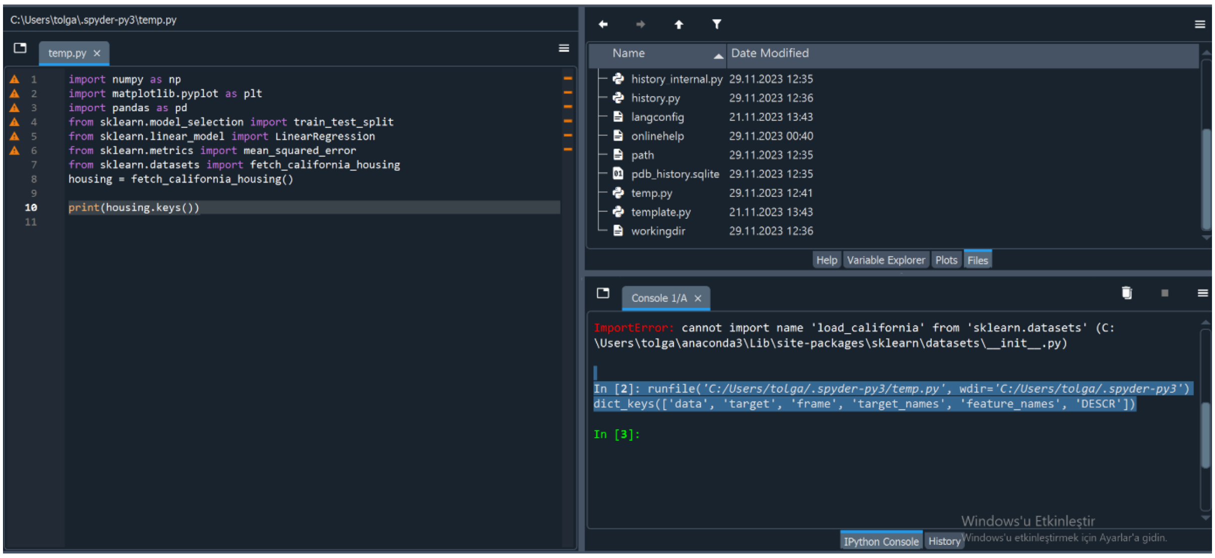
1.1)

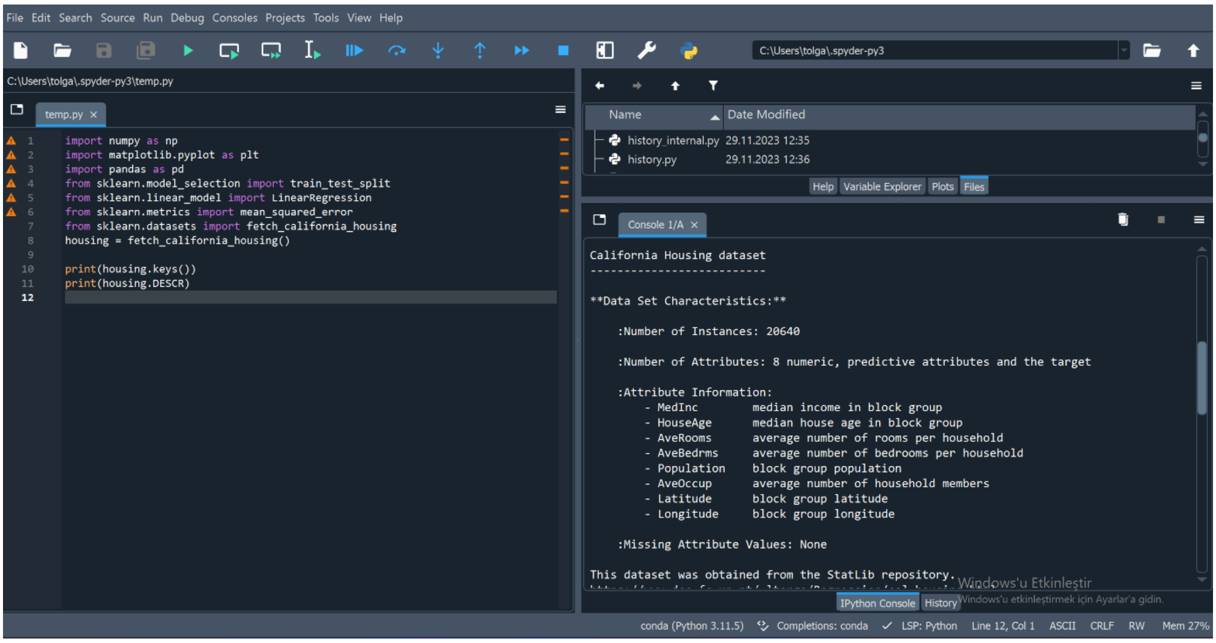


```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

print(housing.keys())

