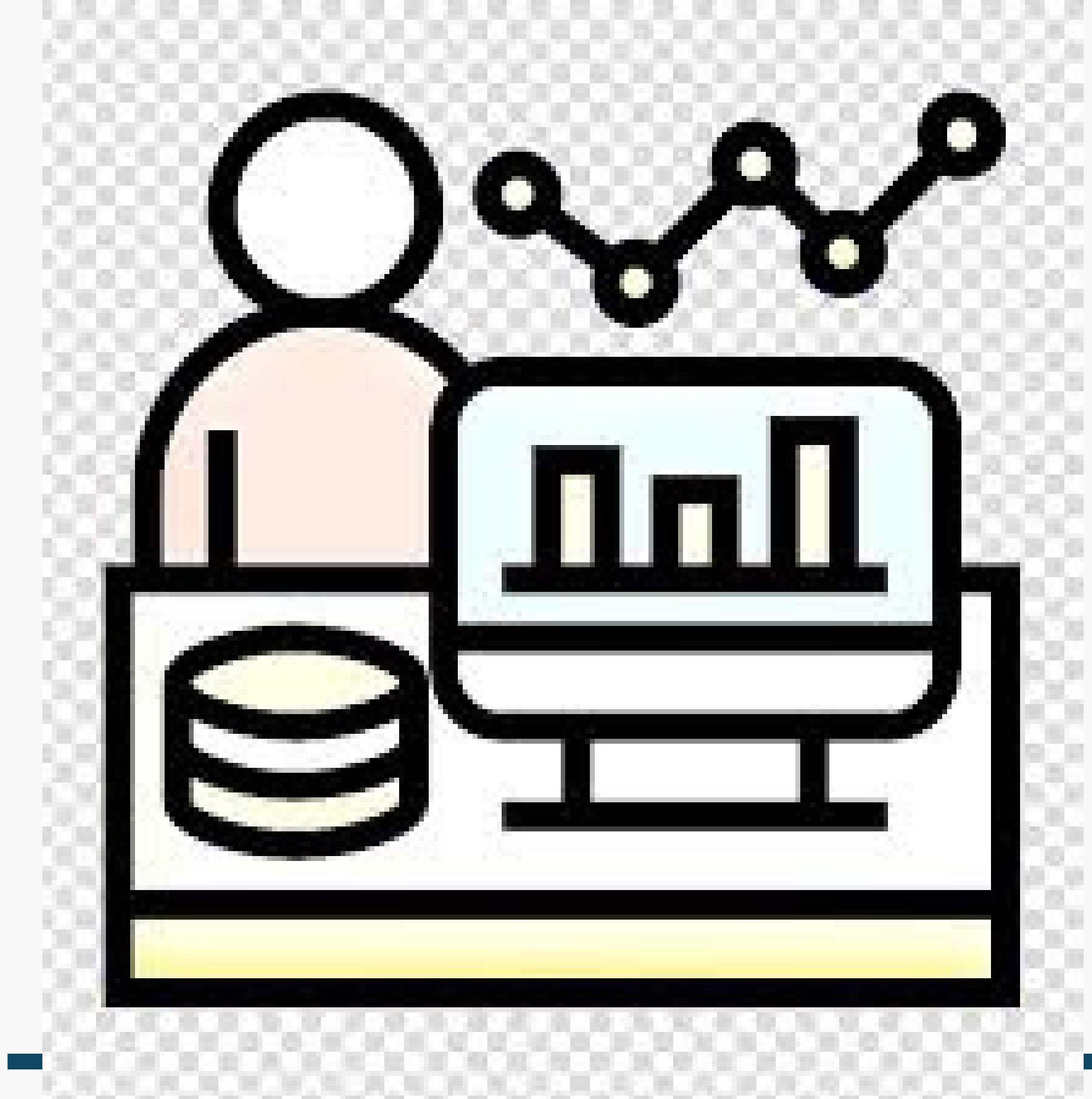
Machine Learning Model

To Car and Heart Datasets



Esakiappan E

20 September, 2024

Machine Learning Algorithm :

Regressor :

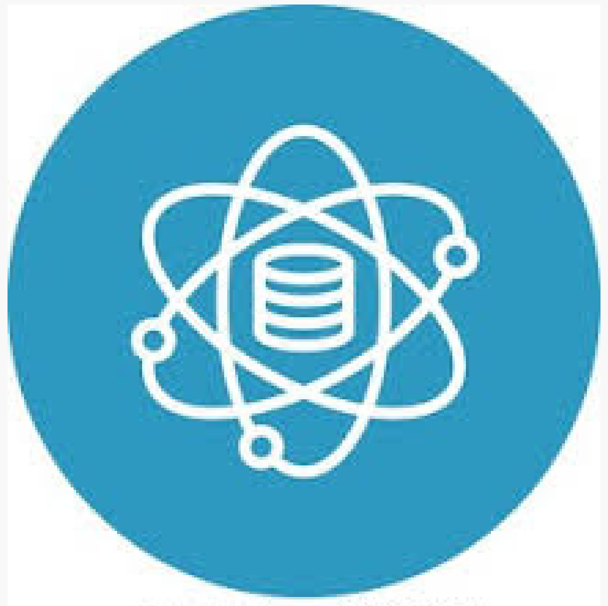
- Linear Regressor
- Ridge Regressor
- Lasso Regressor

Classifier :

- Logistic Classification
- Naive Bayes Classifier

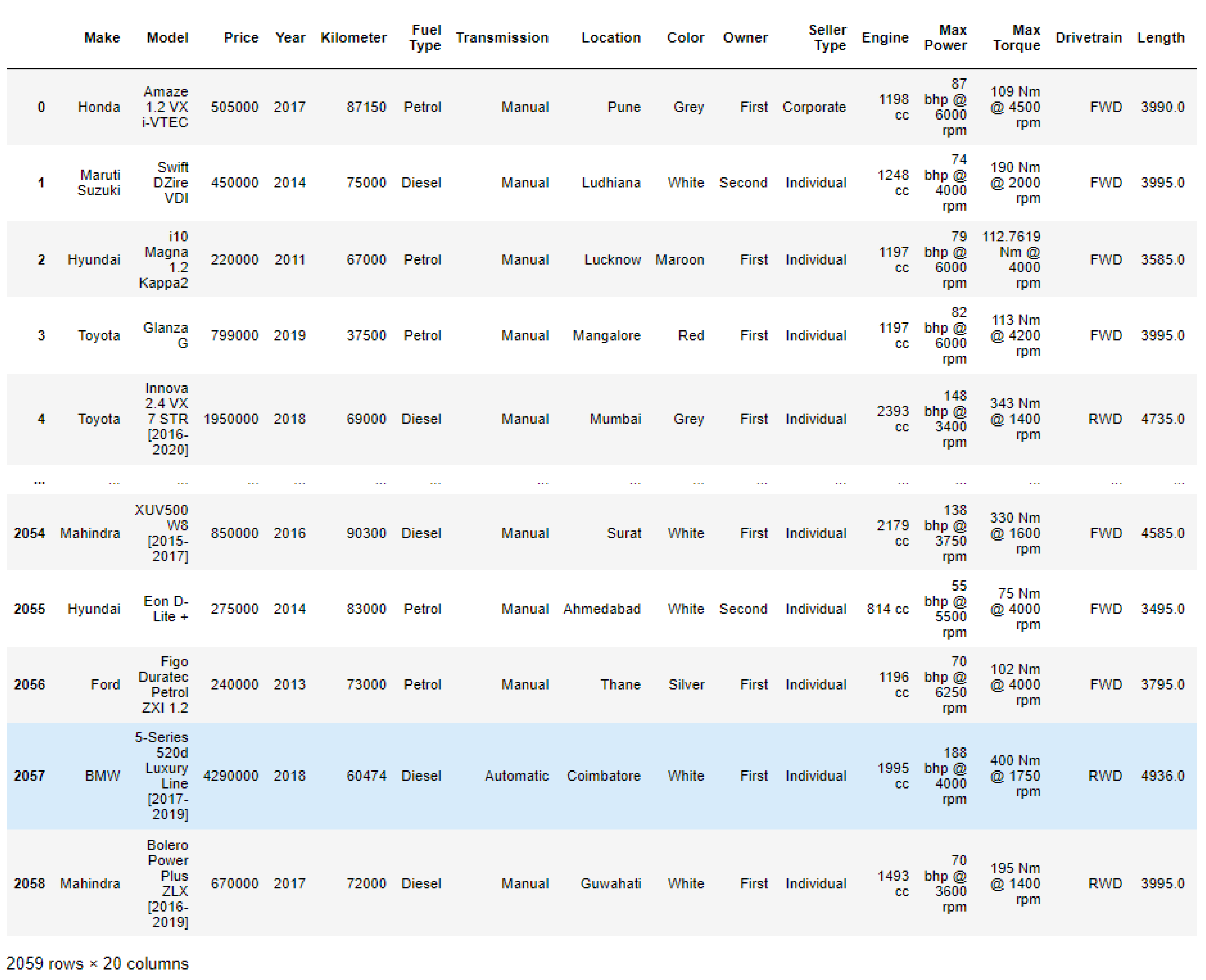
For Both,

- Decision Tree Algorithm
- KNN Algorithm
- Random Forest Algorithm



Regressor(Car Dataset)

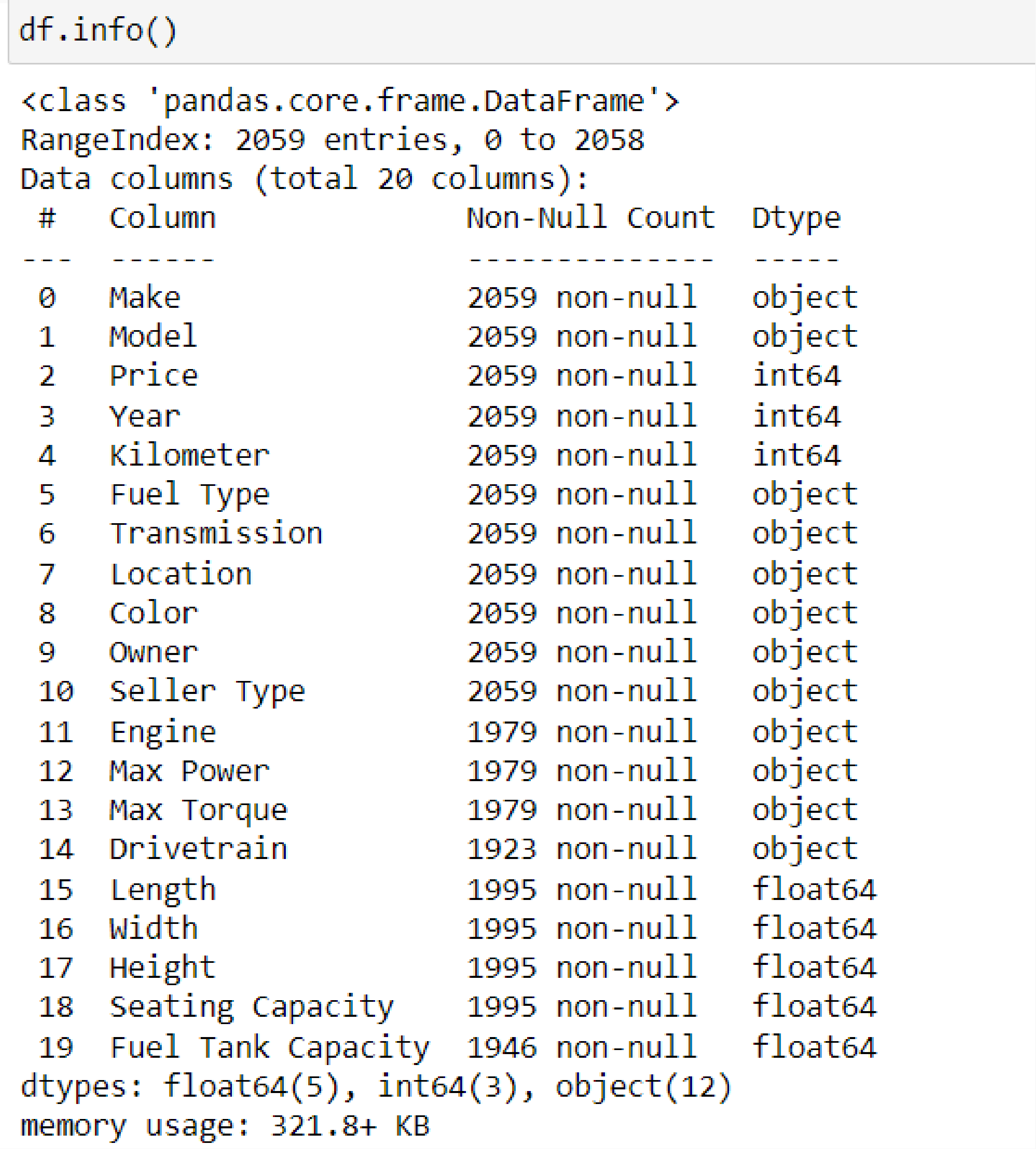
Total Rows and columns 2059 x 20



	Make	Model	Price	Year	Kilometer	Fuel Type	Transmission	Location	Color	Owner	Seller Type	Engine	Max Power	Max Torque	Drivetrain	Length
0	Honda	Amaze 1.2 VX i-VTEC	505000	2017	87150	Petrol	Manual	Pune	Grey	First	Corporate	1198 cc	87 bhp @ 6000 rpm	109 Nm @ 4500 rpm	FWD	3990.0
1	Maruti Suzuki	Swift DZire VDI	450000	2014	75000	Diesel	Manual	Ludhiana	White	Second	Individual	1248 cc	74 bhp @ 4000 rpm	190 Nm @ 2000 rpm	FWD	3995.0
2	Hyundai	i10 Magna 1.2 Kappa2	220000	2011	67000	Petrol	Manual	Lucknow	Maroon	First	Individual	1197 cc	79 bhp @ 6000 rpm	112.7619 Nm @ 4000 rpm	FWD	3585.0
3	Toyota	Glanza G	799000	2019	37500	Petrol	Manual	Mangalore	Red	First	Individual	1197 cc	82 bhp @ 6000 rpm	113 Nm @ 4200 rpm	FWD	3995.0
4	Toyota	Innova 2.4 VX 7 STR [2016-2020]	1950000	2018	69000	Diesel	Manual	Mumbai	Grey	First	Individual	2393 cc	148 bhp @ 3400 rpm	343 Nm @ 1400 rpm	RWD	4735.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2054	Mahindra	XUV500 W8 [2015-2017]	850000	2016	90300	Diesel	Manual	Surat	White	First	Individual	2179 cc	138 bhp @ 3750 rpm	330 Nm @ 1600 rpm	FWD	4585.0
2055	Hyundai	Eon D-Lite +	275000	2014	83000	Petrol	Manual	Ahmedabad	White	Second	Individual	814 cc	55 bhp @ 5500 rpm	75 Nm @ 4000 rpm	FWD	3495.0
2056	Ford	Figo Duratec Petrol ZXI 1.2	240000	2013	73000	Petrol	Manual	Thane	Silver	First	Individual	1196 cc	70 bhp @ 6250 rpm	102 Nm @ 4000 rpm	FWD	3795.0
2057	BMW	5-Series 520d Luxury Line [2017-2019]	4290000	2018	60474	Diesel	Automatic	Coimbatore	White	First	Individual	1995 cc	188 bhp @ 4000 rpm	400 Nm @ 1750 rpm	RWD	4936.0
2058	Mahindra	Bolero Power Plus ZLX [2016-2019]	670000	2017	72000	Diesel	Manual	Guwahati	White	First	Individual	1493 cc	70 bhp @ 3600 rpm	195 Nm @ 1400 rpm	RWD	3995.0

