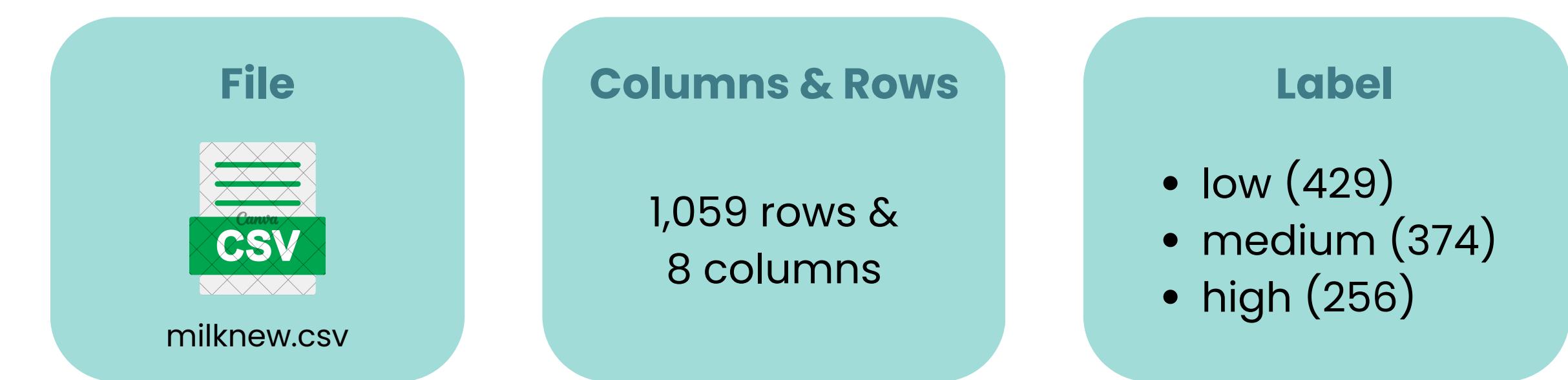
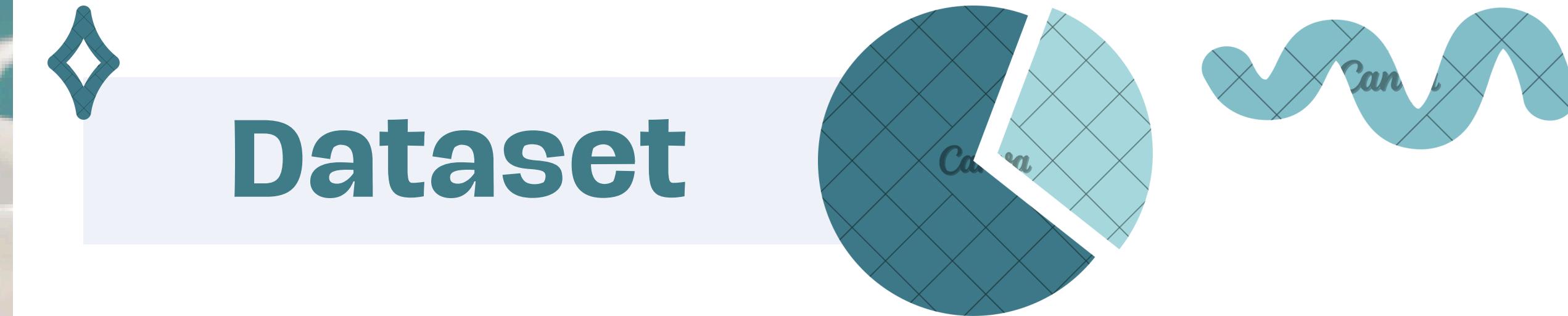


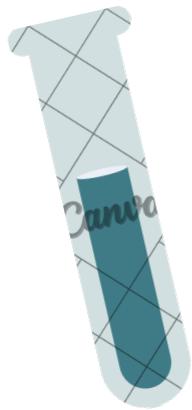
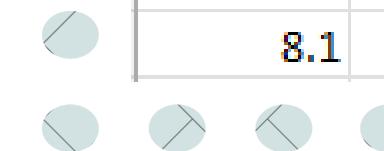


# Unsupervised Machine Learning: **Milk Quality Clustering**

🌐 <https://www.kaggle.com/datasets/cpluzshrijayan/milkquality>

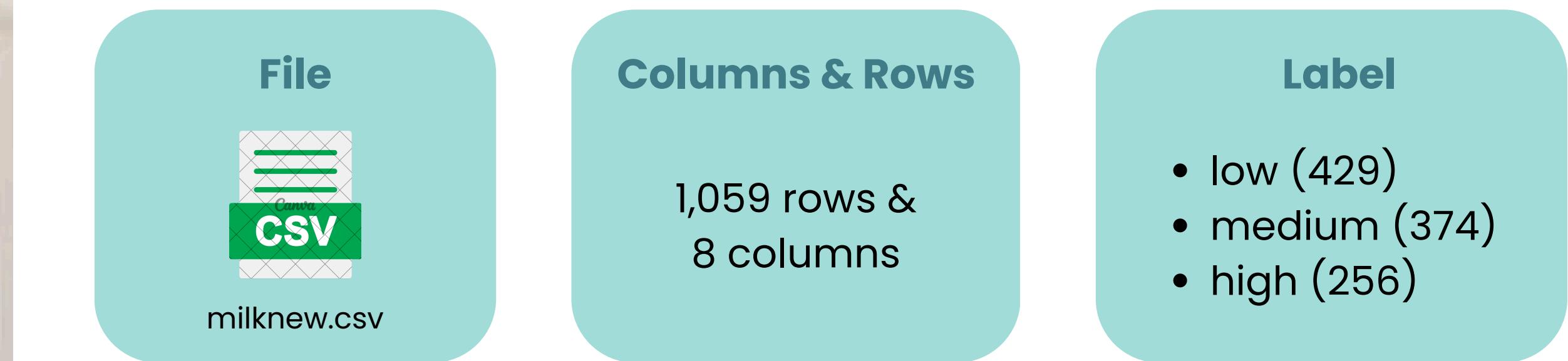


pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
6.6	35	1	0	1	0	254	high
6.6	36	0	1	0	1	253	high
8.5	70	1	1	1	1	246	low
9.5	34	1	1	0	1	255	low
6.6	37	0	0	0	0	255	medium
6.6	37	1	1	1	1	255	high
5.5	45	1	0	1	1	250	low
4.5	60	0	1	1	1	250	low
8.1	66	1	0	1	1	255	low





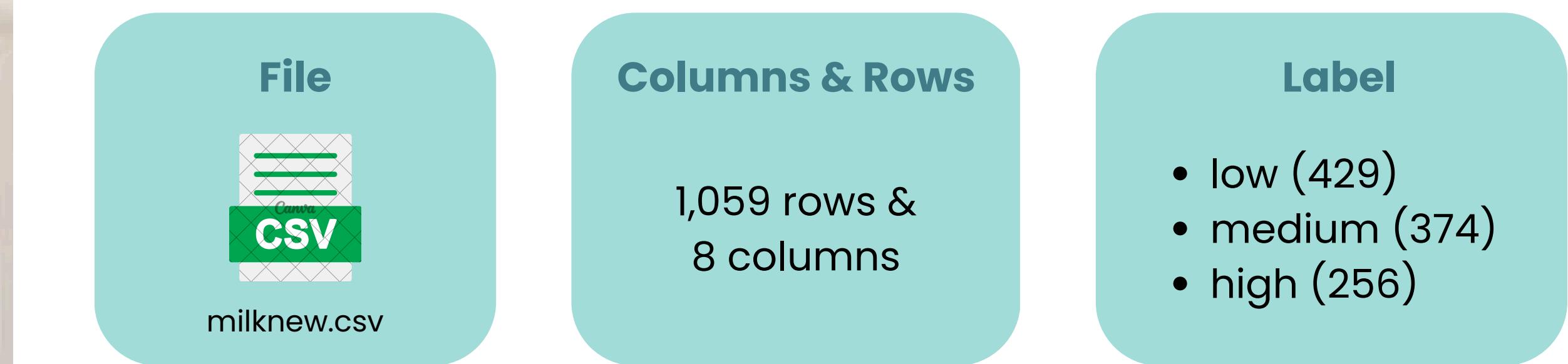
# Dataset



pH	Temp	Chloride	Odor	Fat	Turbidity	Colour	Grade
6.6	35	1	0	1	0	254	high
6.6	36	0	1	0	1	253	high
8.5	70	1	1	1	1	246	low
9.5	34	1	1	0	1	255	low
6.6	37	0	0	0	0	255	medium
6.6	37	1	1	1	1	255	high
5.5	45	1	0	1	1	250	low
4.5	60	0	1	1	1	250	low
8.1	66	1	0	1	1	255	low



