svm

October 16, 2023

[1]:
```
# Loading libraries
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
# Loading dataset
df = pd.read_csv('data.csv')
df.head()
```

[2]:

	Unnamed: 0	feature1	feature2	label_y
0	0	8.660254	-5.00000	0
1	1	8.717792	-4.89899	0
2	2	8.773790	-4.79798	0
3	3	8.828277	-4.69697	0
4	4	8.881281	-4.59596	0

[3]:
```
# Removing unnecessary data
df.drop('Unnamed: 0',axis=1,inplace=True)
```

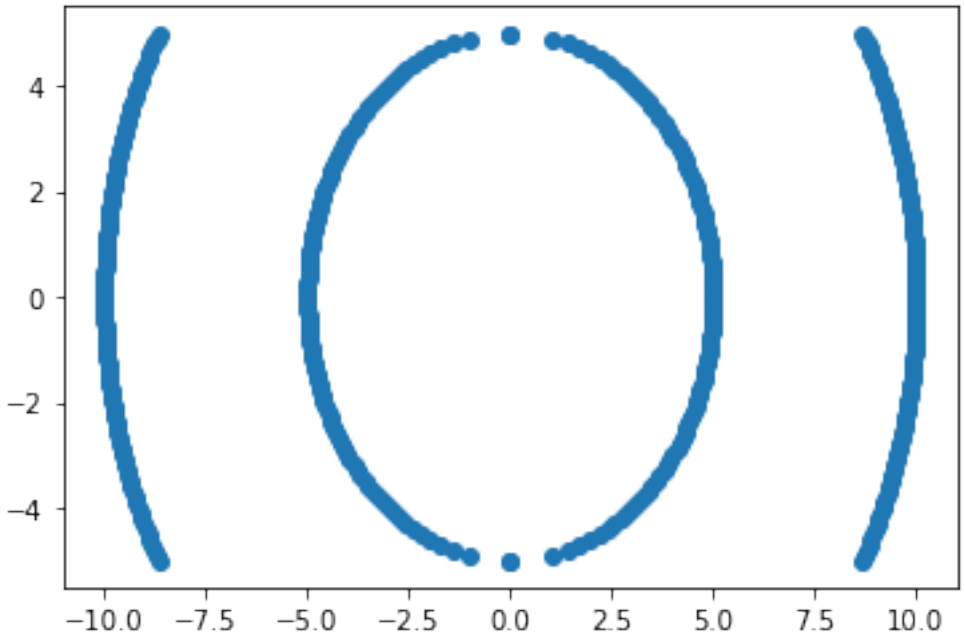
[4]:
```
df.head()
```

[4]:

	feature1	feature2	label_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

[5]:
```
plt.scatter(df.feature1,df.feature2)
```

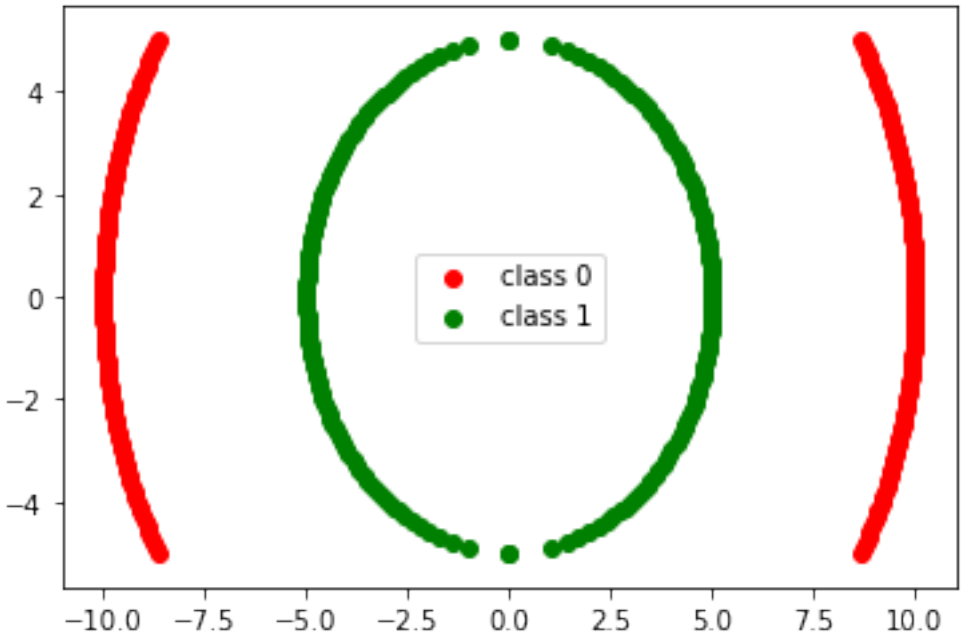
[5]:
```
<matplotlib.collections.PathCollection at 0x22fb43e64c8>
```



