Practical 4

Tutorial

In the 4th Practical we are going to study linear regression model using the Boston dataset.

Step:

1. **For this practical we will not be needing a dataset as the Boston dataset will be already provided in the sklearn library in python.**
2. **First open anaconda and launch spyder**
3. **Import the following libraries:**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

4. **Create a Dataframe with Dependent Variable(x) and independent variable y.**

```
x=np.array([95,85,80,70,60])

y=np.array([85,95,70,65,70])
```

5. **Create Linear Regression Model using Polyfit Function:**

```
model= np.polyfit(x, y, 1)

model
```

6. **Predict the Y value for X and observe the output.**

```
predict = np.poly1d(model)
predict(65)
```

7. **Predict the y_pred for all values of x.**

```
y_pred= predict(x)
y_pred
```

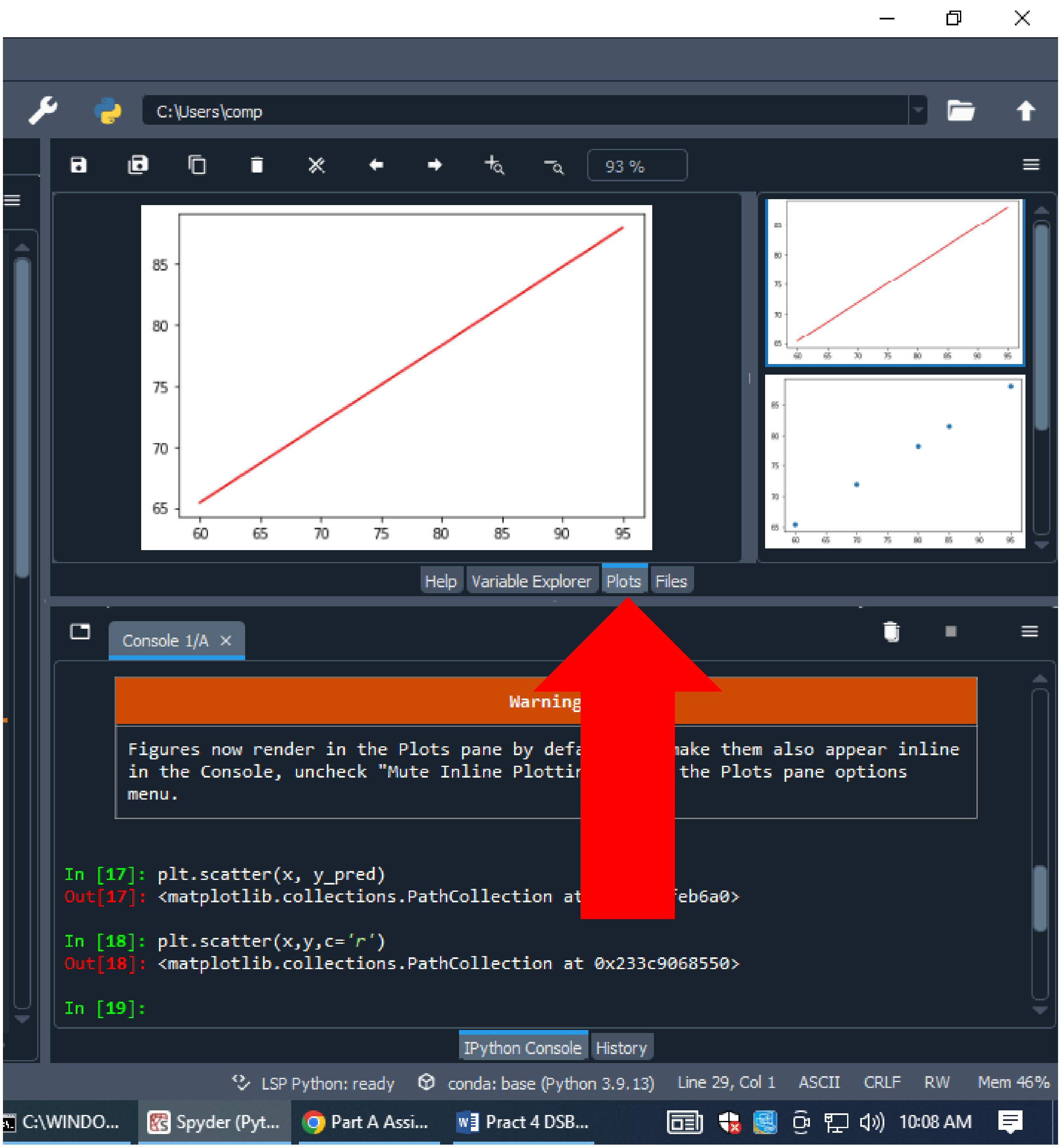
8. **Evaluate the performance of Model (R-Suare) R squared calculation is not implemented in numpy... so that one should be borrowed :**

```
from sklearn.

from sklearn.metrics import r2_score

r2_score(y, y_pred)
```

