Linear regression model using Iris dataset

Importing libraries

In [103]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
import matplotlib.pyplot as plt
```

In [104]:
```
from sklearn import datasets
```

Splitting the data into x array that contains the features to train on and y array that contains the target variable.

In [105]:
```
print(iris.DESCR)
```

```
Iris Plants Database
====================

Notes
-----
Data Set Characteristics:
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica
    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20  0.76     0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

This is a copy of UCI ML iris datasets.
http://archive.ics.uci.edu/ml/datasets/Iris (http://archive.ics.uci.edu/ml/data
sets/Iris)

The famous Iris database, first used by Sir R.A Fisher

This is perhaps the best known database to be found in the
pattern recognition literature.  Fisher's paper is a classic in the field and
is referenced frequently to this day.  (See Duda & Hart, for example.)  The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant.  One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

References
----------
   - Fisher,R.A. "The use of multiple measurements in taxonomic problems"
     Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
     Mathematical Statistics" (John Wiley, NY, 1950).
   - Duda,R.O., & Hart,P.E. (1973) Pattern Classification and Scene Analysis.
     (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.
   - Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
```

```
Structure and Classification Rule for Recognition in Partially Exposed
Environments".  IEEE Transactions on Pattern Analysis and Machine
Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transactions
on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS II
conceptual clustering system finds 3 classes in the data.
- Many, many more ...
```

In [106]:
```
iris.target
```

Out[106]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [107]:
```
y=iris.target
```

In [108]:
```
y
```

Out[108]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [109]:
```
iris.target_names
```

Out[109]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [110]:
```
x=iris.data
```

In [112]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=10
```

In [113]:
```
clf=LinearRegression()
```

In [114]:
```
clf.fit(x_train,y_train)
```

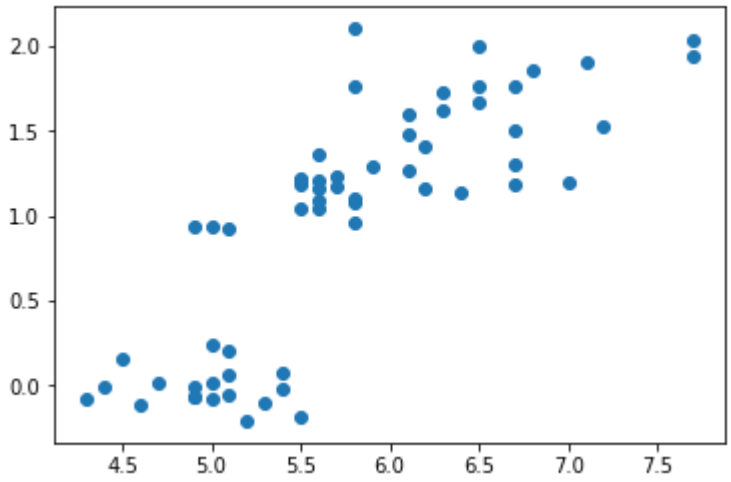
Out[114]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [115]:
```
y_pred=clf.predict(x_test)
```

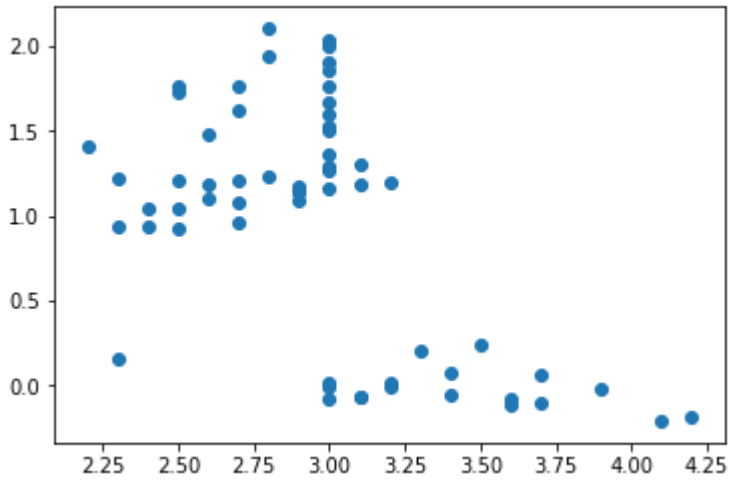
In [116]:
```
y_pred
```

Out[116]:
```
array([-0.18200549, -0.01736407,  0.24485245,  1.52859194,  1.2013094 ,
        1.62007926,  1.40751724,  1.22457535,  1.72881726, -0.00533044,
        1.76143721, -0.20336882,  0.07097943,  1.94228548,  1.60082788,
        1.14231285,  1.15557655,  1.1758841 , -0.00832428,  1.47873408,
        1.04058923, -0.095687  ,  1.18599163,  1.18503991,  1.15655135,
        1.08542227,  0.93588332,  2.09792664,  0.02007734,  0.15517401,
        1.99931409,  1.20224876,  1.66993196,  1.072153  ,  1.85901789,
        1.23583794,  1.30397835,  1.2907781 ,  1.20532995,  2.03282149,
        0.0588473 , -0.1160065 ,  0.01910851,  1.50422798,  1.3609264 ,
       -0.08155336,  1.90271419,  0.96438423, -0.07037812,  0.92779483,
        1.0385702 ,  0.20617541,  1.75835045, -0.07942946,  0.93183928,
        1.76030047,  1.09957981, -0.07037812, -0.04795861,  1.27144575])
```

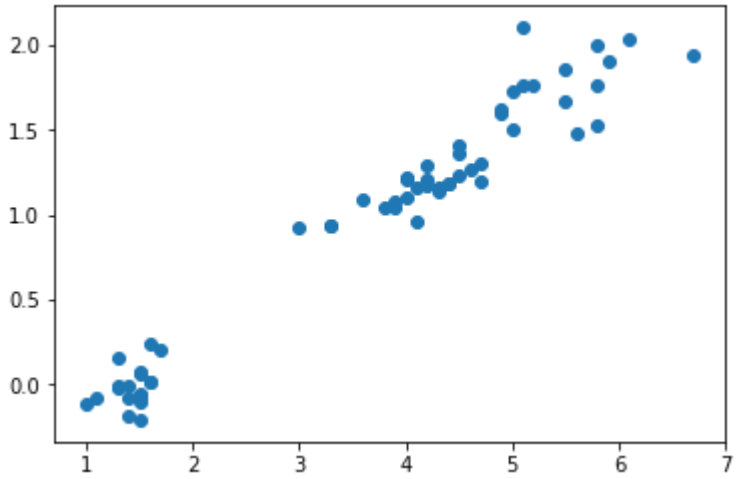
In [117]:
```
#sepal length
plt.scatter(x_test[:,0],y_pred)
plt.show()
```



In [118]:
```
#sepal width
plt.scatter(x_test[:,1],y_pred)
plt.show()
```



In [119]:
```
#petal length
plt.scatter(x_test[:,2],y_pred)
plt.show()
```



In [120]:
```
#petal width
plt.scatter(x_test[:,3],y_pred)
plt.show()
```