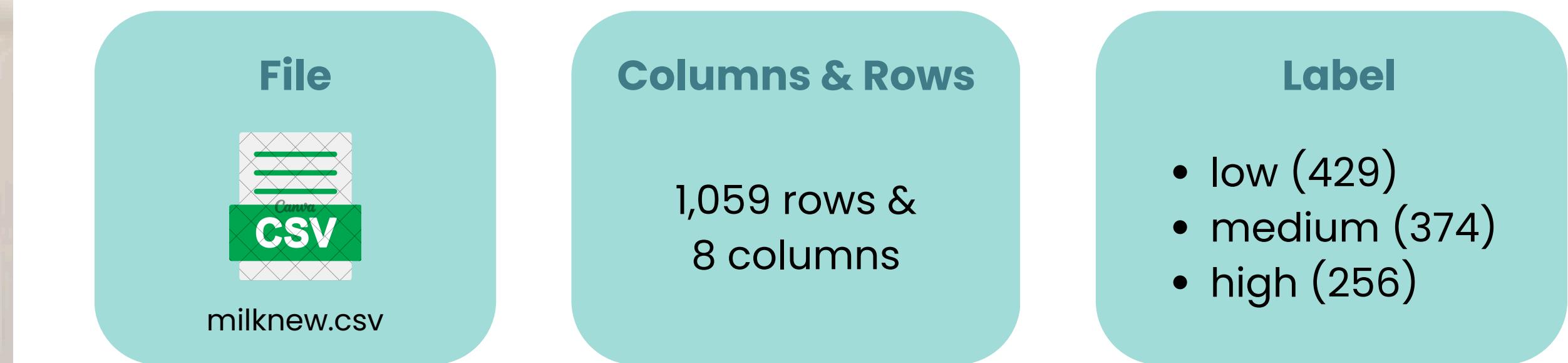
# Dataset



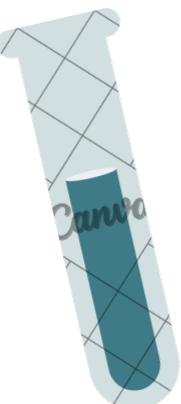
pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
6.6	35	uwa	0	1	0	254	high
6.6	36		1	0	1	253	high
8.5	70		1	1	1	246	low
9.5	34	1	1	0	1	255	low
6.6	37	0	0	0	0	255	medium
6.6	37	1	1	1	1	255	high
5.5	45	1	0	1	1	250	low
4.5	60	0	1	1	1	250	low
8.1	66	1	0	1	1	255	low

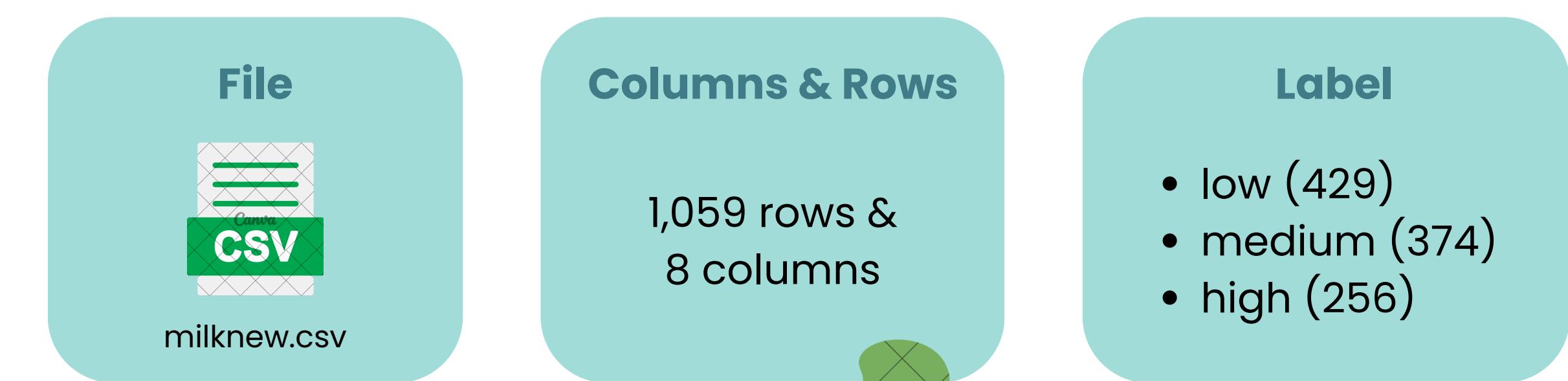


# Dataset

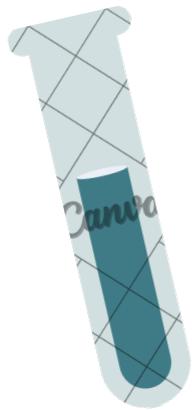


pH	Temprature	Taste	Odor	Turbidity	Colour	Grade
6.6	35	1	1	0	254	high
6.6	36	0	0	1	253	high
8.5	70	1	1	1	246	low
9.5	34	1	1	0	255	low
6.6	37	0	0	0	255	medium
6.6	37	1	1	1	255	high
5.5	45	1	0	1	250	low
4.5	60	0	1	1	250	low
8.1	66	1	0	1	255	low



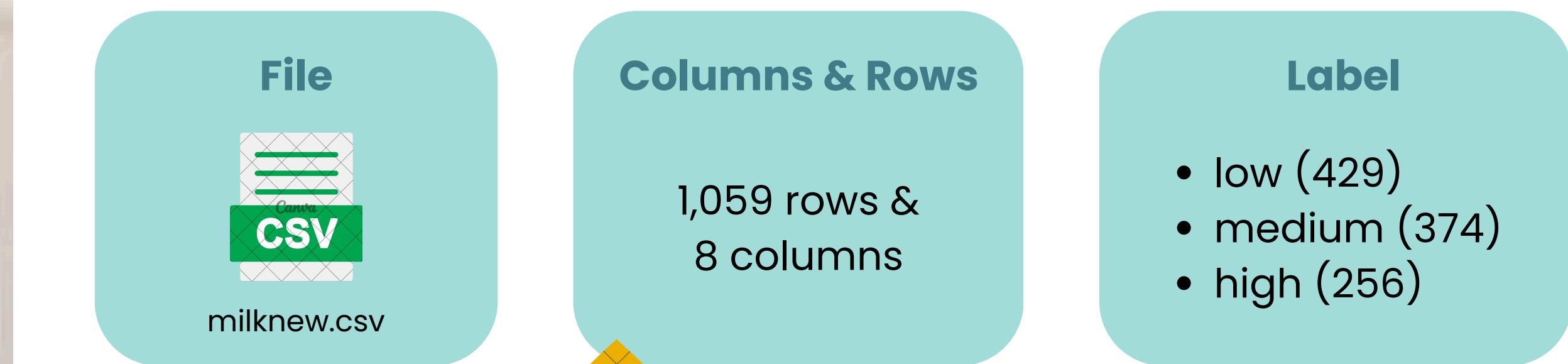


pH	Temprature	Taste	Odor	Fat	Tin	Colour	Grade
6.6	35	1	0	0	1	1	254 high
6.6	36	0	1	1	0	1	253 high
8.5	70	1	1	1	1	1	246 low
9.5	34	1	1	1	0	1	255 low
6.6	37	0	0	0	0	0	255 medium
6.6	37	1	1	1	1	1	255 high
5.5	45	1	0	1	1	1	250 low
4.5	60	0	1	1	1	1	250 low
8.1	66	1	0	1	1	1	255 low

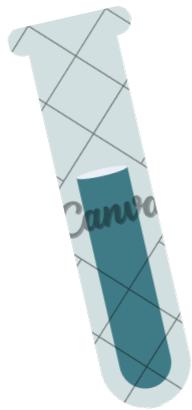
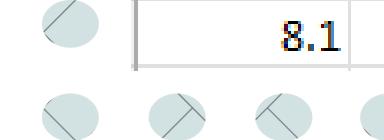




# Dataset

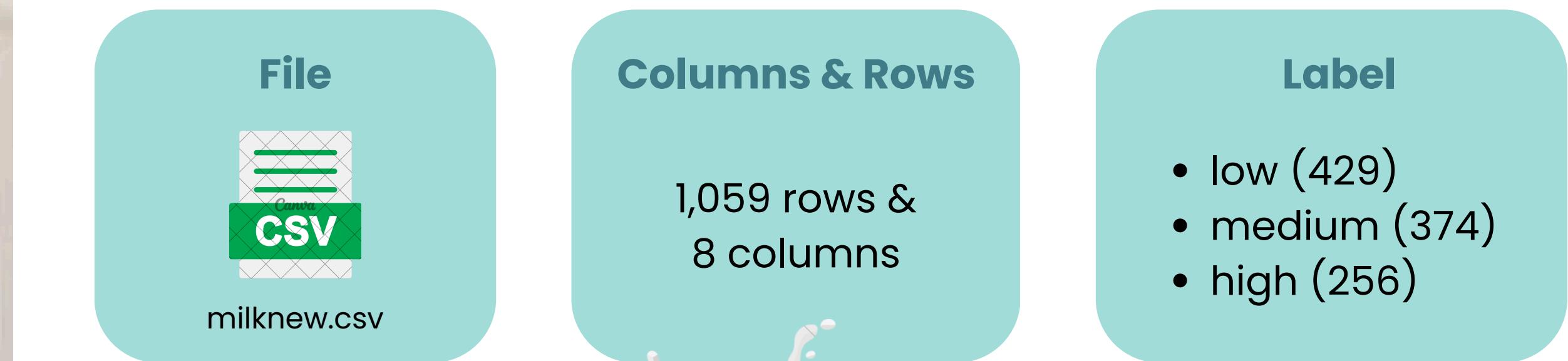


pH	Temprature	Taste	Fat	Turbidity	Colour	Grade
6.6	35		0	1	0	254 high
6.6	36		1	0	1	253 high
8.5	70		1	1	1	246 low
9.5	34	1	1	0	1	255 low
6.6	37	0	0	0	0	255 medium
6.6	37	1	1	1	1	255 high
5.5	45	1	0	1	1	250 low
4.5	60	0	1	1	1	250 low
8.1	66	1	0	1	1	255 low

