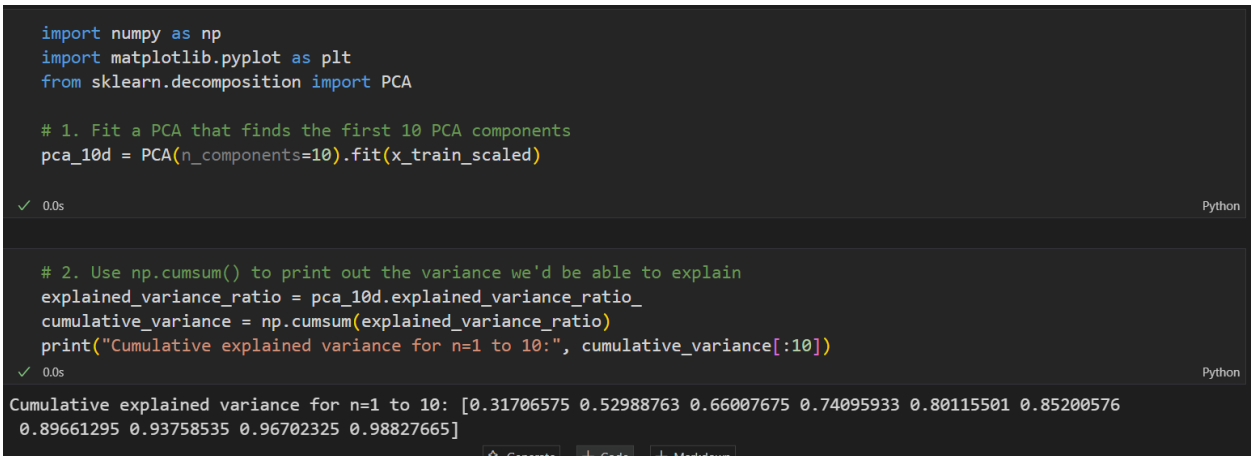
Name – Withanage W.I.N.

Index- 220731M

Assignment- PCA Machine Learning Lab

1. Fit a PCA that finds the first 10 PCA components of our training data



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

# 1. Fit a PCA that finds the first 10 PCA components
pca_10d = PCA(n_components=10).fit(x_train_scaled)
```

✓ 0.0s Python

```
# 2. Use np.cumsum() to print out the variance we'd be able to explain
explained_variance_ratio = pca_10d.explained_variance_ratio_
cumulative_variance = np.cumsum(explained_variance_ratio)
print("Cumulative explained variance for n=1 to 10:", cumulative_variance[:10])
```

✓ 0.0s Python

```
Cumulative explained variance for n=1 to 10: [0.31706575 0.52988763 0.66007675 0.74095933 0.80115501 0.85200576
 0.89661295 0.93758535 0.96702325 0.98827665]
```

2. 2. Use 'np.cumsum()' to print out the variance we'd be able to explain by using n PCA dimensions for n=1 through 10

Number of Component	Cumulative Variance
1	0.31706575
2	0.52988763
3	0.66007675
4	0.74095933
5	0.80115501
6	0.85200576
7	0.89661295
8	0.93758535
9	0.96702325
10	0.98827665

3. Does the 10-dimension PCA agree with the 2d PCA on how much variance the first components explain? **Do the 10d and 2d PCAs find the same first two dimensions? Why or why not?

```
# 3. Compare with 2d PCA (implicitly done via earlier code, but let's verify)
pca_2d = PCA(n_components=2).fit(x_train_scaled)
print("2d PCA first two components' variance:", np.cumsum(pca_2d.explained_variance_ratio_))
print("10d PCA first two components' variance:", cumulative_variance[:2])
# The first two should match, confirming consistency
```

✓ 0.0s Python

```
2d PCA first two components' variance: [0.31706575 0.52988763]
10d PCA first two components' variance: [0.31706575 0.52988763]
```

Agreement on variance: Yes, the 10-dimension PCA should match the 2d PCA's explained variance for the first two components, as both use the same data and PCA algorithm, with the 2d PCA being a subset of the 10d PCA.

Same first two dimensions: Yes, the first two dimensions should be identical, as PCA computes components sequentially based on maximum variance, and reducing to 2d or 10d doesn't alter the initial components unless the data or scaling changes.

Why: The consistency arises because PCA's component calculation is deterministic given fixed input (scaled data), and higher dimensions extend rather than replace earlier ones.

4. Make a plot of number of PCA dimensions against total variance explained. What PCA dimension looks good to you?

```
# 4. Make a plot of number of PCA dimensions against total variance explained
plt.plot(range(1, 11), cumulative_variance, marker='o')
plt.xlabel('Number of PCA Dimensions')
plt.ylabel('Total Variance Explained')
plt.title('PCA Dimensions vs Variance Explained')
plt.grid(True)
plt.show()

# Suggested dimension: Look for the "elbow" where the curve flattens
```

✓ 0.0s Python