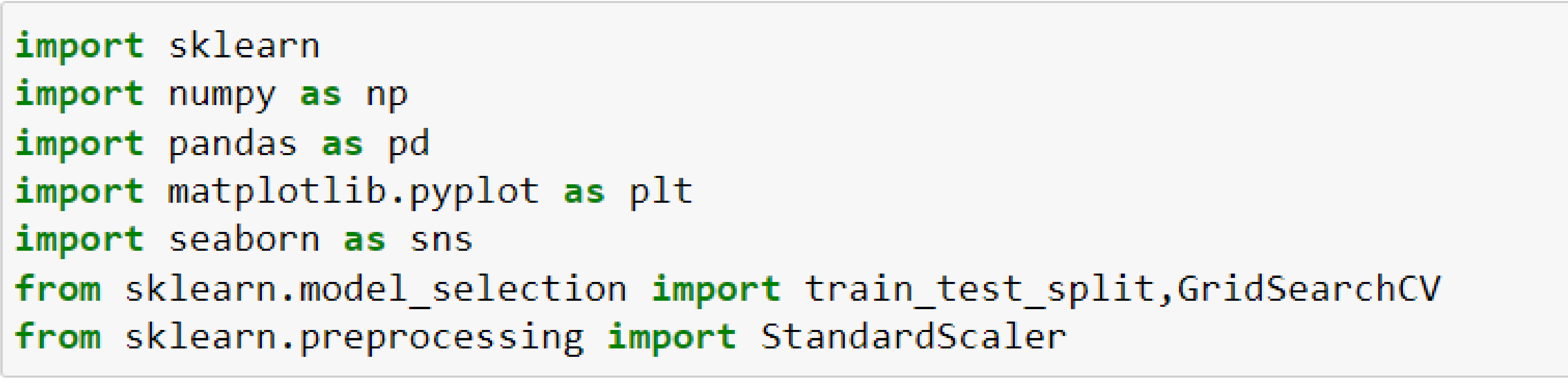
2059 rows × 20 columns

Libraries



```
df.info()
```

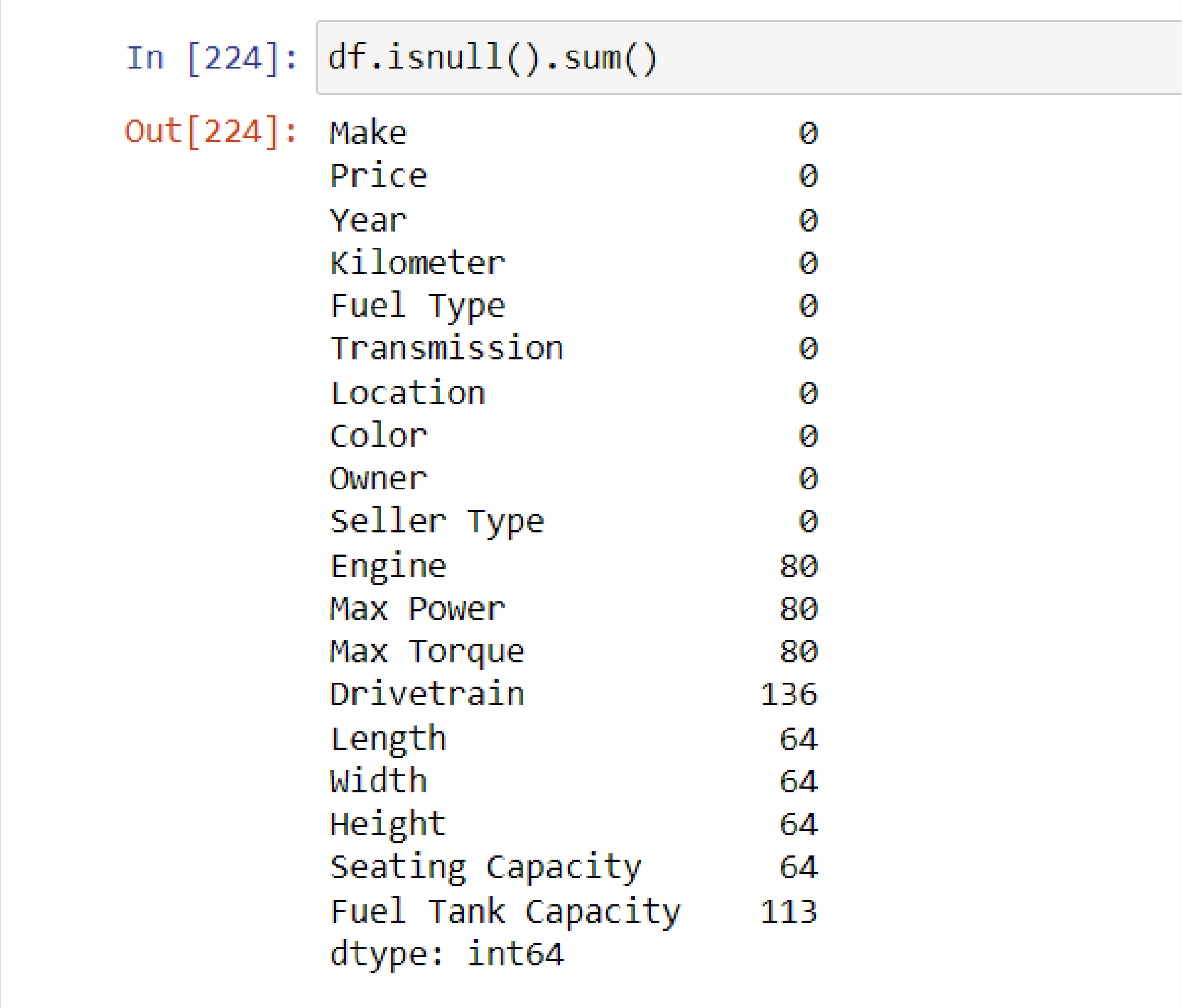
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2059 entries, 0 to 2058
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Make                2059 non-null   object 
 1   Model               2059 non-null   object 
 2   Price               2059 non-null   int64  
 3   Year                2059 non-null   int64  
 4   Kilometer           2059 non-null   int64  
 5   Fuel Type           2059 non-null   object 
 6   Transmission        2059 non-null   object 
 7   Location            2059 non-null   object 
 8   Color               2059 non-null   object 
 9   Owner               2059 non-null   object 
 10  Seller Type         2059 non-null   object 
 11  Engine              1979 non-null   object 
 12  Max Power           1979 non-null   object 
 13  Max Torque          1979 non-null   object 
 14  Drivetrain          1923 non-null   object 
 15  Length              1995 non-null   float64
 16  Width               1995 non-null   float64
 17  Height              1995 non-null   float64
 18  Seating Capacity    1995 non-null   float64
 19  Fuel Tank Capacity  1946 non-null   float64
dtypes: float64(5), int64(3), object(12)
memory usage: 321.8+ KB
```



```
import sklearn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.preprocessing import StandardScaler
```

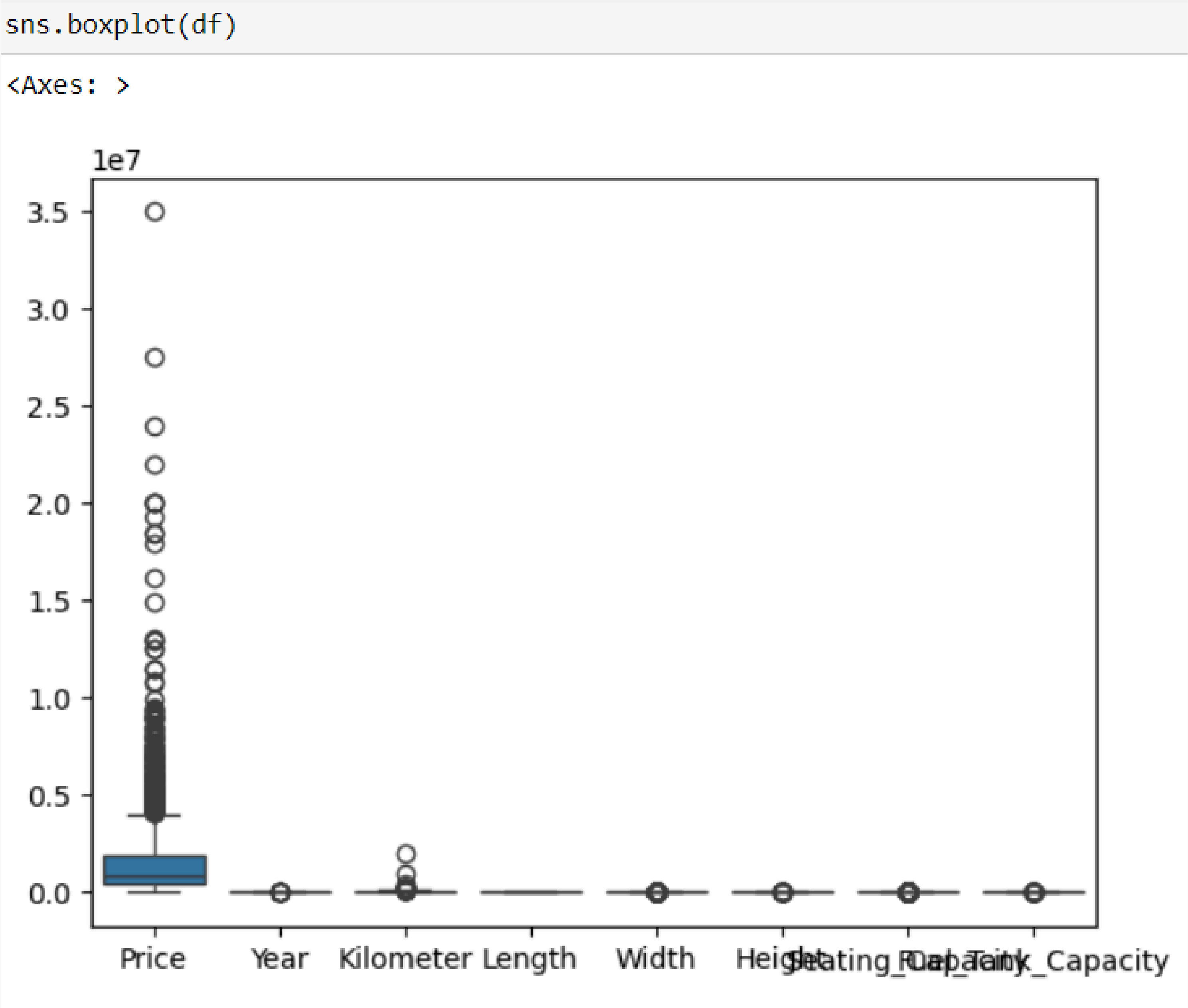
Data Preprocessing

Missing Value Imputation



```
In [224]: df.isnull().sum()

