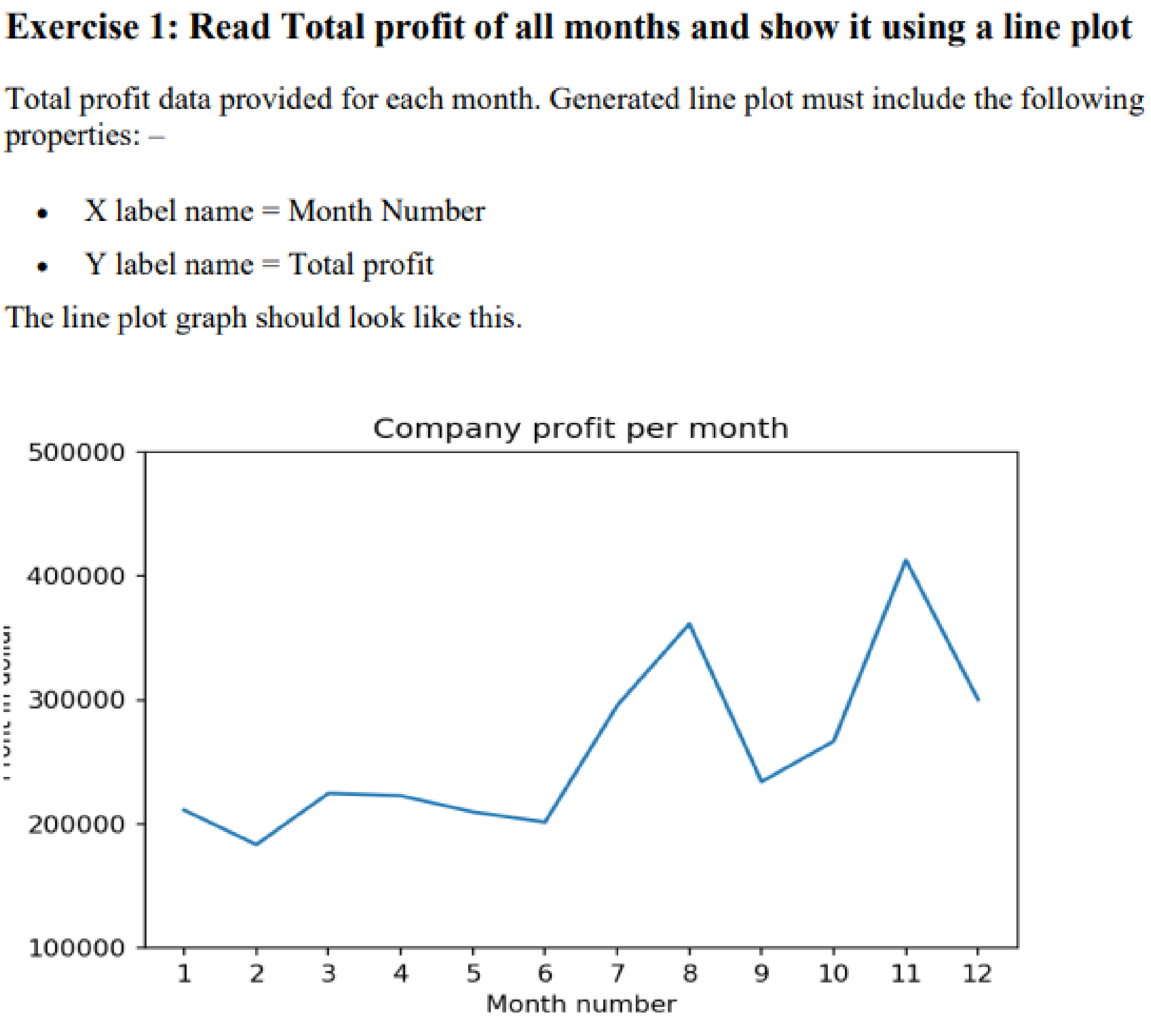
Name of Student : AHMED ALI ANSARI ID No : 1402-2020

Task :

1.



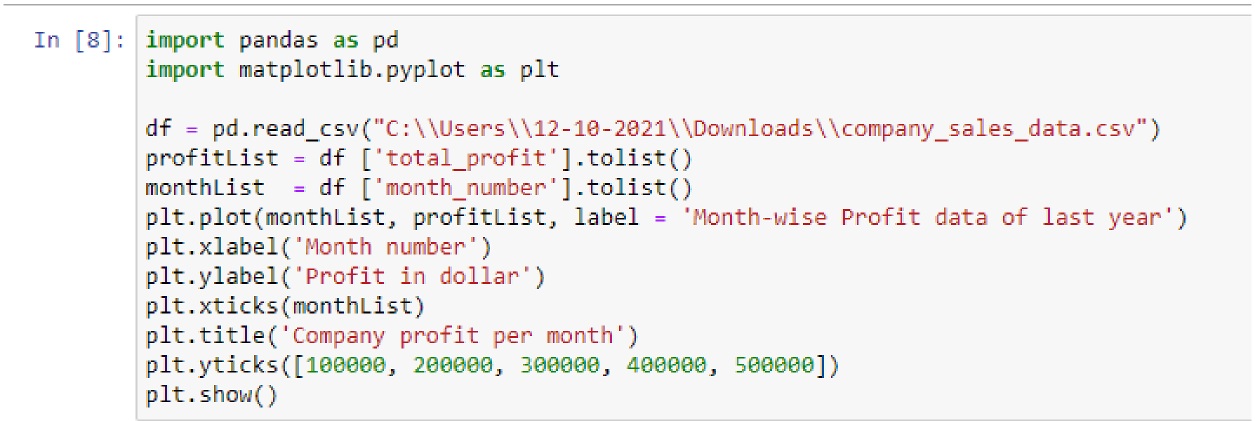
Exercise 1: Read Total profit of all months and show it using a line plot

