Double-click (or enter) to edit

Car Price Prediction using Linear Regression

Importing libraries

```
import pandas as pd
```

```
import numpy as np
```

Importing CSV as DataFrame

```
df=pd.read_csv('/content/sample_data/Car Price.csv')
```

Getting the first five rows of the DataFrame

```
df.head()
```

	Brand	Model	Year	Selling_Price	KM_Driven	Fuel	Seller_Type	Transmission
0	Maruti	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual
1	Maruti	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual
2	Hyundai	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual

Datsun

Option

DTEC

Getting information of DataFrame

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Brand          4340 non-null   object
 1   Model          4340 non-null   object
 2   Year           4340 non-null   int64 
 3   Selling_Price  4340 non-null   int64 
 4   KM_Driven      4340 non-null   int64 
 5   Fuel           4340 non-null   object
 6   Seller_Type    4340 non-null   object
 7   Transmission   4340 non-null   object
 8   Owner          4340 non-null   object
dtypes: int64(3), object(6)
memory usage: 305.3+ KB
```

Getting summary statistics

```
df.describe()
```

	Year	Selling_Price	KM_Driven
count	4340.000000	4.340000e+03	4340.000000
mean	2013.090783	5.041273e+05	66215.777419
std	4.215344	5.785487e+05	46644.102194
min	1992.000000	2.000000e+04	1.000000
25%	2011.000000	2.087498e+05	35000.000000
50%	2014.000000	3.500000e+05	60000.000000
75%	2016.000000	6.000000e+05	90000.000000

Getting categories and counts of categorical variables

```
df[['Brand']].value_counts()
```

```
Brand
Maruti           1280
Hyundai           821
Mahindra          365
Tata              361
Honda             252
Ford              238
Toyota            206
Chevrolet         188
Renault           146
Volkswagen        107
Skoda              68
Nissan             64
Audi               60
BMW                39
Fiat               37
Datsun             37
Mercedes-Benz      35
Mitsubishi          6
Jaguar              6
Land                5
Ambassador          4
Volvo               4
Jeep                3
OpelCorsa           2
MG                  2
Isuzu               1
Force               1
Daewoo              1
Kia                 1
dtype: int64
```

```
df[['Model']].value_counts()
```

```
Model
Maruti Swift Dzire VDI                  69
Maruti Alto 800 LXI                     59
Maruti Alto LXi                         47
Hyundai EON Era Plus                    35
Maruti Alto LX                          35
                                        ..
Mahindra KUV 100 G80 K4 Plus             1
Mahindra KUV 100 mFALCON D75 K8          1
Mahindra KUV 100 mFALCON D75 K8 AW       1
Mahindra KUV 100 mFALCON G80 K2 Plus     1
Volvo XC60 D5 Inscription                1
Length: 1491, dtype: int64
```

```
df[['Fuel']].value_counts()
```

```
Fuel
Diesel      2153
Petrol      2123
CNG           40
LPG           23
Electric       1
dtype: int64
```

```
df[['Seller_Type']].value_counts()
```

```
Seller_Type
Individual          3244
Dealer               994
Trustmark Dealer     102
dtype: int64
```

```
df[['Transmission']].value_counts()
```

```
Transmission
Manual          3892
Automatic        448
dtype: int64
```

```
df[['Owner']].value_counts()
```

```
Owner
First Owner             2832
Second Owner            1106
Third Owner              304
Fourth & Above Owner      81
Test Drive Car            17
dtype: int64
```

Getting column names

```
df.columns
```

```
Index(['Brand', 'Model', 'Year', 'Selling_Price', 'KM_Driven', 'Fuel',
       'Seller_Type', 'Transmission', 'Owner'],
      dtype='object')
```

Getting Shape of DataFrame

```
df.shape
```

```
(4340, 9)
```

Getting encoding of categorical features

```
df.replace({'Fuel':{'Petrol':0, 'Diesel':1, 'CNG':2, 'LPG':3, 'Electric':4}},inplace=True)
```

```
df.replace({'Seller_Type':{'Individual':0,'Dealer':1, 'Trustmark Dealer':2}},inplace=True)
```

```
df.replace({'Transmission':{'Manual':0,'Automatic':1}},inplace=True)
```

```
df.replace({'Owner':{'First Owner':0,'Second Owner':1, 'Third Owner':2, 'Fourth & Above Owner
```

Defining y(dependent/label/target variable) and X(independent/features/attribute variable)

```
y=df['Selling_Price']
```

```
y.shape
```

```
(4340,)
```

```
y
```

```
0        60000
1       135000
2       600000
3       250000
4       450000
        ...
4335    409999
4336    409999
4337    110000
4338    865000
4339    225000
Name: Selling_Price, Length: 4340, dtype: int64
```

```
X=df[['Year','KM_Driven','Fuel','Seller_Type','Transmission', 'Owner']]
```

```
X.shape
```

```
(4340, 6)
```

```
X
```

	Year	KM_Driven	Fuel	Seller_Type	Transmission	Owner
0	2007	70000	0	0	0	0
2	2012	100000	1	0	0	0
4337	2009	83000	0	0	0	1

4340 rows × 6 columns

Getting Train Test Split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.3, random_state=2529)
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((3038, 6), (1302, 6), (3038,), (1302,))
```

Getting Model Train

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()

lr.fit(X_train, y_train)
```

```
LinearRegression()
```

Getting model prediction

```
y_pred=lr.predict(X_test)

y_pred.shape
```

```
(1302,)
```

```
y_pred
```

```
array([502458.82786413, 646333.17428704, 521962.74075836, ...,
       620183.32683781, 315403.8278857 , 731862.54196037])
```

Getting model evaluation

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
mean_squared_error(y_test,y_pred)
```

```
193242972302.19553
```

```
mean_absolute_error(y_test,y_pred)
```

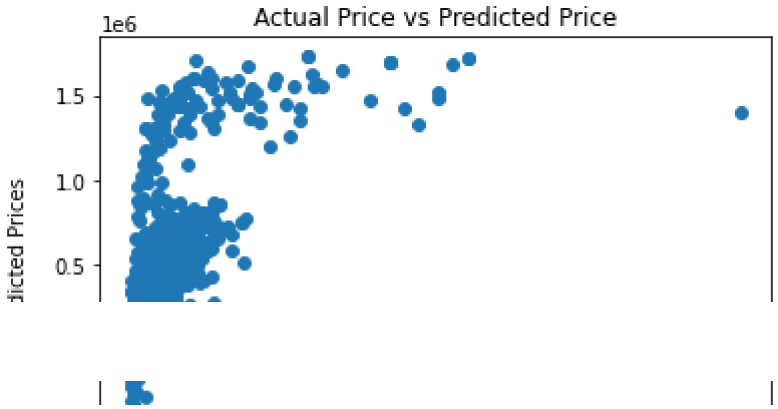
```
228808.95522977872
```

```
r2_score(y_test,y_pred)
```

```
0.4075563394370839
```

Getting visualization of actual vs predicted results

```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual Price vs Predicted Price')
plt.show()
```



Getting future predictions

```
df_new=df.sample(1)
```

```
df_new
```

	Brand	Model	Year	Selling_Price	KM_Driven	Fuel	Seller_Type	Transmission	
3562	Maruti	Maruti Swift VDI Optional	2017	459999	50000	1	0	0	

```
df_new.shape
```

```
(1, 9)
```

```
X_new=df_new.drop(['Brand','Model','Selling_Price'], axis=1)
```

```
y_pred_new=lr.predict(X_new)
```

```
y_pred_new
```

```
array([659625.04339699])
```

✓ 0s completed at 1:11 PM