Dimensionality reduction and classification of breast cancer data

In [157…

```
import pandas as pd
from sklearn.decomposition import PCA
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from matplotlib.patches import Ellipse
```

In [158…

```
dataset = load_breast_cancer()
```

In [159…

```
pca = PCA(n_components=2)
data_pca = pca.fit_transform(dataset.data)
data_pca
```

Out[159…

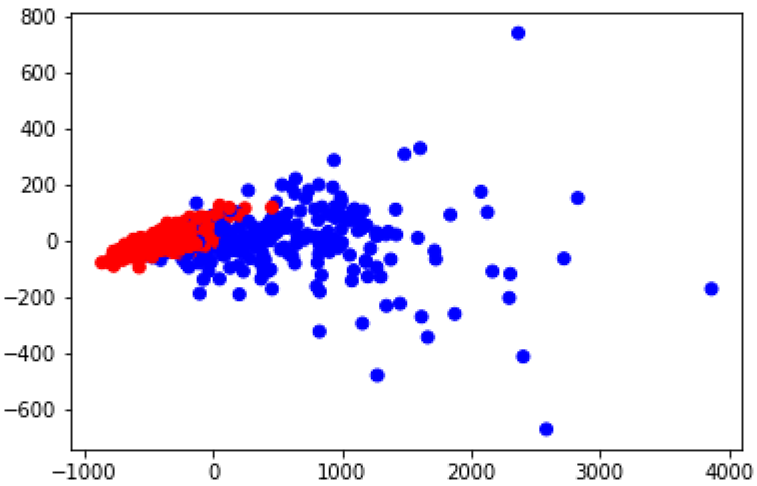
```
array([[1160.1425737 , -293.91754364],
       [1269.12244319,   15.63018184],
       [ 995.79388896,   39.15674324],
       ...,
       [ 314.50175618,   47.55352518],
       [1124.85811531,   34.12922497],
       [-771.52762188,  -88.64310636]])
```

In [160…

```
labels = ['r' if t else 'b' for t in dataset.target]
plt.scatter(data_pca[:,0], data_pca[:,1], c=labels)
```

Out[160…

```
<matplotlib.collections.PathCollection at 0x1a26863908>
```



In [161…

```
labels_adj = np.array([i if i else -1 for i in dataset.target])
labels_adj.shape = (len(dataset.target), 1)
```

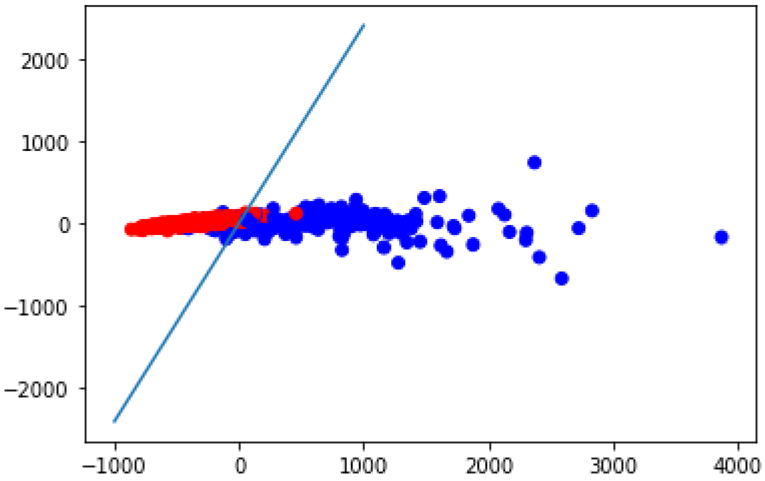
In [162…

```
w = np.linalg.inv(data_pca.T@data_pca)@data_pca.T@labels_adj
```

In [163…

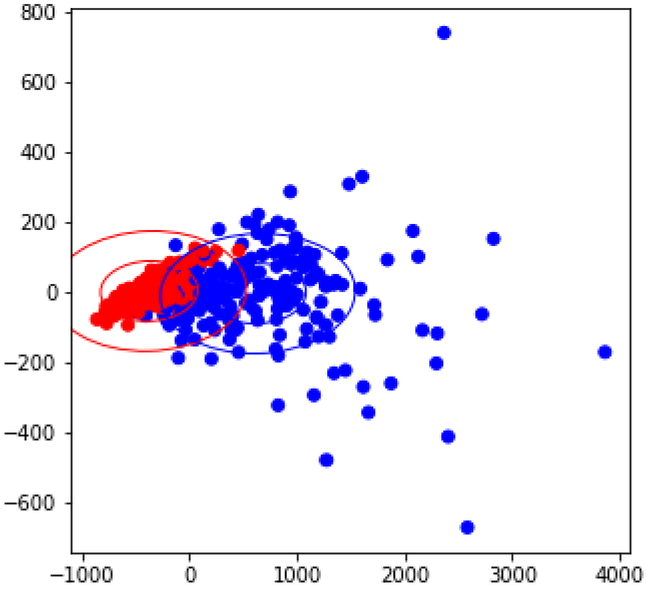
```
x = [i for i in range(-1000, 1000)]
y = x*(-w[0]/w[1])
plt.plot(x,y)
plt.scatter(data_pca[:,0], data_pca[:,1], c=labels)
```

```
Out[163… <matplotlib.collections.PathCollection at 0x1a27058ba8>
```



```
In [164… clf = LinearDiscriminantAnalysis(store_covariance=True, n_components=2)
clf.fit(data_pca, labels)
print ("Covariance Matrix: ")
cov = clf.covariance_
print (cov)
mean_1, mean_2 = clf.means_[0], clf.means_[1]
print ("Means: ")
print (clf.means_)
lambda_, v = np.linalg.eig(cov)
lambda_ = np.sqrt(lambda_)
fig, ax_nstd = plt.subplots(figsize=(5,5))
for j in range(1, 3):
    ell = Ellipse(xy=mean_2,
                  width=lambda_[0]*j*2, height=lambda_[1]*j*2,
                  angle=np.rad2deg(np.arccos(v[0, 0])), color='red')
    ell.set_facecolor('none')
    ax_nstd.add_artist(ell)
for j in range(1, 3):
    ell = Ellipse(xy=mean_1,
                  width=lambda_[0]*j*2, height=lambda_[1]*j*2,
                  angle=np.rad2deg(np.arccos(v[0, 0])), color='blue')
    ell.set_facecolor('none')
    ax_nstd.add_artist(ell)
ax_nstd.scatter(data_pca[:,0], data_pca[:,1], c=labels)
plt.show()
```

```
//anaconda3/lib/python3.7/site-packages/sklearn/discriminant_analysis.py:466: Ch
angedBehaviorWarning: n_components cannot be larger than min(n_features, n_class
es - 1). Using min(n_features, n_classes - 1) = min(2, 2 - 1) = 1 components.
  ChangedBehaviorWarning)
//anaconda3/lib/python3.7/site-packages/sklearn/discriminant_analysis.py:472: Fu
tureWarning: In version 0.23, setting n_components > min(n_features, n_classes -
1) will raise a ValueError. You should set n_components to None (default), or a
value smaller or equal to min(n_features, n_classes - 1).
  warnings.warn(future_msg, FutureWarning)
Covariance Matrix:
[[204958.31608947   1621.51235842]
 [  1621.51235842   7286.20735532]]
Means:
[[ 633.13324872   -4.31278191]
 [-375.97828776    2.56109178]]
```



```
In [125…
```

```
In [ ]:
```