

xmrcuvftv

December 5, 2023

1 Importing Libraries

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

2 Importing Dataset

```
[2]: data = pd.read_csv(r'C:\Users\Shantanu\Desktop\PRO\Car Price Prediction_Quikr\car.csv')
```

```
[3]: data.head()
```

```
[3]:   Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014           3.35           5.59       27000    Petrol
1    sx4    2013           4.75           9.54       43000    Diesel
2    ciaz    2017           7.25           9.85        6900    Petrol
3  wagon r    2011           2.85           4.15        5200    Petrol
4   swift    2014           4.60           6.87      42450    Diesel

   Seller_Type  Transmission  Owner
0     Dealer           Manual      0
1     Dealer           Manual      0
2     Dealer           Manual      0
3     Dealer           Manual      0
4     Dealer           Manual      0
```

3 Data Wrangling

```
[4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 903 entries, 0 to 902
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	Car_Name	903 non-null	object
1	Year	903 non-null	int64
2	Selling_Price	903 non-null	float64
3	Present_Price	903 non-null	float64
4	Kms_Driven	903 non-null	int64
5	Fuel_Type	903 non-null	object
6	Seller_Type	903 non-null	object
7	Transmission	903 non-null	object
8	Owner	903 non-null	int64

dtypes: float64(2), int64(3), object(4)
memory usage: 63.6+ KB

```
[5]: data.dtypes
```

```
[5]: Car_Name      object
      Year          int64
      Selling_Price float64
      Present_Price float64
      Kms_Driven    int64
      Fuel_Type     object
      Seller_Type   object
      Transmission  object
      Owner         int64
      dtype: object
```

```
[6]: data.isnull().sum()
```

```
[6]: 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
```

```
[7]: data.isnull().sum().sum()
```

```
[7]: 0
```

```
[8]: data = data.drop(['Car_Name'], axis=1)
```

```

[9]: data['current_year'] = 2020
    data['no_year'] = data['current_year'] - data['Year']

[10]: data = data.drop(['Year', 'current_year'], axis = 1)

[11]: data = pd.get_dummies(data, drop_first=True)

[12]: data =
↳ data[['Selling_Price', 'Present_Price', 'Kms_Driven', 'no_year', 'Owner', 'Fuel_Type_Diesel', 'Fu

[13]: data

```

```

[13]:
    Selling_Price  Present_Price  Kms_Driven  no_year  Owner  \
0              3.35           5.59       27000         6      0
1              4.75           9.54       43000         7      0
2              7.25           9.85        6900         3      0
3              2.85           4.15        5200         9      0
4              4.60           6.87      42450         6      0
..            ...           ...           ...       ...      ...
898             9.50          11.60       33988         4      0
899             4.00           5.90      60000         5      0
900             3.35          11.00      87934        11      0
901            11.50          12.50        9000         3      0
902             5.30           5.90       5464         4      0

    Fuel_Type_Diesel  Fuel_Type_Petrol  Seller_Type_Individual  \
0                  0                  1                      0
1                  1                  0                      0
2                  0                  1                      0
3                  0                  1                      0
4                  1                  0                      0
..            ...           ...           ...
898                 1                  0                      0
899                 0                  1                      0
900                 0                  1                      0
901                 1                  0                      0
902                 0                  1                      0

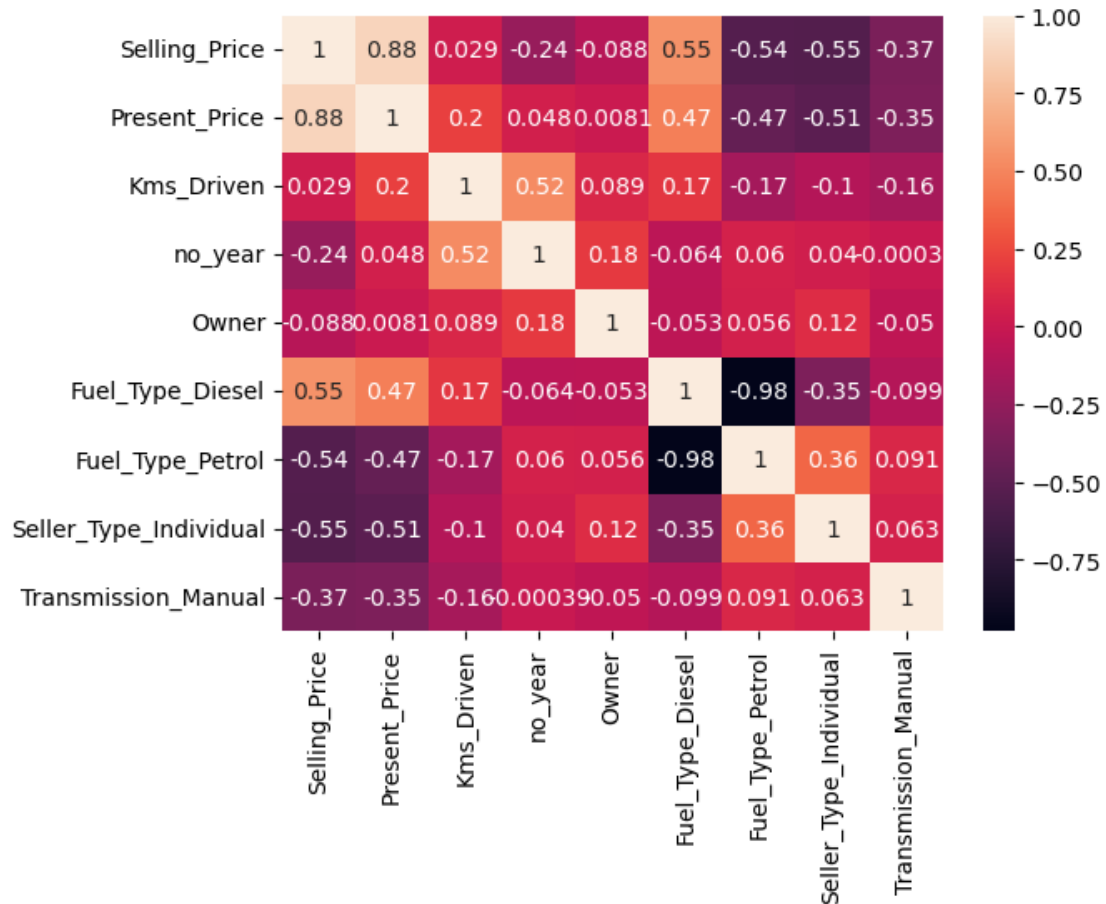
    Transmission_Manual
0                      1
1                      1
2                      1
3                      1
4                      1
..            ...
898                 1
899                 1

```

