```
In [10]: import warnings
```

```
In [11]: warnings.filterwarnings("ignore")
```

```
In [12]: import pandas as pd
```

```
In [13]: data=pd.read_csv("car.csv")
```

```
In [14]: #Display top 5 rows
```

```
In [15]: data.head()
```

```
Out[15]:
     Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  Seller_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 [16]:

```
data.tail()
```

Out[16]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
296	city	2016	9.50	11.6	33988	Diesel	Dealer	Manual	0
297	brio	2015	4.00	5.9	60000	Petrol	Dealer	Manual	0
298	city	2009	3.35	11.0	87934	Petrol	Dealer	Manual	0
299	city	2017	11.50	12.5	9000	Diesel	Dealer	Manual	0
300	brio	2016	5.30	5.9	5464	Petrol	Dealer	Manual	0

In [17]:

```
data.describe()
```

Out[17]:

	Year	Selling_Price	Present_Price	Kms_Driven	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	8.644115	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 [18]:

```
data.shape
```

Out[18]:

```
(301, 9)
```

In [19]:

```
data.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   Kms_Driven     301 non-null    int64  
 5   Fuel_Type      301 non-null    object 
 6   Seller_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 [23]:
```
data.isnull()
```

Out[23]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_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 [24]:
```
data.isnull().sum()
```

Out[24]:
```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

In [21]:
```
data.describe()
```

Out[21]:

	Year	Selling_Price	Present_Price	Kms_Driven	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	8.644115	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 [22]:
```
data.head(1)
```

Out[22]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0

In [25]:
```
import datetime
```

In [27]:
```
date_time=datetime.datetime.now()
```

In [30]:
```
data["Age"]=date_time.year-data["Year"]
```

```
In [31]: data.head()
```

```
Out[31]:
     Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  Seller_Type  Transmission  Owner  Age
0        ritz  2014           3.35           5.59       27000     Petrol       Dealer        Manual      0   10
1         sx4  2013           4.75           9.54       43000     Diesel       Dealer        Manual      0   11
2        ciaz  2017           7.25           9.85        6900     Petrol       Dealer        Manual      0    7
3     wagon r  2011           2.85           4.15        5200     Petrol       Dealer        Manual      0   13
4       swift  2014           4.60           6.87       42450     Diesel       Dealer        Manual      0   10
```

```
In [32]: data.drop("Year",axis=1,inplace=True)
```

```
In [33]: data
```

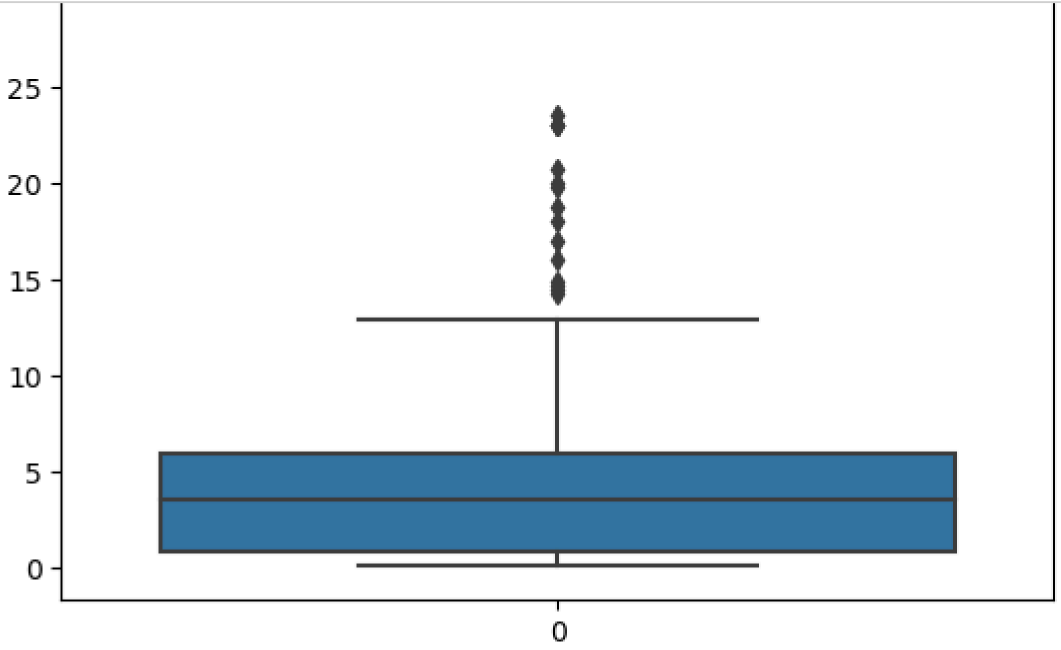
```
Out[33]:
       Car_Name  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  Seller_Type  Transmission  Owner  Age
  0        ritz           3.35           5.59       27000     Petrol       Dealer        Manual      0   10
  1         sx4           4.75           9.54       43000     Diesel       Dealer        Manual      0   11
  2        ciaz           7.25           9.85        6900     Petrol       Dealer        Manual      0    7
  3     wagon r           2.85           4.15        5200     Petrol       Dealer        Manual      0   13
  4       swift           4.60           6.87       42450     Diesel       Dealer        Manual      0   10
...         ...            ...            ...         ...        ...          ...           ...    ...  ...
296        city           9.50          11.60       33988     Diesel       Dealer        Manual      0    8
297        brio           4.00           5.90       60000     Petrol       Dealer        Manual      0    9
298        city           3.35          11.00       87934     Petrol       Dealer        Manual      0   15
299        city          11.50          12.50        9000     Diesel       Dealer        Manual      0    7
300        brio           5.30           5.90        5464     Petrol       Dealer        Manual      0    8