```

Name	Date Modified
history_internal.py	29.11.2023 12:35
history.py	29.11.2023 12:36
langconfig	21.11.2023 13:43
onlinehelp	29.11.2023 00:40
path	29.11.2023 12:35
pdb_history.sqlite	29.11.2023 12:35
temp.py	29.11.2023 12:41
template.py	21.11.2023 13:43
workingdir	29.11.2023 12:36

```
ImportError: cannot import name 'load_california' from 'sklearn.datasets' (C:
\Users\tolga\anaconda3\Lib\site-packages\sklearn\datasets\__init__.py)


In [2]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
dict_keys(['data', 'target', 'frame', 'target_names', 'feature_names', 'DESCR'])

In [3]:
```

1.2)

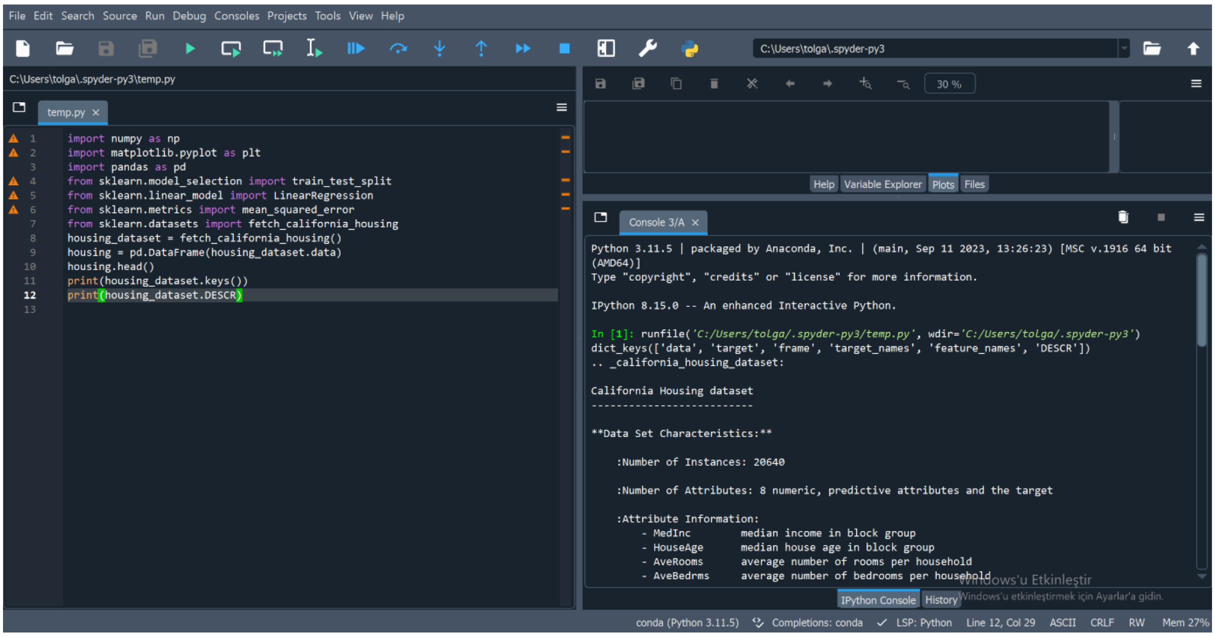


```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

print(housing.keys())
print(housing.DESCR)

```

Name	Date Modified
history_internal.py	29.11.2023 12:35
history.py	29.11.2023 12:36

```
California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude

    :Missing Attribute Values: None

This dataset was obtained from the StatLib repository.
```