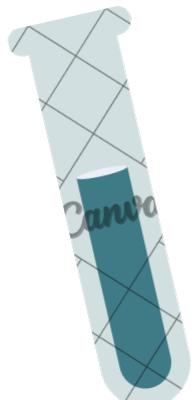


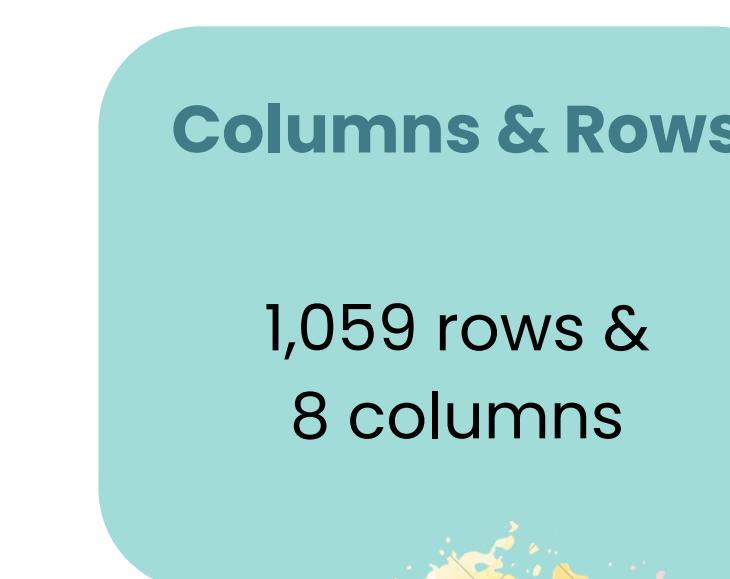
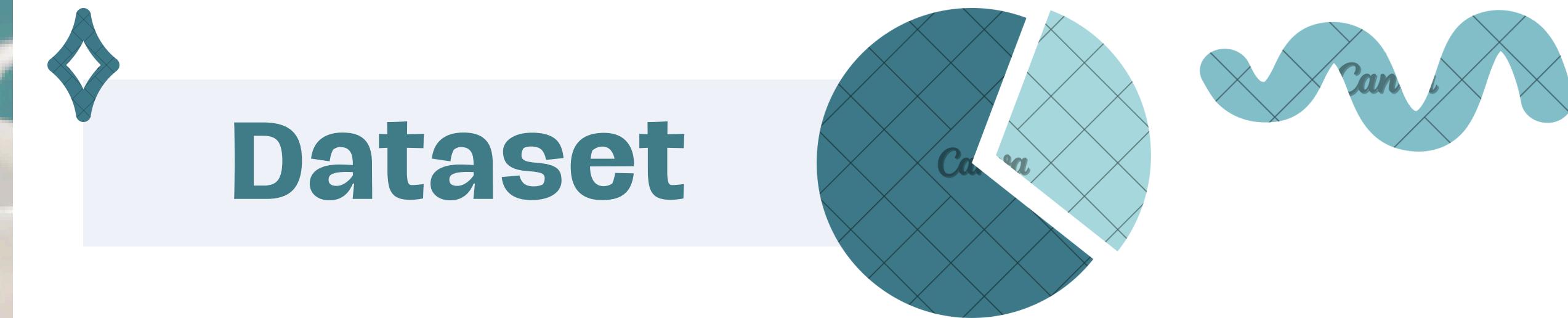


# Dataset

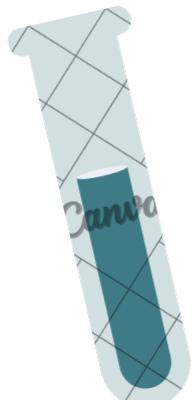


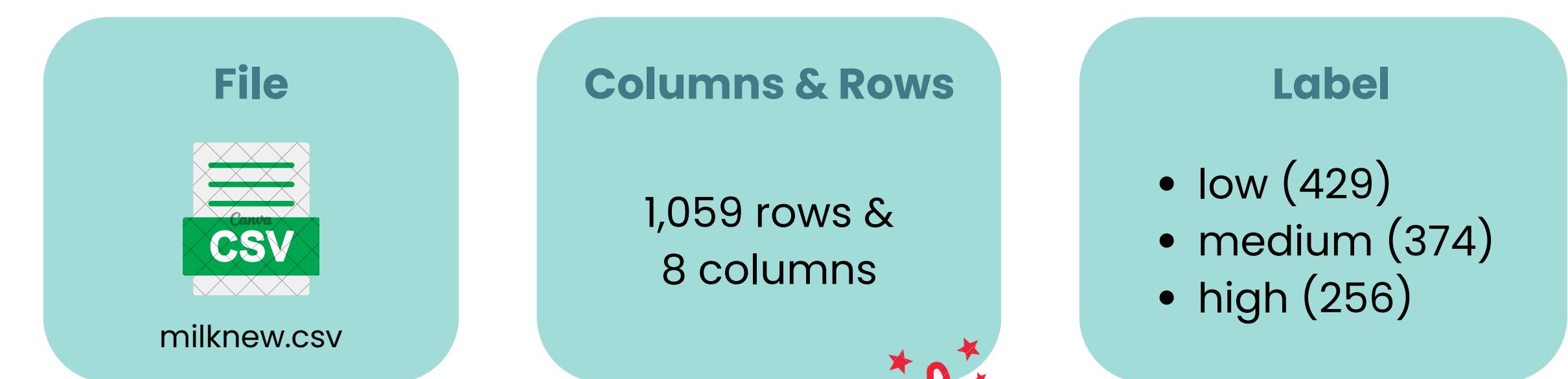
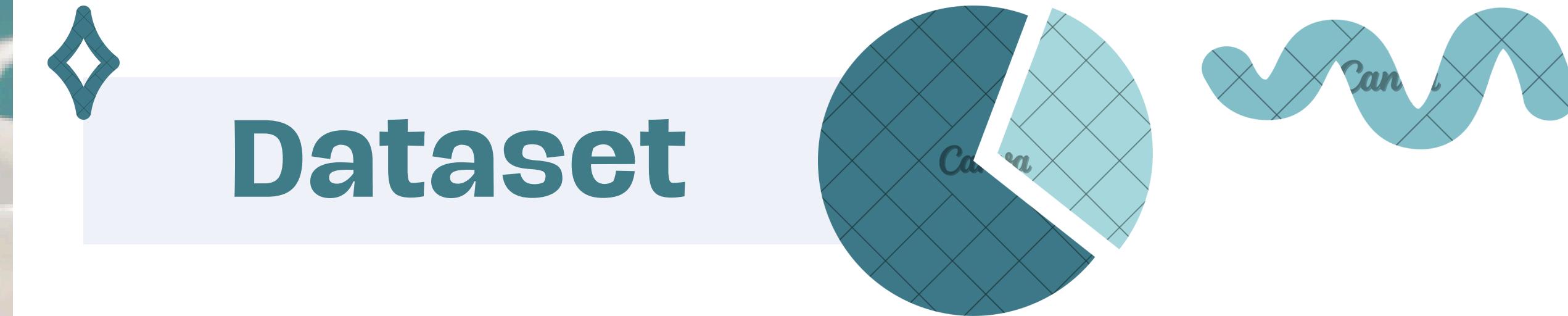
pH	Temprature	Taste		Turbidity	Colour	Grade
6.6	35			1	0	254 high
6.6	36	0	1	0	1	253 high
8.5	70	1	1	1	1	246 low
9.5	34	1	1	0	1	255 low
6.6	37	0	0	0	0	255 medium
6.6	37	1	1	1	1	255 high
5.5	45	1	0	1	1	250 low
4.5	60	0	1	1	1	250 low
8.1	66	1	0	1	1	255 low