9. **Now plotting the regression model:**

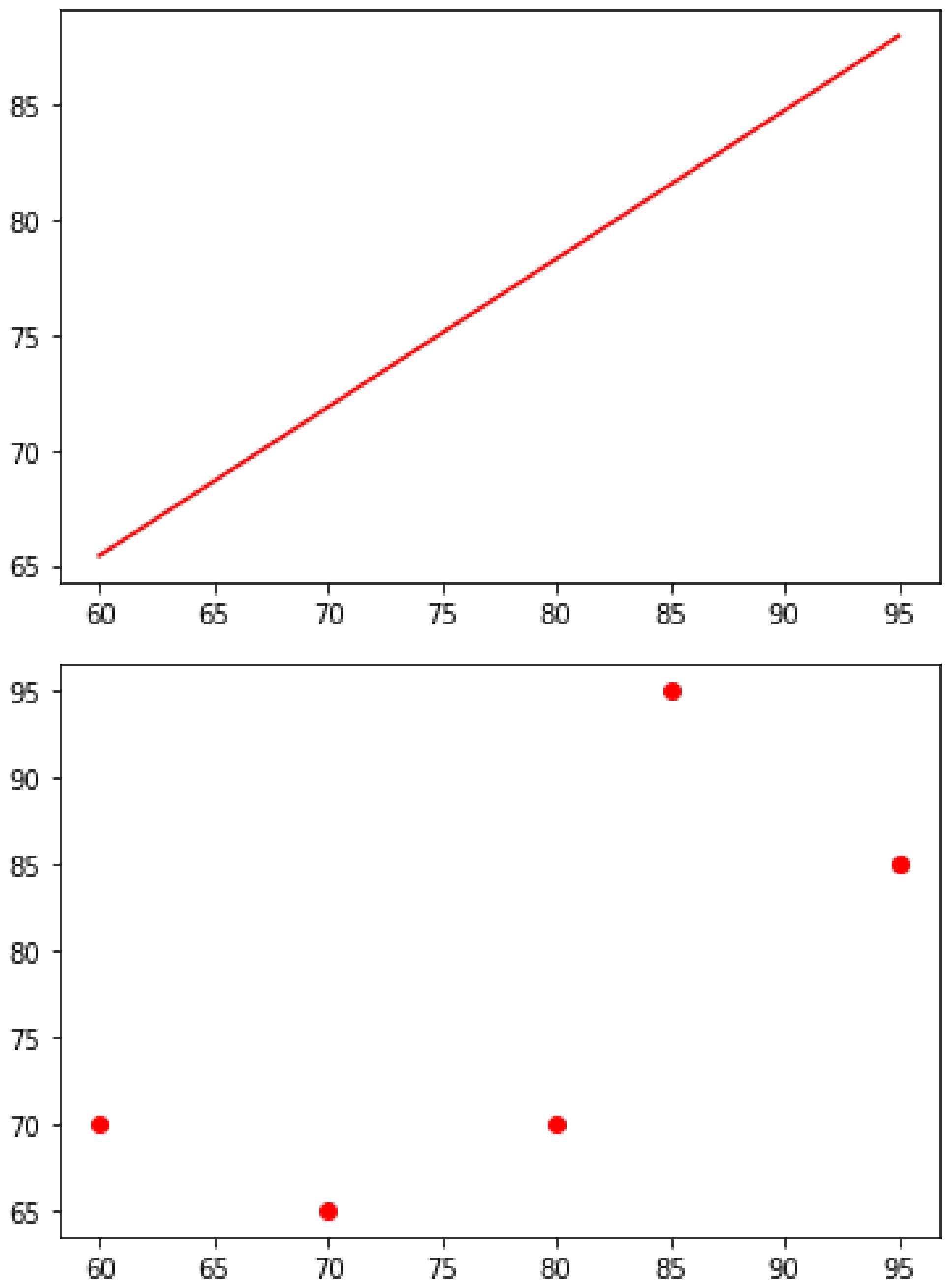
```
y_line = model[1] + model[0]* x
plt.plot(x, y_line, c = 'r')
plt.scatter(x, y_pred)
plt.scatter(x,y,c='r')
```