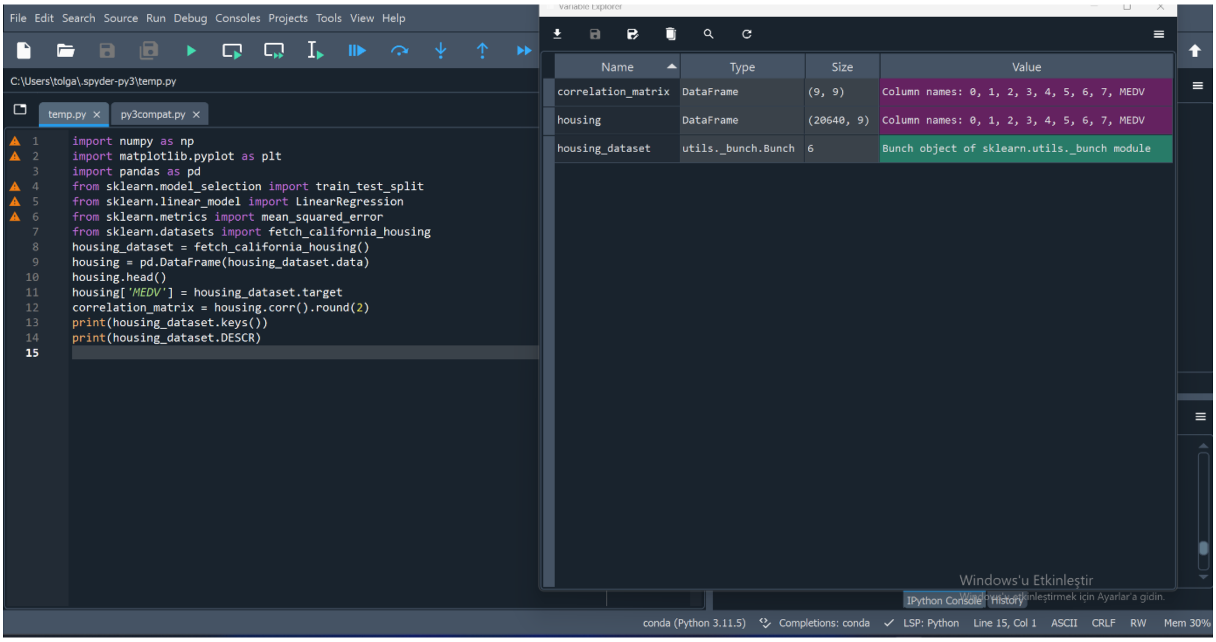
1.3)



```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
print(housing_dataset.keys())
print(housing_dataset.DESCR)
```

```
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
dict_keys(['data', 'target', 'frame', 'target_names', 'feature_names', 'DESCR'])
.. _california_housing_dataset:

California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
```

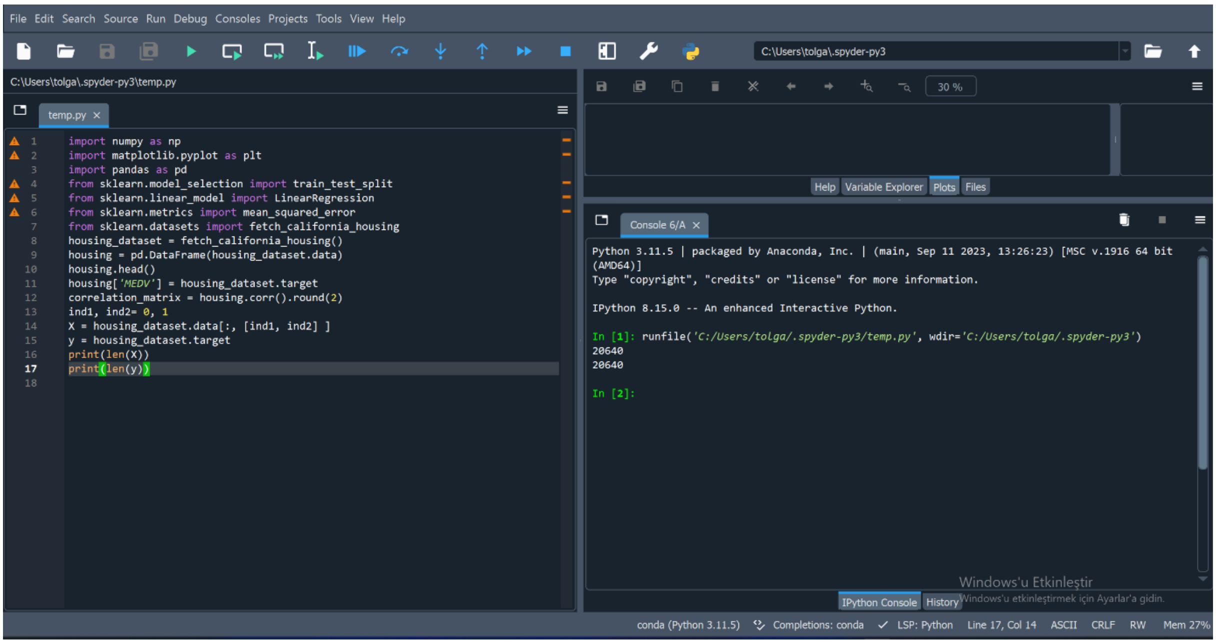
1.4)



```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
print(housing_dataset.keys())
print(housing_dataset.DESCR)
```

Name	Type	Size	Value
correlation_matrix	DataFrame	(9, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, MEDV
housing	DataFrame	(20640, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, MEDV
housing_dataset	utils._bunch.Bunch	6	Bunch object of sklearn.utils._bunch module

1.5)



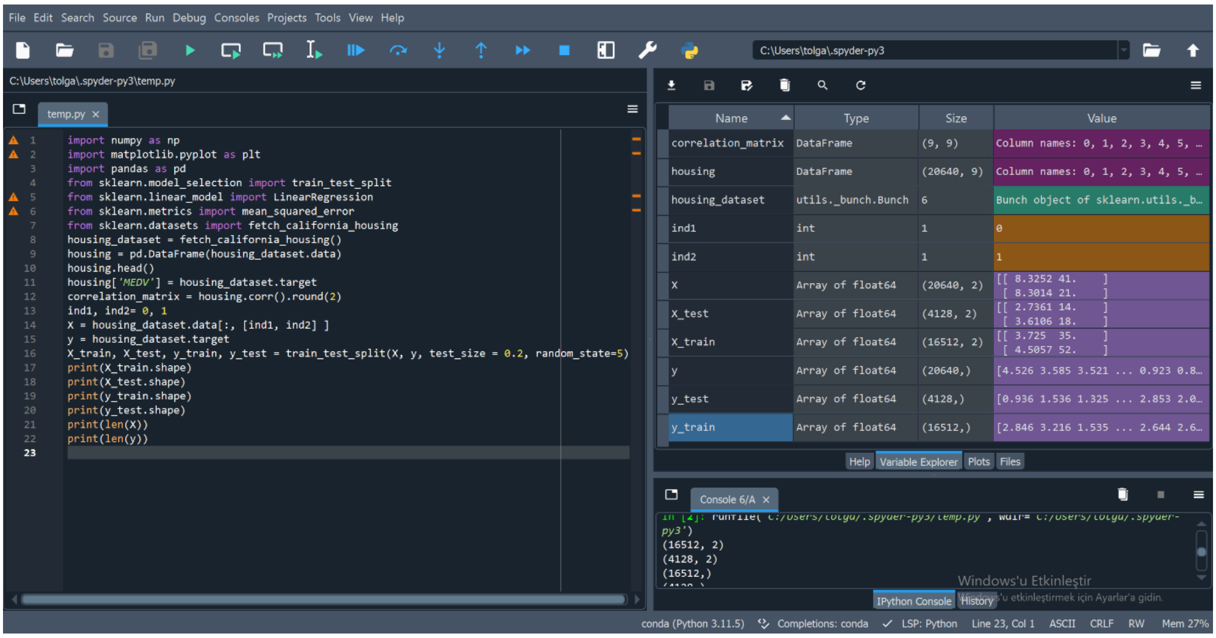
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
print(len(X))
print(len(y))
```

```
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.15.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
20640
20640

In [2]:
```

The reason why the lengths of x and y are equal is that for each sample a vector (X) and a corresponding output value (y) are collected. It is represented by a row, a house or a block. Hence the lengths of x and y vary in data settings.

In this case, the lengths of x and y must be the same because both data are derived from the same sample set. This means you can have accurate input and output data to train and evaluate machine development models. If the lengths match, there will be a mismatch and problems may arise during model training or evaluation. Therefore, it is important that the lengths of x and y are equal.

1.6)

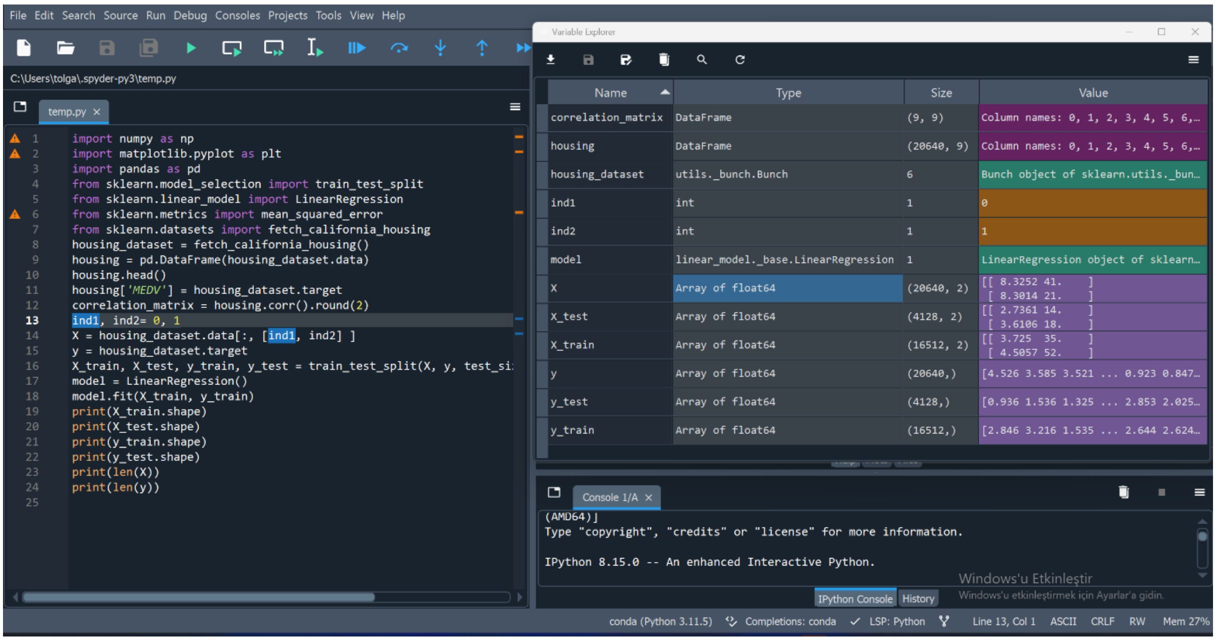