```
900          1
901          1
902          1
```

[903 rows x 9 columns]

```
[14]: sns.heatmap(data.corr(),annot=True)
plt.show()
```

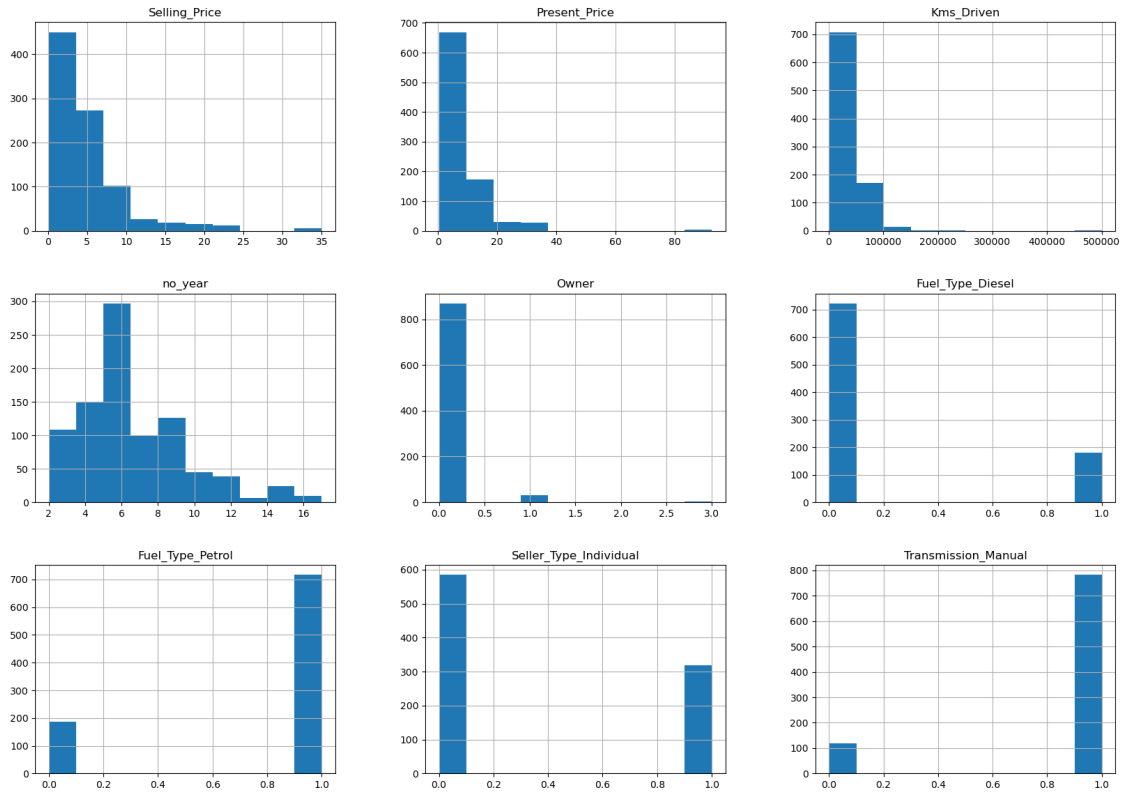


```
[15]: shape=data.shape
print('Number of Rows : ',shape[0])
print('Number of Columns : ',shape[1])
```

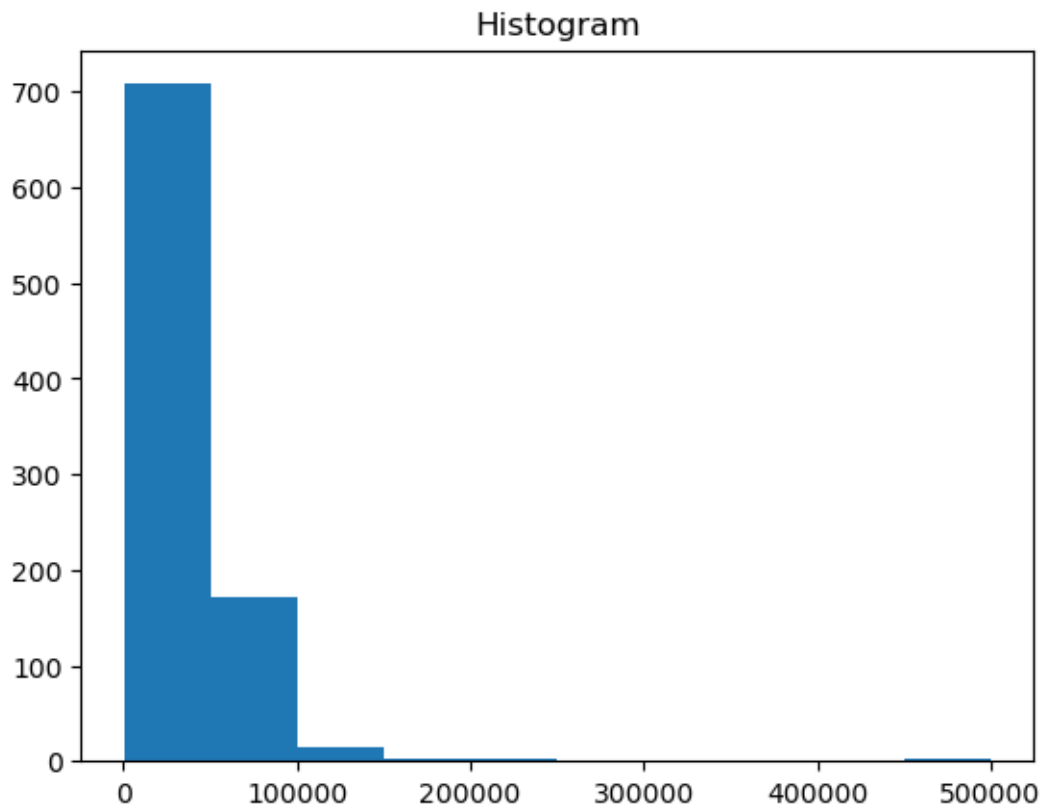
```
Number of Rows : 903
Number of Columns : 9
```

4 Data Analysis

```
[16]: data.hist(figsize=(20,14))  
plt.show()
```

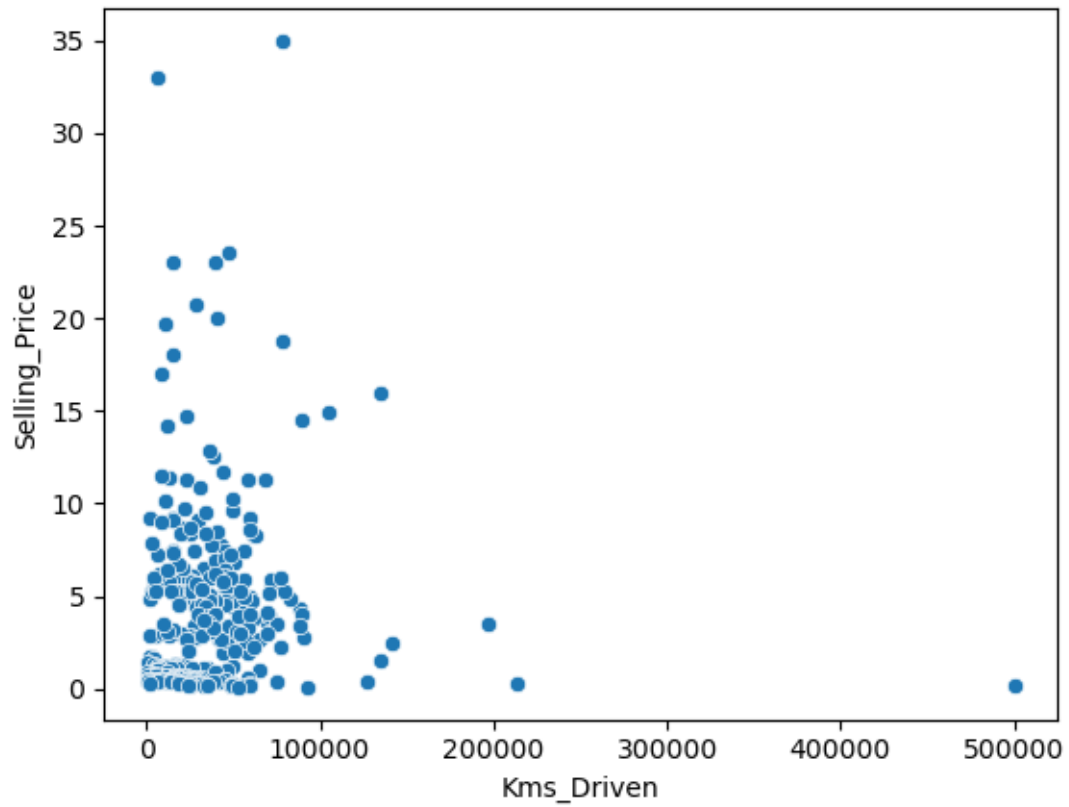


```
[17]: plt.hist(data['Kms_Driven'])  
  
plt.title("Histogram")  
  