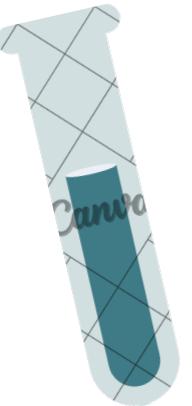


pH	Temprature	Taste	Odor	Colour	Grade
6.6	35	1	1	0	254 high
6.6	36	0	1	1	253 high
8.5	70	1	1	1	246 low
9.5	34	1	1	0	255 low
6.6	37	0	0	0	255 medium
6.6	37	1	1	1	255 high
5.5	45	1	0	1	250 low
4.5	60	0	1	1	250 low
8.1	66	1	0	1	255 low





pH	Temprature	Taste	Odor	Fat	Grade	
6.6	35	1	0	254	high	
6.6	36	0	1	0	253	high
8.5	70	1	1	1	246	low
9.5	34	1	1	0	255	low
6.6	37	0	0	0	255	medium
6.6	37	1	1	1	255	high
5.5	45	1	0	1	250	low
4.5	60	0	1	1	250	low
8.1	66	1	0	1	255	low

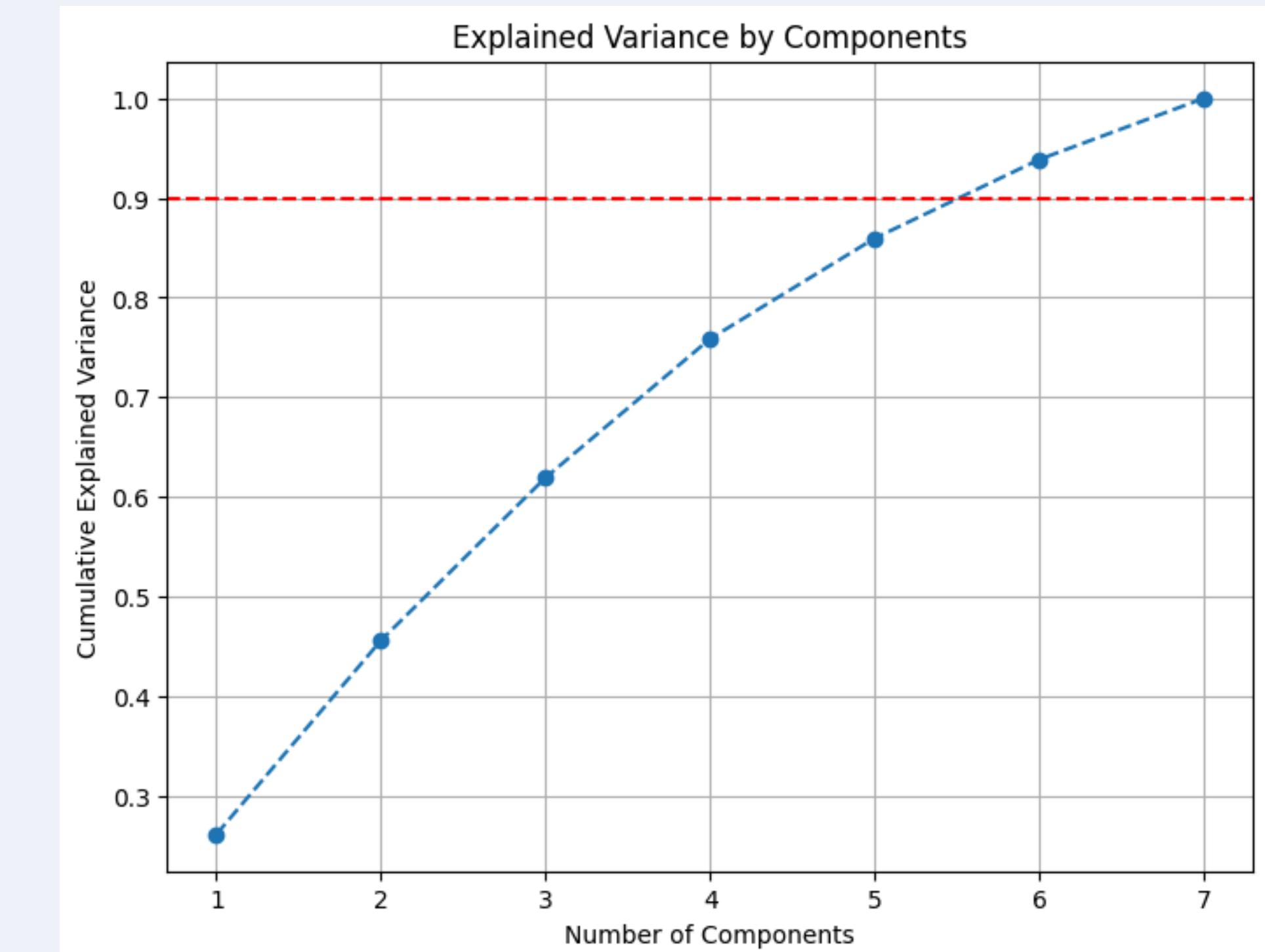




# Principle Component Analysis (PCA)



```
● ● ●  
1 import pandas as pd  
2 from sklearn.decomposition import PCA  
3  
4 data = pd.read_csv('milknew.csv')  
5  
6 X = data.drop('Grade', axis=1)  
7 y = data['Grade']  
8  
9 scaler = StandardScaler()  
10 X_scaled = scaler.fit_transform(X)  
11  
12 pca = PCA(n_components=6)  
13 X_pca = pca.fit_transform(X_scaled)
```

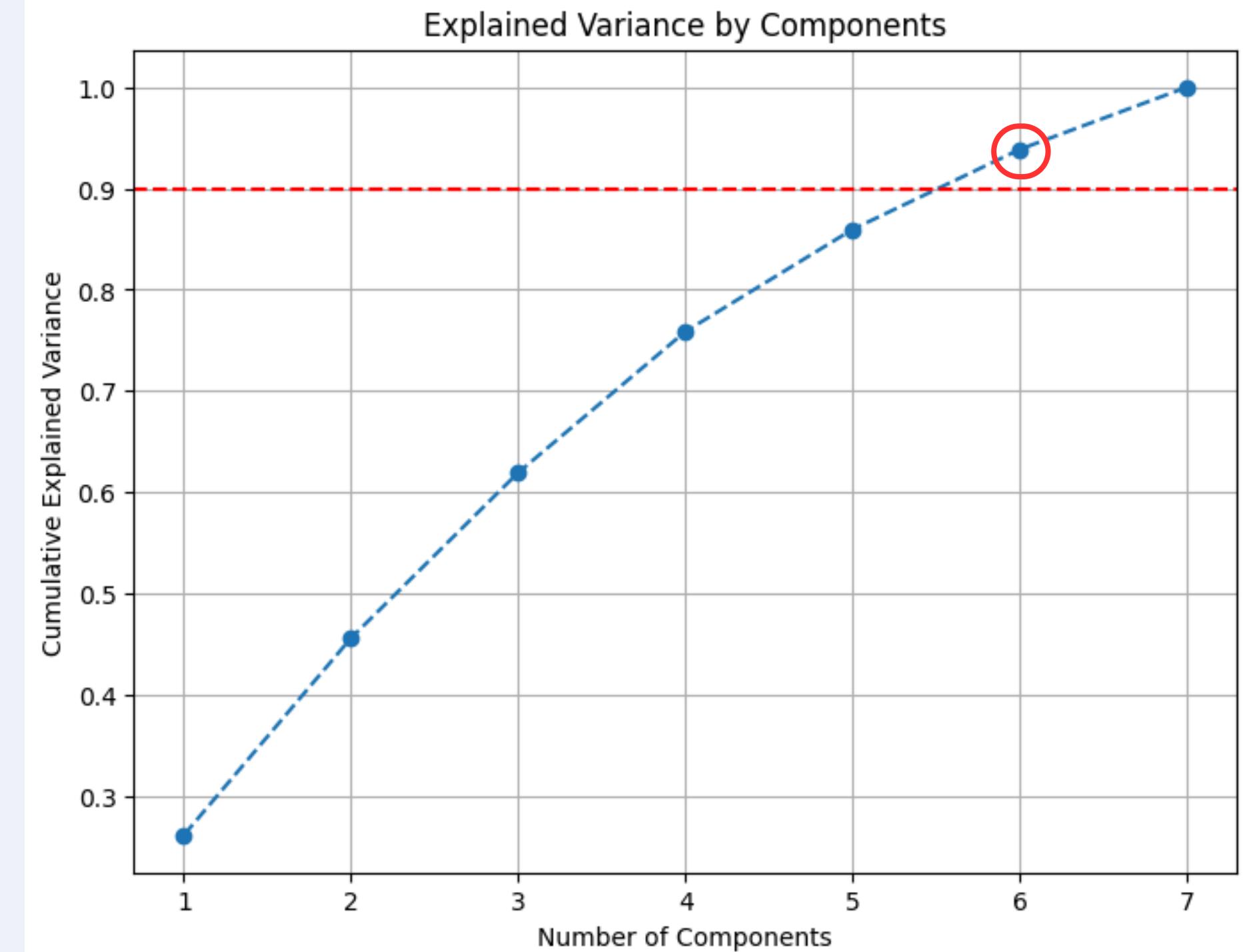




# Principle Component Analysis (PCA)



```
● ● ●  
1 import pandas as pd  
2 from sklearn.decomposition import PCA  
3  
4 data = pd.read_csv('milknew.csv')  
5  
6 X = data.drop('Grade', axis=1)  
7 y = data['Grade']  
8  
9 scaler = StandardScaler()  
10 X_scaled = scaler.fit_transform(X)  
11  
12 pca = PCA(n_components=6)  
13 X_pca = pca.fit_transform(X_scaled)
```