Out[224]: Make                    0
          Price                   0
          Year                    0
          Kilometer               0
          Fuel Type               0
          Transmission            0
          Location                0
          Color                   0
          Owner                   0
          Seller Type             0
          Engine                 80
          Max Power              80
          Max Torque             80
          Drivetrain            136
          Length                 64
          Width                  64
          Height                 64
          Seating Capacity       64
          Fuel Tank Capacity    113
          dtype: int64
```

Outliers

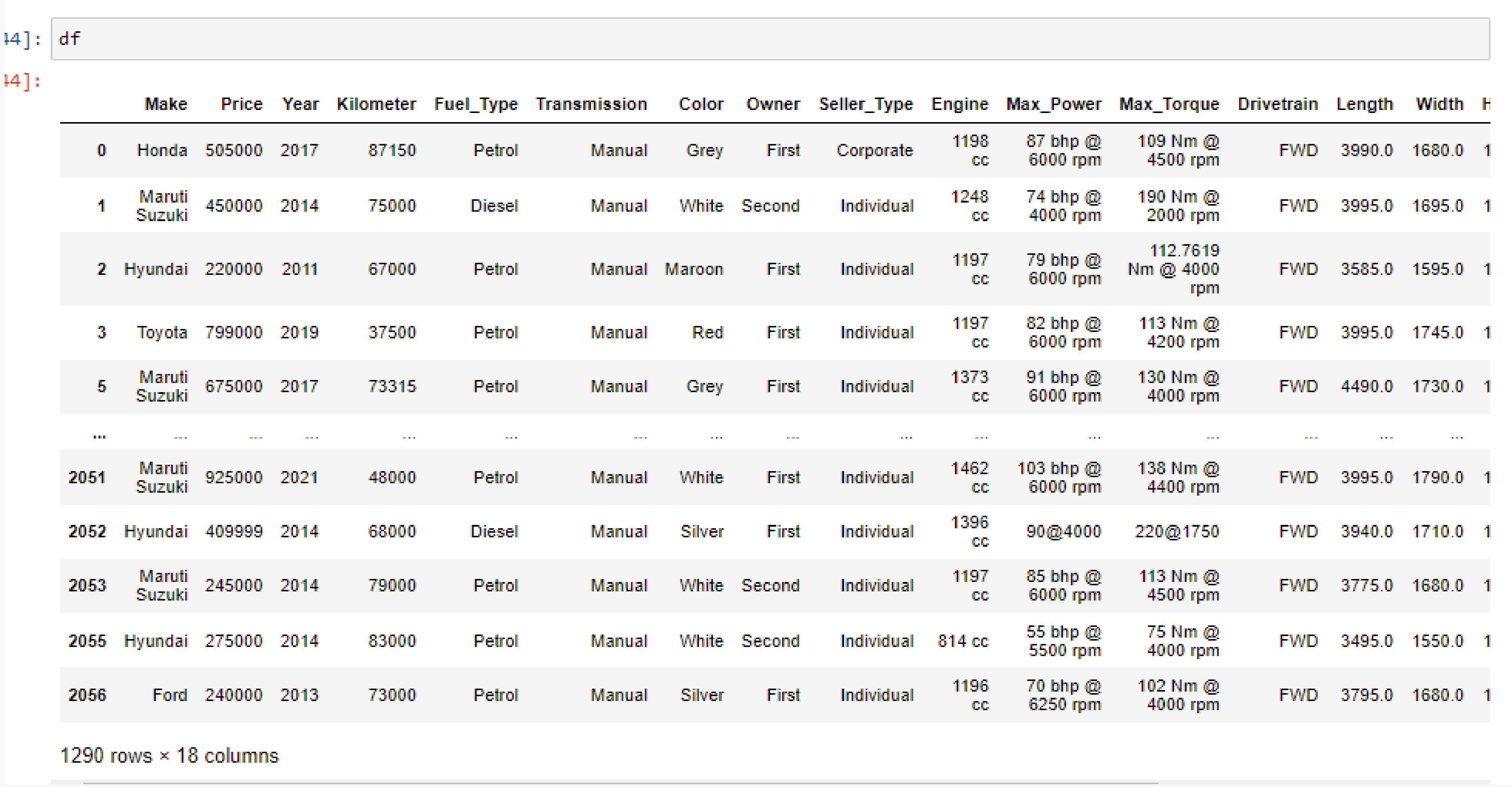


```
sns.boxplot(df)
```

```
<Axes: >
```

After Data Preprocessing

Total Rows and columns 1290 x 18



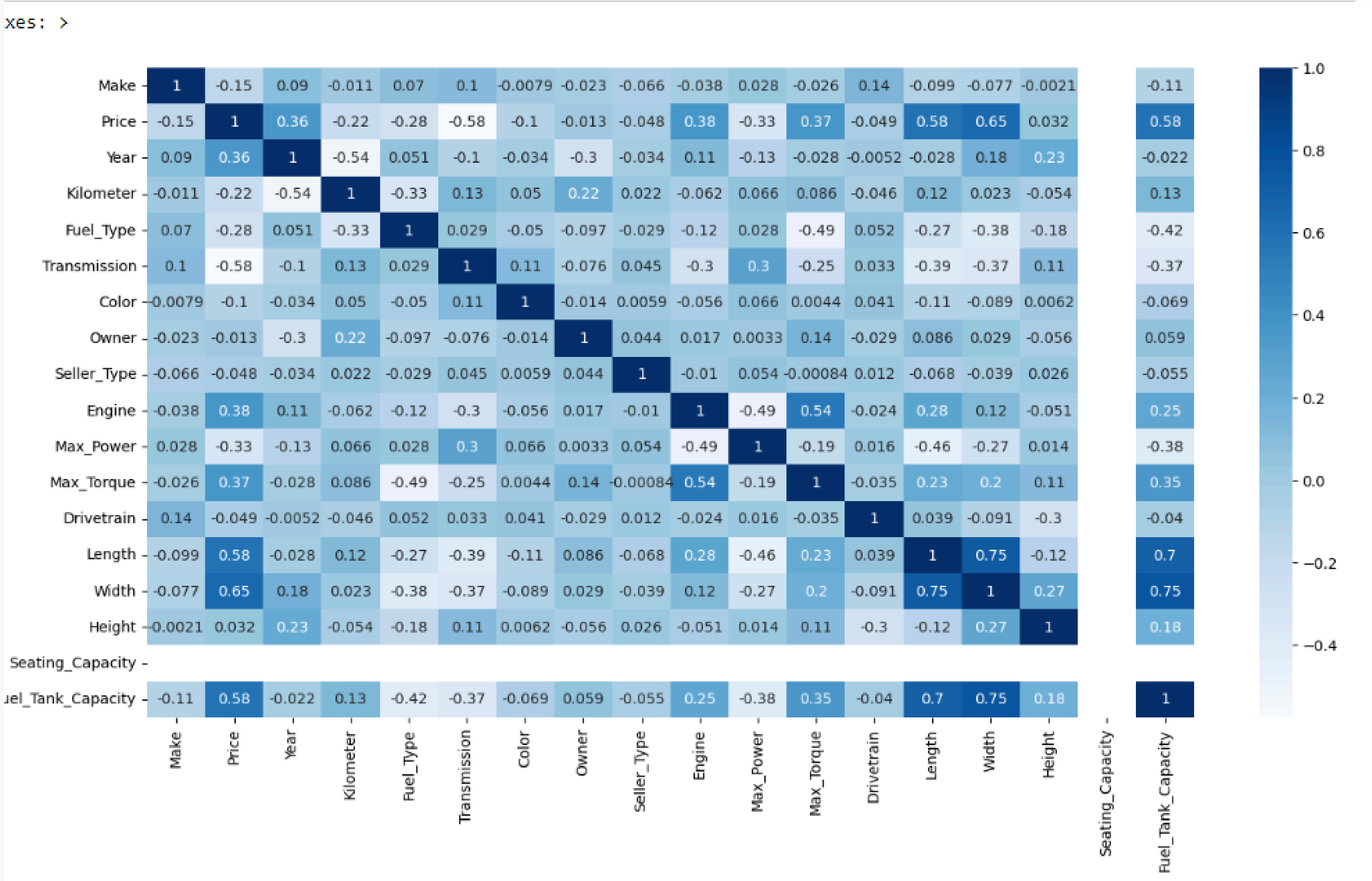
```
44]: df
```

44]:

	Make	Price	Year	Kilometer	Fuel_Type	Transmission	Color	Owner	Seller_Type	Engine	Max_Power	Max_Torque	Drivetrain	Length	Width	H
0	Honda	505000	2017	87150	Petrol	Manual	Grey	First	Corporate	1198 cc	87 bhp @ 6000 rpm	109 Nm @ 4500 rpm	FWD	3990.0	1680.0	1
1	Maruti Suzuki	450000	2014	75000	Diesel	Manual	White	Second	Individual	1248 cc	74 bhp @ 4000 rpm	190 Nm @ 2000 rpm	FWD	3995.0	1695.0	1
2	Hyundai	220000	2011	67000	Petrol	Manual	Maroon	First	Individual	1197 cc	79 bhp @ 6000 rpm	112.7619 Nm @ 4000 rpm	FWD	3585.0	1595.0	1
3	Toyota	799000	2019	37500	Petrol	Manual	Red	First	Individual	1197 cc	82 bhp @ 6000 rpm	113 Nm @ 4200 rpm	FWD	3995.0	1745.0	1
5	Maruti Suzuki	675000	2017	73315	Petrol	Manual	Grey	First	Individual	1373 cc	91 bhp @ 6000 rpm	130 Nm @ 4000 rpm	FWD	4490.0	1730.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2051	Maruti Suzuki	925000	2021	48000	Petrol	Manual	White	First	Individual	1462 cc	103 bhp @ 6000 rpm	138 Nm @ 4400 rpm	FWD	3995.0	1790.0	1
2052	Hyundai	409999	2014	68000	Diesel	Manual	Silver	First	Individual	1396 cc	90@4000	220@1750	FWD	3940.0	1710.0	1
2053	Maruti Suzuki	245000	2014	79000	Petrol	Manual	White	Second	Individual	1197 cc	85 bhp @ 6000 rpm	113 Nm @ 4500 rpm	FWD	3775.0	1680.0	1
2055	Hyundai	275000	2014	83000	Petrol	Manual	White	Second	Individual	814 cc	55 bhp @ 5500 rpm	75 Nm @ 4000 rpm	FWD	3495.0	1550.0	1
2056	Ford	240000	2013	73000	Petrol	Manual	Silver	First	Individual	1196 cc	70 bhp @ 6250 rpm	102 Nm @ 4000 rpm	FWD	3795.0	1680.0	1

1290 rows × 18 columns

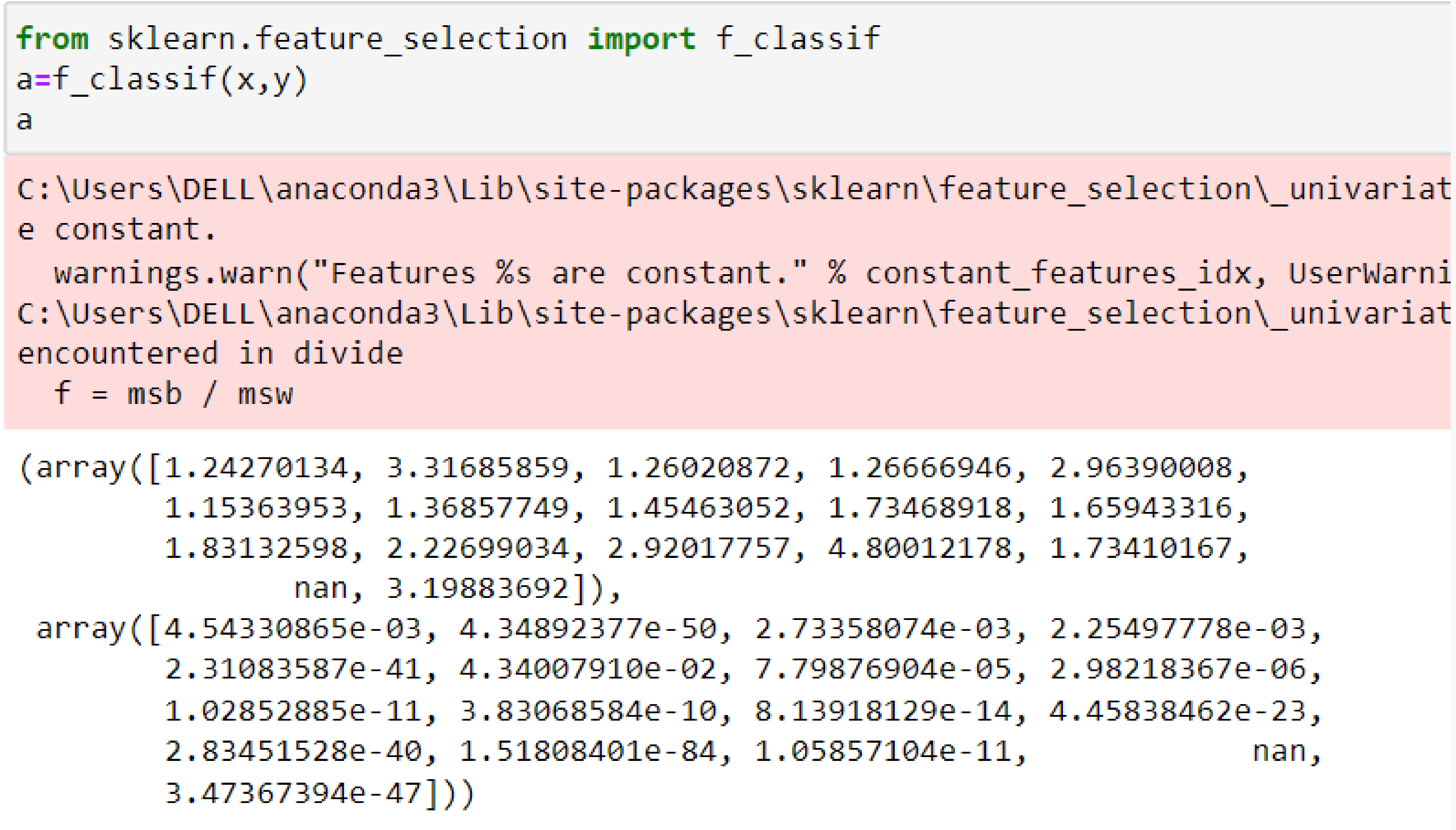
Heatmap



xes: >

	Make	Price	Year	Kilometer	Fuel_Type	Transmission	Color	Owner	Seller_Type	Engine	Max_Power	Max_Torque	Drivetrain	Length	Width	Height	Seating_Capacity	Fuel_Tank_Capacity
Make	1	-0.15	0.09	-0.011	0.07	0.1	-0.0079	-0.023	-0.066	-0.038	0.028	-0.026	0.14	-0.099	-0.077	-0.0021		-0.11
Price	-0.15	1	0.36	-0.22	-0.28	-0.58	-0.1	-0.013	-0.048	0.38	-0.33	0.37	-0.049	0.58	0.65	0.032		0.58
Year	0.09	0.36	1	-0.54	0.051	-0.1	-0.034	-0.3	-0.034	0.11	-0.13	-0.028	-0.0052	-0.028	0.18	0.23		-0.022
Kilometer	-0.011	-0.22	-0.54	1	-0.33	0.13	0.05	0.22	0.022	-0.062	0.066	0.086	-0.046	0.12	0.023	-0.054		0.13
Fuel_Type	0.07	-0.28	0.051	-0.33	1	0.029	-0.05	-0.097	-0.029	-0.12	0.028	-0.49	0.052	-0.27	-0.38	-0.18		-0.42
Transmission	0.1	-0.58	-0.1	0.13	0.029	1	0.11	-0.076	0.045	-0.3	0.3	-0.25	0.033	-0.39	-0.37	0.11		-0.37
Color	-0.0079	-0.1	-0.034	0.05	-0.05	0.11	1	-0.014	0.0059	-0.056	0.066	0.0044	0.041	-0.11	-0.089	0.0062		-0.069
Owner	-0.023	-0.013	-0.3	0.22	-0.097	-0.076	-0.014	1	0.044	0.017	0.0033	0.14	-0.029	0.086	0.029	-0.056		0.059
Seller_Type	-0.066	-0.048	-0.034	0.022	-0.029	0.045	0.0059	0.044	1	-0.01	0.054	-0.00084	0.012	-0.068	-0.039	0.026		-0.055
Engine	-0.038	0.38	0.11	-0.062	-0.12	-0.3	-0.056	0.017	-0.01	1	-0.49	0.54	-0.024	0.28	0.12	-0.051		0.25
Max_Power	0.028	-0.33	-0.13	0.066	0.028	0.3	0.066	0.0033	0.054	-0.49	1	-0.19	0.016	-0.46	-0.27	0.014		-0.38
Max_Torque	-0.026	0.37	-0.028	0.086	-0.49	-0.25	0.0044	0.14	-0.00084	0.54	-0.19	1	-0.035	0.23	0.2	0.11		0.35
Drivetrain	0.14	-0.049	-0.0052	-0.046	0.052	0.033	0.041	-0.029	0.012	-0.024	0.016	-0.035	1	0.039	-0.091	-0.3		-0.04
Length	-0.099	0.58	-0.028	0.12	-0.27	-0.39	-0.11	0.086	-0.068	0.28	-0.46	0.23	0.039	1	0.75	-0.12		0.7
Width	-0.077	0.65	0.18	0.023	-0.38	-0.37	-0.089	0.029	-0.039	0.12	-0.27	0.2	-0.091	0.75	1	0.27		0.75
Height	-0.0021	0.032	0.23	-0.054	-0.18	0.11	0.0062	-0.056	0.026	-0.051	0.014	0.11	-0.3	-0.12	0.27	1		0.18
Seating_Capacity																		
uel_Tank_Capacity	-0.11	0.58	-0.022	0.13	-0.42	-0.37	-0.069	0.059	-0.055	0.25	-0.38	0.35	-0.04	0.7	0.75	0.18		1

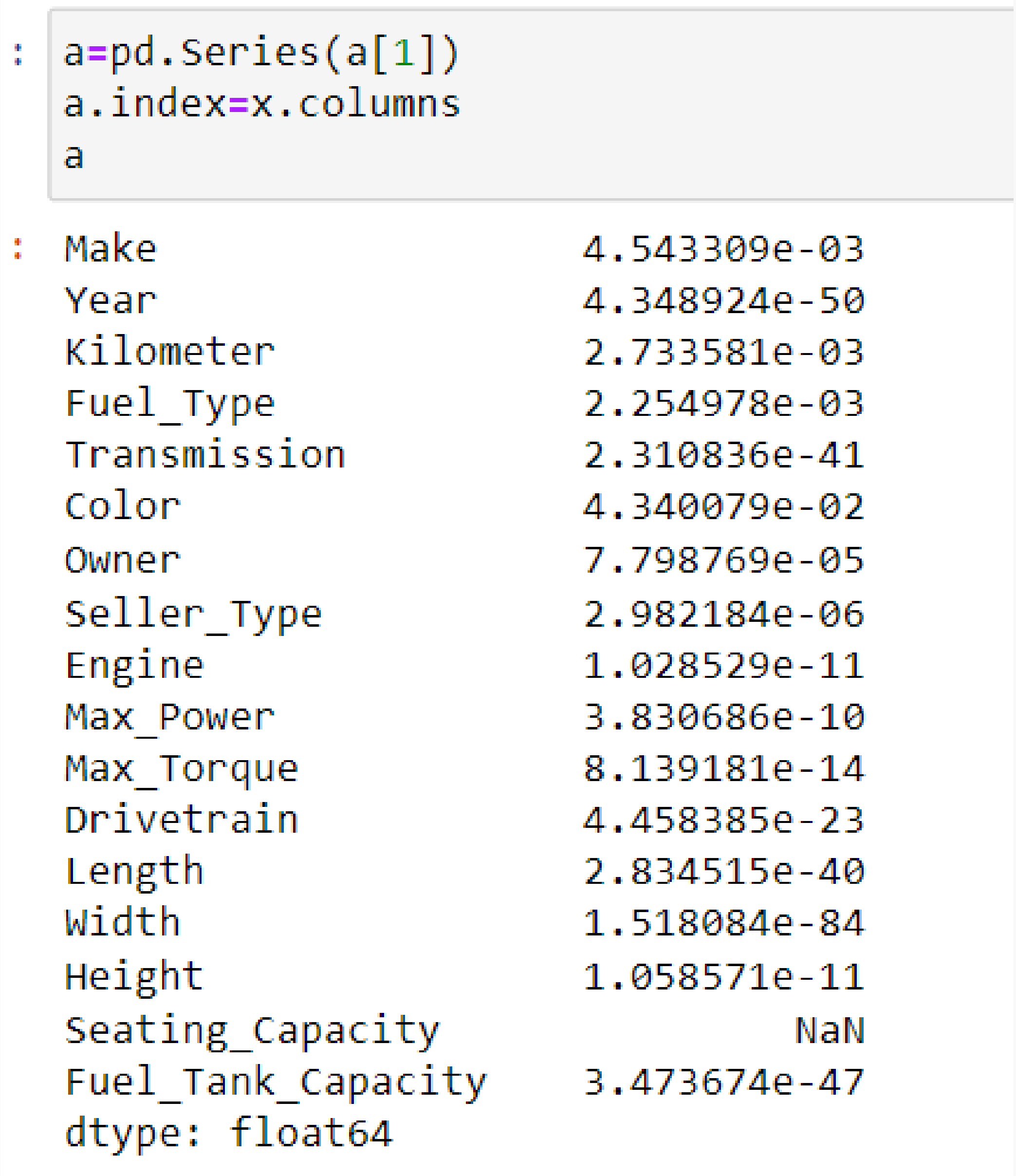
Anova Testing