Total profit data provided for each month. Generated line plot must include the following properties: –

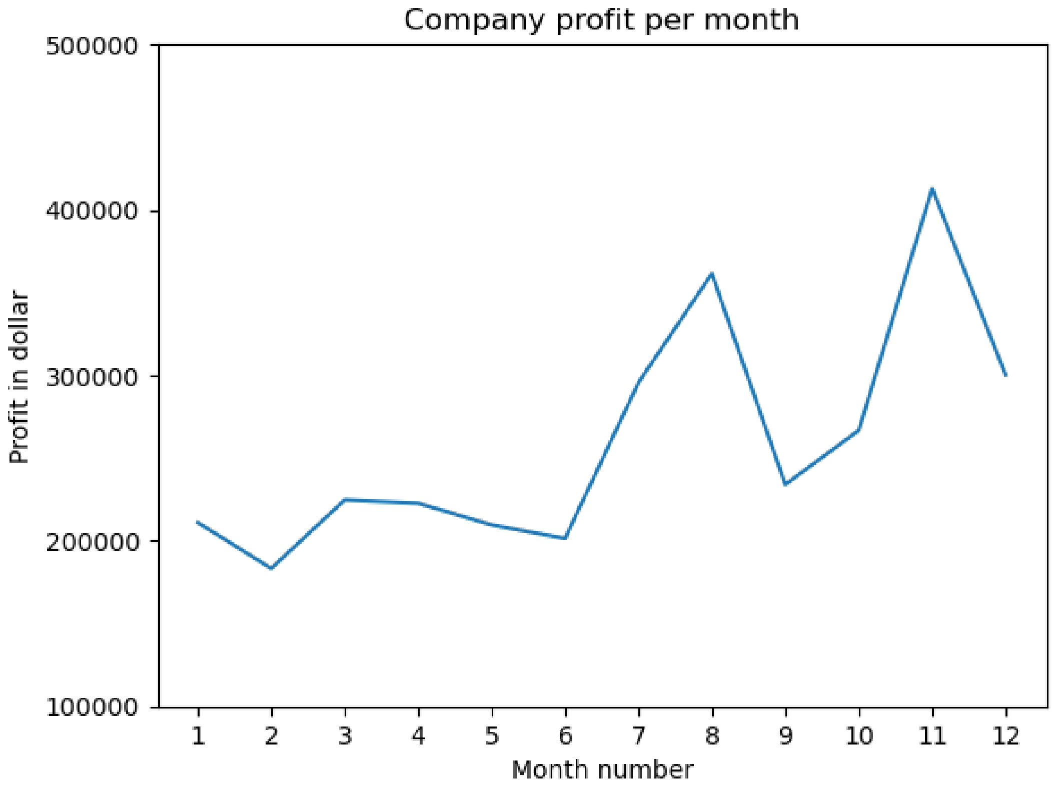
- X label name = Month Number
- Y label name = Total profit

The line plot graph should look like this.

Answer :

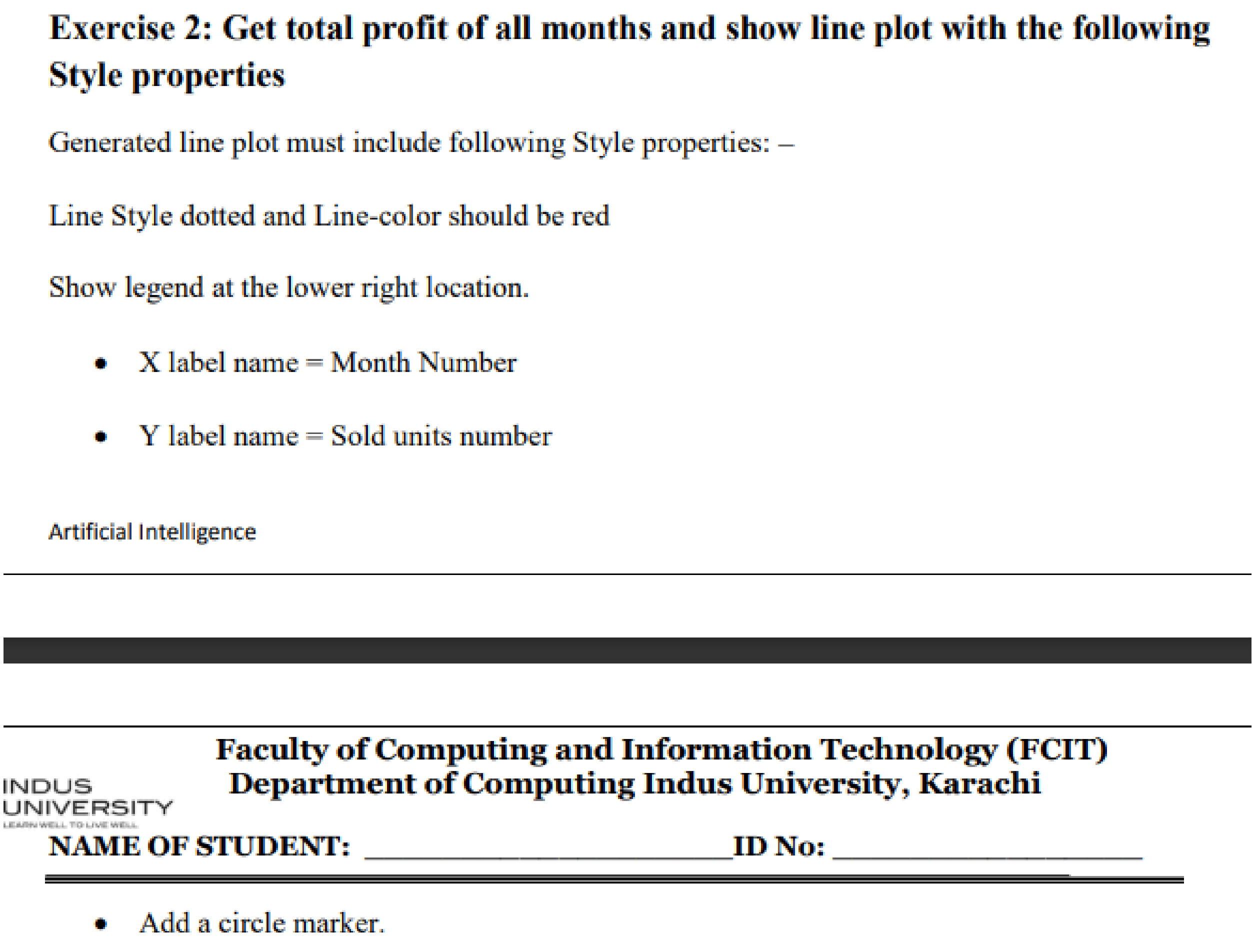


```
In [8]: import pandas as pd
        import matplotlib.pyplot as plt

        df = pd.read_csv("C:\\Users\\12-10-2021\\Downloads\\company_sales_data.csv")
        profitList = df ['total_profit'].tolist()
        monthList  = df ['month_number'].tolist()
        plt.plot(monthList, profitList, label = 'Month-wise Profit data of last year')
        plt.xlabel('Month number')
        plt.ylabel('Profit in dollar')
        plt.xticks(monthList)
        plt.title('Company profit per month')
        plt.yticks([100000, 200000, 300000, 400000, 500000])
        plt.show()
```



