ejrubins 1)

a) u * v = (1,2,1) x (2,1,0) = 1 x 2 + 2 x 1 + 1 x 0 = 4

b) A *u^t = [[1,3,1], [2,1,1]]* [[1] , [2], [1]] = [[1 x 1 + 3 x 2 + 1 x 1], [2 x 1 + 1 x 2 + 1 x 1]] = [[8], [5]]

c) A * B = [[1,3,1], [2,1,1]] x [[1, 2], [0, 1], [4, 0]] = [[1 x 1 + 3 x 0 + 1 x 4, 1 x 2 + 3 x 1 + 1 x 0], [2 x 1 + 1 x 0 + 1 x 4, 2 x 2 + 1 x 1 + 1 x 0]] = [[5, 5], [6, 5]]

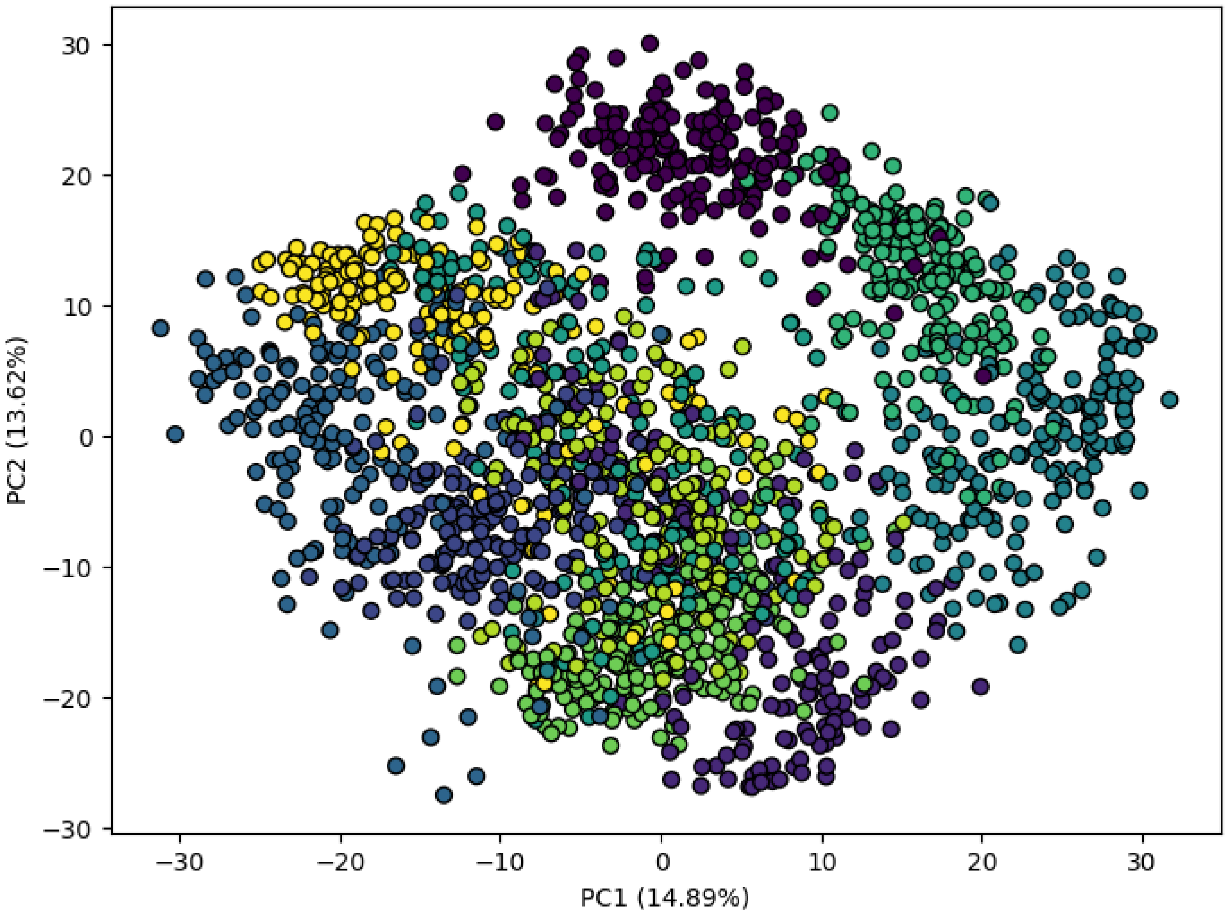
d) cos = (v *x) / (||v|| ||x||) v* x = 2 x 20 + 1 x 10 + 0 x 0 = 50 ||v|| = sqrt(2^2 + 1^2 + 0^2) = sqrt(5) ||x|| = sqrt(20^2 + 10^2 + 0^2) = sqrt(500) cos = 50 / (sqrt(5) x sqrt(500)) = 50/50 = 1

In [1]:
```
'''
ejrubins HW 3

2)
'''


from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

(data, label) = load_digits(as_frame=True, return_X_y=True)

pca = PCA(n_components=2)
data_2D = pca.fit_transform(data)

print("pca.explained_variance_ratio_.sum")
print(pca.explained_variance_ratio_.sum())

plt.figure(figsize=(8,6))
plt.scatter(
  x=data_2D[:, 0],
  y=data_2D[:, 1],
  c=label,
  edgecolor='k',
  s=40
)

pc1, pc2 = pca.explained_variance_ratio_
plt.xlabel(f"PC1 ({pc1:.2%})")
plt.ylabel(f"PC2 ({pc2:.2%})")
```

```
pca.explained_variance_ratio_.sum
0.2850936482369848
```

Out[1]:

```
Text(0, 0.5, 'PC2 (13.62%)')
```



PCA is not a good approach for dimentionality reduction in this dataset. Around 70% of the information is lost and reducing an image to a few parameters does not help in visualization. only 13-14% of the variability is captured when PCA is used. The data does cluster according to the digits the represent.