plt.show()
```



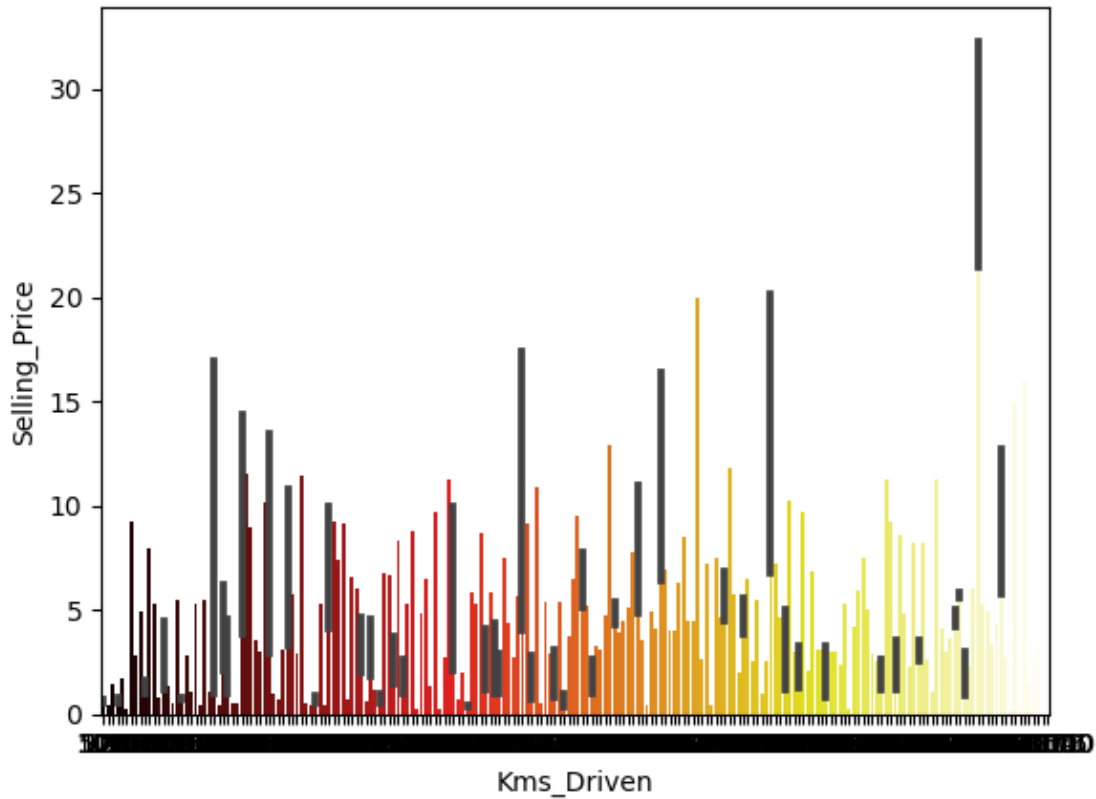
```
[18]: sns.scatterplot(x = data['Kms_Driven'] , y = data['Selling_Price'])
```

```
[18]: <Axes: xlabel='Kms_Driven', ylabel='Selling_Price'>
```



```
[19]: sns.barplot(x = data['Kms_Driven'] , y = data['Selling_Price'], palette =_
      ↪ 'hot')
```

```
[19]: <Axes: xlabel='Kms_Driven', ylabel='Selling_Price'>
```



```
[20]: import plotly.express as px
import pandas as pd

# plotting the scatter chart
fig = px.histogram(data, x='Selling_Price', color='Kms_Driven')

# showing the plot
fig.show()
```

```
[21]: data.corr()
```

```
[21]:
```

	Selling_Price	Present_Price	Kms_Driven	no_year	\
Selling_Price	1.000000	0.878983	0.029187	-0.236141	
Present_Price	0.878983	1.000000	0.203647	0.047584	
Kms_Driven	0.029187	0.203647	1.000000	0.524342	
no_year	-0.236141	0.047584	0.524342	1.000000	
Owner	-0.088344	0.008057	0.089216	0.182104	
Fuel_Type_Diesel	0.552339	0.473306	0.172515	-0.064315	
Fuel_Type_Petrol	-0.540571	-0.465244	-0.172874	0.059959	
Seller_Type_Individual	-0.550724	-0.512030	-0.101419	0.039896	
Transmission_Manual	-0.367128	-0.348715	-0.162510	-0.000394	

	Owner	Fuel_Type_Diesel	Fuel_Type_Petrol \
Selling_Price	-0.088344	0.552339	-0.540571
Present_Price	0.008057	0.473306	-0.465244
Kms_Driven	0.089216	0.172515	-0.172874
no_year	0.182104	-0.064315	0.059959
Owner	1.000000	-0.053469	0.055687
Fuel_Type_Diesel	-0.053469	1.000000	-0.979648
Fuel_Type_Petrol	0.055687	-0.979648	1.000000
Seller_Type_Individual	0.124269	-0.350467	0.358321
Transmission_Manual	-0.050316	-0.098643	0.091013

	Seller_Type_Individual	Transmission_Manual
Selling_Price	-0.550724	-0.367128
Present_Price	-0.512030	-0.348715
Kms_Driven	-0.101419	-0.162510
no_year	0.039896	-0.000394
Owner	0.124269	-0.050316
Fuel_Type_Diesel	-0.350467	-0.098643
Fuel_Type_Petrol	0.358321	0.091013
Seller_Type_Individual	1.000000	0.063240
Transmission_Manual	0.063240	1.000000

```
[22]: from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df = scaler.fit_transform(data[['Kms_Driven']])
```

5 Training and Testing Data

```
[23]: x = data.iloc[:,1:]
y = data.iloc[:,0]
```

```
[24]: from sklearn.ensemble import ExtraTreesRegressor
model = ExtraTreesRegressor()
model.fit(x,y)
```

```
[24]: ExtraTreesRegressor()
```

```
[25]: model.feature_importances_
```

```
[25]: array([4.02387438e-01, 3.70722363e-02, 7.83679890e-02, 3.70983498e-04,
2.14518971e-01, 1.30737989e-02, 1.18732102e-01, 1.35476481e-01])
```

6 Splitting Train and Test data

```
[26]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = \
    train_test_split(x, y, random_state=0, test_size=0.2)
```

```
[27]: print(len(x_train))
print(len(x_test))
print(len(y_train))
print(len(y_test))
```