```python
from sklearn.feature_selection import f_classif
a=f_classif(x,y)
a
```

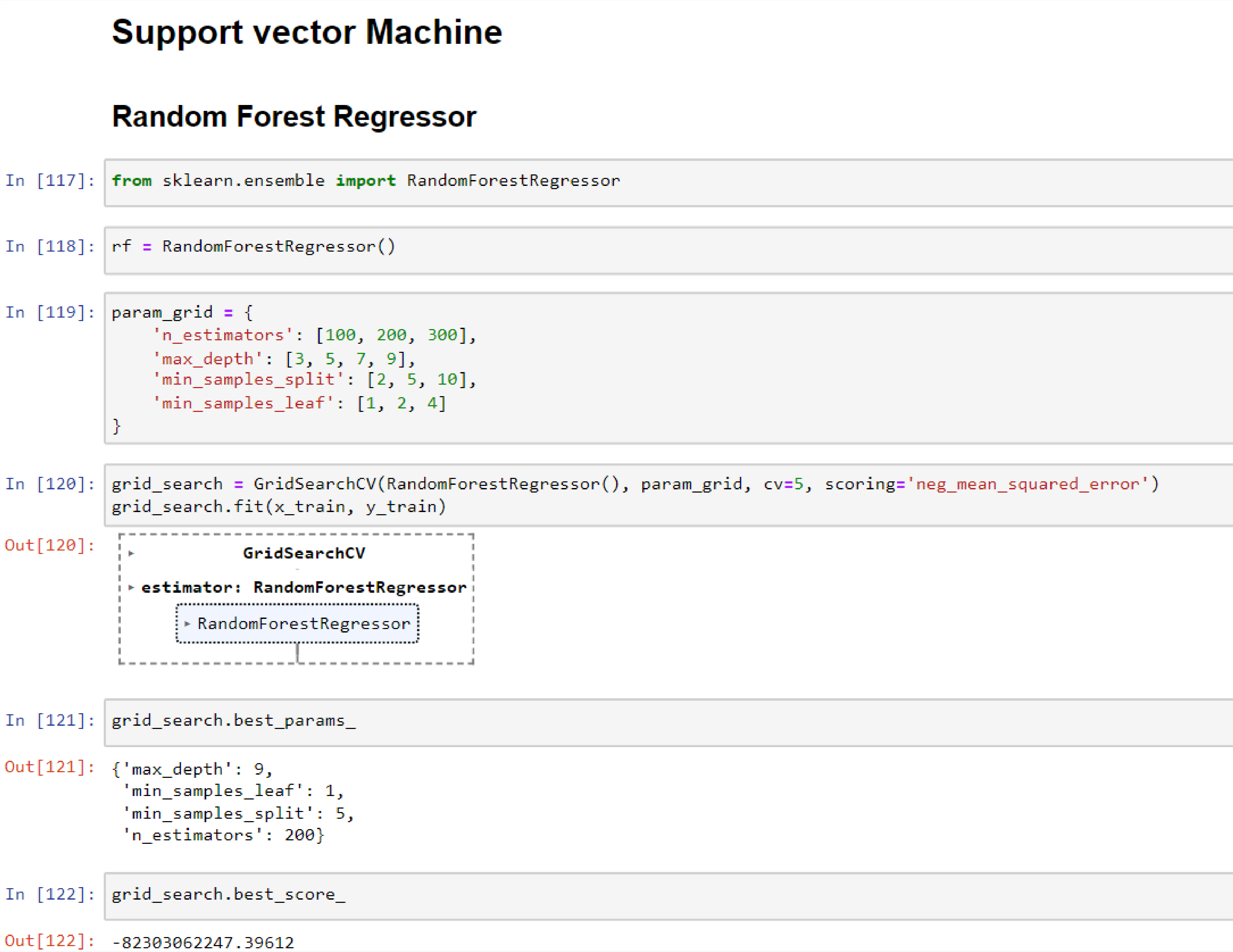
```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\feature_selection\_univariat
e constant.
  warnings.warn("Features %s are constant." % constant_features_idx, UserWarni
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\feature_selection\_univariat
encountered in divide
  f = msb / msw
```

```
(array([1.24270134, 3.31685859, 1.26020872, 1.26666946, 2.96390008,
        1.15363953, 1.36857749, 1.45463052, 1.73468918, 1.65943316,
        1.83132598, 2.22699034, 2.92017757, 4.80012178, 1.73410167,
               nan, 3.19883692]),
 array([4.54330865e-03, 4.34892377e-50, 2.73358074e-03, 2.25497778e-03,
        2.31083587e-41, 4.34007910e-02, 7.79876904e-05, 2.98218367e-06,
        1.02852885e-11, 3.83068584e-10, 8.13918129e-14, 4.45838462e-23,
        2.83451528e-40, 1.51808401e-84, 1.05857104e-11,            nan,
        3.47367394e-47]))
```



```
: a=pd.Series(a[1])
  a.index=x.columns
  a
```

```
: Make                  4.543309e-03
  Year                  4.348924e-50
  Kilometer             2.733581e-03
  Fuel_Type             2.254978e-03
  Transmission          2.310836e-41
  Color                 4.340079e-02
  Owner                 7.798769e-05
  Seller_Type           2.982184e-06
  Engine                1.028529e-11
  Max_Power             3.830686e-10
  Max_Torque            8.139181e-14
  Drivetrain            4.458385e-23
  Length                2.834515e-40
  Width                 1.518084e-84
  Height                1.058571e-11
  Seating_Capacity               NaN
  Fuel_Tank_Capacity    3.473674e-47
  dtype: float64
```



Support vector Machine

Random Forest Regressor

```
In [117]: from sklearn.ensemble import RandomForestRegressor
```

```
In [118]: rf = RandomForestRegressor()
```

```
In [119]: param_grid = {
              'n_estimators': [100, 200, 300],
              'max_depth': [3, 5, 7, 9],
              'min_samples_split': [2, 5, 10],
              'min_samples_leaf': [1, 2, 4]
          }
```