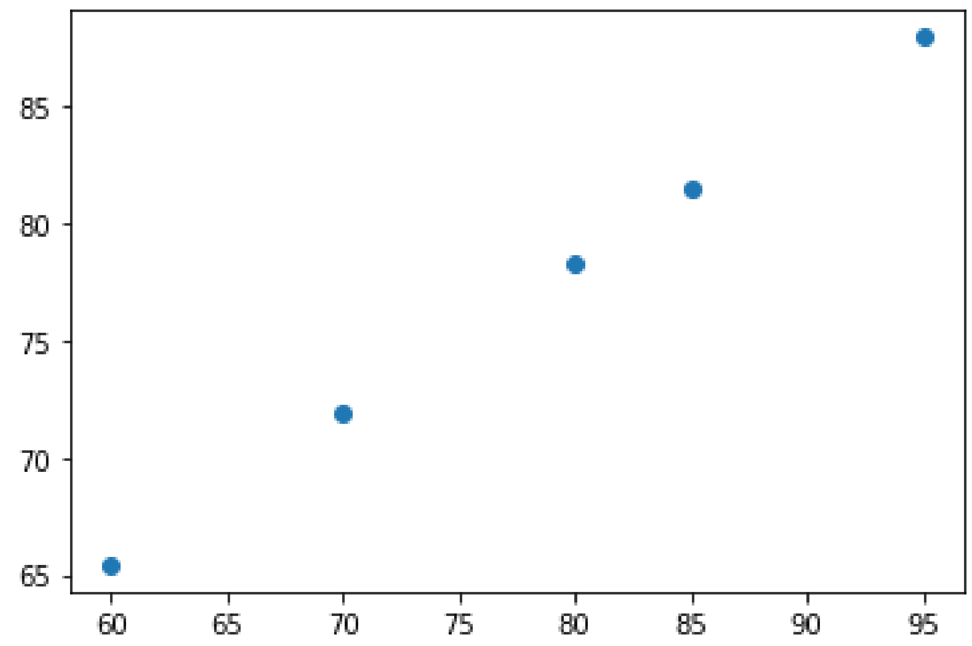
the output will be in the plots section in the top right hand side of the spyder GUI. That is above the console.



```
Warning

Figures now render in the Plots pane by defa      make them also appear inline
in the Console, uncheck "Mute Inline Plottir      the Plots pane options
menu.


In [17]: plt.scatter(x, y_pred)
Out[17]: <matplotlib.collections.PathCollection at         feb6a0>

In [18]: plt.scatter(x,y,c='r')
Out[18]: <matplotlib.collections.PathCollection at 0x233c9068550>

In [19]:
```

Output:


85
80
75
70
65
60
65
70
75
80
85
90
95
95
90
85
80
75
70
65
60
65
70
75
80
85
90
95



We will now move on to the Boston dataset
Steps:

1. **Import the Boston Housing dataset**

```
from sklearn.datasets import load_boston
boston = load_boston()
```

2. **Initialize the data frame**

```
data = pd.DataFrame(boston.data)
```

3. **Add the feature names to the dataframe**

```
data.columns = boston.feature_names
data.head()
```

4. **Adding target variable to dataframe**

```
data['PRICE'] = boston.target
```

5. **Perform Data Preprocessing(Check for missing values)**

```
data.isnull().sum()
```

6. Split dependent variable and independent variables

```
x = data.drop(['PRICE'], axis = 1)
y = data['PRICE']
```

7. splitting data to training and testing dataset.

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest =
train_test_split(x, y, test_size =0.2,random_state = 0)
```

8. Use linear regression(Train the Machine) to Create Model

```
import sklearn
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model=lm.fit(xtrain, ytrain)
```

9. Predict the y_pred for all values of train_x and test_x

```
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)
```

10. Evaluate the performance of Model for train_y and test_y