Name of Student : AHMED ALI ANSARI ID No : 1402-2020

2.



Exercise 2: Get total profit of all months and show line plot with the following Style properties

Generated line plot must include following Style properties: –

Line Style dotted and Line-color should be red

Show legend at the lower right location.

- X label name = Month Number
- Y label name = Sold units number

Artificial Intelligence

INDUS UNIVERSITY LEARN WELL TO LIVE WELL

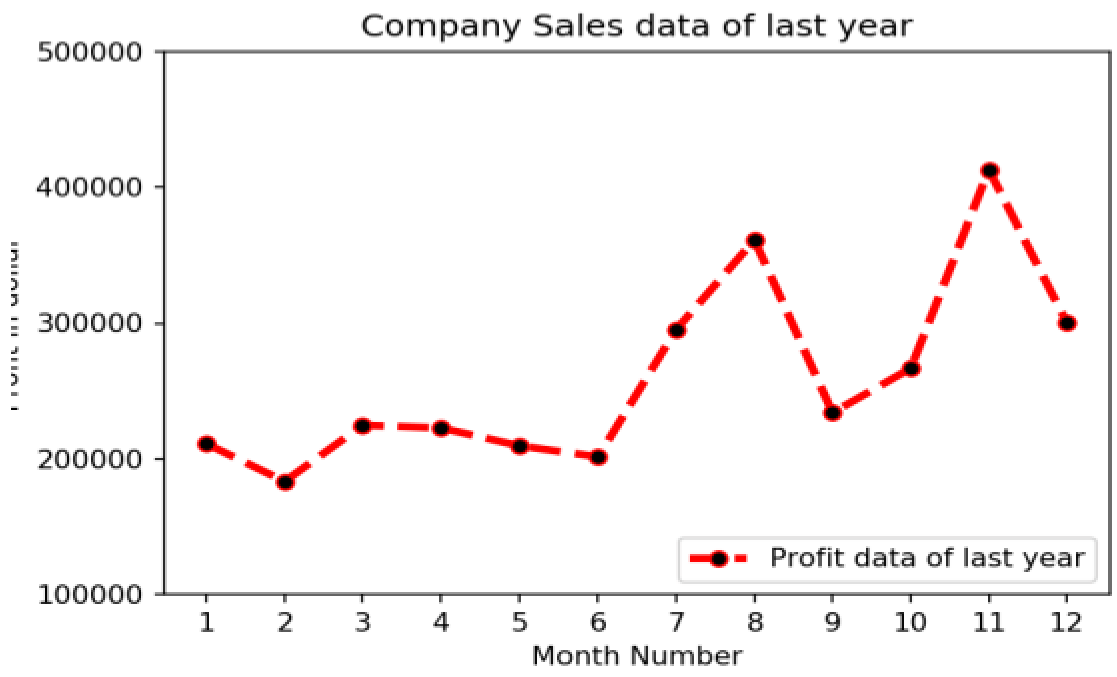
Faculty of Computing and Information Technology (FCIT) Department of Computing Indus University, Karachi