301 rows × 9 columns
```

In [34]:

```
import seaborn as sns
```

In [35]:

```
sns.boxplot(data["Selling_Price"])
```



In [39]:

```
data[["Selling_Price"]]
```

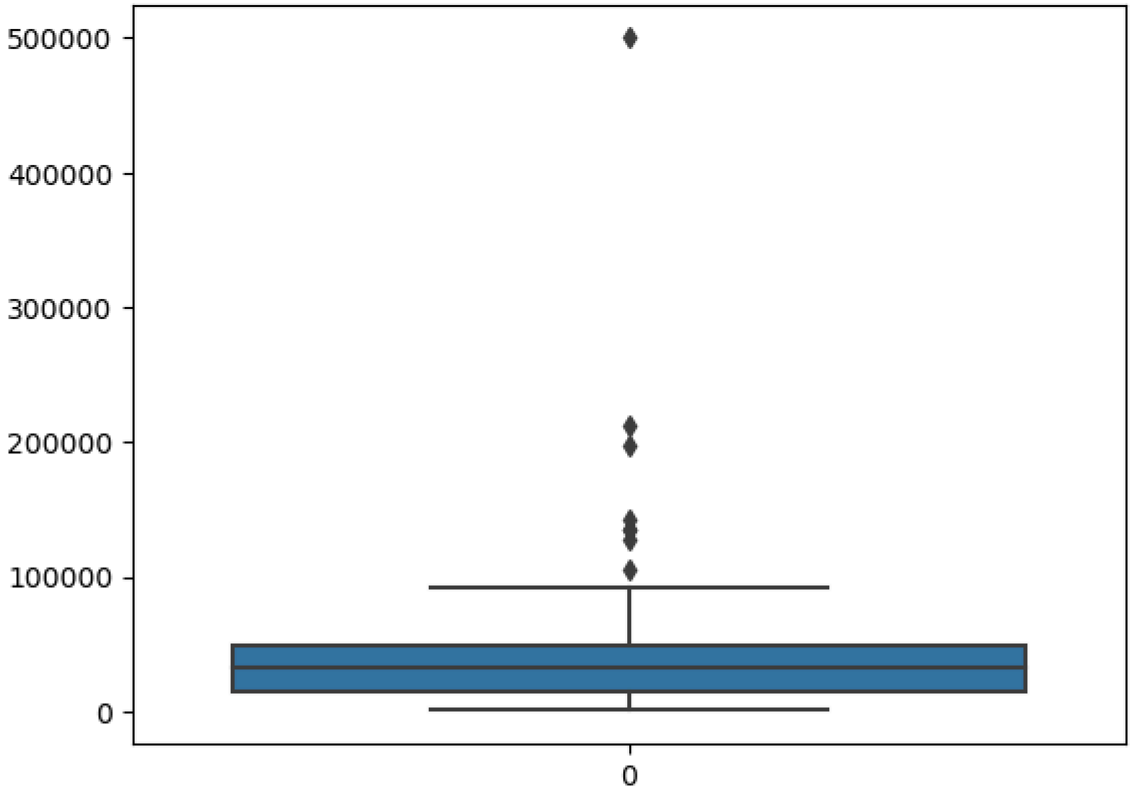
Out[39]:

	Selling_Price
0	3.35
1	4.75
2	7.25
3	2.85
4	4.60
...	...
296	9.50
297	4.00
298	3.35
299	11.50
300	5.30

301 rows × 1 columns

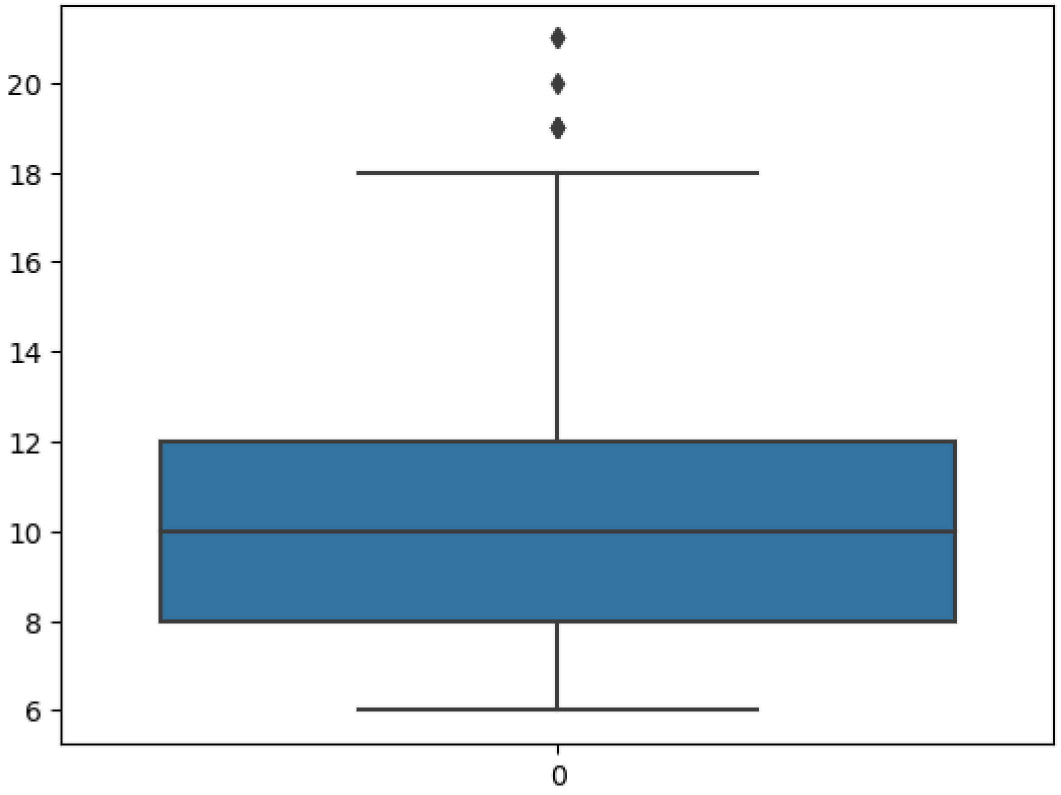
```
In [36]: sns.boxplot(data["Kms_Driven"])
```

```
Out[36]: <Axes: >
```



```
In [37]: sns.boxplot(data["Age"])
```

```
Out[37]: <Axes: >
```



```
In [40]: sorted(data["Selling_Price"],reverse=True)
```

```
Out[40]: [35.0,
          33.0,
          23.5,
          23.0,
          23.0,
          23.0,
          20.75,
          19.99,
          19.75,
          18.75,
          18.0,
          17.0,
          16.0,
          14.9,
          14.73,
          14.5,
          14.25,
          12.9,
          12.5,
```