```
df=pd.DataFrame(ytrain_pred,ytrain)
df=pd.DataFrame(ytest_pred,ytest)
```

11. Calculate Mean Square Paper for train_y and test_y

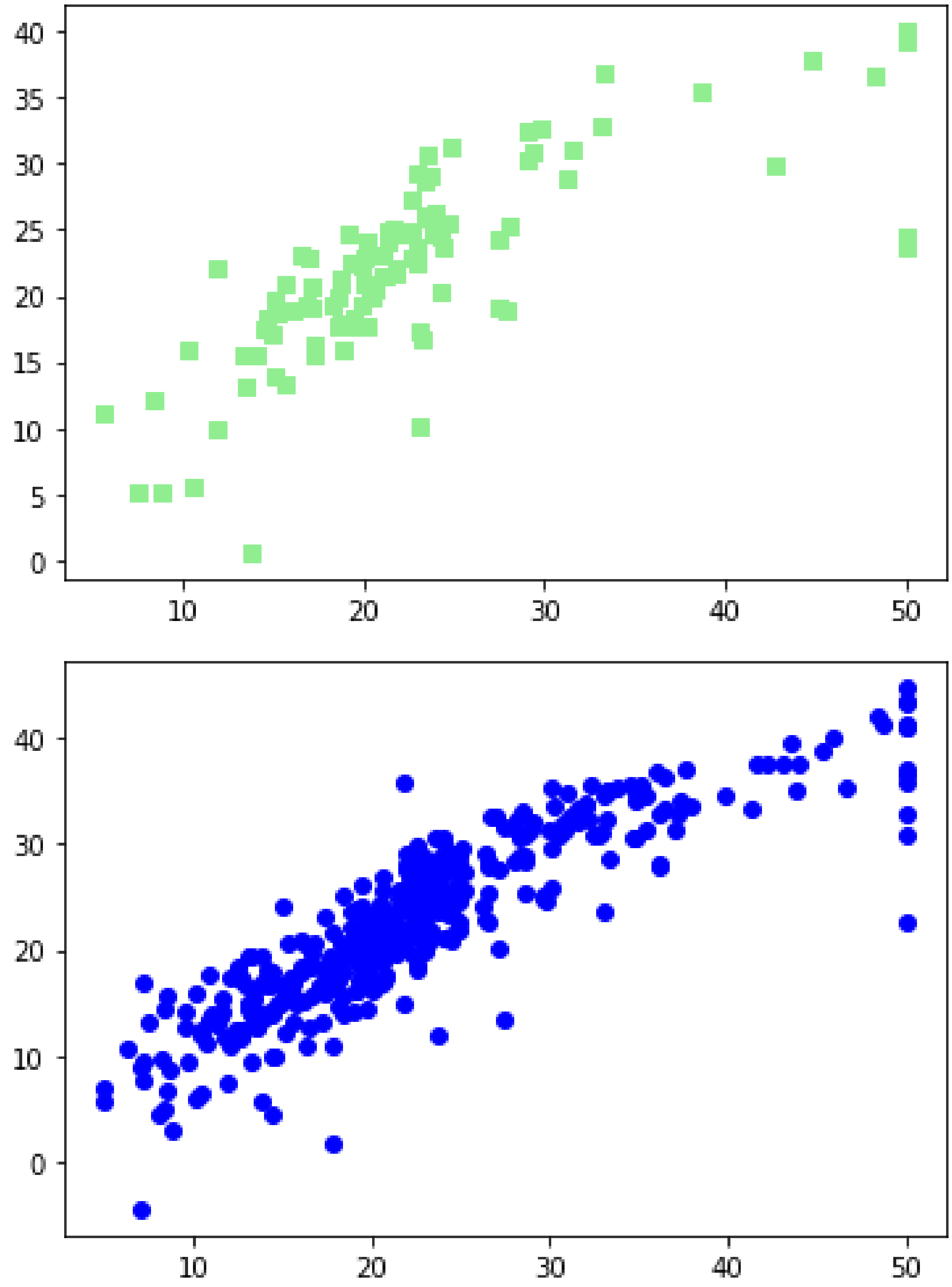
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
mse = mean_squared_error(ytrain_pred,ytrain)
print(mse)
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
```

12. Plotting the linear regression model

```
lt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Training data')
plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Test data')
plt.xlabel('True values')
plt.ylabel('Predicted')
plt.title("True value vs Predicted value")
plt.legend(loc= 'upper left')
#plt.hlines(y=0,xmin=0,xmax=50)
plt.plot()
plt.show()
```

The output will be two graph plots : (given on the next page)

Finally save the file and create another text document including the final code and the output.


40
35
30
25
20
15
10
5
0
10
20
30
40
50
40
30
20
10
0
10
20
30
40
50