Simple Linear Regression

In [118…]
```
import pandas as pd
import numpy as np
```

In [119…]
```
df=pd.DataFrame()
```

In [120…]
```
a1=pd.read_csv('cars_details.csv')
a1
```

Out[120]:

	name	year	km_driven	fuel	seller_type	transmission	owner	selling_price
0	Maruti 800 AC	2007	70000	Petrol	Individual	Manual	First Owner	60000
1	Maruti Wagon R LXI Minor	2007	50000	Petrol	Individual	Manual	First Owner	135000
2	Hyundai Verna 1.6 SX	2012	100000	Diesel	Individual	Manual	First Owner	600000
3	Datsun RediGO T Option	2017	46000	Petrol	Individual	Manual	First Owner	250000
4	Honda Amaze VX i-DTEC	2014	141000	Diesel	Individual	Manual	Second Owner	450000
...	...	...	...	...	...	...	...	...
4335	Hyundai i20 Magna 1.4 CRDi (Diesel)	2014	80000	Diesel	Individual	Manual	Second Owner	409999
4336	Hyundai i20 Magna 1.4 CRDi	2014	80000	Diesel	Individual	Manual	Second Owner	409999
4337	Maruti 800 AC BSIII	2009	83000	Petrol	Individual	Manual	Second Owner	110000
4338	Hyundai Creta 1.6 CRDi SX Option	2016	90000	Diesel	Individual	Manual	First Owner	865000
4339	Renault KWID RXT	2016	40000	Petrol	Individual	Manual	First Owner	225000

4340 rows × 8 columns

In [121…]
```
df1=pd.DataFrame()
```

In [122…]
```
df1=a1[["km_driven","selling_price"]]  # new dataframe containing one input and one output
df1
```

Out[122]:

	km_driven	selling_price
0	70000	60000
1	50000	135000
2	100000	600000
3	46000	250000
4	141000	450000
...	...	...
4335	80000	409999
4336	80000	409999
4337	83000	110000
4338	90000	865000
4339	40000	225000

4340 rows × 2 columns

In [123…]
```
X = pd.DataFrame(df1.iloc[:400, 0:1].values)
Y = pd.DataFrame(df1.iloc[:400, 1:2].values)
```

In [124…]
```
X
```

Out[124]:

	0
0	70000
1	50000
2	100000
3	46000
4	141000
...	...
395	50000
396	45000
397	110000
398	49000
399	35000

400 rows × 1 columns

In [125…

```
Y
```

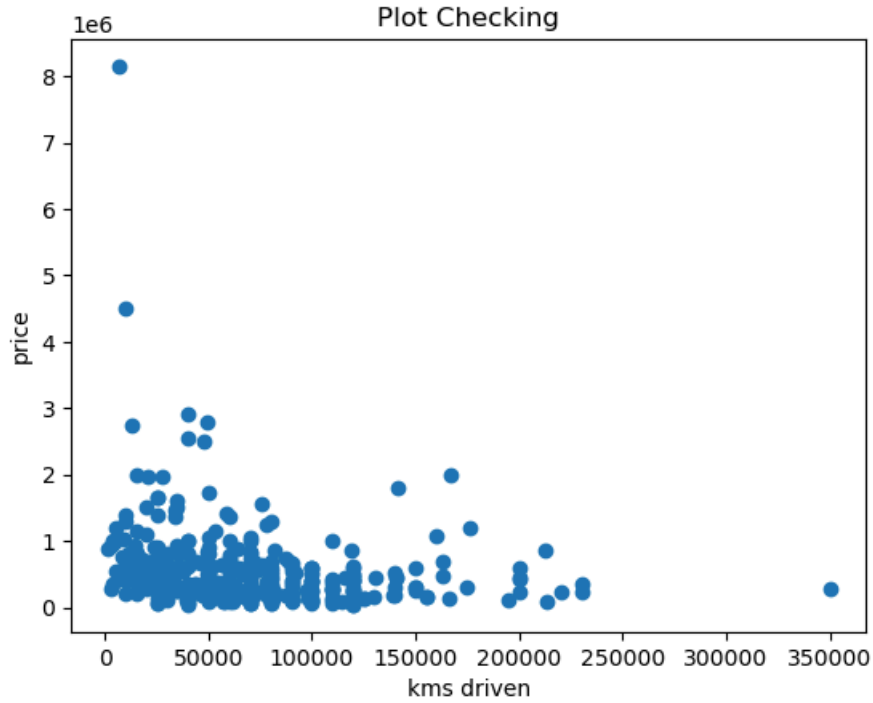
Out[125]:

	0
0	60000
1	135000
2	600000
3	250000
4	450000
...	...
395	400000
396	315000
397	1000000
398	500000
399	1600000

400 rows × 1 columns

In [126…

```
import matplotlib.pyplot as plt
#plt.plot(X,Y, color = 'red')
plt.scatter(X,Y)
# naming the x axis
plt.xlabel('kms driven')
# naming the y axis
plt.ylabel('price')

# giving a title to my graph
plt.title('Plot Checking')

# function to show the plot
plt.show()
```



In [129…
```
from sklearn.model_selection import train_test_split
```

In [130…
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=3)
```

In [131…
```
X_train
```

Out[131]:

	0
236	163000
65	73300
286	49000
180	23000
323	140000
…	…
256	49000
131	81000
249	32000
152	53772
362	81925

280 rows × 1 columns

In [132…
```
X_test
```

Out[132]:

	0
376	97000
16	46000
365	90000
82	120000
107	100000
...	...
25	25000
220	56000
67	66764
229	80000
238	120000

120 rows × 1 columns

In [133…
```
Y_train
```

Out[133]:

	0
236	700000
65	240000
286	395000
180	640000
323	525000
...	...
256	290000
131	125000
249	500000
152	210000
362	240000

280 rows × 1 columns

In [134…
```
Y_test
```

Out[134]:

	0
376	260000
16	250000
365	400000
82	80000
107	600000
...	...
25	1650000
220	640000
67	151000
229	285000
238	120000

120 rows × 1 columns

In [135…
```
from sklearn.linear_model import LinearRegression
```

In [136…
```
model = LinearRegression()
```

In [137…
```
model.fit(X_train, Y_train)
```

Out[137]:
```
▾ LinearRegression
LinearRegression()
```

In [138…
```
intercept=model.intercept_
intercept
```

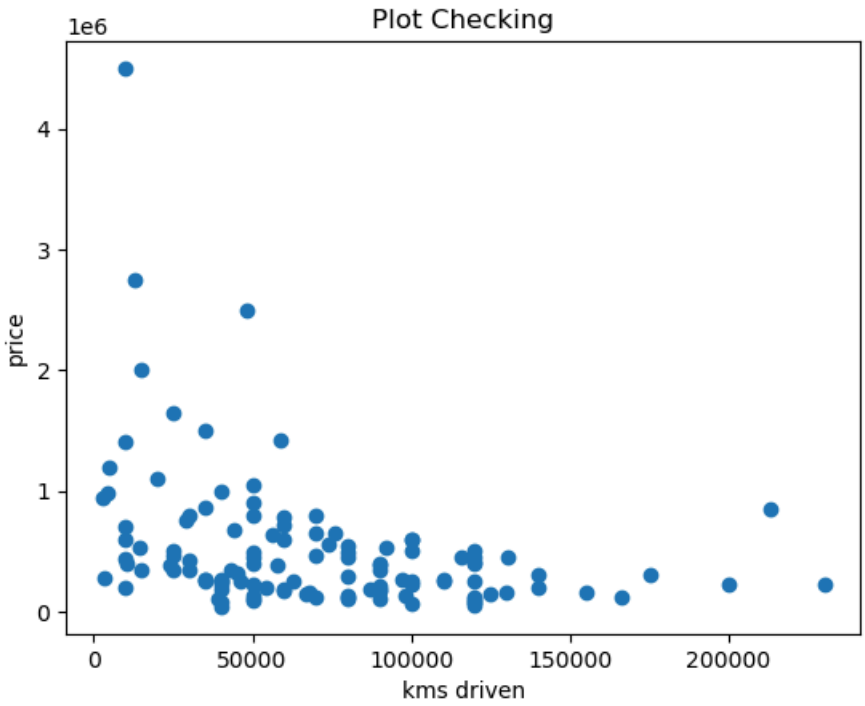
Out[138]:
```
array([691739.15086697])
```

In [139…
```
coef=model.coef_
coef
```

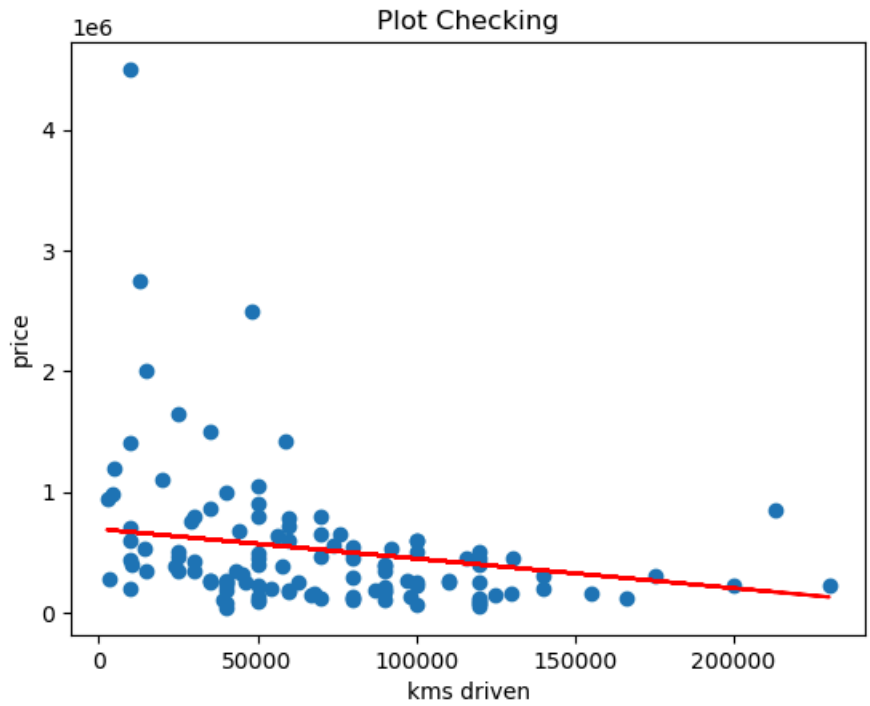
Out[139]:
```
array([[-2.44877617]])
```

In [140…
```
Y_pred=model.predict(X_test)
```

In [141…
```
import matplotlib.pyplot as plt
#plt.plot(X_test,Y_test, color = 'red')
plt.scatter(X_test,Y_test)
# naming the x axis
plt.xlabel('kms driven')
# naming the y axis
plt.ylabel('price')

