Car Price Prediction

In [1]:
```
# importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# loading the data from csv
df = pd.read_csv('car data.csv')
```

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

Out[3]:

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	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

In [4]:
```
df.shape
```

Out[4]:
```
(301, 9)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Driven_kms     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Selling_type   301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Driven_kms',
       'Fuel_Type', 'Selling_type', 'Transmission', 'Owner'],
      dtype='object')
```

In [7]:
```
df.isnull()
```

Out[7]:

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_type	Transmission	Owner
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
296	False	False	False	False	False	False	False	False	False
297	False	False	False	False	False	False	False	False	False
298	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False

301 rows × 9 columns

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

Out[8]:
```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Driven_kms       0
Fuel_Type        0
Selling_type     0
Transmission     0
Owner            0
dtype: int64
```

In [9]:
```
df.describe()
```

Out[9]:

	Year	Selling_Price	Present_Price	Driven_kms	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	8.642584	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

In [10]:
```
df.Fuel_Type.value_counts()
```

Out[10]:
```
Petrol    239
Diesel     60
CNG         2
Name: Fuel_Type, dtype: int64
```

In [11]:
```
df.Selling_type.value_counts()
```

Out[11]:
```
Dealer        195
Individual    106
Name: Selling_type, dtype: int64
```

In [12]:
```
df.Transmission.value_counts()
```

Out[12]:
```
Manual       261
Automatic     40
Name: Transmission, dtype: int64
```

In [13]:
```
# encolding values
df.replace({'Fuel_Type':{'Petrol':0, 'Diesel':1, 'CNG':2}}, inplace=True)

df.replace({'Selling_type':{'Dealer':0, 'Individual':1}}, inplace=True)

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

In [14]:
```
df.head()
```

Out[14]:

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	0	0	0	0
1	sx4	2013	4.75	9.54	43000	1	0	0	0
2	ciaz	2017	7.25	9.85	6900	0	0	0	0
3	wagon r	2011	2.85	4.15	5200	0	0	0	0
4	swift	2014	4.60	6.87	42450	1	0	0	0

In [15]:
```
X = df.drop(['Car_Name', 'Selling_Price'], axis=1)
y = df['Selling_Price']
```

In [17]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=2)
```

In [18]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[18]:
```
LinearRegression
LinearRegression()
```

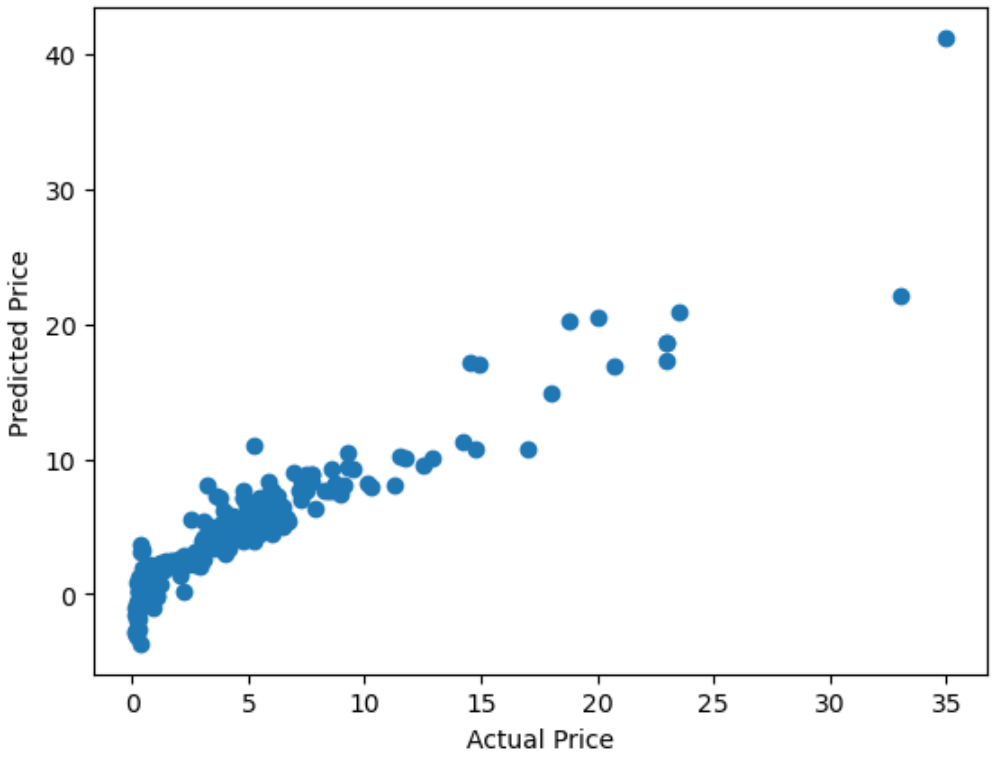
In [19]:
```
pred = lr.predict(X_train)
```

In [24]:
```
from sklearn.metrics import r2_score
rsq = r2_score(y_train, pred)
print("R square Error: ", rsq)
```