```
In [120]: grid_search = GridSearchCV(RandomForestRegressor(), param_grid, cv=5, scoring='neg_mean_squared_error')
          grid_search.fit(x_train, y_train)
```

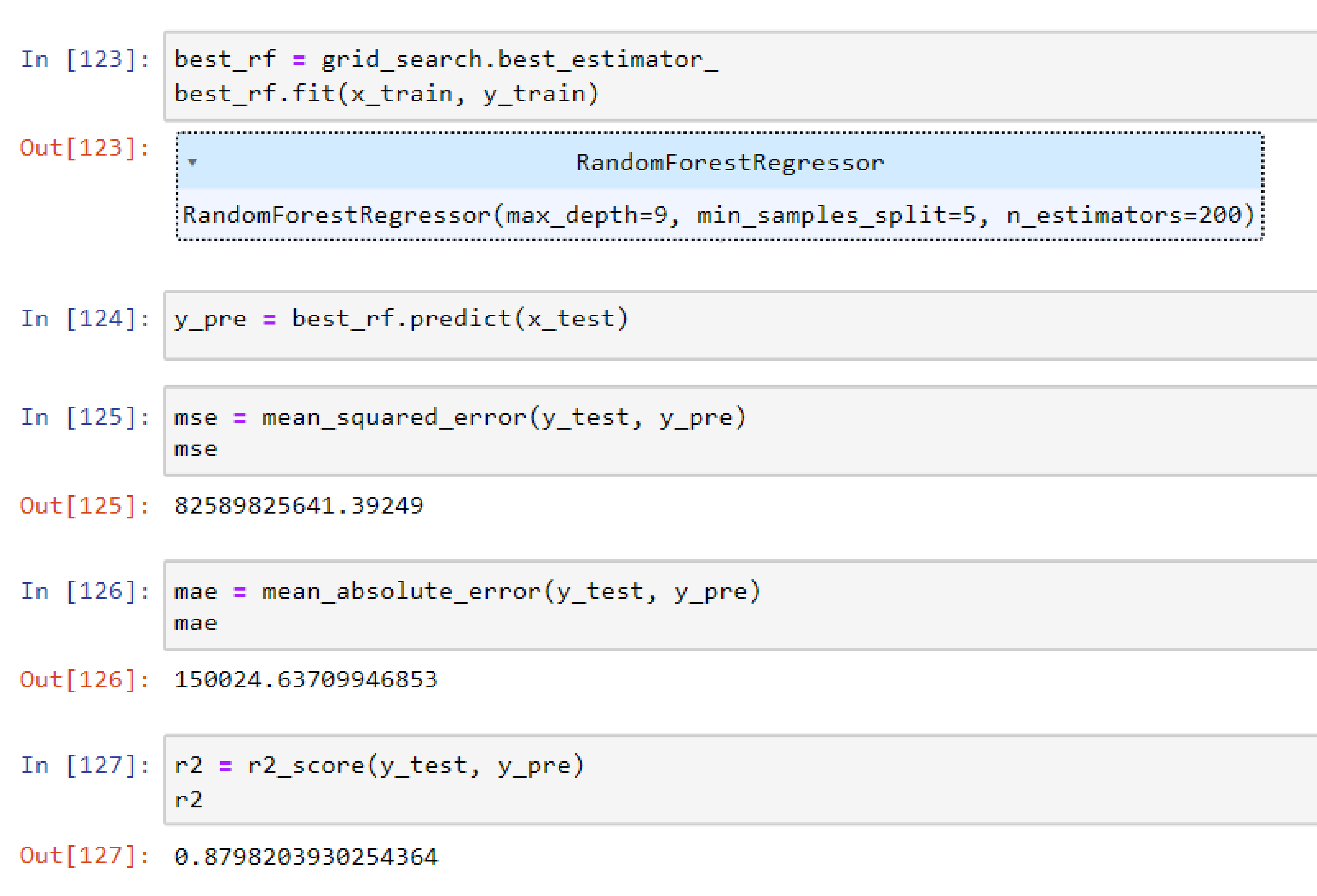
```
Out[120]:
▸ GridSearchCV
▸ estimator: RandomForestRegressor
    ▸ RandomForestRegressor
```

```
In [121]: grid_search.best_params_
```

```
Out[121]: {'max_depth': 9,
           'min_samples_leaf': 1,
           'min_samples_split': 5,
           'n_estimators': 200}
```

```
In [122]: grid_search.best_score_
```

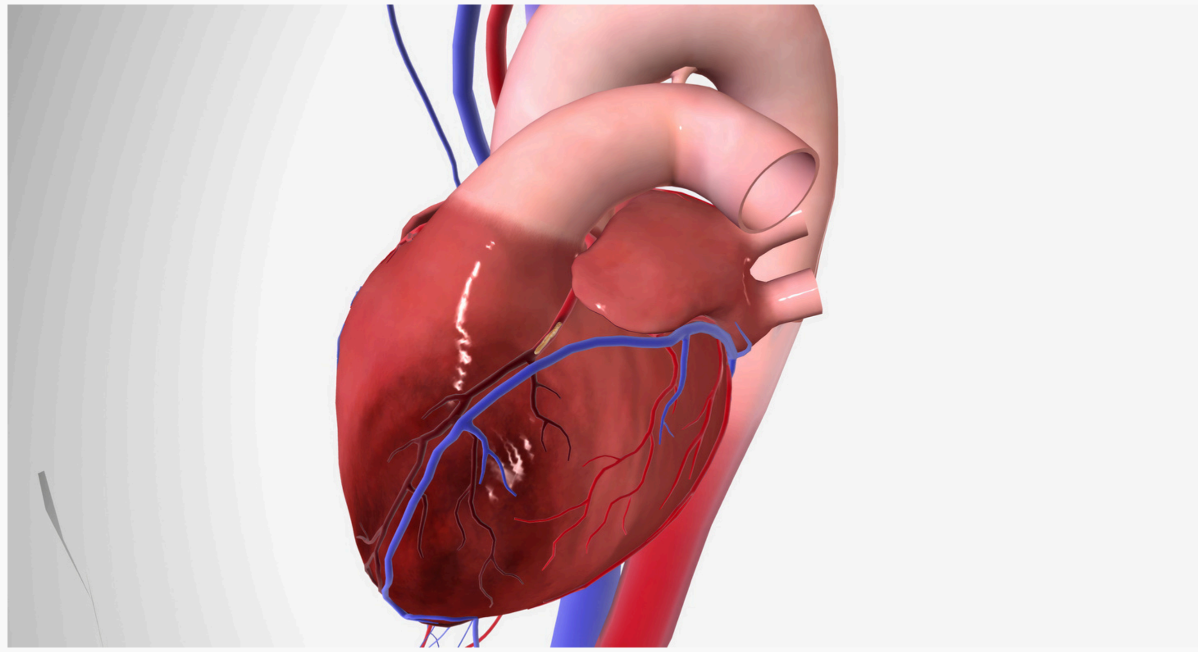
```
Out[122]: -82303062247.39612
```

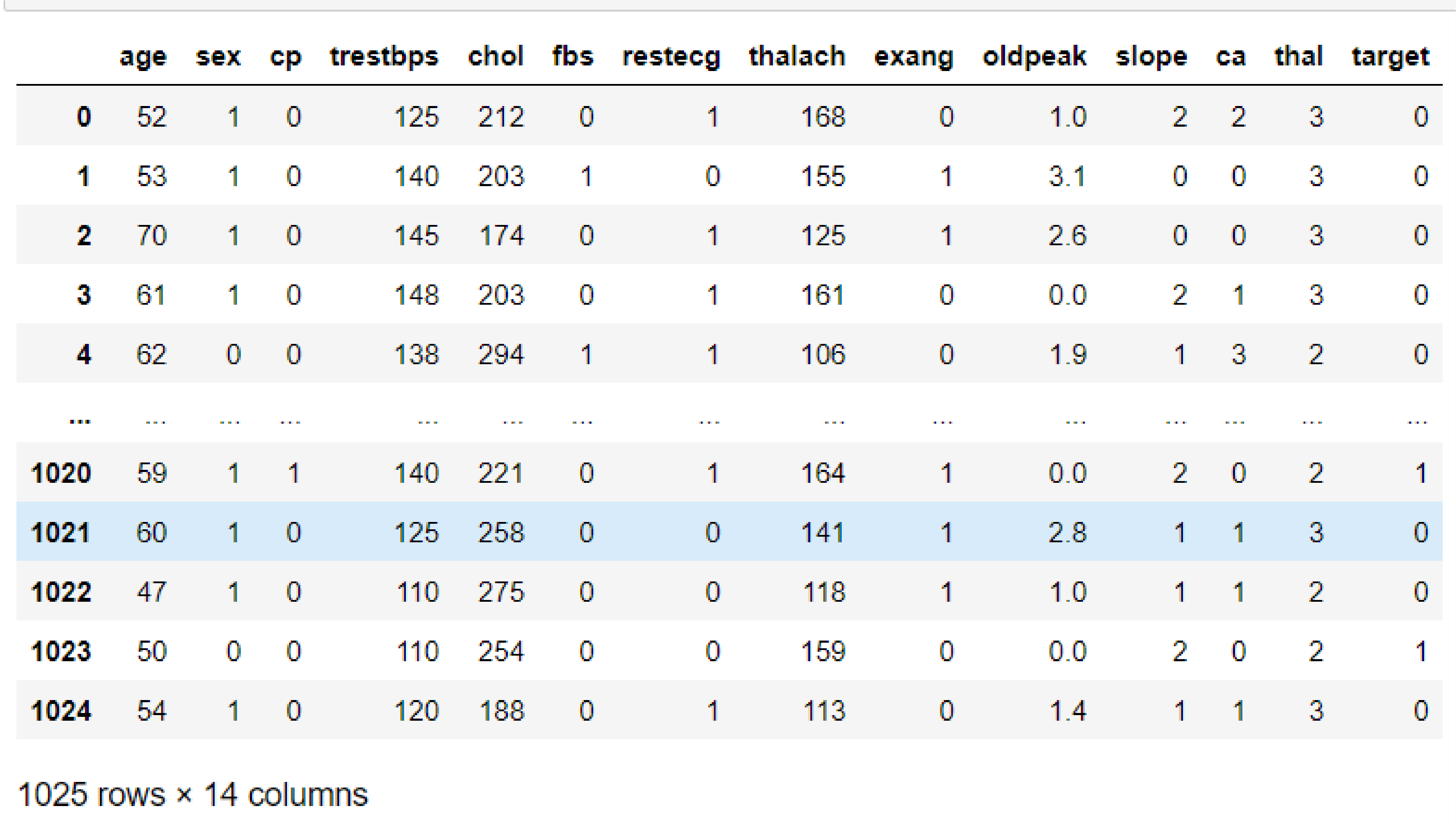


```
In [123]: best_rf = grid_search.best_estimator_
          best_rf.fit(x_train, y_train)

Out[123]: ▾                RandomForestRegressor
          RandomForestRegressor(max_depth=9, min_samples_split=5, n_estimators=200)

In [124]: y_pre = best_rf.predict(x_test)

In [125]: mse = mean_squared_error(y_test, y_pre)
          mse

Out[125]: 82589825641.39249

In [126]: mae = mean_absolute_error(y_test, y_pre)
          mae

Out[126]: 150024.63709946853

In [127]: r2 = r2_score(y_test, y_pre)
          r2

Out[127]: 0.8798203930254364
```

Classifier(Heart Dataset)

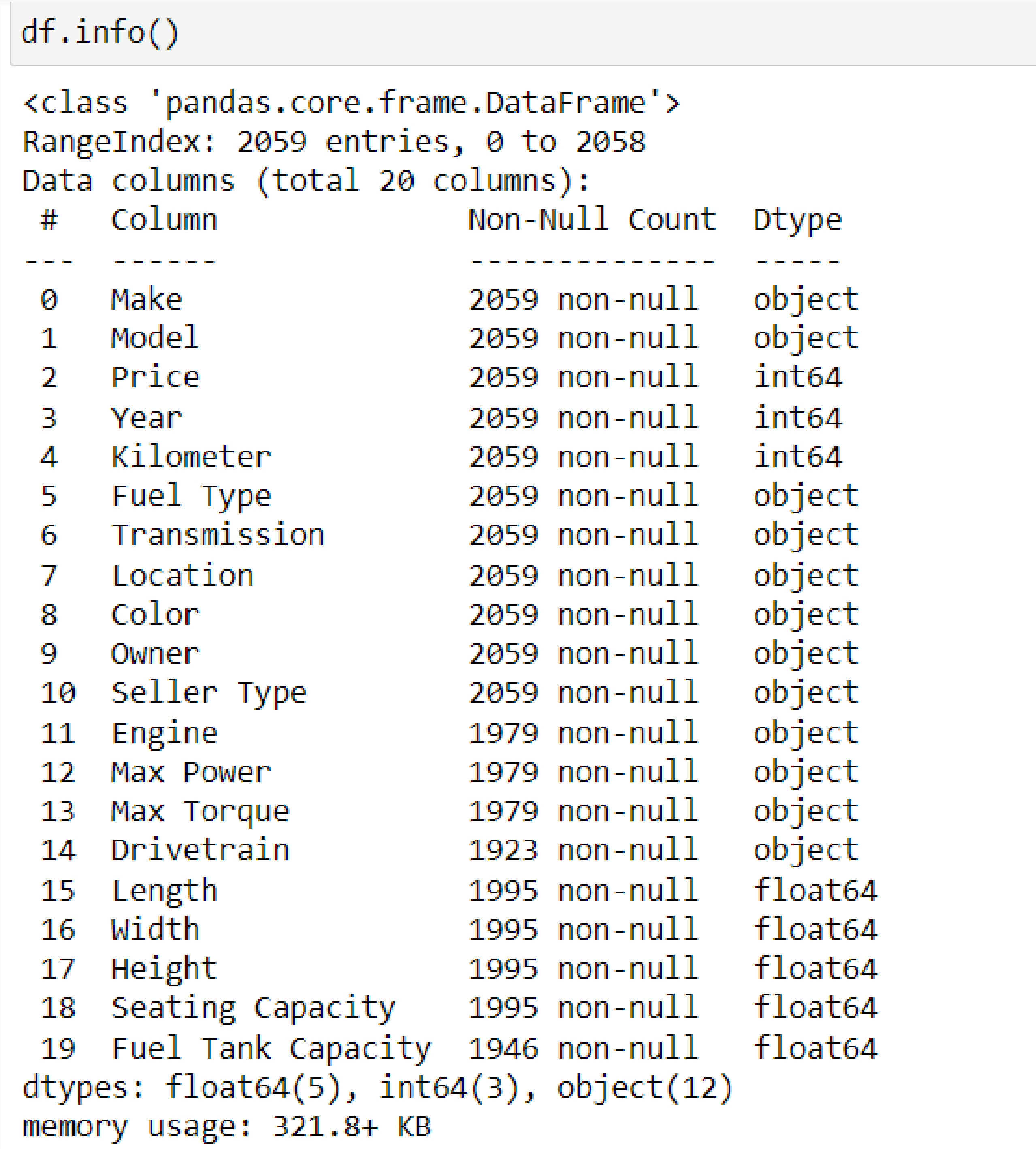
Total Rows and columns
1025 x 14





	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

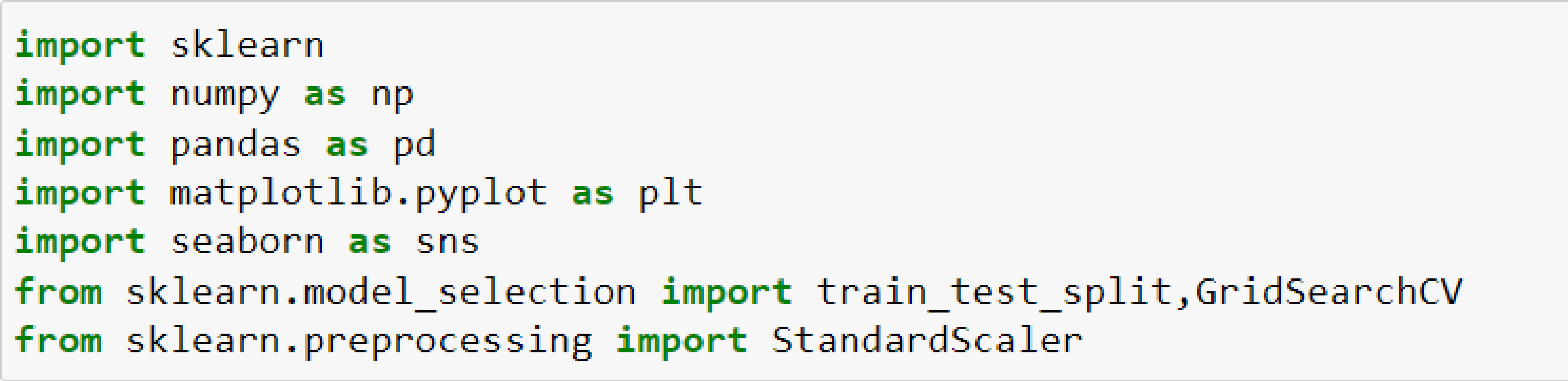
1025 rows × 14 columns



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2059 entries, 0 to 2058
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Make                2059 non-null   object 
 1   Model               2059 non-null   object 
 2   Price               2059 non-null   int64  
 3   Year                2059 non-null   int64  
 4   Kilometer           2059 non-null   int64  
 5   Fuel Type           2059 non-null   object 
 6   Transmission        2059 non-null   object 
 7   Location            2059 non-null   object 
 8   Color               2059 non-null   object 
 9   Owner               2059 non-null   object 
 10  Seller Type         2059 non-null   object 
 11  Engine              1979 non-null   object 
 12  Max Power           1979 non-null   object 
 13  Max Torque          1979 non-null   object 
 14  Drivetrain          1923 non-null   object 
 15  Length              1995 non-null   float64
 16  Width               1995 non-null   float64
 17  Height              1995 non-null   float64
 18  Seating Capacity    1995 non-null   float64
 19  Fuel Tank Capacity  1946 non-null   float64
dtypes: float64(5), int64(3), object(12)
memory usage: 321.8+ KB
```