```
722
181
722
181
```

7 Different Algorithms and Its Evaluation

8 1.Random Forest Regressor

```
[28]: n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
min_samples_split = [2, 5, 10, 15, 100]
min_samples_leaf = [1, 2, 5, 10]
```

```
[29]: from sklearn.model_selection import RandomizedSearchCV
```

```
[30]: grid = {'n_estimators': n_estimators,
            'max_features': max_features,
            'max_depth': max_depth,
            'min_samples_split': min_samples_split,
            'min_samples_leaf': min_samples_leaf}
print(grid)
```

```
{'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100,
1200], 'max_features': ['auto', 'sqrt'], 'max_depth': [5, 10, 15, 20, 25, 30],
'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_leaf': [1, 2, 5, 10]}
```

```
[31]: from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()

hyp = RandomizedSearchCV(estimator = model, param_distributions=grid,
n_iter=10,
scoring= 'neg_mean_squared_error',
cv=5, verbose = 2, random_state = 42, n_jobs = 1)
```

```
[32]: hyp.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5,
min_samples_split=5, n_estimators=900; total time= 1.3s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5,
min_samples_split=5, n_estimators=900; total time= 1.1s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5,
min_samples_split=5, n_estimators=900; total time= 1.2s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5,
min_samples_split=5, n_estimators=900; total time= 1.2s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5,
min_samples_split=5, n_estimators=900; total time= 1.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=1100; total time= 1.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=1100; total time= 2.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=1100; total time= 2.1s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=1100; total time= 1.7s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=1100; total time= 1.7s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this
parameter as it is also the default value for RandomForestRegressors and
ExtraTreesRegressors.
```

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=100, n_estimators=300; total time= 0.7s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features=1.0` or remove this
parameter as it is also the default value for RandomForestRegressors and
ExtraTreesRegressors.
```

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=100, n_estimators=300; total time= 0.6s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:
FutureWarning:
```

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=100, n_estimators=300; total time= 0.6s

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble_forest.py:413:
FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=100, n_estimators=300; total time= 0.5s

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble_forest.py:413:
FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=100, n_estimators=300; total time= 0.4s

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble_forest.py:413:
FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,
min_samples_split=5, n_estimators=400; total time= 0.8s

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble_forest.py:413:
FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and

ExtraTreesRegressors.

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,  
min_samples_split=5, n_estimators=400; total time= 0.6s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,  
min_samples_split=5, n_estimators=400; total time= 0.7s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,  
min_samples_split=5, n_estimators=400; total time= 0.6s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5,  
min_samples_split=5, n_estimators=400; total time= 0.7s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,  
min_samples_split=5, n_estimators=700; total time= 1.1s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,  
min_samples_split=5, n_estimators=700; total time= 1.0s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,  
min_samples_split=5, n_estimators=700; total time= 1.6s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,  
min_samples_split=5, n_estimators=700; total time= 1.2s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10,  
min_samples_split=5, n_estimators=700; total time= 1.1s
```

```
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1,  
min_samples_split=2, n_estimators=1000; total time= 1.9s
```

```

[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1,
min_samples_split=2, n_estimators=1000; total time= 1.6s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1,
min_samples_split=2, n_estimators=1000; total time= 1.7s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1,
min_samples_split=2, n_estimators=1000; total time= 1.7s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1,
min_samples_split=2, n_estimators=1000; total time= 1.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10,
min_samples_split=15, n_estimators=1100; total time= 1.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10,
min_samples_split=15, n_estimators=1100; total time= 1.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10,
min_samples_split=15, n_estimators=1100; total time= 1.3s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10,
min_samples_split=15, n_estimators=1100; total time= 1.4s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10,
min_samples_split=15, n_estimators=1100; total time= 1.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=300; total time= 0.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=300; total time= 0.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=300; total time= 0.4s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=300; total time= 0.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1,
min_samples_split=15, n_estimators=300; total time= 0.3s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=700; total time= 0.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=700; total time= 0.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=700; total time= 0.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=700; total time= 0.9s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2,
min_samples_split=10, n_estimators=700; total time= 1.0s

```

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble_forest.py:413:
FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features=1.0` or remove this parameter as it is also the default value for RandomForestRegressors and ExtraTreesRegressors.

```

[CV] END max_depth=20, max_features=auto, min_samples_leaf=1,

```

```
min_samples_split=15, n_estimators=700; total time= 1.2s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1,  
min_samples_split=15, n_estimators=700; total time= 1.1s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1,  
min_samples_split=15, n_estimators=700; total time= 1.3s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1,  
min_samples_split=15, n_estimators=700; total time= 1.4s
```

```
C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:413:  
FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To  
keep the past behaviour, explicitly set `max_features=1.0` or remove this  
parameter as it is also the default value for RandomForestRegressors and  
ExtraTreesRegressors.
```

```
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1,  
min_samples_split=15, n_estimators=700; total time= 1.2s
```



```
[32]: RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=1,
                        param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                           'max_features': ['auto', 'sqrt'],
                                           'min_samples_leaf': [1, 2, 5, 10],
                                           'min_samples_split': [2, 5, 10, 15,
                                                                100],
                                           'n_estimators': [100, 200, 300, 400,
                                                            500, 600, 700, 800,
                                                            900, 1000, 1100,
                                                            1200]}},
                        random_state=42, scoring='neg_mean_squared_error',
                        verbose=2)
```

```
[33]: y_pred = hyp.predict(x_test)
```

```
[57]: test_mae = mean_absolute_error(y_test , y_pred)
test_mse = mean_squared_error(y_test , y_pred)
test_accuracy = r2_score(y_test , y_pred)
```

```
[59]: print("Mean Absolute Error for test data is:",test_mae)
print("Mean Squared Error for test data is:",test_mse)
print("Accuracy for Test Data:{:.2f}% ".format(test_accuracy*100))
```