```
In [43]: data=data[~(data["Selling_Price"]>=33.0) & (data["Selling_Price"]<=35.0)]
```

In [44]:
```
data
```

Out[44]:

	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	ritz	3.35	5.59	27000	Petrol	Dealer	Manual	0	10
1	sx4	4.75	9.54	43000	Diesel	Dealer	Manual	0	11
2	ciaz	7.25	9.85	6900	Petrol	Dealer	Manual	0	7
3	wagon r	2.85	4.15	5200	Petrol	Dealer	Manual	0	13
4	swift	4.60	6.87	42450	Diesel	Dealer	Manual	0	10
...	...	...	...	...	...	...	...	...	...
296	city	9.50	11.60	33988	Diesel	Dealer	Manual	0	8
297	brio	4.00	5.90	60000	Petrol	Dealer	Manual	0	9
298	city	3.35	11.00	87934	Petrol	Dealer	Manual	0	15
299	city	11.50	12.50	9000	Diesel	Dealer	Manual	0	7
300	brio	5.30	5.90	5464	Petrol	Dealer	Manual	0	8

299 rows × 9 columns

In [45]:
```
#Encoding the categorical Columns
```

In [47]:
```
data.head(1)
```

Out[47]:

	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	ritz	3.35	5.59	27000	Petrol	Dealer	Manual	0	10

In [49]:
```
data["Fuel_Type"].unique()
```

Out[49]:
```
array(['Petrol', 'Diesel', 'CNG'], dtype=object)
```

In [50]:
```
data["Fuel_Type"]=data["Fuel_Type"].map({"Petrol":0,"Diesel":1,"CNG":2})
```

In [51]:
```
data
```

Out[51]:

	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	ritz	3.35	5.59	27000	0	Dealer	Manual	0	10
1	sx4	4.75	9.54	43000	1	Dealer	Manual	0	11
2	ciaz	7.25	9.85	6900	0	Dealer	Manual	0	7
3	wagon r	2.85	4.15	5200	0	Dealer	Manual	0	13
4	swift	4.60	6.87	42450	1	Dealer	Manual	0	10
...	...	...	...	...	...	...	...	...	...
296	city	9.50	11.60	33988	1	Dealer	Manual	0	8
297	brio	4.00	5.90	60000	0	Dealer	Manual	0	9
298	city	3.35	11.00	87934	0	Dealer	Manual	0	15
299	city	11.50	12.50	9000	1	Dealer	Manual	0	7
300	brio	5.30	5.90	5464	0	Dealer	Manual	0	8

299 rows × 9 columns

In [52]:
```
data["Fuel_Type"].unique()
```

Out[52]:
```
array([0, 1, 2], dtype=int64)
```

In [53]:
```
data["Seller_Type"].unique()
```

Out[53]:
```
array(['Dealer', 'Individual'], dtype=object)
```

```
In [54]: data["Seller_Type"]=data["Seller_Type"].map({"Dealer":0,"Individual":1})
```

```
In [55]: data
```

```
Out[55]:
          Car_Name  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  Seller_Type  Transmission  Owner  Age
      0       ritz           3.35           5.59       27000          0            0        Manual      0   10
      1        sx4           4.75           9.54       43000          1            0        Manual      0   11
      2       ciaz           7.25           9.85        6900          0            0        Manual      0    7
      3    wagon r           2.85           4.15        5200          0            0        Manual      0   13
      4      swift           4.60           6.87       42450          1            0        Manual      0   10
    ...        ...            ...            ...         ...        ...          ...           ...    ...  ...
    296       city           9.50          11.60       33988          1            0        Manual      0    8
    297       brio           4.00           5.90       60000          0            0        Manual      0    9
    298       city           3.35          11.00       87934          0            0        Manual      0   15
    299       city          11.50          12.50        9000          1            0        Manual      0    7
    300       brio           5.30           5.90        5464          0            0        Manual      0    8

299 rows × 9 columns
```

```
In [56]: data["Transmission"].unique()
```

```
Out[56]: array(['Manual', 'Automatic'], dtype=object)
```

```
In [57]: data["Transmission"]=data["Transmission"].map({"Manual":0,"Automatic":1})
In [58]: data
```