NAME OF STUDENT: ________________________ID No: ____________________

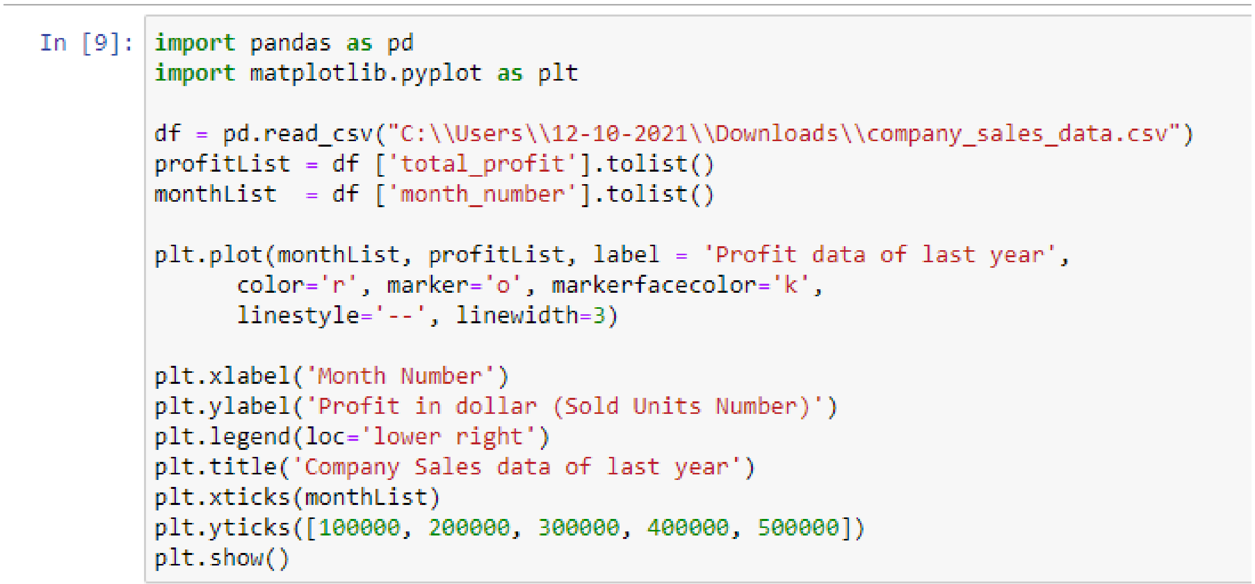
- Add a circle marker.

- Line marker color as read
- Line width should be 3

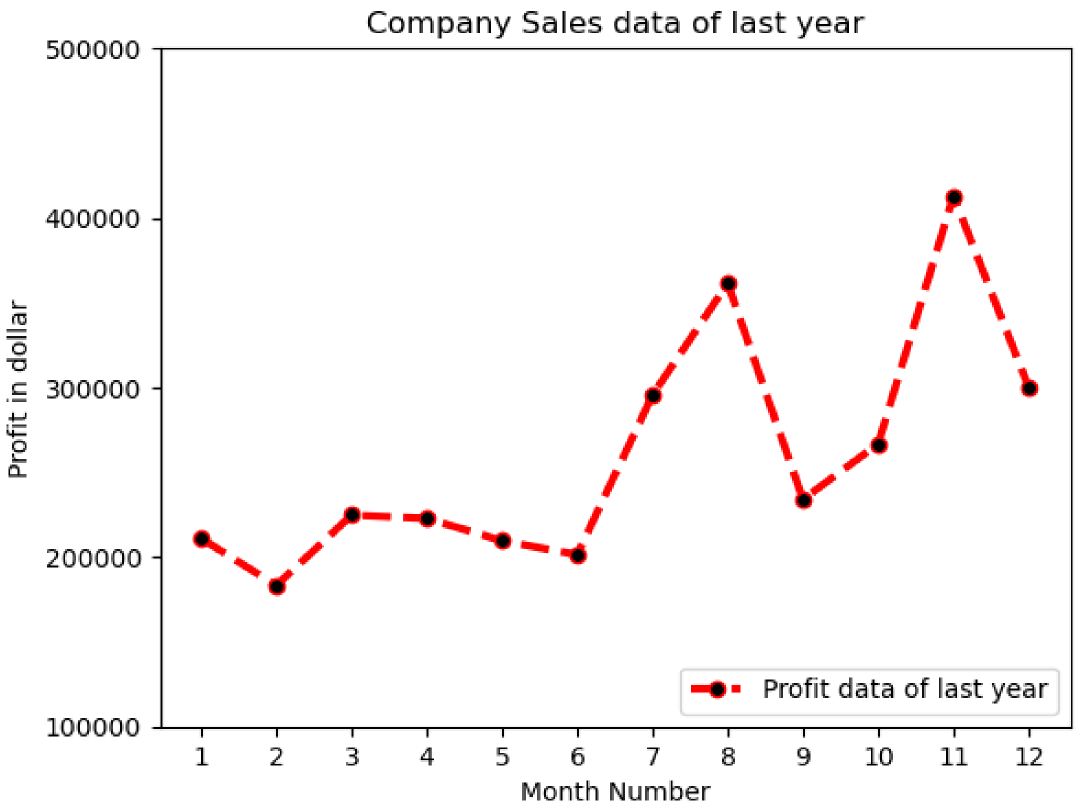
The line plot graph should look like this.