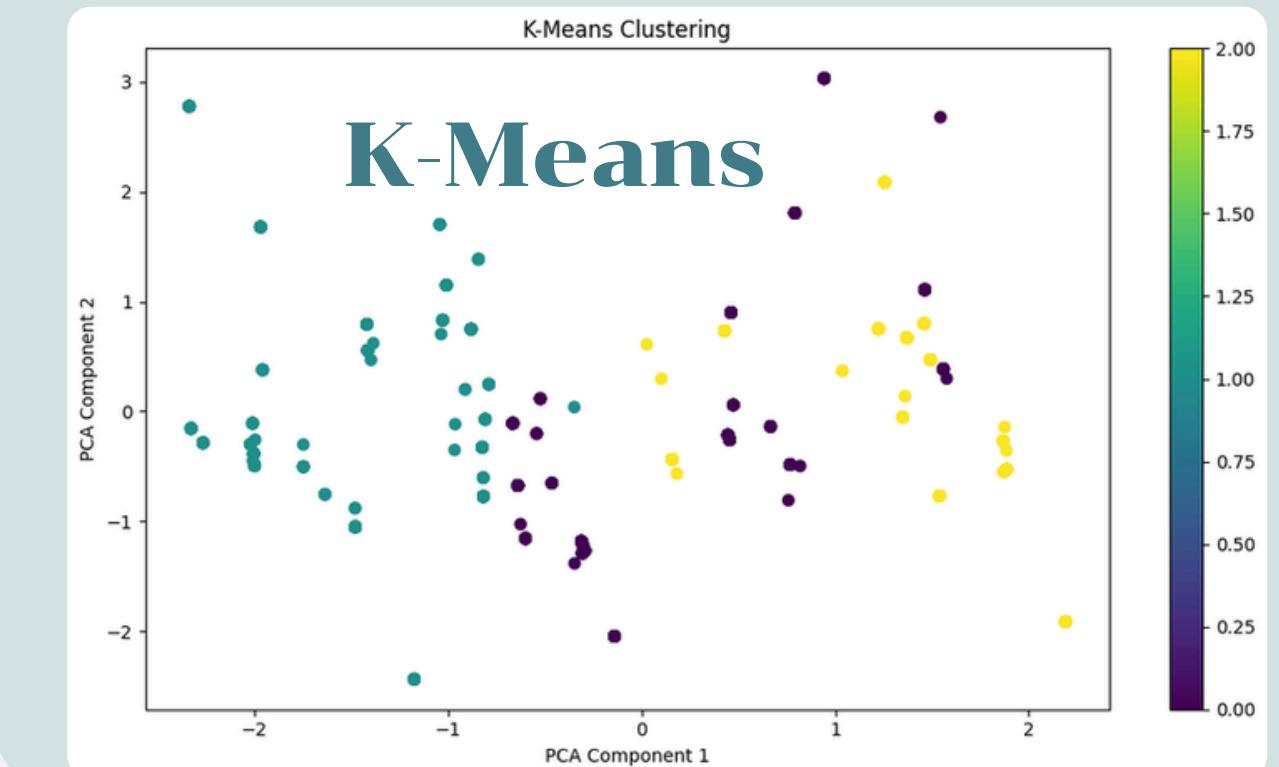
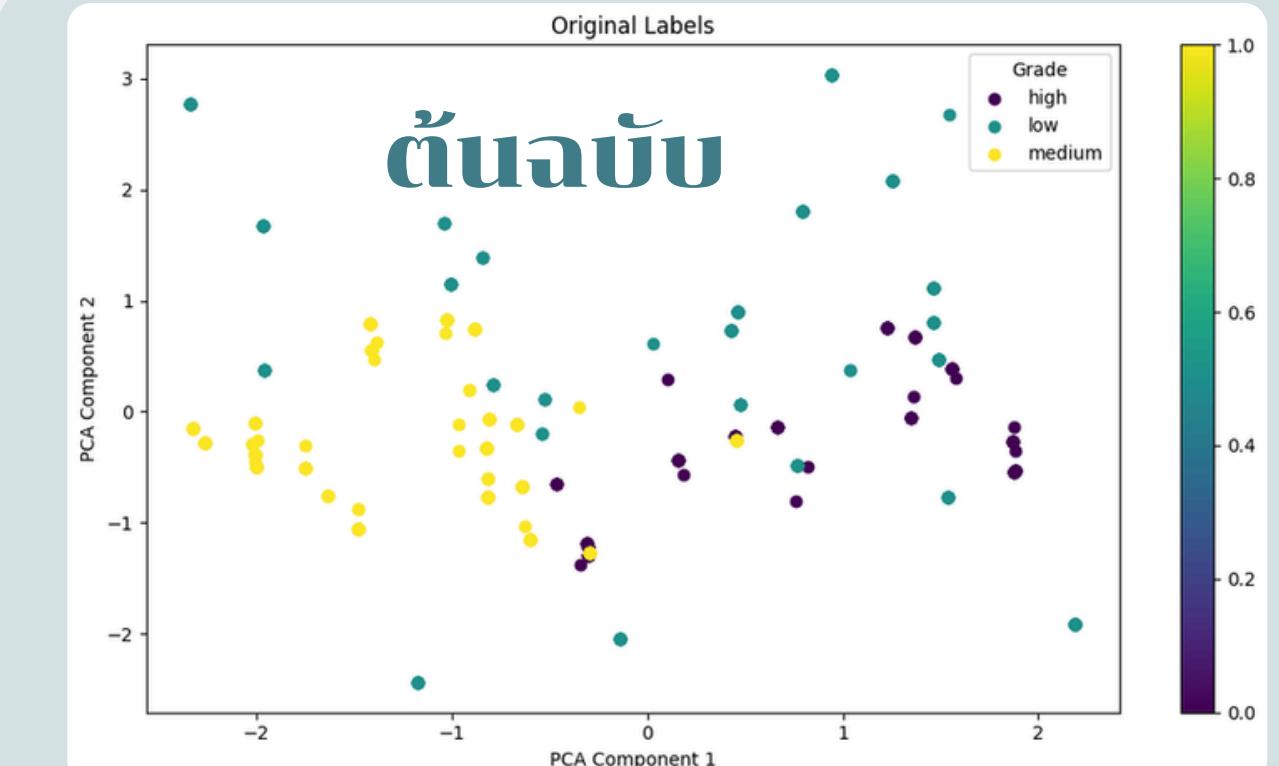
# K-mean Clustering



```
1  data = pd.read_csv('milknnew.csv')
2
3  X = data.drop('Grade', axis=1)
4  y = data['Grade']
5
6  scaler = StandardScaler()
7  X_scaled = scaler.fit_transform(X)
8
9  pca = PCA(n_components=6)
10 X_pca = pca.fit_transform(X_scaled)
11
12 kmeans = KMeans(n_clusters=3, random_state=42)
13 kmeans_labels = kmeans.fit_predict(X_pca)
```



```
1  kmeans = KMeans(n_clusters=3, random_state=42)
```



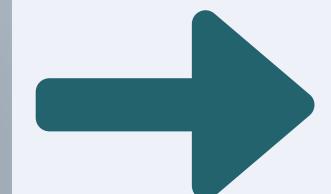


# Hierarchical Clustering



```
● ● ●  
1 data = pd.read_csv('milknew.csv')  
2  
3 X = data.drop('Grade', axis=1)  
4 y = data['Grade']  
5  
6 scaler = StandardScaler()  
7 X_scaled = scaler.fit_transform(X)  
8  
9 pca = PCA(n_components=6)  
10 X_pca = pca.fit_transform(X_scaled)  
11  
12 hierarchical = AgglomerativeClustering(n_clusters=3, metric='euclidean', Linkage='ward')  
13 hierarchical_labels = hierarchical.fit_predict(X_pca)
```