# giving a title to my graph
plt.title('Plot Checking')

# function to show the plot
plt.show()
```



In [142…
```
import matplotlib.pyplot as plt
#plt.plot(X_test,Y_test, color = 'red')
plt.scatter(X_test,Y_test)
plt.plot(X_test,Y_pred,color = 'red')
# naming the x axis
plt.xlabel('kms driven')
# naming the y axis
plt.ylabel('price')

# giving a title to my graph
plt.title('Plot Checking')

# function to show the plot
plt.show()
```



In [143…
```
from sklearn.metrics import mean_squared_error,mean_absolute_error, r2_score
```

In [147…
```
# Mean Square Error
mse = mean_squared_error(Y_test, Y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 304937373452.3693
```

In [148…
```
# Mean Absolute Error
mae = mean_absolute_error(Y_test, Y_pred)
print("Mean Absolute Error:", mae)
```

```
Mean Absolute Error: 332048.5841150246
```

In [149…
```
# R-squared score
r_squared = r2_score(Y_test, Y_pred)
print("R-squared score:", r_squared)
```

```
R-squared score: 0.09382296921253097
```

Multiple Linear Regression

In [150…
```
a2=a1.iloc[:400,:]
a2
```

Out[150]:

	name	year	km_driven	fuel	seller_type	transmission	owner	selling_price
0	Maruti 800 AC	2007	70000	Petrol	Individual	Manual	First Owner	60000
1	Maruti Wagon R LXI Minor	2007	50000	Petrol	Individual	Manual	First Owner	135000
2	Hyundai Verna 1.6 SX	2012	100000	Diesel	Individual	Manual	First Owner	600000
3	Datsun RediGO T Option	2017	46000	Petrol	Individual	Manual	First Owner	250000
4	Honda Amaze VX i-DTEC	2014	141000	Diesel	Individual	Manual	Second Owner	450000
...	...	...	...	...	...	...	...	...
395	Mahindra Bolero SLX	2010	50000	Diesel	Individual	Manual	First Owner	400000
396	Datsun GO Plus A	2015	45000	Petrol	Individual	Manual	Second Owner	315000
397	Toyota Fortuner 4x4 MT	2014	110000	Diesel	Individual	Manual	First Owner	1000000
398	Ford Ecosport 1.5 DV5 MT Titanium	2014	49000	Diesel	Individual	Manual	First Owner	500000
399	Mahindra XUV500 W11 AT BSIV	2018	35000	Diesel	Individual	Automatic	First Owner	1600000

400 rows × 8 columns

In [151…]
```
df2=pd.DataFrame()
df2
```

Out[151]: —

In [152…]
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [153…]
```
df2['name'] = encoder.fit_transform(a2['name'])
df2['year']=a2['year']
df2['km_driven']=a2['km_driven']
df2['fuel'] = encoder.fit_transform(a2['fuel'])
df2['seller_type'] = encoder.fit_transform(a2['seller_type'])
df2['transmission'] = encoder.fit_transform(a2['transmission'])
df2['owner'] = encoder.fit_transform(a2['owner'])
df2['selling_price']=a2['selling_price']
```

In [154…]
```
df2
```

Out[154]:

	name	year	km_driven	fuel	seller_type	transmission	owner	selling_price
0	159	2007	70000	3	1	1	0	60000
1	228	2007	50000	3	1	1	0	135000
2	101	2012	100000	1	1	1	0	600000
3	35	2017	46000	3	1	1	0	250000
4	54	2014	141000	1	1	1	2	450000
...	...	...	...	...	...	...	...	...
395	134	2010	50000	1	1	1	0	400000
396	32	2015	45000	3	1	1	2	315000
397	293	2014	110000	1	1	1	0	1000000
398	41	2014	49000	1	1	1	0	500000
399	151	2018	35000	1	1	0	0	1600000

400 rows × 8 columns

In [155…]
```
X=df2.iloc[:,1:3]
X
```

Out[155]:

	year	km_driven
0	2007	70000
1	2007	50000
2	2012	100000
3	2017	46000
4	2014	141000
...	...	...
395	2010	50000
396	2015	45000
397	2014	110000
398	2014	49000
399	2018	35000

400 rows × 2 columns

In [156…]
```
Y=df2.iloc[:,7:8]
Y
```

Out[156]:

	selling_price
0	60000
1	135000
2	600000
3	250000
4	450000
...	...
395	400000
396	315000
397	1000000
398	500000
399	1600000

400 rows × 1 columns

In [157…
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

In [158…
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, Y_train)
```