Answer :

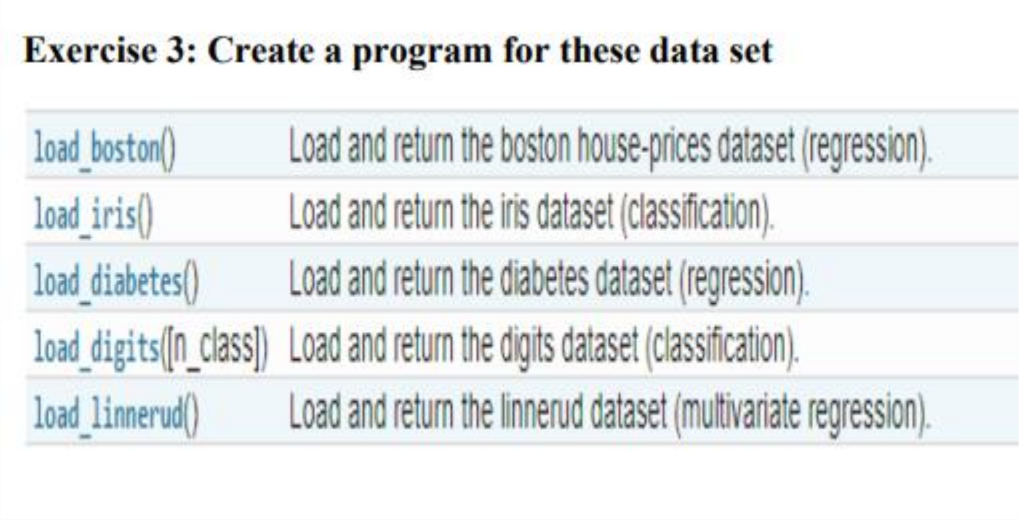


In [9]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("C:\\Users\\12-10-2021\\Downloads\\company_sales_data.csv")
profitList = df ['total_profit'].tolist()
monthList  = df ['month_number'].tolist()

plt.plot(monthList, profitList, label = 'Profit data of last year',
      color='r', marker='o', markerfacecolor='k',
      linestyle='--', linewidth=3)

plt.xlabel('Month Number')
plt.ylabel('Profit in dollar (Sold Units Number)')
plt.legend(loc='lower right')
plt.title('Company Sales data of last year')
plt.xticks(monthList)
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



3.

Name of Student : AHMED ALI ANSARI **ID No : 1402-2020**

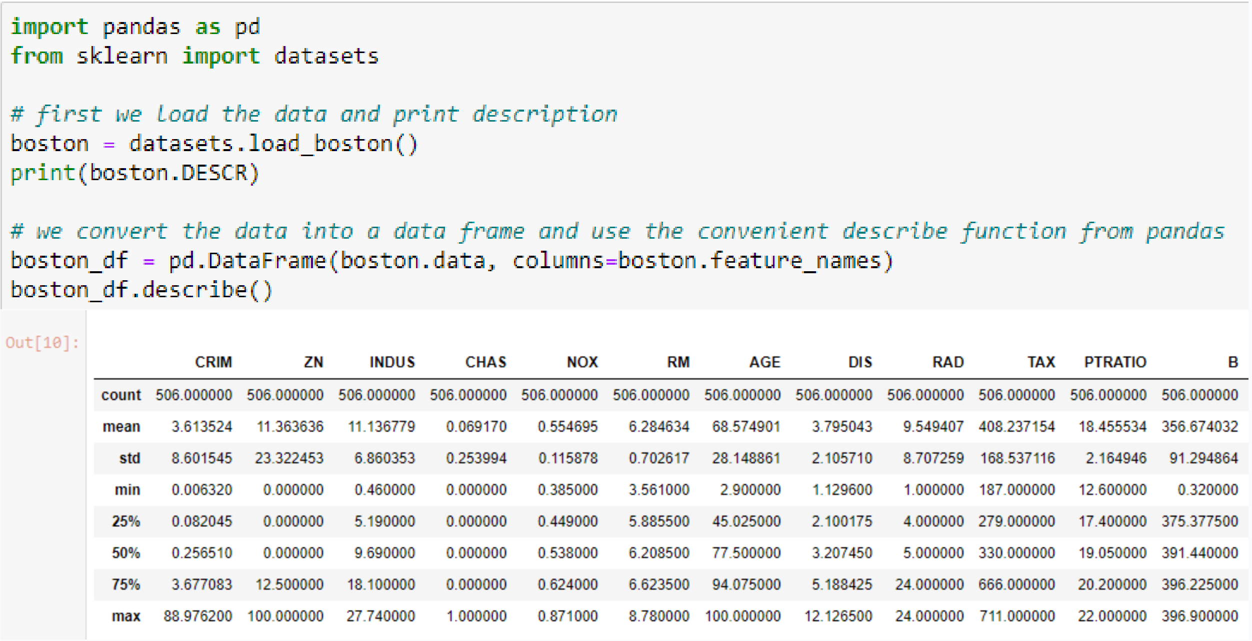


Exercise 3: Create a program for these data set

load_boston()	Load and return the boston house-prices dataset (regression).
load_iris()	Load and return the iris dataset (classification).
load_diabetes()	Load and return the diabetes dataset (regression).
load_digits([n_class])	Load and return the digits dataset (classification).
load_linnerud()	Load and return the linnerud dataset (multivariate regression).

Answer:

Load_boston()

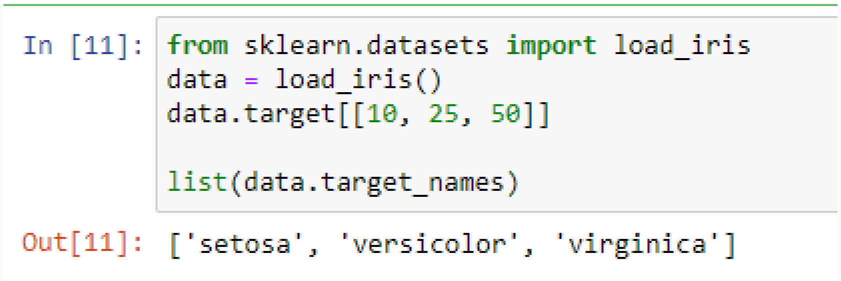


```
import pandas as pd
from sklearn import datasets

# first we load the data and print description
boston = datasets.load_boston()
print(boston.DESCR)

# we convert the data into a data frame and use the convenient describe function from pandas
boston_df = pd.DataFrame(boston.data, columns=boston.feature_names)
boston_df.describe()
```