```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=5)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
print(len(X))
print(len(y))
```

Name	Type	Size	Value
correlation_matrix	DataFrame	(9, 9)	Column names: 0, 1, 2, 3, 4, 5, …
housing	DataFrame	(20640, 9)	Column names: 0, 1, 2, 3, 4, 5, …
housing_dataset	utils._bunch.Bunch	6	Bunch object of sklearn.utils._b…
ind1	int	1	0
ind2	int	1	1
X	Array of float64	(20640, 2)	[[8.3252 41.] [8.3014 21.]
X_test	Array of float64	(4128, 2)	[[2.7361 14.] [3.6106 18.]
X_train	Array of float64	(16512, 2)	[[3.725 35.] [4.5057 52.]
y	Array of float64	(20640,)	[4.526 3.585 3.521 ... 0.923 0.8…
y_test	Array of float64	(4128,)	[0.936 1.536 1.325 ... 2.853 2.0…
y_train	Array of float64	(16512,)	[2.846 3.216 1.535 ... 2.644 2.6…

```
In [2]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/tolga/.spyder-py3')
(16512, 2)
(4128, 2)
(16512,)
(4128,)
```

1.7)

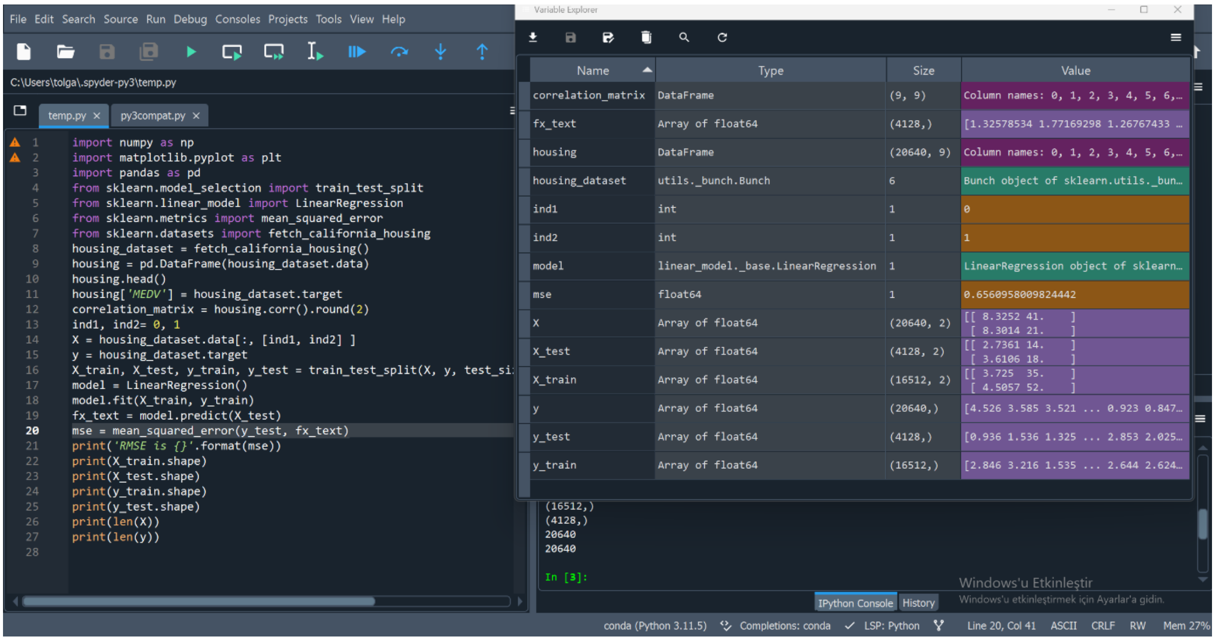