```
Out[58]:
     Car_Name  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  Seller_Type  Transmission  Owner  Age
  0      ritz           3.35           5.59       27000          0            0             0      0   10
  1       sx4           4.75           9.54       43000          1            0             0      0   11
  2      ciaz           7.25           9.85        6900          0            0             0      0    7
  3   wagon r           2.85           4.15        5200          0            0             0      0   13
  4     swift           4.60           6.87       42450          1            0             0      0   10
...       ...            ...            ...         ...        ...          ...           ...    ...  ...
296      city           9.50          11.60       33988          1            0             0      0    8
297      brio           4.00           5.90       60000          0            0             0      0    9
298      city           3.35          11.00       87934          0            0             0      0   15
299      city          11.50          12.50        9000          1            0             0      0    7
300      brio           5.30           5.90        5464          0            0             0      0    8

299 rows × 9 columns
```

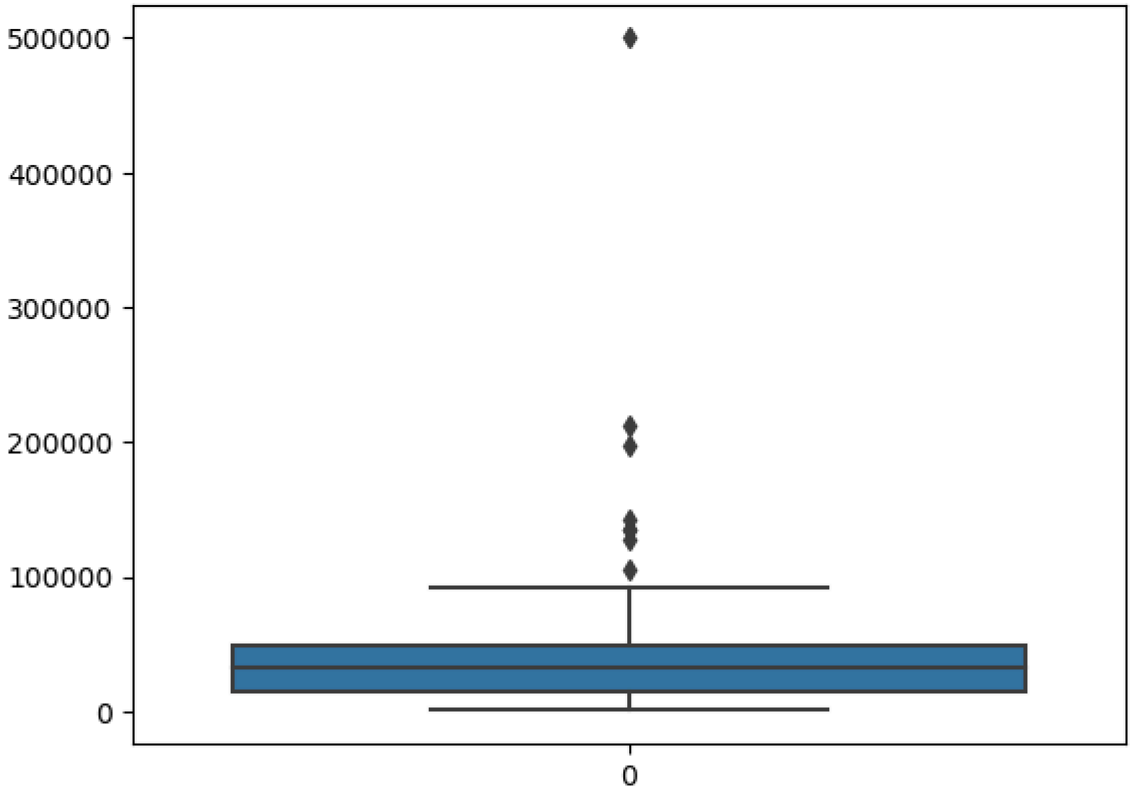
In [59]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Car_Name       299 non-null    object 
 1   Selling_Price  299 non-null    float64
 2   Present_Price  299 non-null    float64
 3   Kms_Driven     299 non-null    int64  
 4   Fuel_Type      299 non-null    int64  
 5   Seller_Type    299 non-null    int64  
 6   Transmission   299 non-null    int64  
 7   Owner          299 non-null    int64  
 8   Age            299 non-null    int64  
dtypes: float64(2), int64(6), object(1)
memory usage: 23.4+ KB
```