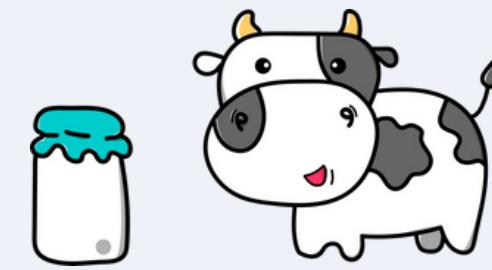
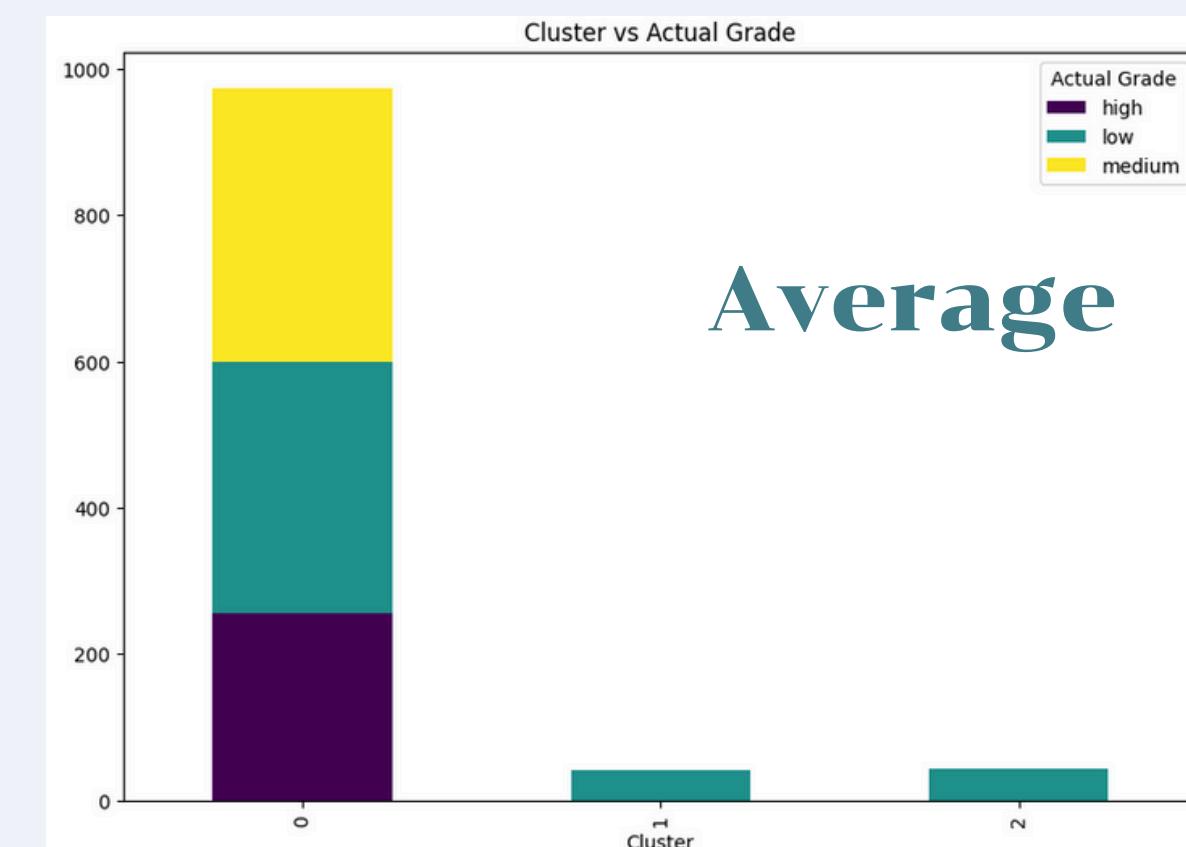
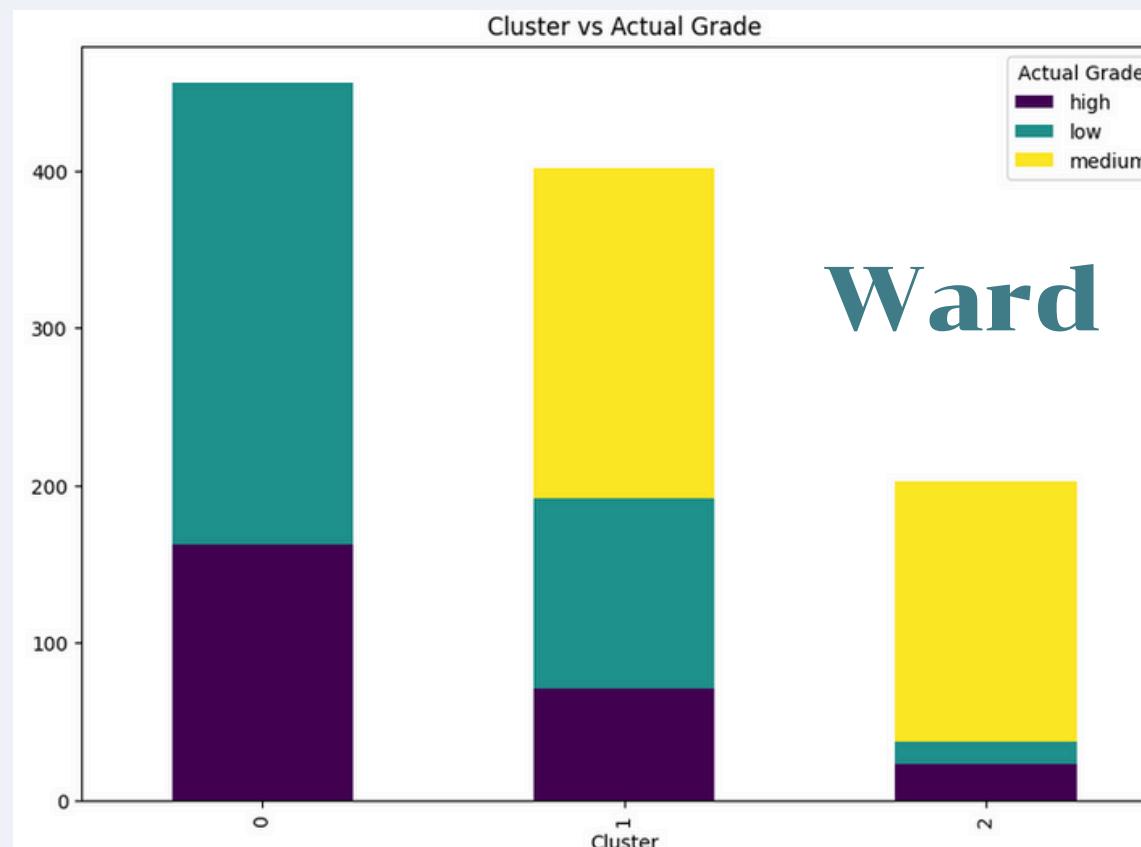
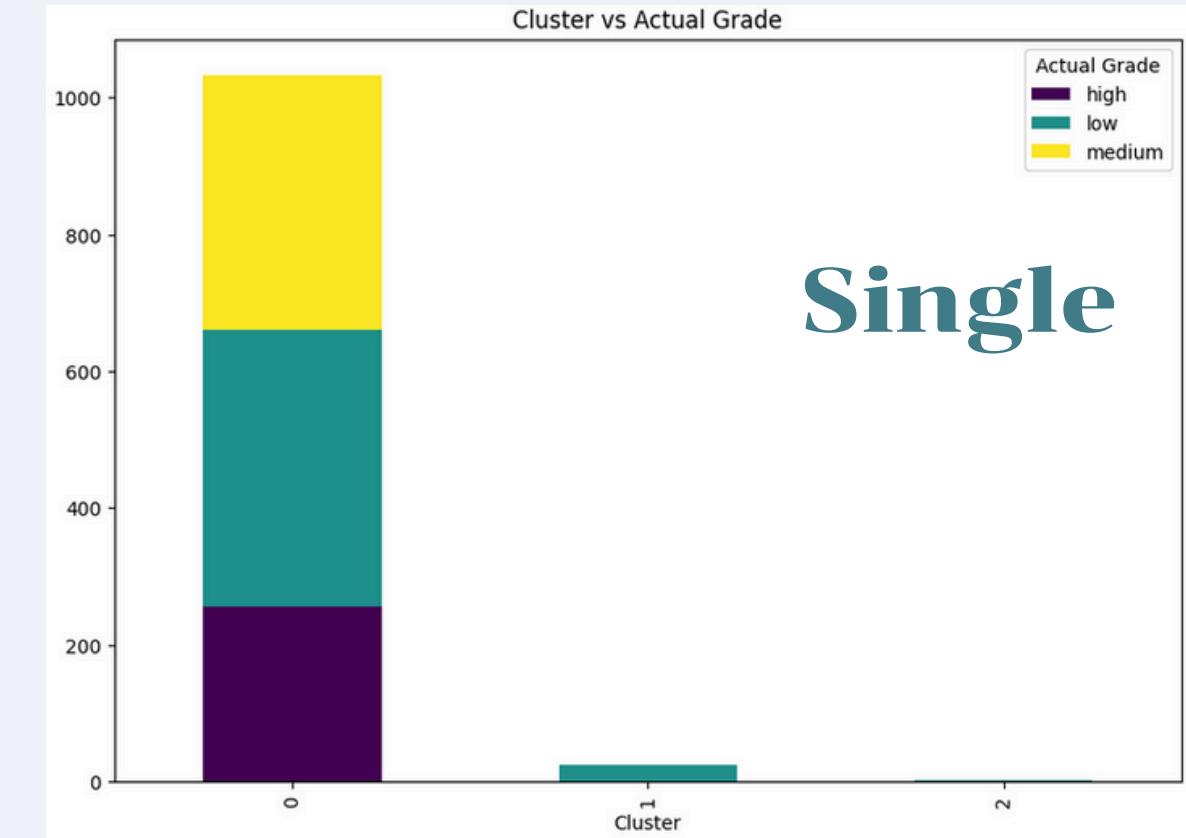
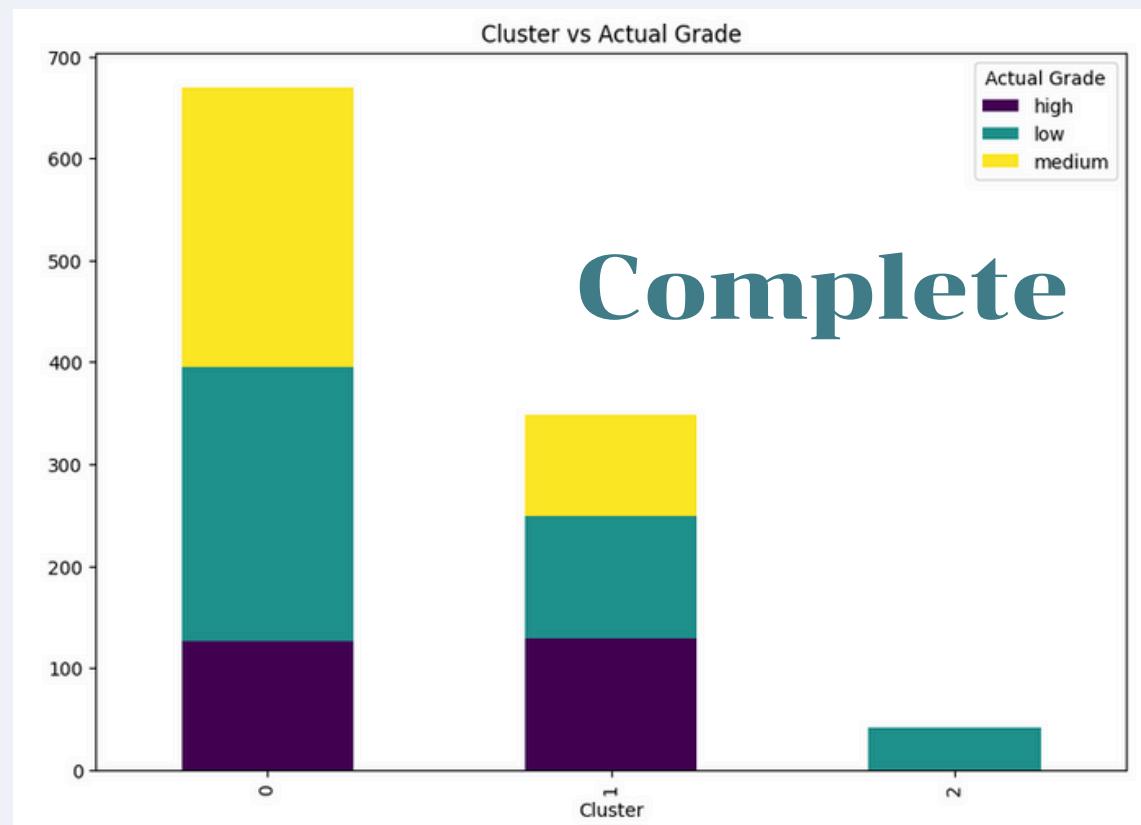
```
● ● ●  
1 hierarchical = AgglomerativeClustering(n_clusters=3, metric='euclidean', Linkage='ward')
```