```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_si
model = LinearRegression()
model.fit(X_train, y_train)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
print(len(X))
print(len(y))
```

Name	Type	Size	Value
correlation_matrix	DataFrame	(9, 9)	Column names: 0, 1, 2, 3, 4, 5, 6,…
housing	DataFrame	(20640, 9)	Column names: 0, 1, 2, 3, 4, 5, 6,…
housing_dataset	utils._bunch.Bunch	6	Bunch object of sklearn.utils._bun…
ind1	int	1	0
ind2	int	1	1
model	linear_model._base.LinearRegression	1	LinearRegression object of sklearn…
X	Array of float64	(20640, 2)	[[8.3252 41.] [8.3014 21.]
X_test	Array of float64	(4128, 2)	[[2.7361 14.] [3.6106 18.]
X_train	Array of float64	(16512, 2)	[[3.725 35.] [4.5057 52.]
y	Array of float64	(20640,)	[4.526 3.585 3.521 ... 0.923 0.847…
y_test	Array of float64	(4128,)	[0.936 1.536 1.325 ... 2.853 2.025…
y_train	Array of float64	(16512,)	[2.846 3.216 1.535 ... 2.644 2.624…

```
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.15.0 -- An enhanced Interactive Python.
```

1.8)

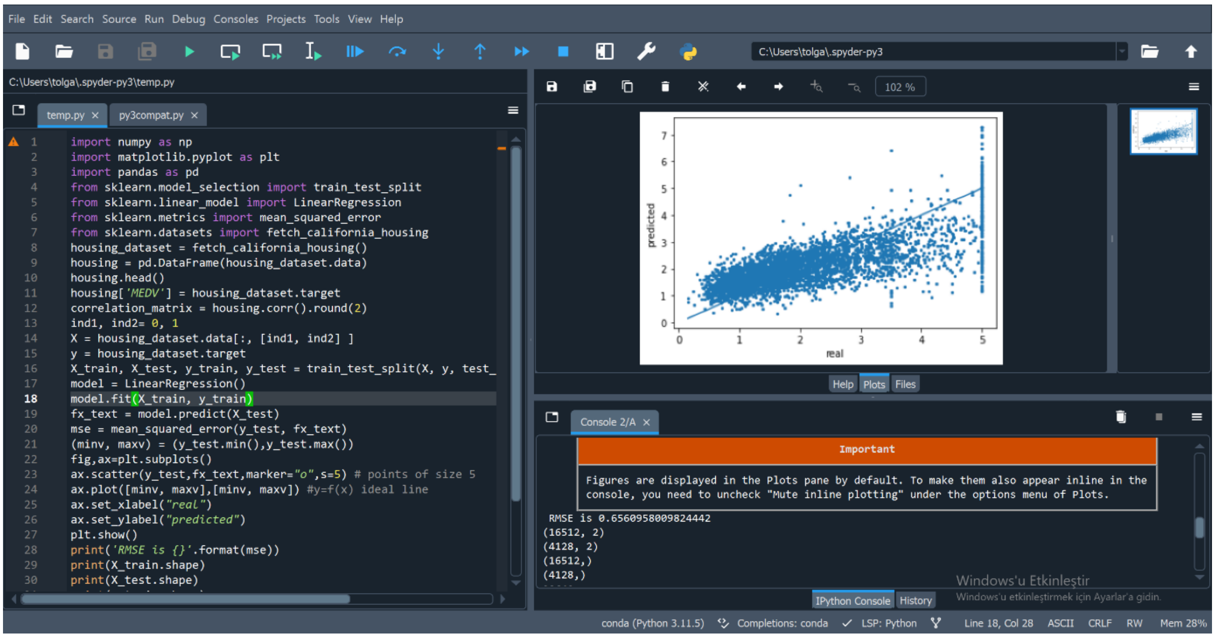


```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_si
model = LinearRegression()
model.fit(X_train, y_train)
fx_text = model.predict(X_test)
mse = mean_squared_error(y_test, fx_text)
print('RMSE is {}'.format(mse))
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
print(len(X))
print(len(y))
```

Name	Type	Size	Value
correlation_matrix	DataFrame	(9, 9)	Column names: 0, 1, 2, 3, 4, 5, 6,…
fx_text	Array of float64	(4128,)	[1.32578534 1.77169298 1.26767433 …
housing	DataFrame	(20640, 9)	Column names: 0, 1, 2, 3, 4, 5, 6,…
housing_dataset	utils._bunch.Bunch	6	Bunch object of sklearn.utils._bun…
ind1	int	1	0
ind2	int	1	1
model	linear_model._base.LinearRegression	1	LinearRegression object of sklearn…
mse	float64	1	0.6560958009824442
X	Array of float64	(20640, 2)	[[8.3252 41.] [8.3014 21.]
X_test	Array of float64	(4128, 2)	[[2.7361 14.] [3.6106 18.]
X_train	Array of float64	(16512, 2)	[[3.725 35.] [4.5057 52.]
y	Array of float64	(20640,)	[4.526 3.585 3.521 ... 0.923 0.847…
y_test	Array of float64	(4128,)	[0.936 1.536 1.325 ... 2.853 2.025…
y_train	Array of float64	(16512,)	[2.846 3.216 1.535 ... 2.644 2.624…