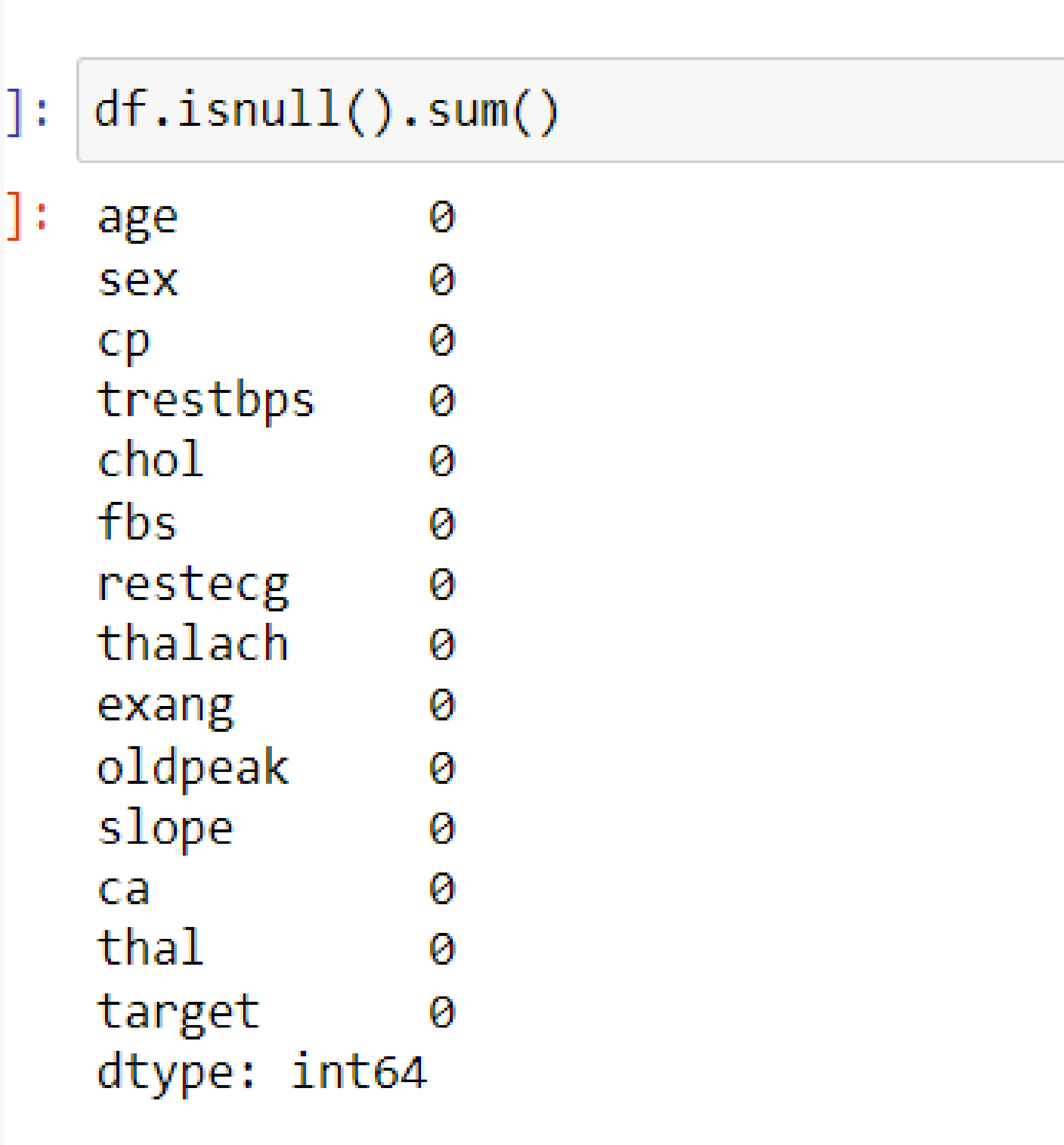
Libraries



```
import sklearn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.preprocessing import StandardScaler
```

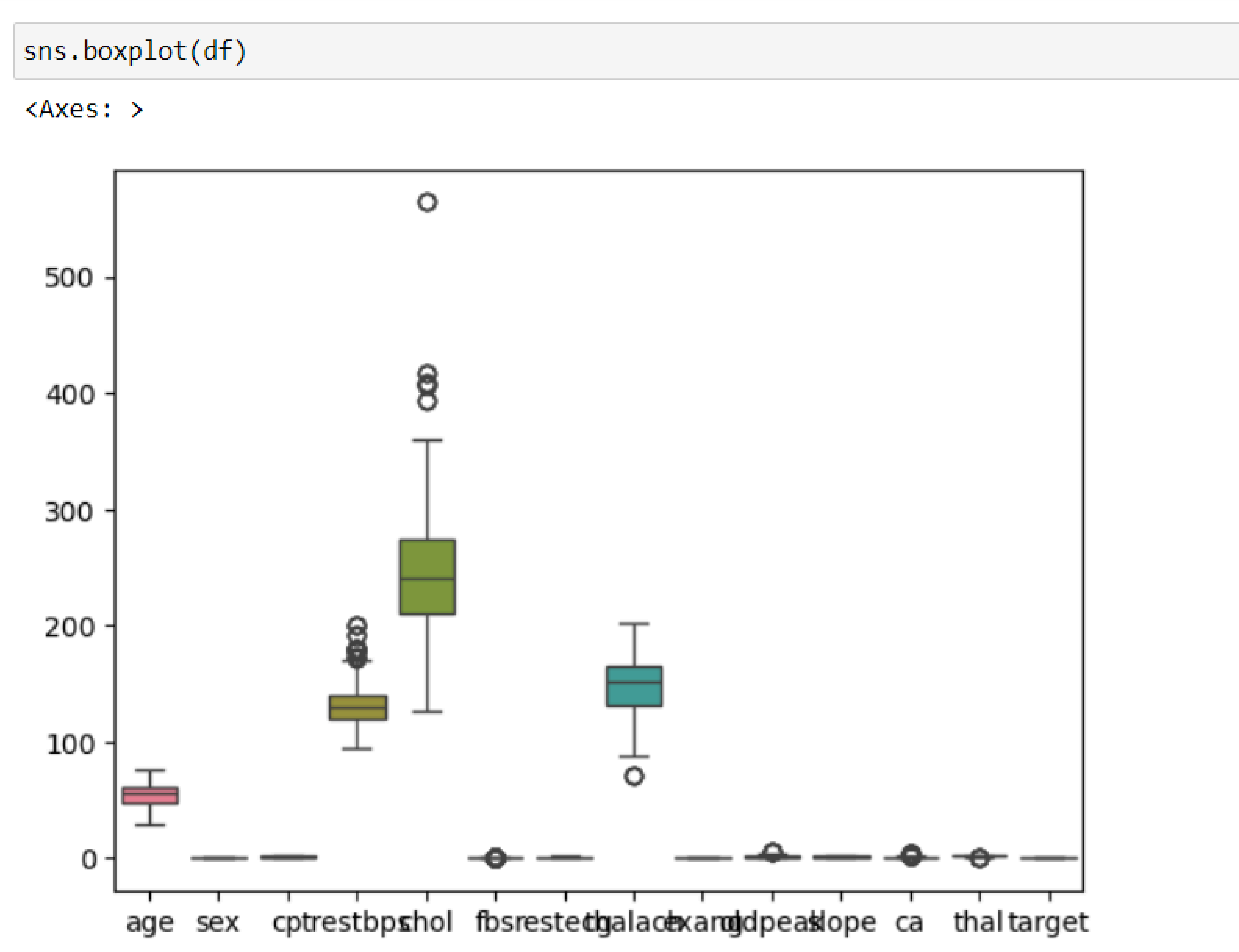
Data Preprocessing

Missing Value Imputation



```
]: df.isnull().sum()

]: age         0
   sex         0
   cp          0
   trestbps    0
   chol        0
   fbs         0
   restecg     0
   thalach     0
   exang       0
   oldpeak     0
   slope       0
   ca          0
   thal        0
   target      0
   dtype: int64
```

Outliers

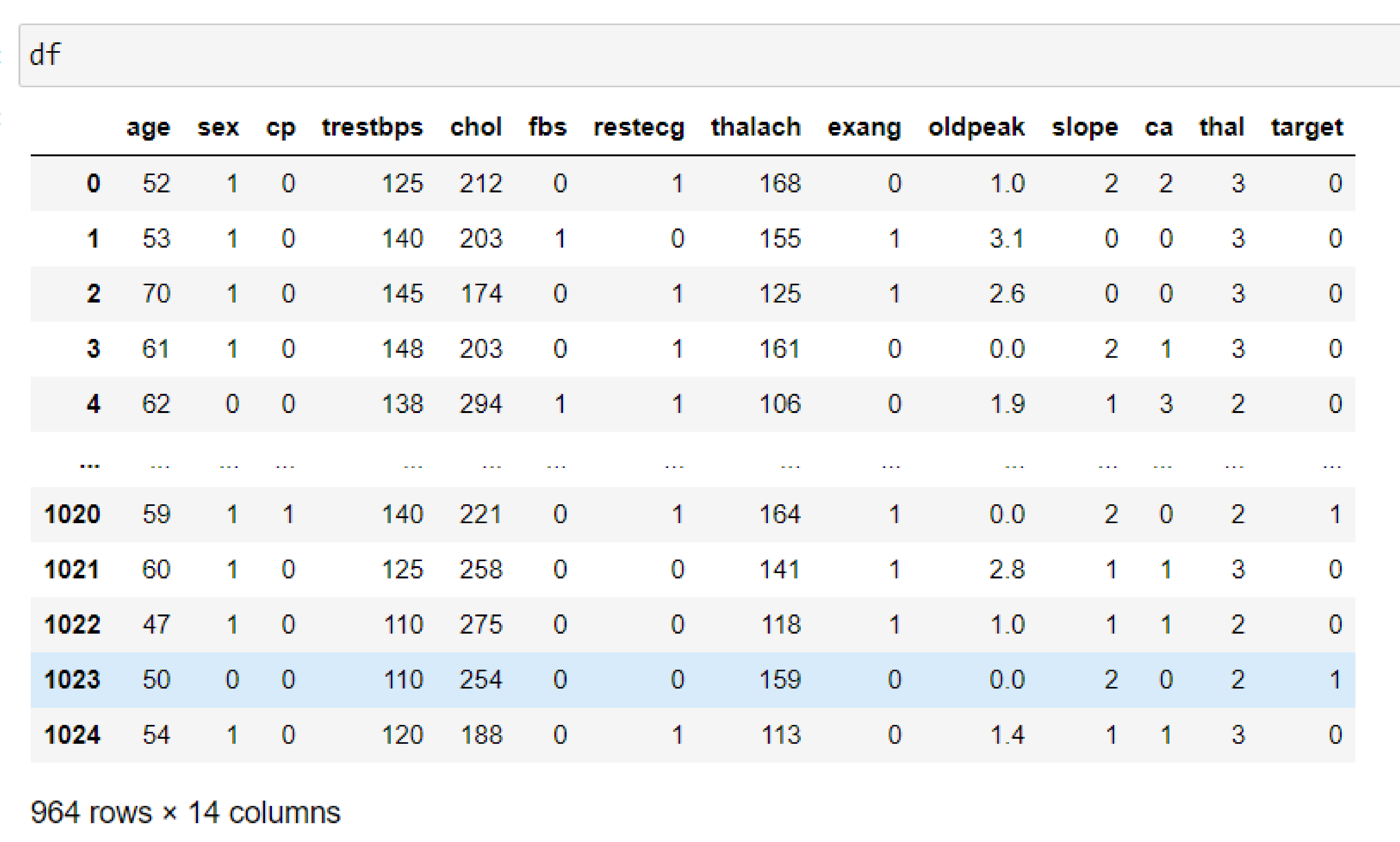


```
sns.boxplot(df)
```

```
<Axes: >
```

