: `data.head()`

```
Out[36]:
```

	S.No.	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
0	0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First
1	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First
2	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First
3	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First
4	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second

```
In [37]: plt.figure(figsize=(15,8))
ax=da["Price"].hist(bins=10,density=True,stacked=True,color='blue',alpha=0.6)
da["Price"].plot(kind='density',color='teal')
ax.legend(['Mileage','Seats'])
ax.set(xlabel='Price')
plt.xlim(-9,80)
plt.show()
```



```
In [38]: training=pd.get_dummies(data,columns=["S.No."])
final_train=training
final_train.head()
```

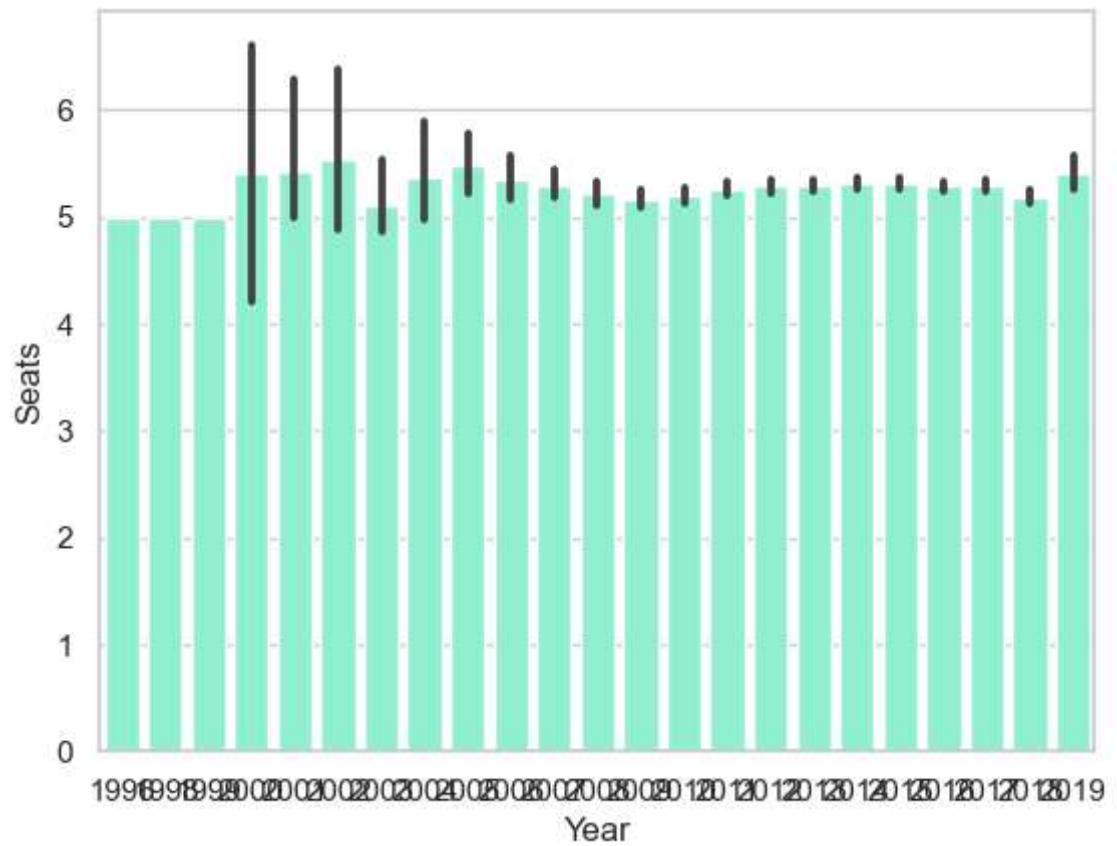
Out[38]:

Owner_Type	Mileage	Seats	Price	...	S.No._7243	S.No._7244	S.No._7245	S.No._7246	S.No._724
First	26.6 km/kg	5.0	1.75	...	False	False	False	False	Fals
First	19.67 kmpl	5.0	12.50	...	False	False	False	False	Fals
First	18.2 kmpl	5.0	4.50	...	False	False	False	False	Fals
First	20.77 kmpl	7.0	6.00	...	False	False	False	False	Fals
Second	15.2 kmpl	5.0	17.74	...	False	False	False	False	Fals

```
In [39]: sns.barplot(x='Price',y='Year',data=final_train,color='mediumturquoise')  
plt.show()
```



```
In [40]: import seaborn as sns
import matplotlib.pyplot as plt
sns.barplot(x='Year',y='Seats',data=da,color='aquamarine')
plt.show()
```



```
In [ ]:
```