In [60]:
```
sns.boxplot(data["Kms_Driven"])
```

Out[60]:
```
<Axes: >
```



In [67]:
```
data=data[~(data["Kms_Driven"]>100000)]
```

In [68]:

```
data
```

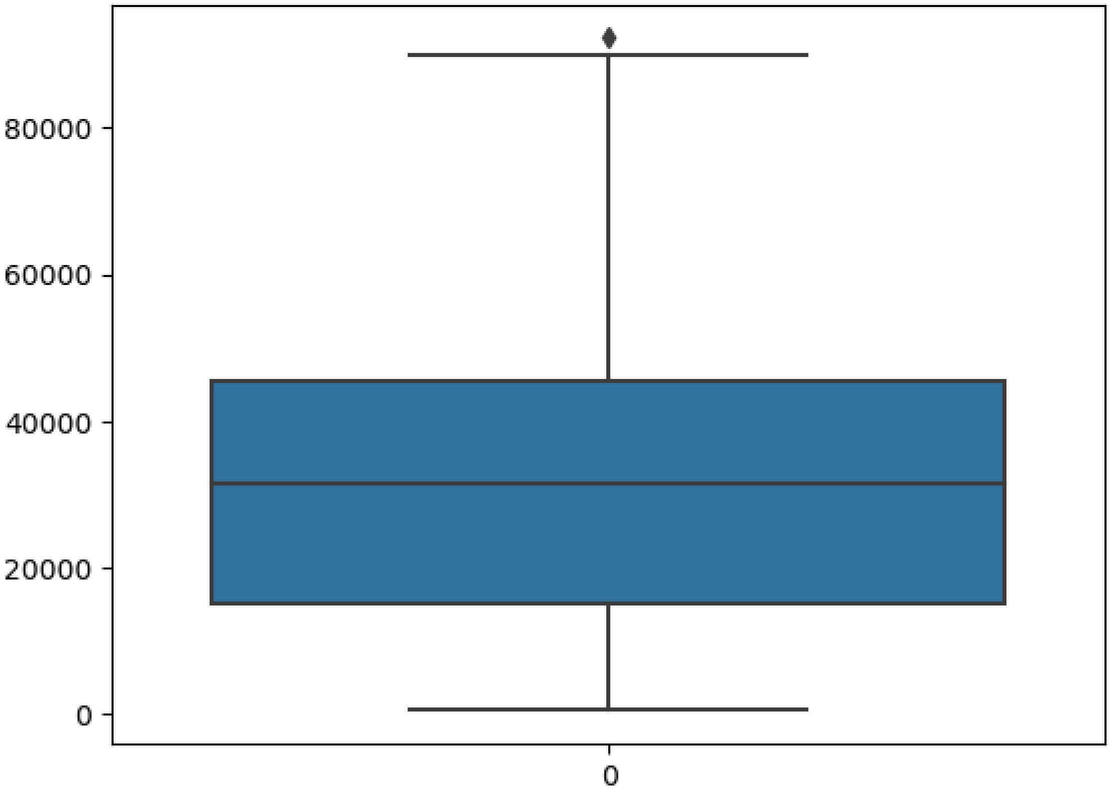
Out[68]:

	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	ritz	3.35	5.59	27000	0	0	0	0	10
1	sx4	4.75	9.54	43000	1	0	0	0	11
2	ciaz	7.25	9.85	6900	0	0	0	0	7
3	wagon r	2.85	4.15	5200	0	0	0	0	13
4	swift	4.60	6.87	42450	1	0	0	0	10
...	...	...	...	...	...	...	...	...	...
296	city	9.50	11.60	33988	1	0	0	0	8
297	brio	4.00	5.90	60000	0	0	0	0	9
298	city	3.35	11.00	87934	0	0	0	0	15
299	city	11.50	12.50	9000	1	0	0	0	7
300	brio	5.30	5.90	5464	0	0	0	0	8