After Data Preprocessing

Total Rows and columns
964 x 14



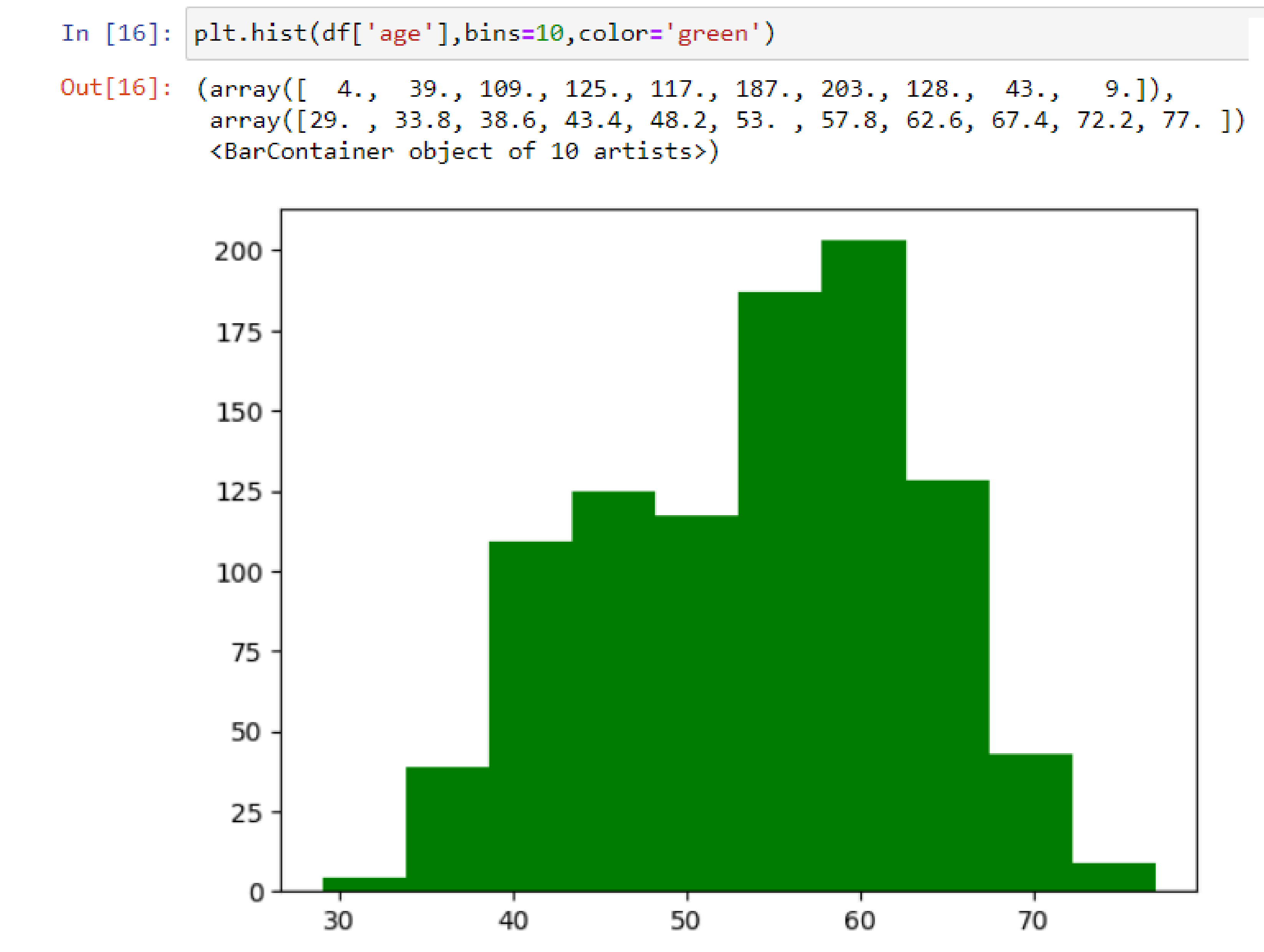
```
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

964 rows × 14 columns

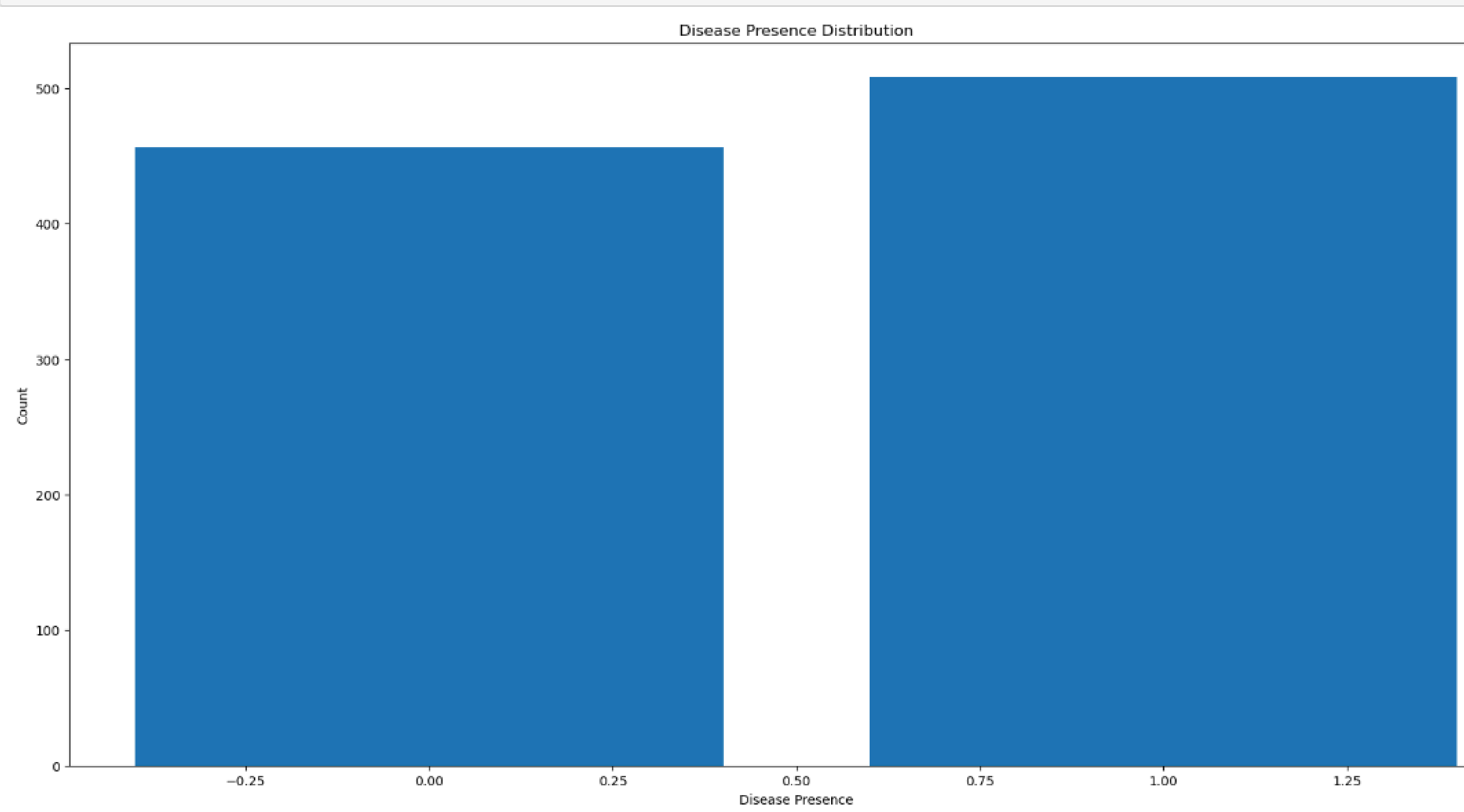
Univariate Analysis

Histogram



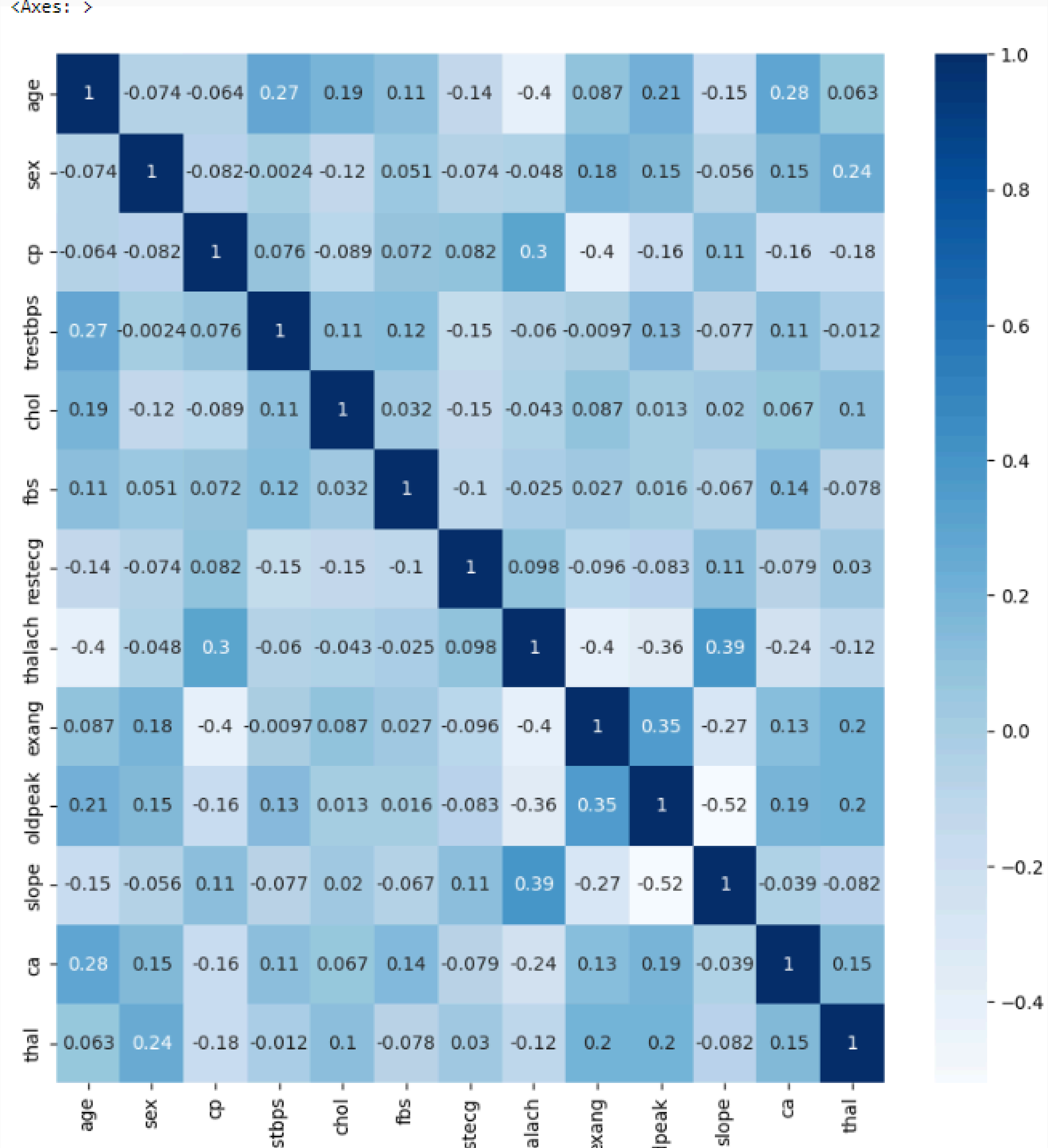
```
In [16]: plt.hist(df['age'],bins=10,color='green')

Out[16]: (array([  4.,  39., 109., 125., 117., 187., 203., 128.,  43.,   9.]),
          array([29. , 33.8, 38.6, 43.4, 48.2, 53. , 57.8, 62.6, 67.4, 72.2, 77. ]),
          <BarContainer object of 10 artists>)
```

Bar Chart



Disease Presence Distribution

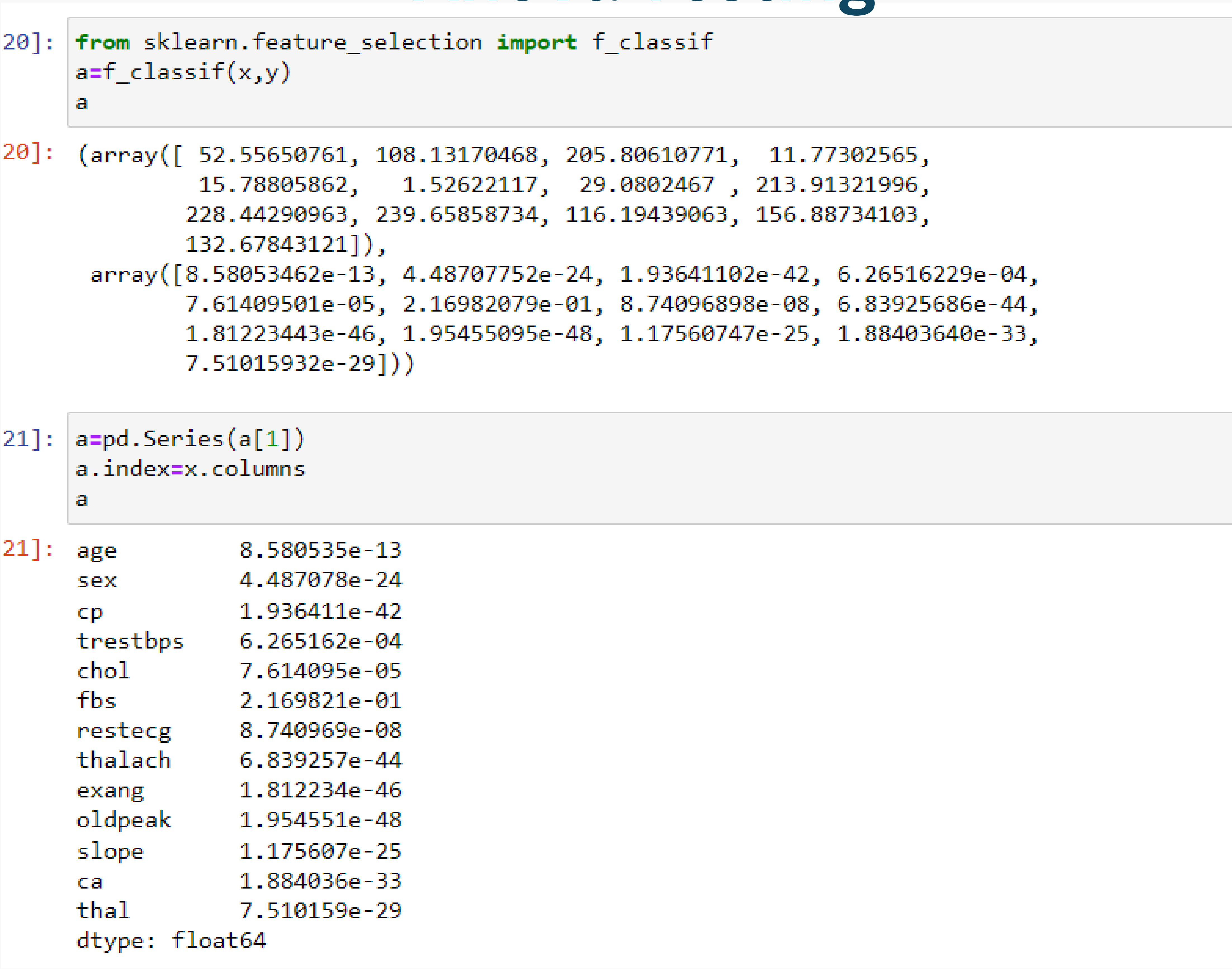
Heatmap