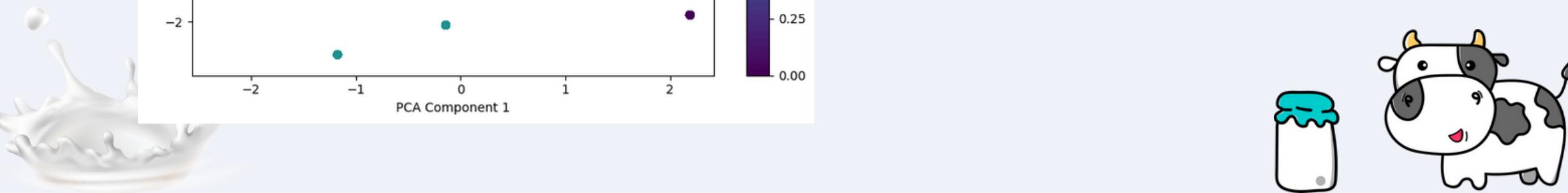
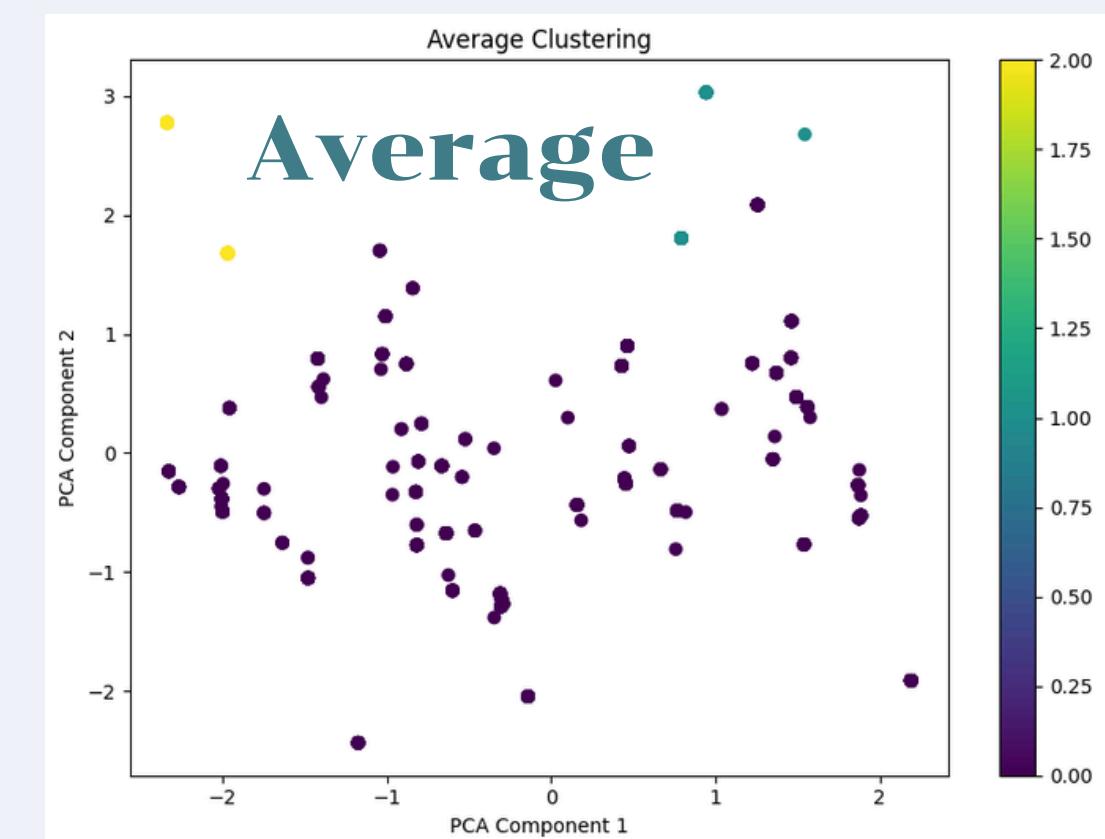
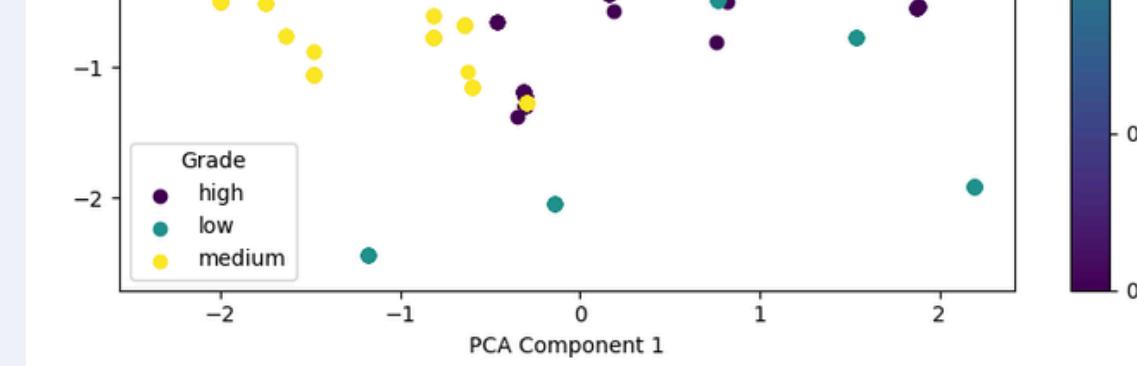
