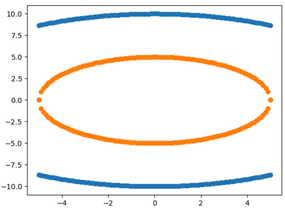
In [154...]
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-5.0, 5.0, 100)
y = np.sqrt(10**2 - x**2)
y=np.hstack((y,-y))
x=np.hstack((x,-x))
```

In [156...]
```
x1 = np.linspace(-5.0, 5.0, 100)
y1 = np.sqrt(5**2 - x1**2)
y1=np.hstack((y1,-y1))
x1=np.hstack((x1,-x1))
```

In [160...]
```
plt.scatter(x,y)
plt.scatter(x1,y1)
```

Out[160...]
```
<matplotlib.collections.PathCollection at 0x1d794bdce00>
```



In [161...]
```
import pandas as pd
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 = pd.concat([df1, df2], ignore_index=True)
df.head(5)
```

Out[161...]

	X1	X2	Y
0	8.660254	-5.00000	0
1	8.717792	-4.89899	0
2	8.773790	-4.79798	0
3	8.828277	-4.69697	0
4	8.881281	-4.59596	0

In [164...]
```
df.tail()
```

Out[164...]

	X1	X2	Y
395	-1.969049	-4.59596	1
396	-1.714198	-4.69697	1
397	-1.406908	-4.79798	1
398	-0.999949	-4.89899	1
399	-0.000000	-5.00000	1

In [166...]
```
### Independent and Dependent features
x = df.iloc[:, :2]
y = df.Y
```

In [168...]
```
y
```

Out[168...]
```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    1
399    1
Name: Y, Length: 400, dtype: int64
```

In [170...]
```
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=0)
```

In [172...]
```
y_train
```

Out[172...]
```
250    1
63     0
312    1
159    0
283    1
      ..
323    1
192    0
117    0
47     0
172    0
Name: Y, Length: 300, dtype: int64
```

In [174...]
```
### X1_X2,x1_square,x2_square,x1*x2

df['X1_Square']= df['X1']**2
df['X2_Square']= df['X2']**2
df['X1*X2']= (df['X1'] *df['X2'])
df.head()
```

Out[174...]

	X1	X2	Y	X1_Square	X2_Square	X1*X2
0	8.660254	-5.00000	0	75.000000	25.000000	-43.301270
1	8.717792	-4.89899	0	75.999898	24.000102	-42.708375
2	8.773790	-4.79798	0	76.979390	23.020610	-42.096467
3	8.828277	-4.69697	0	77.938476	22.061524	-41.466150
4	8.881281	-4.59596	0	78.877155	21.122845	-40.818009

In [176...]
```
x = df[['X1','X2','X1_Square','X2_Square','X1*X2']]
y = df['Y']
```

In [178...]
```
y
```

Out[178...]
```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    1
399    1
Name: Y, Length: 400, dtype: int64
```

In [180...]
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [182...]
```
x_train
```

Out[182...]

	X1	X2	X1_Square	X2_Square	X1*X2
250	4.999745	0.050505	24.997449	0.002551	0.252512
63	9.906589	1.363636	98.140496	1.859504	13.508984
312	-3.263736	3.787879	10.651974	14.348026	-12.362637
159	-9.953852	-0.959596	99.079176	0.920824	9.551676
283	3.680983	3.383838	13.549638	11.450362	12.455852
...	...	...	...	...	...
323	-4.223140	2.676768	17.834915	7.165085	-11.304366
192	-9.031653	-4.292929	81.570758	18.429242	38.772248
117	-9.445795	3.282828	89.223038	10.776962	-31.008922
47	9.996811	-0.252525	99.936231	0.063769	-2.524447
172	-9.738311	-2.272727	94.834711	5.165289	22.132526

300 rows × 5 columns

In [188...]
```
import plotly.express as px

fig = px.scatter_3d(df, x='X1', y='X2', z='X1*X2',color='Y')
fig.show()
```

In [189...]
```
fig = px.scatter_3d(df, x='X1_Square', y='X1_Square', z='X1*X2',color='Y')
fig.show()
```

In [191...]
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
classifier = SVC(kernel="linear")
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
accuracy_score(y_test, y_pred)
```

Out[191...]
```
1.0
```

In [197...]
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
classifier = SVC(kernel="poly")
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
accuracy_score(y_test, y_pred)
```

Out[197...]
```
1.0
```

In [201...]
```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
classifier = SVC(kernel="rbf")
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
accuracy_score(y_test, y_pred)
```

Out[201...]
```
1.0
```

In [ ]:

In [ ]: