Naveen Kumar Meena

UE208066

In [8]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
```

In [9]:

```
x = np.array([[0,1,2,3], [0,1,4,9]])
xx = np.array([[0,0,1],
               [1,1,1],
               [2,4,1],])
yy = np.array([1,-1,-1])
y = np.array([1,-1,-1,1])
w = np.linalg.solve(xx,yy)
print(w)
```

```
[-3.  1.  1.]
```

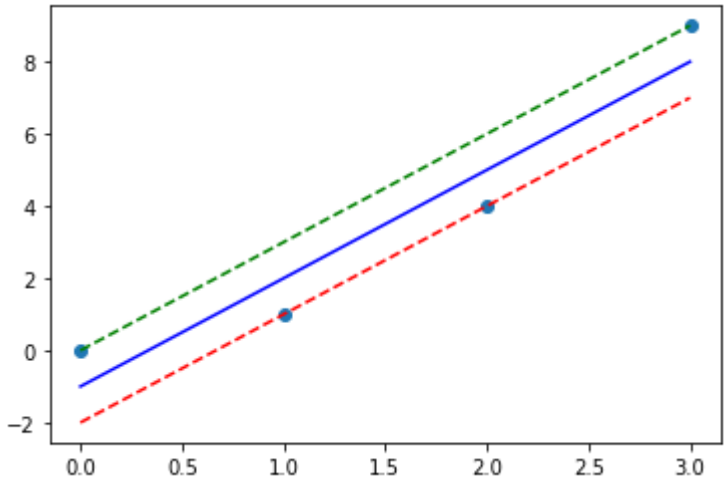
In [10]:

```
plt.scatter(x[0],x[1])
# w1*x1 + w2*x2 + b=0 => x2=-(w1*x1  + b)/w2
hp = -(w[0]*x[0] + w[2])/w[1]
plt.plot(x[0],hp, color="b")

#w1*x1 + w2*x2 + b=1 => x2=(1-(w1*x1  + b))/w2
pp = (1-(w[0]*x[0] + w[2]))/w[1]
plt.plot(x[0], pp, color="g", linestyle="--")

#w1*x1 + w2*x2 + b=-1 => x2=-(1+w1*x1  + b))/w2
np = -(1+ w[0]*x[0] + w[2])/w[1]
plt.plot(x[0], np, color="r",linestyle="--")

plt.show()
```



Assignment 2

In [12]:

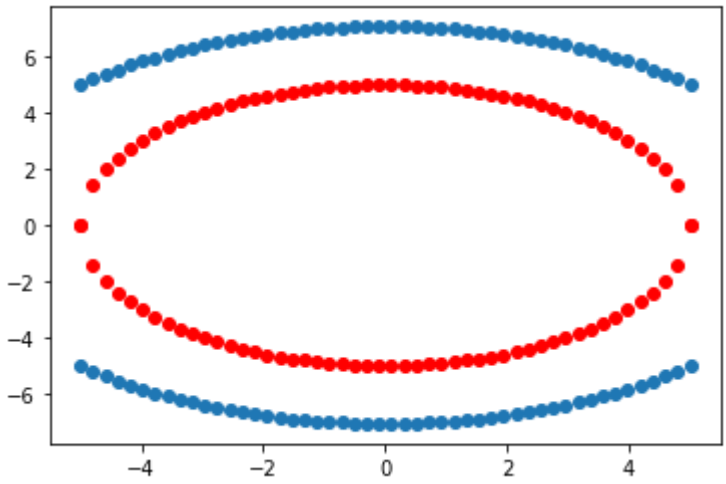
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [13]:

```
x = np.linspace(-5, 5, 50)
y = np.sqrt(50 - x**2)
y = np.hstack([y,-y])
x = np.hstack([x,-x])
x1 = np.linspace(-5, 5, 50)
y1 = np.sqrt(25 - x1**2)
y1 = np.hstack([y1,-y1])
x1 = np.hstack([x1,-x1])
plt.scatter(x,y)
plt.scatter(x1,y1,color='r')
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x17c62255b20>
```



In [14]:

```
df1 = pd.DataFrame(np.vstack([y,x]).T, columns =['x1','x2'] )
df1['Y'] = 0
df2 = pd.DataFrame(np.vstack([y1,x1]).T, columns =['x1','x2'] )
df2['Y'] = 1
df = df1.append(df2)
df.head(5)
```

Out[14]:

	x1	x2	Y
0	5.000000	-5.000000	0
1	5.196072	-4.795918	0
2	5.377270	-4.591837	0
3	5.545052	-4.387755	0
4	5.700603	-4.183673	0

In [15]:

```
X = df.iloc[:,:2]
y = df.Y
y
```

Out[15]:

```
0     0
1     0
2     0
3     0
4     0
     ..
95    1
96    1
97    1
98    1
99    1
Name: Y, Length: 200, dtype: int64
```

In [16]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.25, random_state=

#Import svm model
from sklearn import svm
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

In [17]:

```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy: how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.46
```

In [18]:

```
# polynomial kernel
df['x1cube']= df['x1']**3
df['x2cube']= df['x2']**3
df['x1*x2']= df['x1']*df['x2']
df.head(5)
```

Out[18]:

	x1	x2	Y	x1cube	x2cube	x1*x2
0	5.000000	-5.000000	0	125.000000	-125.000000	-25.000000
1	5.196072	-4.795918	0	140.289623	-110.310117	-24.919938
2	5.377270	-4.591837	0	155.483960	-96.818715	-24.691547
3	5.545052	-4.387755	0	170.497079	-84.474794	-24.330332
4	5.700603	-4.183673	0	185.251795	-73.227354	-23.849462

In [19]:

```
X = df[['x1','x2','x1cube','x2cube','x1*x2']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.25, random_state=
X_train
```

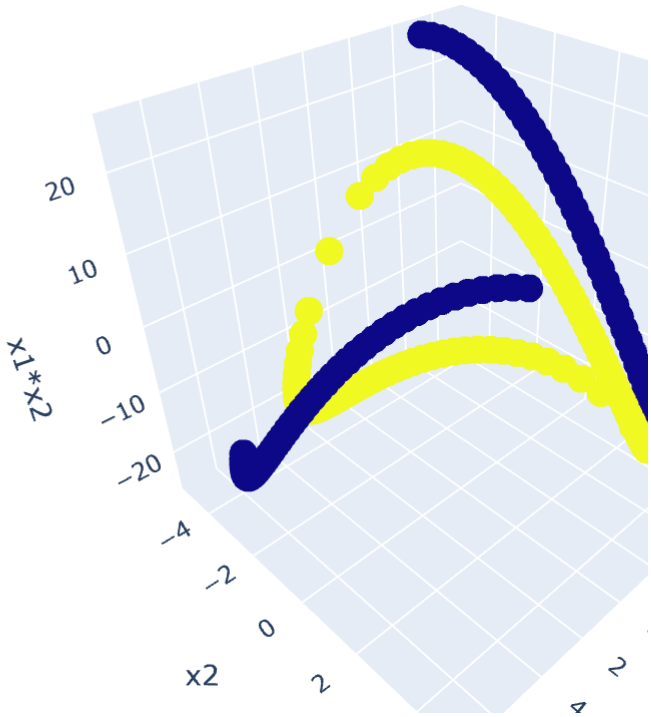
Out[19]:

	x1	x2	x1cube	x2cube	x1*x2
71	-7.034898	0.714286	-348.155688	0.364431	-5.024927
24	4.998959	-0.102041	124.921916	-0.001062	-0.510098
84	-4.608812	-1.938776	-97.896468	-7.287567	8.935452
97	-5.377270	-4.591837	-155.483960	-96.818715	24.691547
49	0.000000	5.000000	0.000000	125.000000	0.000000
...	...	...	...	...	...
67	-6.903421	1.530612	-328.997912	3.585878	-10.566461
92	-3.499271	-3.571429	-42.848217	-45.553936	12.497397
17	4.759961	-1.530612	107.847507	-3.585878	-7.285654
47	5.377270	4.591837	155.483960	96.818715	24.691547
72	-4.973901	0.510204	-123.052779	0.132810	-2.537705

150 rows × 5 columns

In [20]:

```
import plotly.express as px
fig = px.scatter_3d(df, x='x1', y = 'x2', z='x1*x2',color='Y')
fig.show()
```



In [21]:

```
import plotly.express as px
fig = px.scatter_3d(df, x='x1cube', y = 'x2cube', z='x1*x2',color='Y')
fig.show()
```