Mean Absolute Error for test data is: 0.13211436464088167

Mean Squared Error for test data is: 0.05303169555027679

Accuracy for Test Data:99.78%

```
[56]: a = [[8.20,25000,6,0,0,1,0,1]]
df = pd.DataFrame(a)
hyp.predict(df)
```

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning:

X does not have valid feature names, but RandomForestRegressor was fitted with feature names

```
[56]: array([5.54238])
```

```
[35]: from sklearn.metrics import r2_score
res1= r2_score(y_test,y_pred)*100
print("Accuracy score: ",res1)
```

Accuracy score: 99.77881052867768

9 2.Linear Regression

```
[36]: from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error , mean_squared_error , r2_score
```

```
[37]: classifier = LinearRegression()
```

```
[38]: classifier.fit(x_train, y_train)
```

```
[38]: LinearRegression()
```

```
[39]: y_test_pred = classifier.predict(x_test)
```

```
[40]: test_mae = mean_absolute_error(y_test , y_test_pred)
      test_mse = mean_squared_error(y_test , y_test_pred)
      test_accuracy = r2_score(y_test , y_test_pred)
```

```
[41]: print("Mean Absolute Error for test data is:\n",test_mae)
      print("Mean Squared Error for test data is:\n",test_mse)
      print("Accuracy for Test Data:{:.2f}% ".format(test_accuracy*100))
```

Mean Absolute Error for test data is:

1.2103166187723802

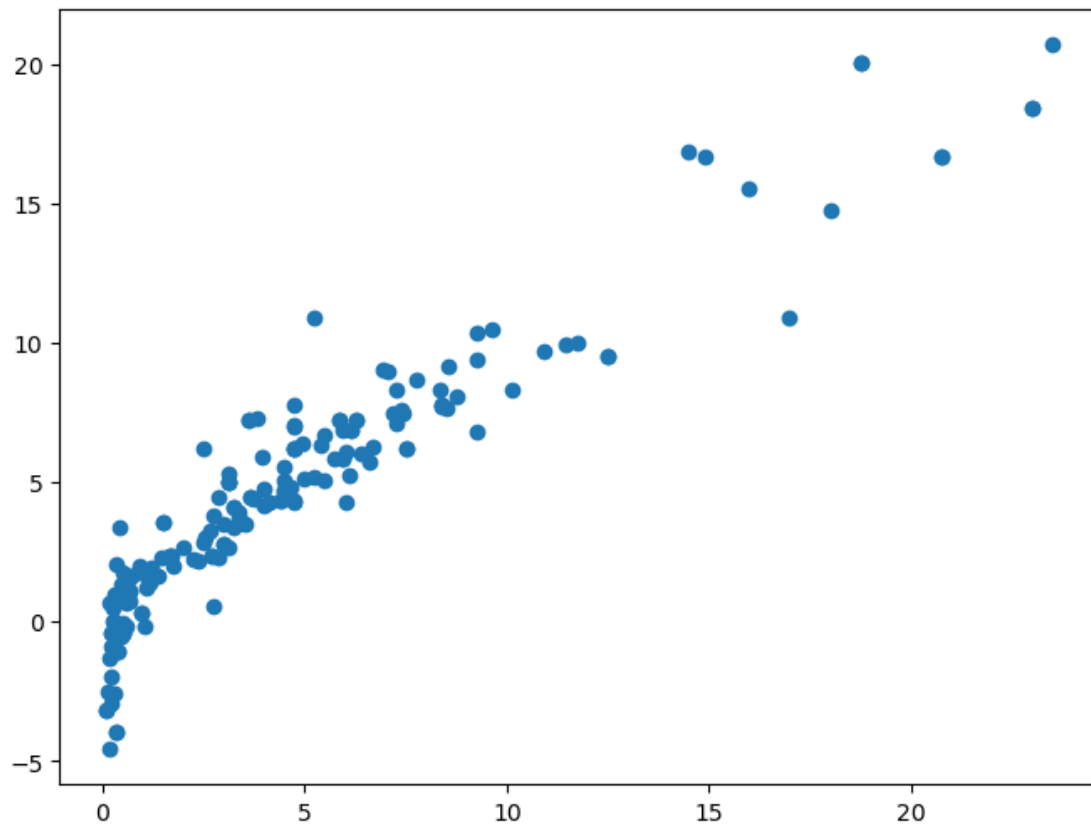
Mean Squared Error for test data is:

2.8741103894889632

Accuracy for Test Data:88.01%

```
[42]: plt.figure (figsize=(8,6))
      plt.scatter(y_test,y_test_pred,)
```

```
[42]: <matplotlib.collections.PathCollection at 0x2b5e0d042e0>
```



```
[43]: data = [[5.59,27000,6,0,0,1,0,1]]
      new_df = pd.DataFrame(data)
      classifier.predict(new_df)
```

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning:

X does not have valid feature names, but LinearRegression was fitted with feature names

```
[43]: array([3.91798344])
```

```
[44]: from sklearn.metrics import r2_score
      res2=r2_score(y_test_pred,y_pred)*100
      print("Accuracy score: ",res2)
```