```
<Axes: >
```

	age	sex	cp	stbps	chol	fbs	stecg	alach	xang	lpeak	slope	ca	thal
age	1	-0.074	-0.064	0.27	0.19	0.11	-0.14	-0.4	0.087	0.21	-0.15	0.28	0.063
sex	-0.074	1	-0.082	-0.0024	-0.12	0.051	-0.074	-0.048	0.18	0.15	-0.056	0.15	0.24
cp	-0.064	-0.082	1	0.076	-0.089	0.072	0.082	0.3	-0.4	-0.16	0.11	-0.16	-0.18
trestbps	0.27	-0.0024	0.076	1	0.11	0.12	-0.15	-0.06	-0.0097	0.13	-0.077	0.11	-0.012
chol	0.19	-0.12	-0.089	0.11	1	0.032	-0.15	-0.043	0.087	0.013	0.02	0.067	0.1
fbs	0.11	0.051	0.072	0.12	0.032	1	-0.1	-0.025	0.027	0.016	-0.067	0.14	-0.078
restecg	-0.14	-0.074	0.082	-0.15	-0.15	-0.1	1	0.098	-0.096	-0.083	0.11	-0.079	0.03
thalach	-0.4	-0.048	0.3	-0.06	-0.043	-0.025	0.098	1	-0.4	-0.36	0.39	-0.24	-0.12
exang	0.087	0.18	-0.4	-0.0097	0.087	0.027	-0.096	-0.4	1	0.35	-0.27	0.13	0.2
oldpeak	0.21	0.15	-0.16	0.13	0.013	0.016	-0.083	-0.36	0.35	1	-0.52	0.19	0.2
slope	-0.15	-0.056	0.11	-0.077	0.02	-0.067	0.11	0.39	-0.27	-0.52	1	-0.039	-0.082
ca	0.28	0.15	-0.16	0.11	0.067	0.14	-0.079	-0.24	0.13	0.19	-0.039	1	0.15
thal	0.063	0.24	-0.18	-0.012	0.1	-0.078	0.03	-0.12	0.2	0.2	-0.082	0.15	1

Anova Testing



```
20]: from sklearn.feature_selection import f_classif
     a=f_classif(x,y)
     a
```

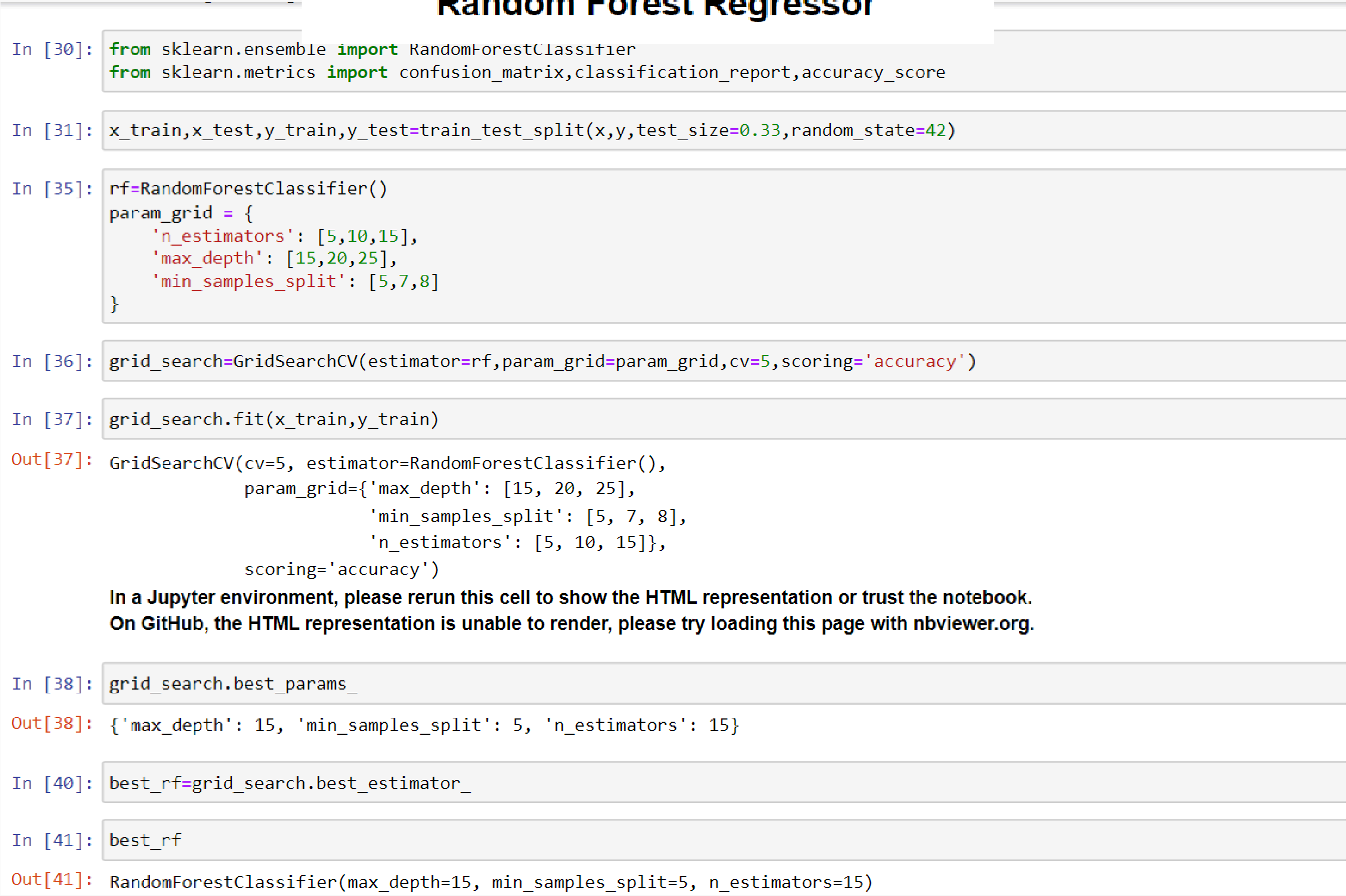
```
20]: (array([ 52.55650761, 108.13170468, 205.80610771,  11.77302565,
              15.78805862,   1.52622117,  29.0802467 , 213.91321996,
             228.44290963, 239.65858734, 116.19439063, 156.88734103,
             132.67843121]),
      array([8.58053462e-13, 4.48707752e-24, 1.93641102e-42, 6.26516229e-04,
             7.61409501e-05, 2.16982079e-01, 8.74096898e-08, 6.83925686e-44,
             1.81223443e-46, 1.95455095e-48, 1.17560747e-25, 1.88403640e-33,
             7.51015932e-29]))
```

```
21]: a=pd.Series(a[1])
     a.index=x.columns
     a
```

```
21]: age         8.580535e-13
     sex         4.487078e-24
     cp          1.936411e-42
     trestbps    6.265162e-04
     chol        7.614095e-05
     fbs         2.169821e-01
     restecg     8.740969e-08
     thalach     6.839257e-44
     exang       1.812234e-46
     oldpeak     1.954551e-48
     slope       1.175607e-25
     ca          1.884036e-33
     thal        7.510159e-29
     dtype: float64
```

Support vector Machine

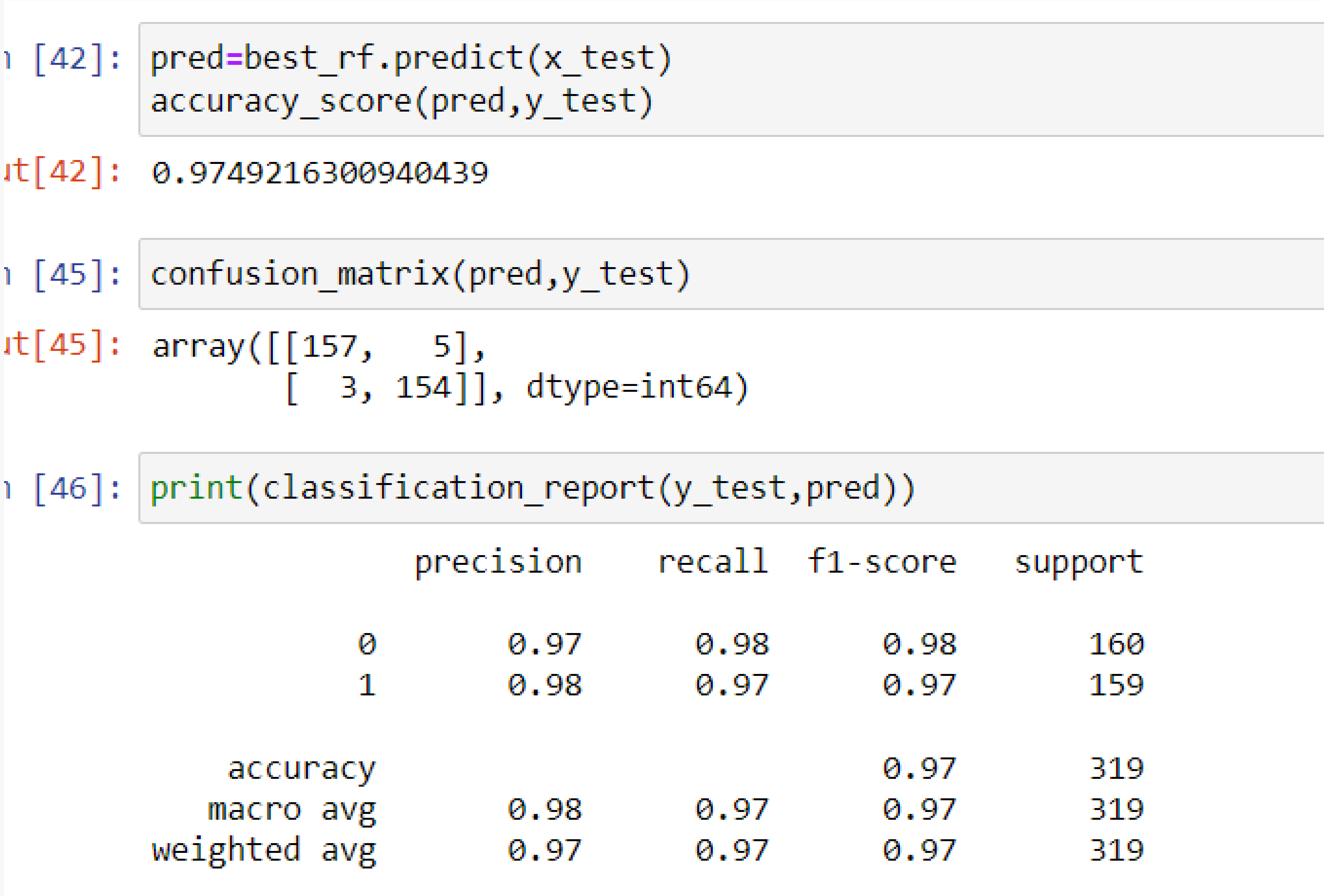
Random Forest Regressor



```
In [30]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import confusion_matrix,classification_report,accuracy_score

In [31]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)

In [35]: rf=RandomForestClassifier()
         param_grid = {
             'n_estimators': [5,10,15],
             'max_depth': [15,20,25],
             'min_samples_split': [5,7,8]
         }

In [36]: grid_search=GridSearchCV(estimator=rf,param_grid=param_grid,cv=5,scoring='accuracy')

In [37]: grid_search.fit(x_train,y_train)

Out[37]: GridSearchCV(cv=5, estimator=RandomForestClassifier(),
                      param_grid={'max_depth': [15, 20, 25],
                                  'min_samples_split': [5, 7, 8],
                                  'n_estimators': [5, 10, 15]},
                      scoring='accuracy')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [38]: grid_search.best_params_

Out[38]: {'max_depth': 15, 'min_samples_split': 5, 'n_estimators': 15}

In [40]: best_rf=grid_search.best_estimator_

In [41]: best_rf

Out[41]: RandomForestClassifier(max_depth=15, min_samples_split=5, n_estimators=15)
```



```
 [42]: pred=best_rf.predict(x_test)
       accuracy_score(pred,y_test)

t[42]: 0.9749216300940439

 [45]: confusion_matrix(pred,y_test)

t[45]: array([[157,   5],
              [  3, 154]], dtype=int64)

 [46]: print(classification_report(y_test,pred))

                     precision    recall  f1-score   support

                0       0.97      0.98      0.98       160
                1       0.98      0.97      0.97       159

         accuracy                           0.97       319
        macro avg       0.98      0.97      0.97       319
     weighted avg       0.97      0.97      0.97       319
```

Accuracy Score - 0.9749316