```
(16512,)
(4128,)
20640
20640

In [3]:
```

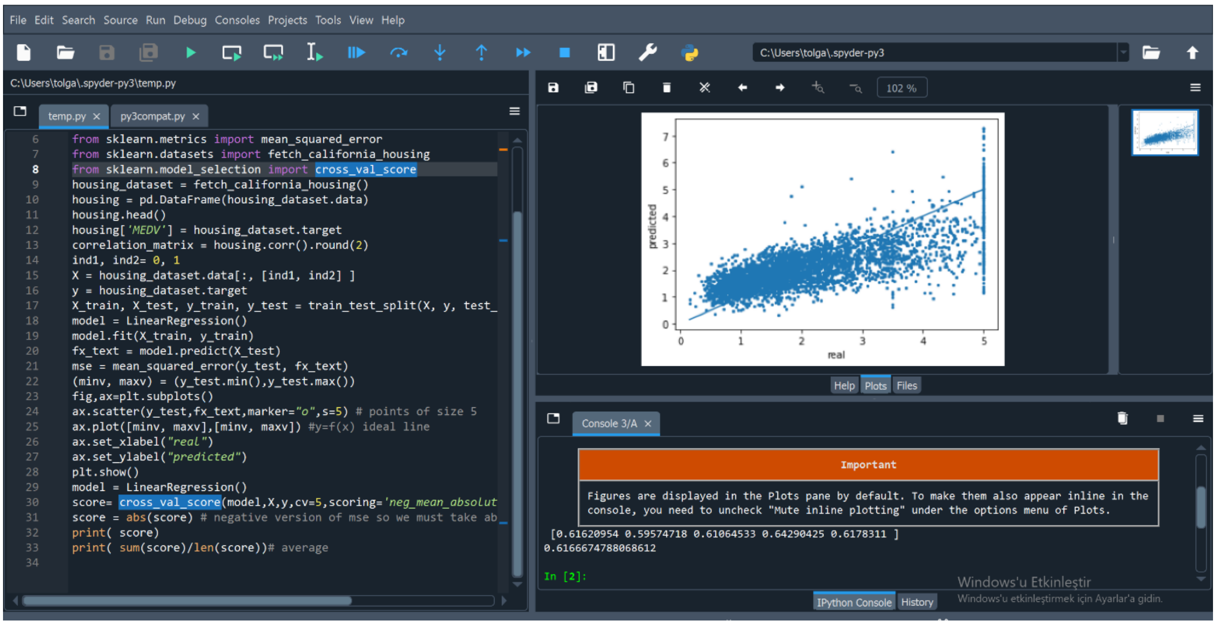
1.9)



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_
model = LinearRegression()
model.fit(X_train, y_train)
fx_text = model.predict(X_test)
mse = mean_squared_error(y_test, fx_text)
(minv, maxv) = (y_test.min(),y_test.max())
fig,ax=plt.subplots()
ax.scatter(y_test,fx_text,marker="o",s=5) # points of size 5
ax.plot([minv, maxv],[minv, maxv]) #y=f(x) ideal line
ax.set_xlabel("real")
ax.set_ylabel("predicted")
plt.show()
print('RMSE is {}'.format(mse))
print(X_train.shape)
print(X_test.shape)
```

```
                              Important

 Figures are displayed in the Plots pane by default. To make them also appear inline in the
 console, you need to uncheck "Mute inline plotting" under the options menu of Plots.

 RMSE is 0.6560958009824442
(16512, 2)
(4128, 2)
(16512,)
(4128,)
```

2.1)


File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\tolga\.spyder-py3
C:\Users\tolga\.spyder-py3\temp.py
temp.py
py3compat.py
102 %
predicted
real
0
1
2
3
4
5
6
7
Help
Plots
Files
Console 3/A
IPython Console
History
Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