[8]:
```
# Mixed data - Non separable
df1 = df[df.label_y==0]
df2 = df[df.label_y==1]

plt.scatter(df1.feature1,df1.feature2,c='red',label='class 0')
plt.scatter(df2.feature1,df2.feature2,c='green', label = 'class 1')
plt.legend()
```

[8]:
```
<matplotlib.legend.Legend at 0x22fb45e8c88>
```



[9]:
```
from sklearn.model_selection import train_test_split
train,test = train_test_split(df,train_size=.7,random_state=100)
```

0.1 Linear karnel

[12]:
```
from sklearn.svm import SVC
svm = SVC(kernel='linear')
```

[13]:
```
svm.fit(train.drop('label_y', axis=1) , train.label_y) # fit(x,y)
```

[13]:
```
SVC(kernel='linear')
```

[14]:
```
svm.score(test.drop('label_y', axis=1) , test.label_y)
```

[14]:
```
0.55
```

0.2 Polynomial karnel

[15]:
```
# polynomial kernel k(a,b) = (a^T * b)^2
# F1, F2, F1^2 , F2^2 , F1F2
df['f1_square'] = df.feature1 ** 2
df['f2_square'] = df.feature2 ** 2
df['f1f2'] = (df.feature1)*(df.feature2)

df.head()
```

[15]:

	feature1	feature2	label_y	f1_square	f2_square	f1f2
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

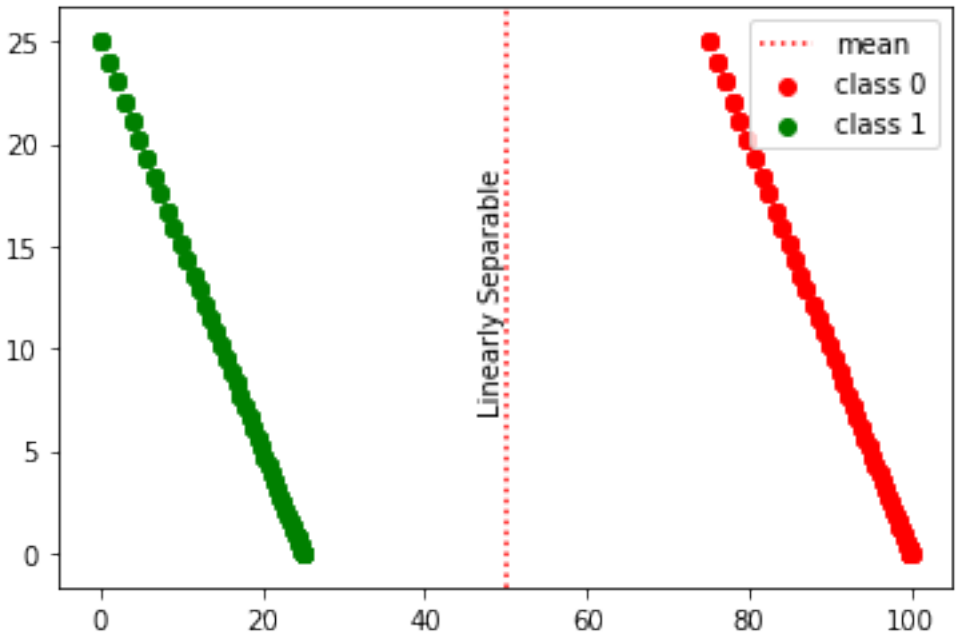
[33]:
```
df1 = df[df.label_y==0]
df2 = df[df.label_y==1]

plt.scatter(df1.f1_square,df1.f2_square,c='red',label='class 0')
plt.scatter(df2.f1_square,df2.f2_square,c='green', label = 'class 1')

plt.axvline(x=50, color='red', linestyle='dotted',label= 'mean')
plt.text(50, 13, 'Linearly Separable␣
 ↪',horizontalalignment='right',verticalalignment='center',rotation='vertical')

plt.legend()
```

[33]: <matplotlib.legend.Legend at 0x22fb8c51448>



[23]:
```
# 3D display
import plotly.express as p
p.scatter_3d(df, x='f1_square', y='f2_square', z='f1f2', color='label_y')
```

[24]:
```
x = df.drop('label_y',axis=1)
y = df.label_y
```

[25]:
```
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=.3,random_state=42)
xtest.shape
```

[25]: (120, 5)

[26]:
```
svm.fit(xtrain,ytrain)
```

[26]: SVC(kernel='linear')

[27]:
```
svm.score(xtest,ytest)
```

[27]: 1.0

0.3 Radial basis function (RBF) kernel

[30]:
```
svm2 = SVC(kernel='rbf')
svm2.fit(xtrain,ytrain)
```

[30]: SVC()

[31]:
```
svm.score(xtest,ytest)
```

[31]: 1.0