Out[10]:

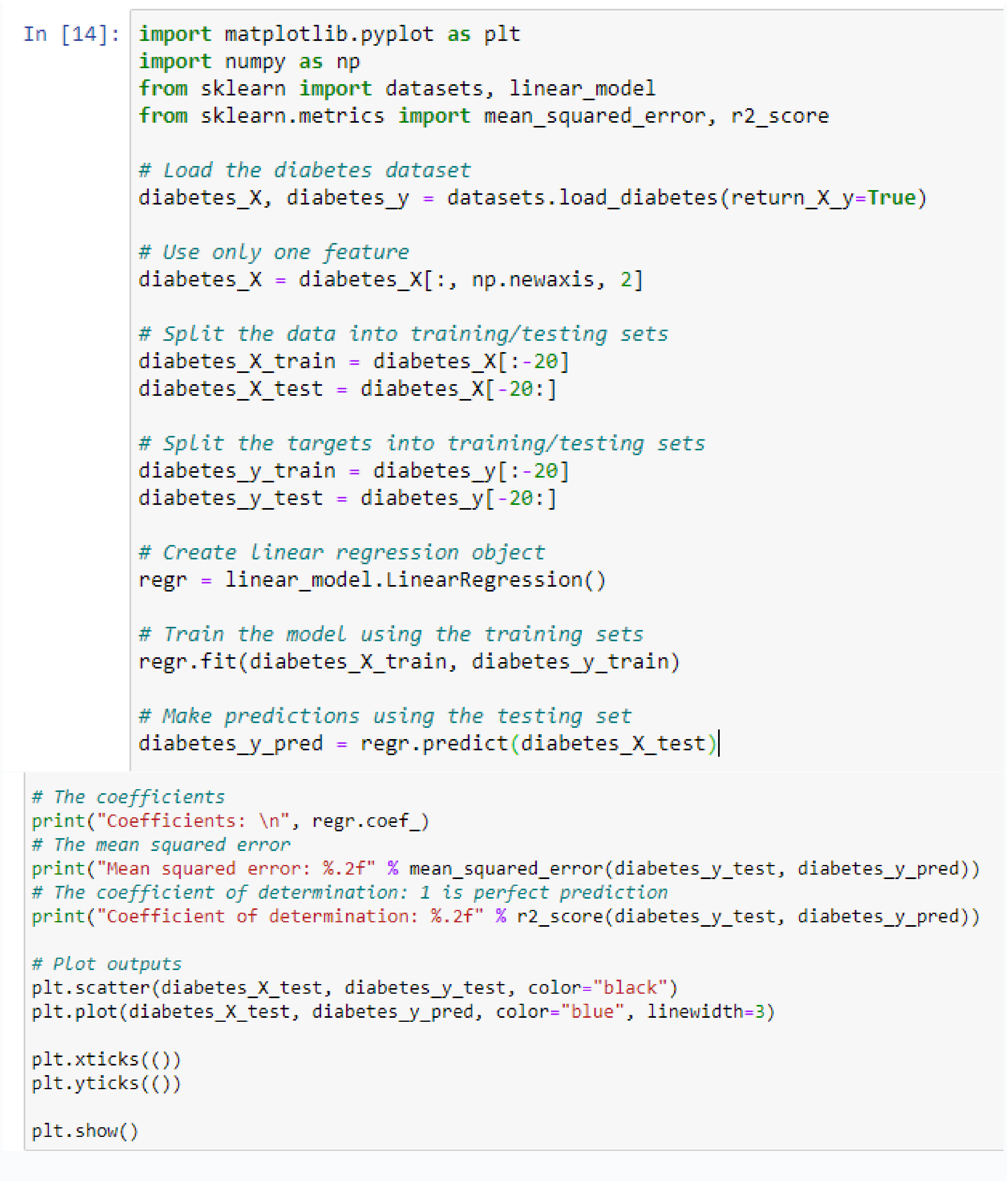
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000

Load_iris()



```
In [11]: from sklearn.datasets import load_iris
         data = load_iris()
         data.target[[10, 25, 50]]

         list(data.target_names)

Out[11]: ['setosa', 'versicolor', 'virginica']
```

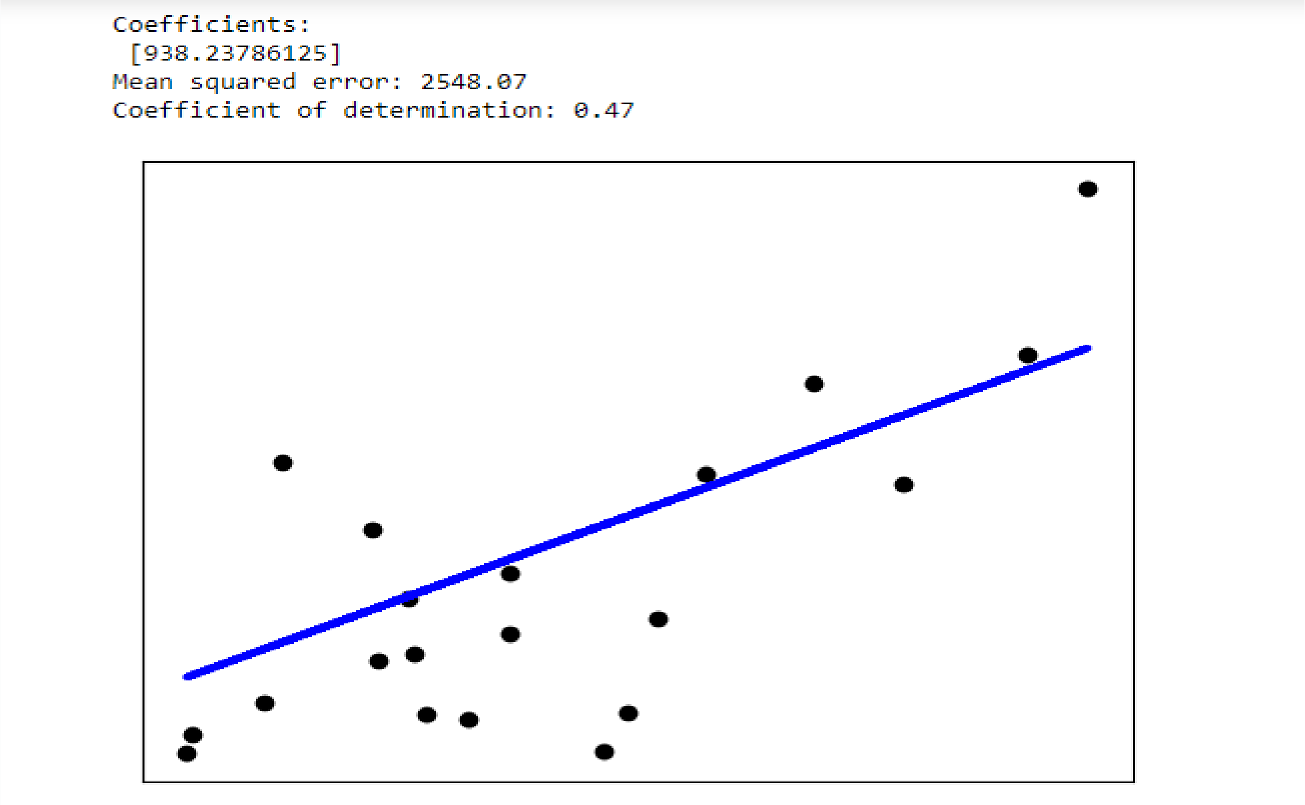
Load_diabetes()



In [14]:

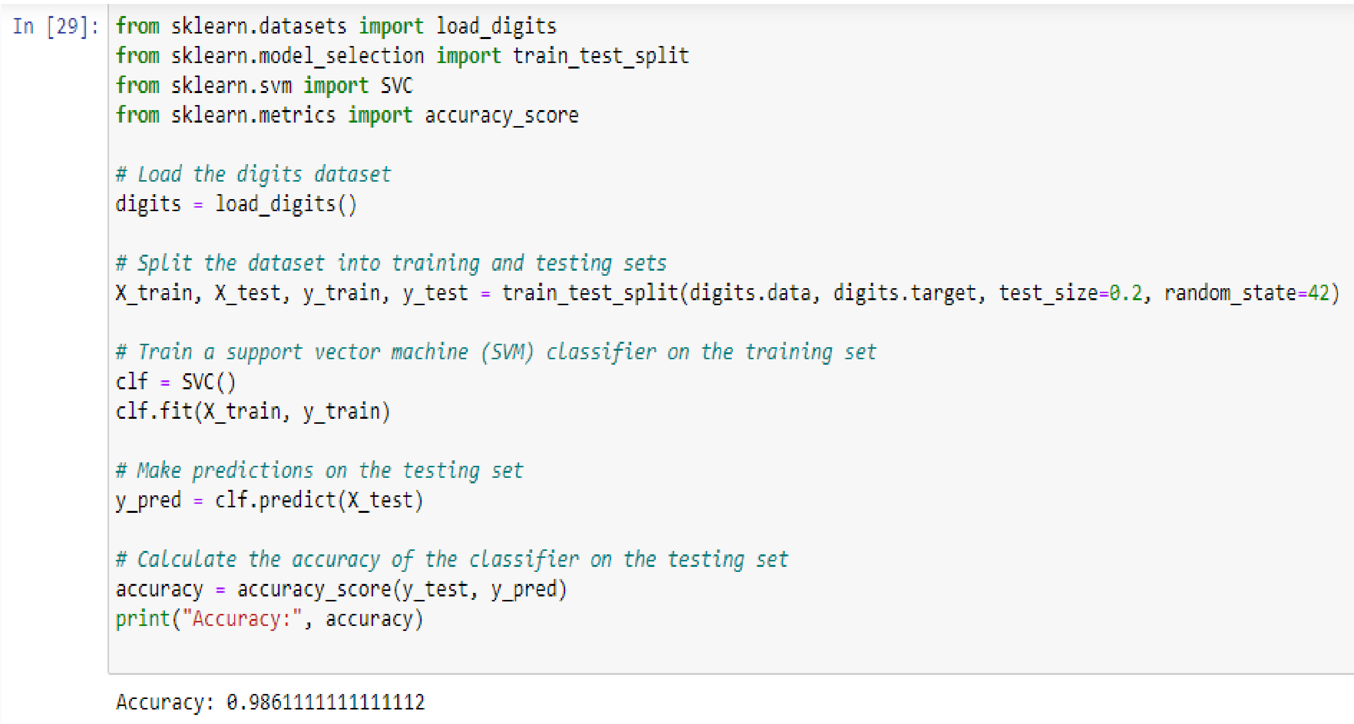
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

# Use only one feature
diabetes_X = diabetes_X[:, np.newaxis, 2]

# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

# Split the targets into training/testing sets
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]

# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)

# Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)
```

```
# The coefficients
print("Coefficients: \n", regr.coef_)
# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test, diabetes_y_pred))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination: %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))

# Plot outputs
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```



```
Coefficients:
 [938.23786125]
Mean squared error: 2548.07
Coefficient of determination: 0.47
```

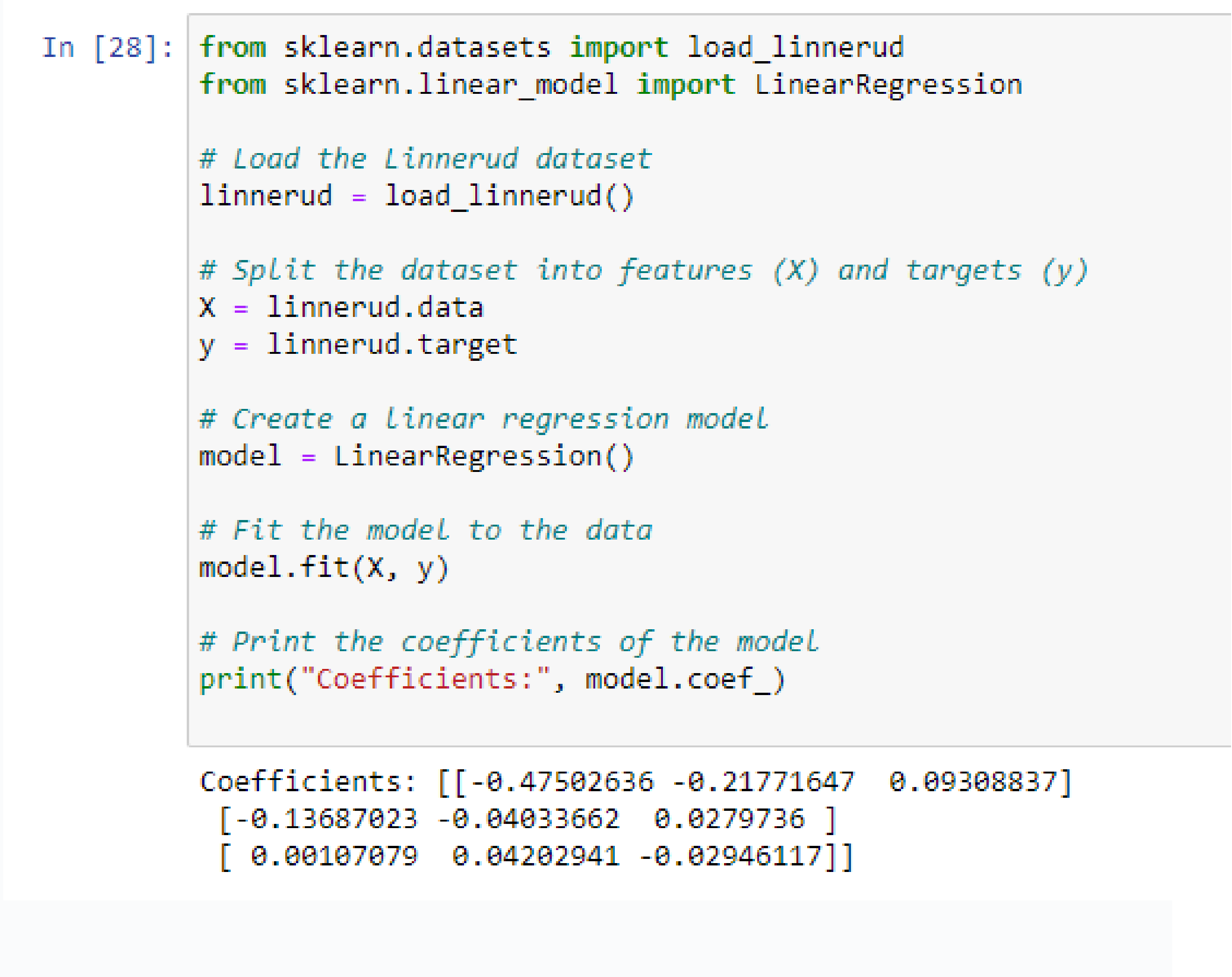
Load_digits()



```
In [29]: from sklearn.datasets import load_digits
         from sklearn.model_selection import train_test_split
         from sklearn.svm import SVC
         from sklearn.metrics import accuracy_score

         # Load the digits dataset
         digits = load_digits()

         # Split the dataset into training and testing sets
         X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.2, random_state=42)

         # Train a support vector machine (SVM) classifier on the training set
         clf = SVC()
         clf.fit(X_train, y_train)

         # Make predictions on the testing set
         y_pred = clf.predict(X_test)

         # Calculate the accuracy of the classifier on the testing set
         accuracy = accuracy_score(y_test, y_pred)
         print("Accuracy:", accuracy)

         Accuracy: 0.9861111111111112
```

Load_linnerud()

Name of Student : AHMED ALI ANSARI **ID No : 1402-2020**



```
In [28]: from sklearn.datasets import load_linnerud
         from sklearn.linear_model import LinearRegression

         # Load the Linnerud dataset
         linnerud = load_linnerud()

         # Split the dataset into features (X) and targets (y)
         X = linnerud.data
         y = linnerud.target

         # Create a linear regression model
         model = LinearRegression()

         # Fit the model to the data
         model.fit(X, y)

         # Print the coefficients of the model
         print("Coefficients:", model.coef_)
```

```
Coefficients: [[-0.47502636 -0.21771647  0.09308837]
 [-0.13687023 -0.04033662  0.0279736 ]
 [ 0.00107079  0.04202941 -0.02946117]]
```