Accuracy score: 88.1552575133944

10 3.Decision Tree Regressor

```
[45]: from sklearn.tree import DecisionTreeRegressor
dtree = DecisionTreeRegressor(max_depth=6, random_state=1)

dtree.fit(x_train,y_train)
```

```
[45]: DecisionTreeRegressor(max_depth=6, random_state=1)
```

```
[46]: y_test_pred1=dtree.predict(x_test)

test_mae = mean_absolute_error(y_test , y_test_pred)
test_mse = mean_squared_error(y_test , y_test_pred)
test_accuracy = r2_score(y_test , y_test_pred)
```

```
[47]: print("Mean Absolute Error for test data is:\n",test_mae)
print("Mean Squared Error for test data is:\n",test_mse)
print("Accuracy for Test Data:{:.2f}% ".format(test_accuracy*100))
```

Mean Absolute Error for test data is:

1.2103166187723802

Mean Squared Error for test data is:

2.8741103894889632

Accuracy for Test Data:88.01%

```
[48]: data = [[5.59,27000,6,0,0,1,0,1]]
new_df = pd.DataFrame(data)
classifier.predict(new_df)
```

C:\Users\Shantanu\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning:

X does not have valid feature names, but LinearRegression was fitted with
feature names

```
[48]: array([3.91798344])
```

```
[49]: from sklearn.metrics import r2_score
res3=r2_score(y_test_pred1,y_pred)*100
print("Accuracy score: ",res3)
```

Accuracy score: 97.45344639019265

11 Conclusion

```
[50]: print("1. Random Forest Regressor:",res1)
      print("2. Linear Regression:",res2)
      print("3. Decision Tree:",res3)
```

```
1. Random Forest Regressor: 99.77881052867768
2. Linear Regression: 88.1552575133944
3. Decision Tree: 97.45344639019265
```

```
[51]: import matplotlib.pyplot as plt

      algo = ['Random Forest', 'Linear Regression', 'Decision Tree']
      accuracy = [res1, res2, res3]

      res11=(round(res1, 2))
      res21=(round(res2, 2))
      res31=(round(res3, 2))

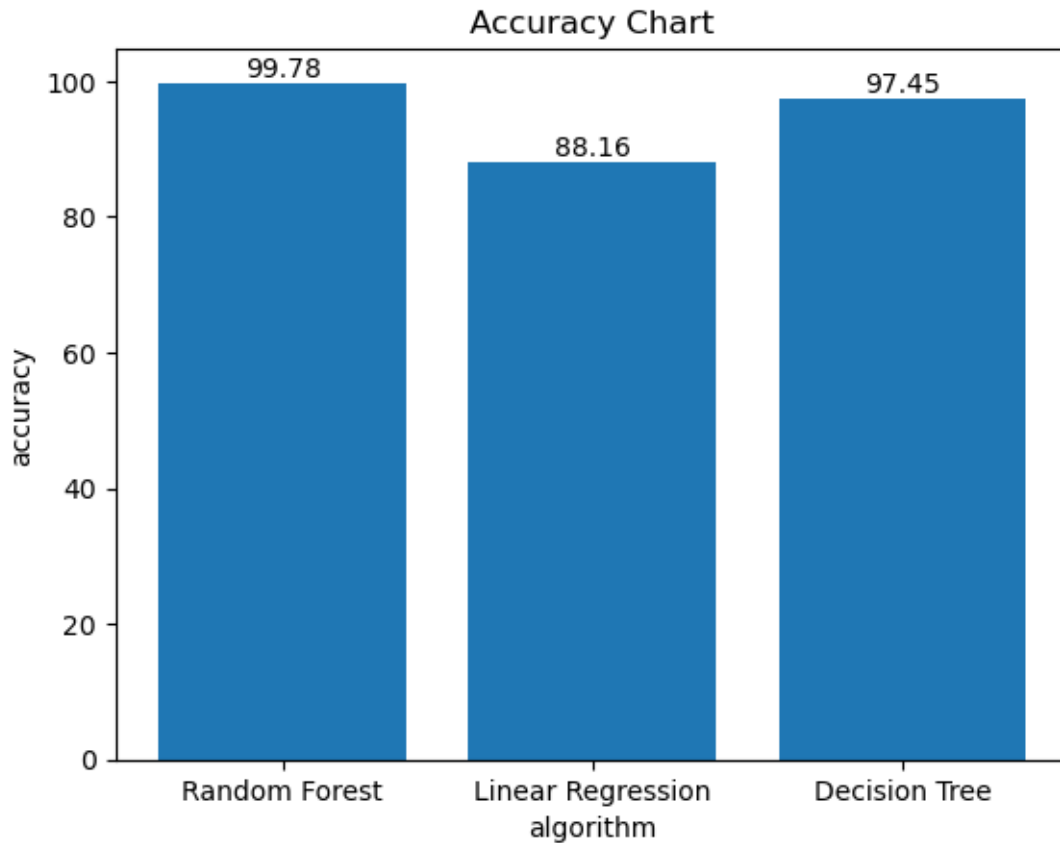
      accuracy1 = [res11, res21, res31]

      # Create a bar chart
      plt.bar(algo, accuracy)

      for i, v in enumerate(accuracy1):
          plt.text(i, v, str(v), ha='center', va='bottom')

      # Add labels and a title
      plt.xlabel('algorithm')
      plt.ylabel('accuracy')
      plt.title('Accuracy Chart')

      # Show the plot
      plt.show()
```