291 rows × 9 columns

```
In [69]: sns.boxplot(data["Kms_Driven"])
```

```
Out[69]: <Axes: >
```



```
In [70]: X=data.drop(["Car_Name","Selling_Price"],axis=1)
```

```
In [71]: y=data["Selling_Price"]
```

In [72]:
```
X
```

Out[72]:

	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	5.59	27000	0	0	0	0	10
1	9.54	43000	1	0	0	0	11
2	9.85	6900	0	0	0	0	7
3	4.15	5200	0	0	0	0	13
4	6.87	42450	1	0	0	0	10
...	...	...	...	...	...	...	...
296	11.60	33988	1	0	0	0	8
297	5.90	60000	0	0	0	0	9
298	11.00	87934	0	0	0	0	15
299	12.50	9000	1	0	0	0	7
300	5.90	5464	0	0	0	0	8

291 rows × 7 columns

In [74]:
```
y
```

Out[74]:
```
0        3.35
1        4.75
2        7.25
3        2.85
4        4.60
       ...  
296      9.50
297      4.00
298      3.35
299     11.50
300      5.30
Name: Selling_Price, Length: 291, dtype: float64
```

In [75]:

```
from sklearn.model_selection import train_test_split
```

In [76]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,random_state=42)
```

In [77]:

```
data.head()
```

Out[77]:

	Car_Name	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Age
0	ritz	3.35	5.59	27000	0	0	0	0	10
1	sx4	4.75	9.54	43000	1	0	0	0	11
2	ciaz	7.25	9.85	6900	0	0	0	0	7
3	wagon r	2.85	4.15	5200	0	0	0	0	13
4	swift	4.60	6.87	42450	1	0	0	0	10

In [80]:

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor
```

In [81]:

```
lr=LinearRegression()
lr.fit(X_train,y_train)
```

Out[81]:

```
▾ LinearRegression ⓘ ?
(https://scikit-learn.org/1.4/modules/generated/sklearn.linear_model.LinearRegression.html)

LinearRegression()
```

In [83]:
```
rf=RandomForestRegressor()
rf.fit(X_train,y_train)
```

Out[83]:
```
▾ RandomForestRegressor i ?
(https://scikit-learn.org/1.4/modules/generated/sklearn.ensemble.RandomForestRegressor.html)

RandomForestRegressor()
```

In [84]:
```
xg=XGBRegressor()
xg.fit(X_train,y_train)
```

Out[84]:
```
▾                         XGBRegressor                                  i

             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

In [86]:
```
xgb=GradientBoostingRegressor()
xgb.fit(X_train,y_train)
```

Out[86]:
```
▾ GradientBoostingRegressor i ?
(https://scikit-learn.org/1.4/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html)

GradientBoostingRegressor()
```

In [87]:

```
Y_pred1=lr.predict(X_test)
Y_pred2=rf.predict(X_test)
Y_pred3=xg.predict(X_test)
Y_pred4=xgb.predict(X_test)
```

In [89]:

```
from sklearn import metrics
```

In [90]:

```
score1=metrics.r2_score(y_test,Y_pred1)
score2=metrics.r2_score(y_test,Y_pred2)
score3=metrics.r2_score(y_test,Y_pred3)
score4=metrics.r2_score(y_test,Y_pred4)
```

In [91]:

```
print(score1,score2,score3,score4)
```

```
0.9305603810622383 0.9716419188476204 0.9092836108103018 0.979725911421164
```

In [92]:

```
xgb=GradientBoostingRegressor()
xgb_final=xgb.fit(X,y)
```

In [93]:

```
import joblib
```

In [94]:

```
joblib.dump(xgb_final,"Car_Price_Predictor")
```

Out[94]:

```
['Car_Price_Predictor']
```

In [95]:

```
model=joblib.load("Car_Price_Predictor")
```

In [96]:

```
xgb_final.save_model("xgb_model.json")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[96], line 1
----> 1 xgb_final.save_model("xgb_model.json")

AttributeError: 'GradientBoostingRegressor' object has no attribute 'save_model'
```

In [97]:

```
import pickle


model_filename = "xgb_model.pkl"
with open(model_filename, "wb") as file:
    pickle.dump(xgb_final, file)

print(f"Model saved as {model_filename}")
```

```
Model saved as xgb_model.pkl
```