Out[158]:
```
▾ LinearRegression
LinearRegression()
```

In [159…
```
Y_pred = model.predict(X_test)
```

In [160…
```
coef_year, coef_km_driven = model.coef_[0]
print(coef_year, coef_km_driven)
```

```
43880.50176799953 -1.2071176049068386
```

In [161…
```
intercept = model.intercept_
intercept
```

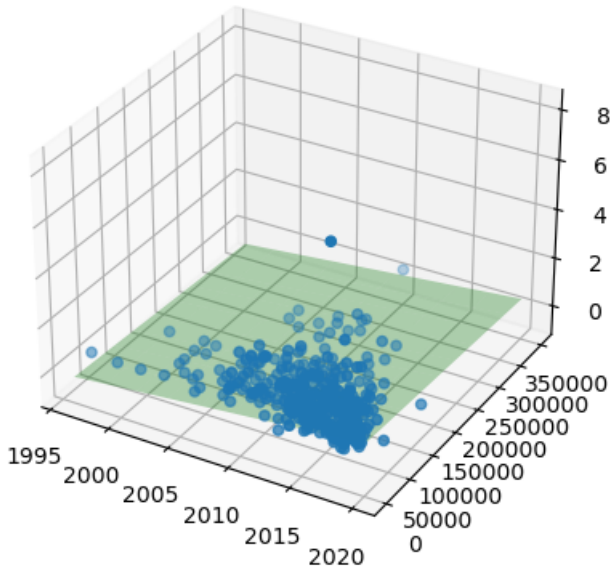
Out[161]:
```
array([-87735502.86920957])
```

In [162…
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
```

In [163…
```
# Creating a meshgrid for the plane
x_values = np.linspace(min(X["year"]), max(X["year"]), 10)
y_values = np.linspace(min(X["km_driven"]), max(X["km_driven"]), 10)
x_mesh, y_mesh = np.meshgrid(x_values, y_values)
```

In [164…
```
# Calculating z values for the plane using the plane equation
z_mesh = coef_year * x_mesh + coef_km_driven * y_mesh + intercept
```

In [165…
```
# Plotting the scatter plot
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(X["year"], X["km_driven"], Y)

# Plotting the best fit plane
ax.plot_surface(x_mesh, y_mesh, z_mesh, alpha = 0.3, color='green')

plt.show()
```



In [166…
```
from sklearn.metrics import mean_squared_error,mean_absolute_error, r2_score
```

In [167…
```
# Mean Square Error
mse = mean_squared_error(Y_test, Y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 296249482191.81506
```

In [168…
```
# Mean Absolute Error
mae = mean_absolute_error(Y_test, Y_pred)
print("Mean Absolute Error:", mae)
```

```
Mean Absolute Error: 299086.9784888638
```

In [169…
```
# R-squared score
r_squared = r2_score(Y_test, Y_pred)
print("R-squared score:", r_squared)
```

```
R-squared score: 0.19865556486397873
```