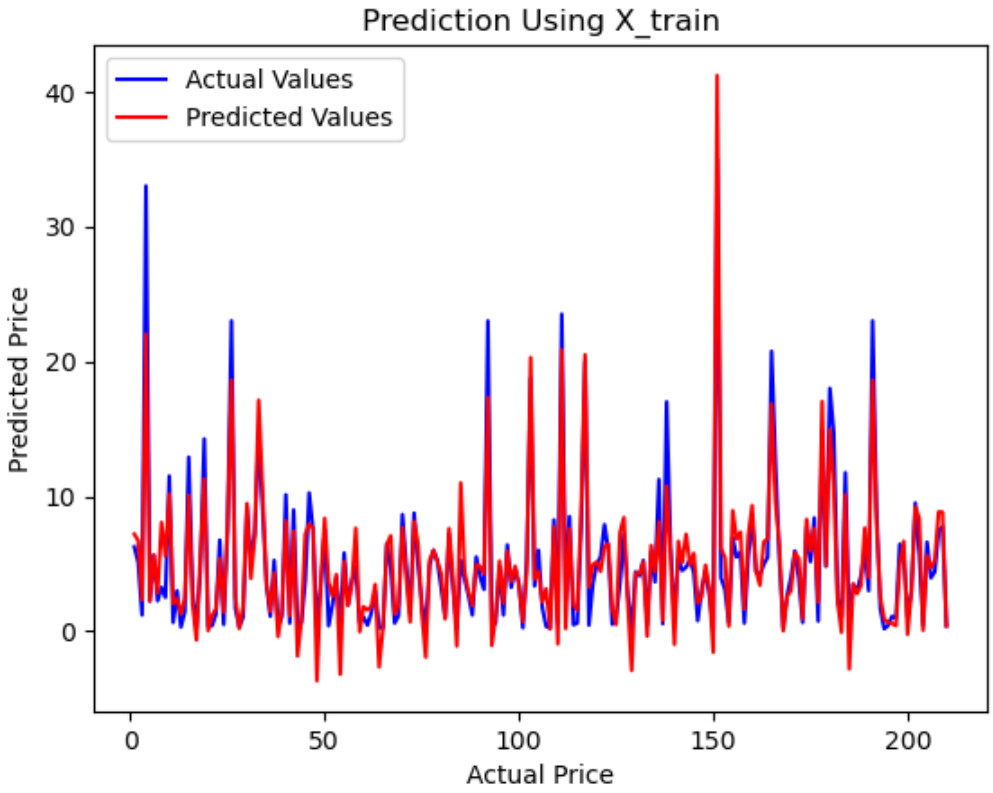
```
R square Error:  0.8860116999099485
```

In [26]:
```
plt.scatter(y_train, pred)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
```

Out[26]:
```
Text(0, 0.5, 'Predicted Price')
```



In [29]:
```
# plotting the actual and predicted value
c = [i for i in range(1, len(y_train)+1, 1)]
plt.plot(c, y_train, color='b', linestyle='-', label='Actual Values')
plt.plot(c, pred, color='r', linestyle='-', label='Predicted Values')
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Prediction Using X_train')
plt.legend()
plt.show()
```

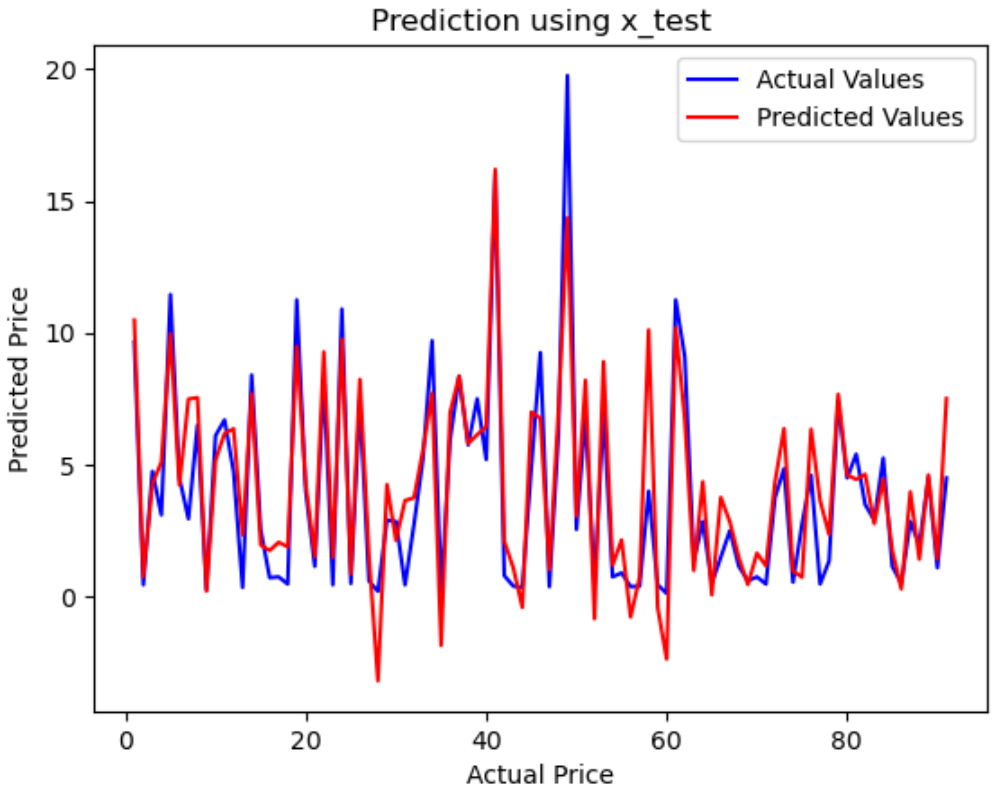


In [30]:
```
pred_2 = lr.predict(X_test)

rsq = r2_score(y_test, pred_2)
print('R Square Error y_test: ',rsq)
```

```
R Square Error y_test:  0.8191491844928422
```

In [32]:
```
c = [i for i in range (1,len(y_test)+1,1)]
plt.plot(c,y_test,color='b',linestyle='-',label="Actual Values")
plt.plot(c,pred_2,color='r',linestyle='-',label="Predicted Values")
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Prediction using x_test')
plt.legend()
plt.show()
```



You can find this project on **GitHub.**