12 Data Visualization

```
[52]: import seaborn as sns
```

```
[53]: sns.distplot(y_test-y_pred)
```

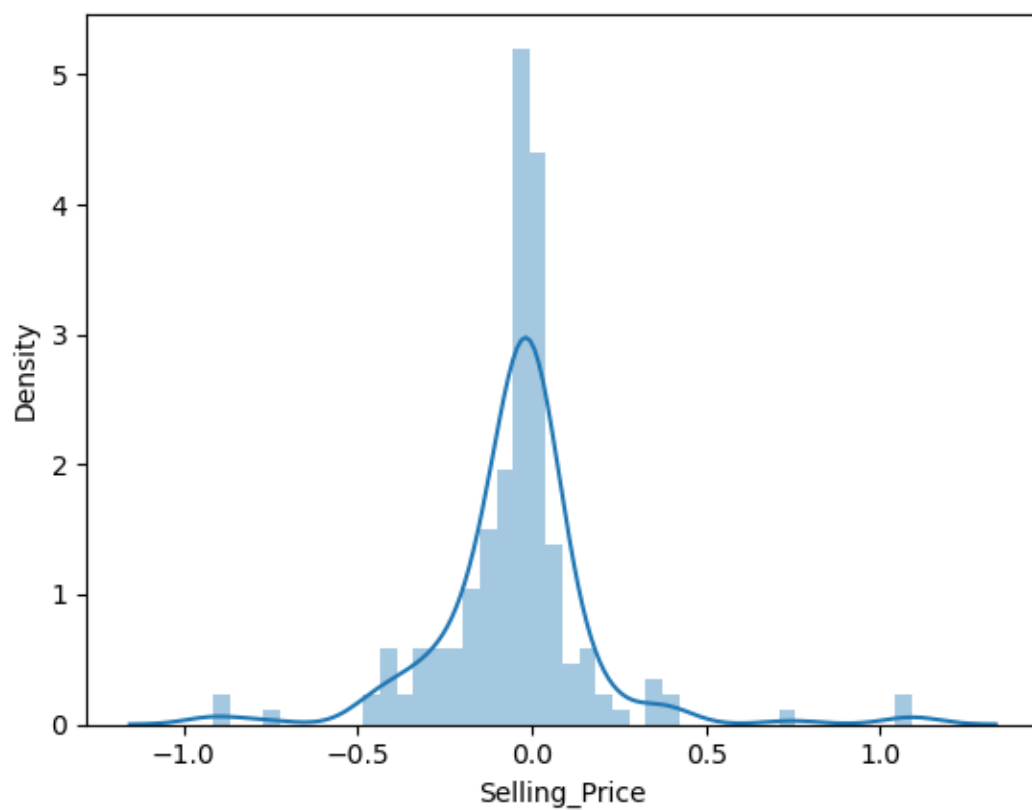
C:\Users\Shantanu\AppData\Local\Temp\ipykernel_16216\520176394.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

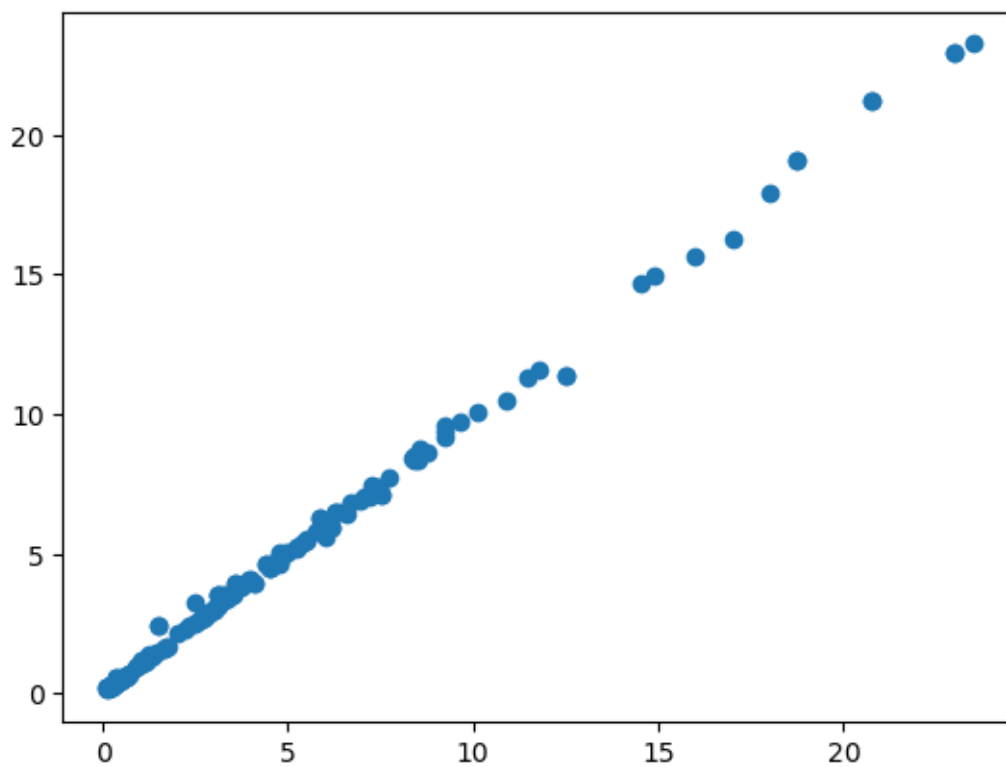
For a guide to updating your code to use the new functions, please see
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

[53]: <Axes: xlabel='Selling_Price', ylabel='Density'>



```
[54]: import matplotlib.pyplot as plt  
plt.scatter(y_test,y_pred)
```

[54]: <matplotlib.collections.PathCollection at 0x2b5e2f43ac0>



```
[55]: import pickle
      file = open('file.pkl','wb')

      pickle.dump(hyp,file)
```

```
[ ]:
```

```
[ ]:
```