Lab 1: Splitting dataset into training and testing data to find the Accuracy of AI model.

Q.n) The data below shows no of hours studied and marks obtained by 13 students. Use Linear Regression Model to split given data into traning set and test set and find the accuracy of the model.

Hours studied	20	50	32	65	23	43	10	5	22	35	29	5	56
Marks obtained	56	83	47	93	47	82	45	23	55	67	57	4	89

Code:

```
import numpy as np
import matplotlib.pyplot as plt.
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
marks = np.array ( [56,83,47,93, 47, 82,45,23,55,67,57, 4, 89]).
 reshape(-1,1)
hours = np.array([20,50,32,65,23,43,10,5,22,35,29,5,56]).reshape(-1,1)
time_train, time_test, score_train, score_test = train_test_split(hours
                                                  marks, test size = 0.2)
model = LinearRegression( )
model.fit (time_train, score_train)
print ( f'Accuracy of model = {model.score (time_test, score_test)}
                                                              )
```

Output

Accuracy of model = 0.6267918872117746

Lab 2: Prediction using Linear Regression Model and representing data in graphical form.

Q) In same lab 1, use Linear Regression Model to predict the score of a student who studied 56 hours.

Code:

```
prediction = model.predict(np.array([56]).reshape(-1,1))
print(f'Prediction for 56 hours: {prediction[0][0]}')
plt.scatter(hours_studied, marks_obtained, color='purple')
plt.ylim(0,100)
plt.xlim(0,70)
plt.plot(np.linspace(0,70,100).reshape(-1,1),model.predict(np.linspace
        (0,70,100).reshape(-1,1)),color='red', label='Regression
                                                      Line')
plt.xlabel('Hours studied')
plt.ylabel('Marks obtained')
plt.title('Linear Regression: Hours studied Vs Marks obtained')
plt.legend()
plt.show()
```

Output

Prediction of 56 hours: 94.96115519468536

Lab3: Using Support Vector Machine for Classification.

Q) The data below shows attribute (diameter in cm and wt. in gm) of 3 types of balls train an AI model using SVM to classify new balls of attributes:

a) diameter 15 cm and 200 gm

b) diameter 21 cm and 7100 gm

Cricket ball		Football		Bowling Ball	
Diameter	Weight	Diameter	Weight	Diameter	Weight
7.3	160	22	450	21.8	7000
7.5	155	23	460	22	7100
7.2	165	22.5	440	21.9	6950

Code:

```
import numpy as np
import matplotlib.pyplot as plt
from sklear import SVM
cricket_ball = np.array ( [ [7.3,160], [7.5,155], [7.2,165] )
football = np.array( [22,450], [23,460], [22.5,440])
bowling_ball = np.array( [21.8,7000], [22,7100], [21.9,6950])
data = np.vstack ( (cricket_ball. football, bowling_ball) )
labels= np.array ( [0]*3 + [1]*[3] + [2]*3)
classifier= svm.SVC (kernel='linear')
classifier.fit (data,labels)
a=np.array ( [[15,200]])
b= np.array ([[21,7100]])
prediction1 = classifier.predict(a)
prediction2= classifier.predict ( b)
~~prediction1~~ =
ball_types = ['cricket', 'Football', 'Bowling']
print (f' New ball with diameter 15 cm and weight 200 gm is:
                {ball_types [prediction1[0]]}')
plt.figure( figsize = (8,6))
colors = ['red', 'green', 'blue']
markers = [ 'o', 's', 'd']
for i, color in enumerate (colors):
    plt.scatter (data [labels == i] [:,0],
                 data [labels == i] [:, i],
                 color = color, marker = markers [i], label=ball_types[i],
```

```
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()
xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 500), np.linspace
                     (ylim[0], ylim[1], 500))
Z = classifier.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = np.argmax(Z, axis=1).reshape(xx.shape)
plt.contour(xx, yy, Z, levels = [-0.5, 0.5, 1.5], colors='k',
                        linestyles = ['--', '-', '--'], alpha = 0.5)
plt.scatter(classifier.support_vectors_[:,0],
            classifier.support_vectors_[:,1], s=1,
            linewidth = 1,
            facecolors = 'none', edgecolors='k')
plt.scatter(a[:,0], a[:,1], c='yellow', label='Point1(15,200)'
            edgecolors='k', s=100)
plt.scatter(b[:,0], b[:,1], c='cyan', label='Point2(21,7100)',
            edgecolors='k', s=100)
plt.xlabel('Diameter(cm)')
plt.ylabel('Weight(gm)')
plt.legend()
plt.title('SVM Decision Boundary and Predictions')
plt.show()
```

Output

```
New ball with diameter 15 cm and Weight 200gm is : Cricket
New ball with diameter 21 cm and weight 7100gm is: Bowling
```