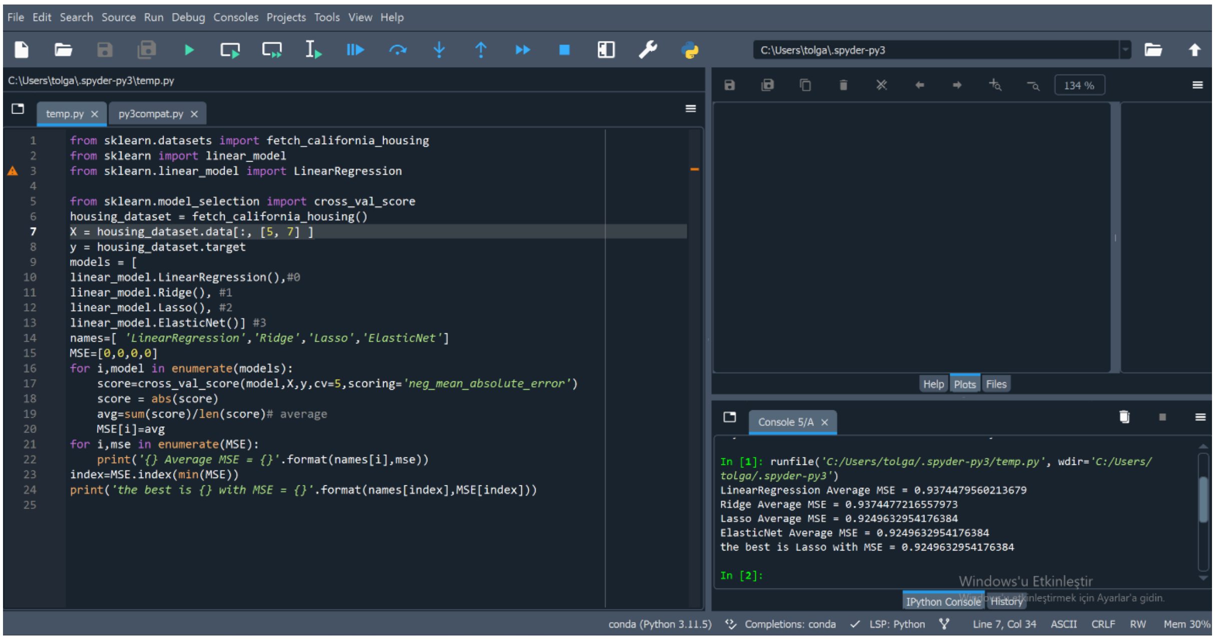
```python
from sklearn.metrics import mean_squared_error
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import cross_val_score
housing_dataset = fetch_california_housing()
housing = pd.DataFrame(housing_dataset.data)
housing.head()
housing['MEDV'] = housing_dataset.target
correlation_matrix = housing.corr().round(2)
ind1, ind2= 0, 1
X = housing_dataset.data[:, [ind1, ind2] ]
y = housing_dataset.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_
model = LinearRegression()
model.fit(X_train, y_train)
fx_text = model.predict(X_test)
mse = mean_squared_error(y_test, fx_text)
(minv, maxv) = (y_test.min(),y_test.max())
fig,ax=plt.subplots()
ax.scatter(y_test,fx_text,marker="o",s=5) # points of size 5
ax.plot([minv, maxv],[minv, maxv]) #y=f(x) ideal line
ax.set_xlabel("real")
ax.set_ylabel("predicted")
plt.show()
model = LinearRegression()
score= cross_val_score(model,X,y,cv=5,scoring='neg_mean_absolut
score = abs(score) # negative version of mse so we must take ab
print( score)
print( sum(score)/len(score))# average
```

```
Important

Figures are displayed in the Plots pane by default. To make them also appear inline in the
console, you need to uncheck "Mute inline plotting" under the options menu of Plots.
 [0.61620954 0.59574718 0.61064533 0.64290425 0.6178311 ]
0.6166674788068612

In [2]:
```

If the average cross-validation MSE is close to the validation set MSE, it suggests that the model is generalizing well to different subsets of the data.

If the average cross-validation MSE is significantly lower than the validation set MSE, it could indicate that the model is overfitting to the training set, and its performance doesn't generalize well to

2.2)



```python
from sklearn.datasets import fetch_california_housing
from sklearn import linear_model
from sklearn.linear_model import LinearRegression

from sklearn.model_selection import cross_val_score
housing_dataset = fetch_california_housing()
X = housing_dataset.data[:, [5, 7] ]
y = housing_dataset.target
models = [
linear_model.LinearRegression(),#0
linear_model.Ridge(), #1
linear_model.Lasso(), #2
linear_model.ElasticNet()] #3
names=[ 'LinearRegression','Ridge','Lasso','ElasticNet']
MSE=[0,0,0,0]
for i,model in enumerate(models):
    score=cross_val_score(model,X,y,cv=5,scoring='neg_mean_absolute_error')
    score = abs(score)
    avg=sum(score)/len(score)# average
    MSE[i]=avg
for i,mse in enumerate(MSE):
    print('{} Average MSE = {}'.format(names[i],mse))
index=MSE.index(min(MSE))
print('the best is {} with MSE = {}'.format(names[index],MSE[index]))
```

```
In [1]: runfile('C:/Users/tolga/.spyder-py3/temp.py', wdir='C:/Users/
tolga/.spyder-py3')
LinearRegression Average MSE = 0.9374479560213679
Ridge Average MSE = 0.9374477216557973
Lasso Average MSE = 0.9249632954176384
ElasticNet Average MSE = 0.9249632954176384
the best is Lasso with MSE = 0.9249632954176384

In [2]:
```