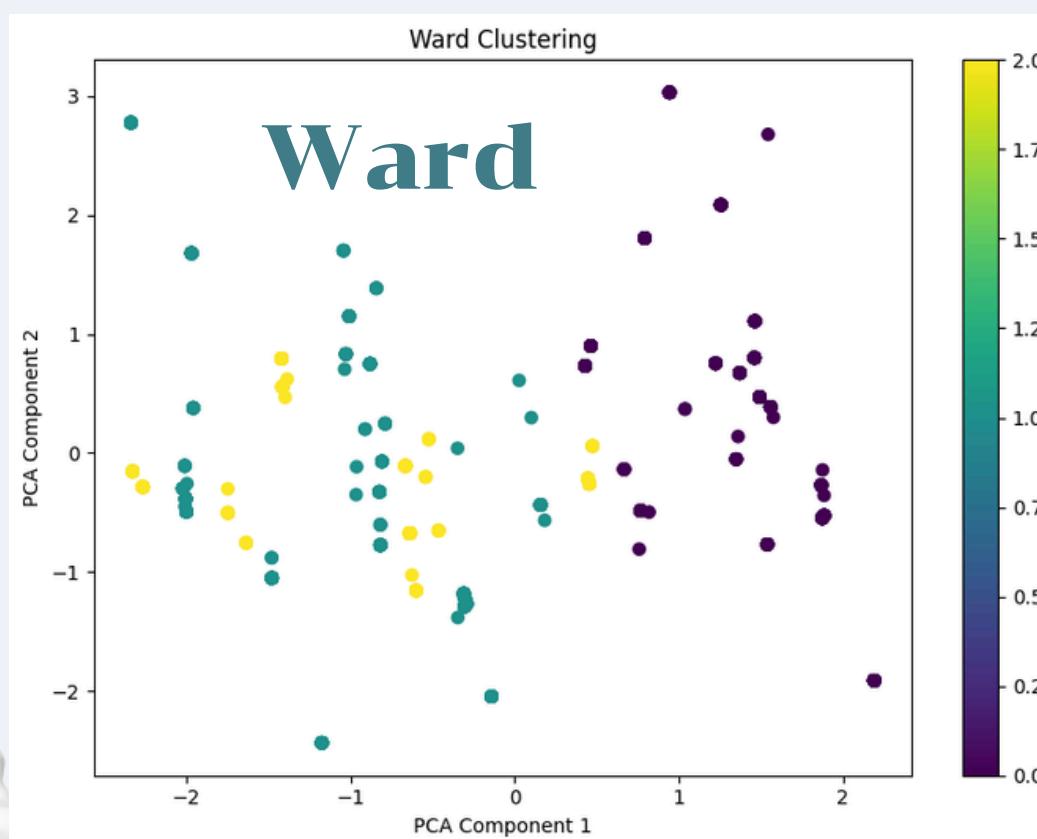
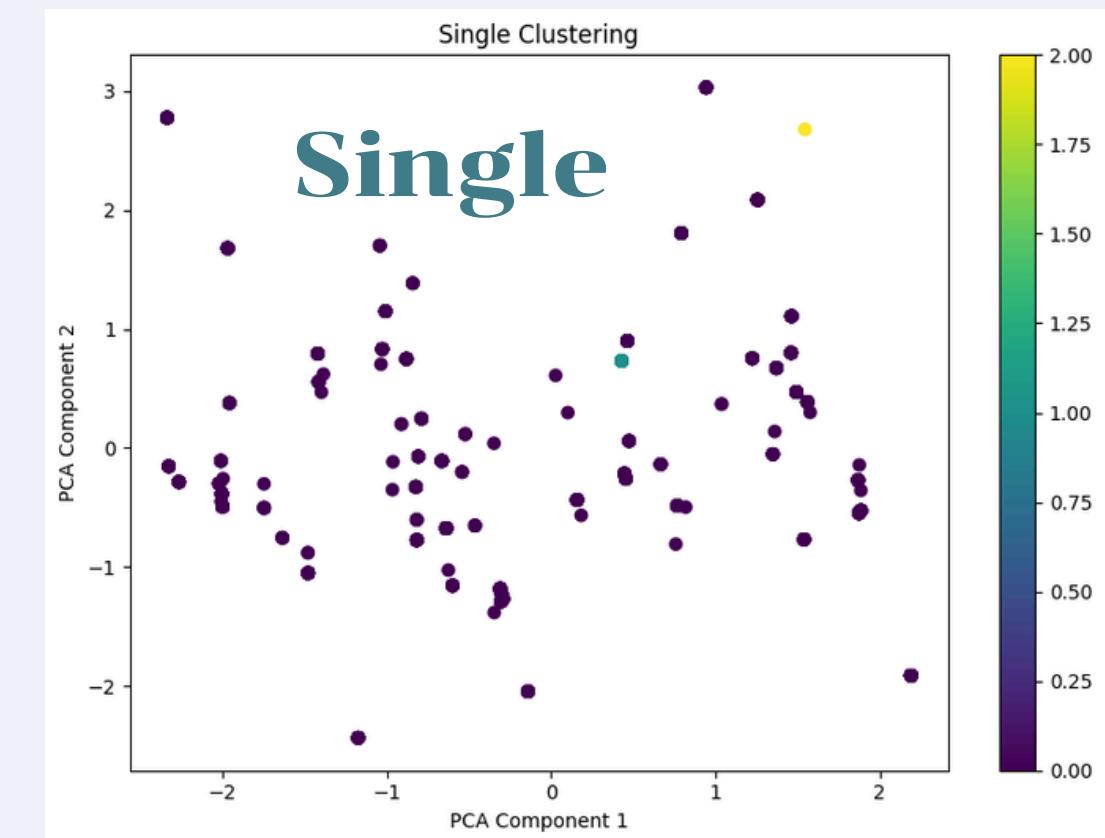
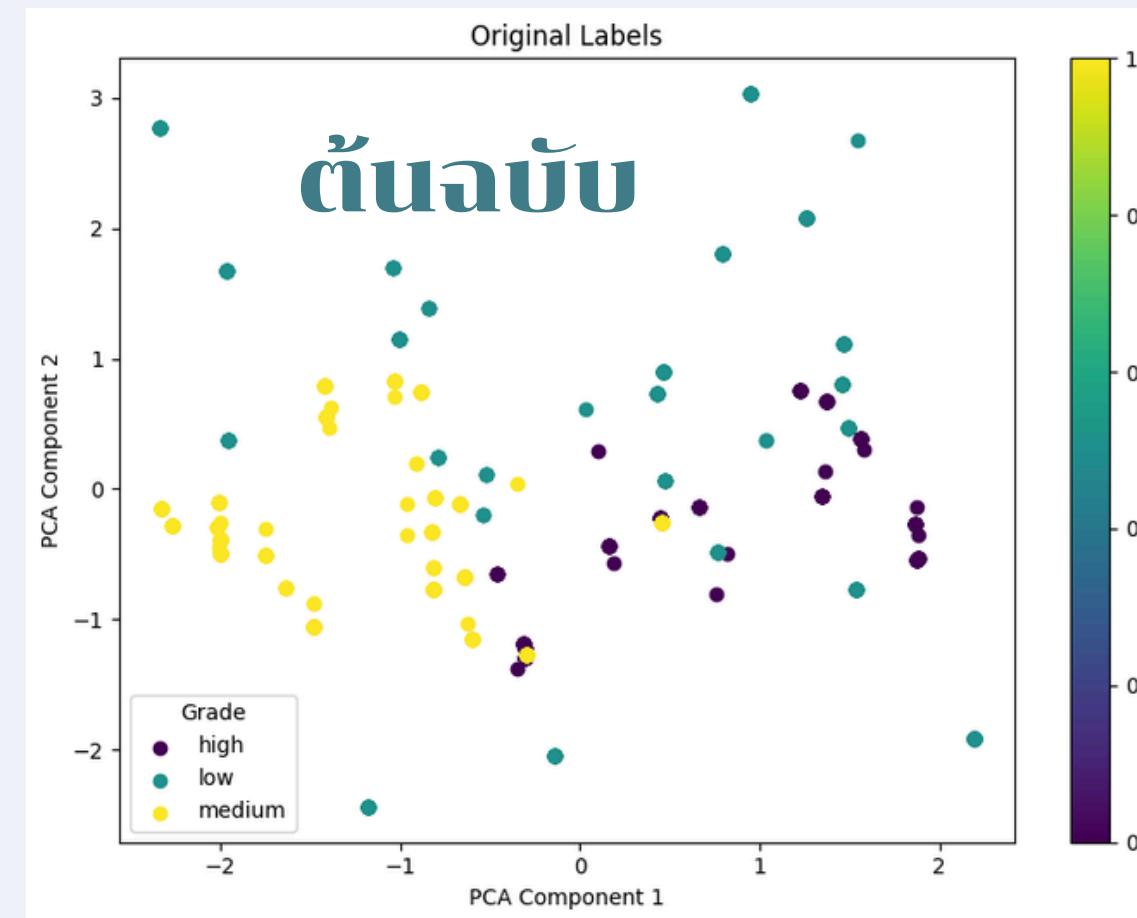
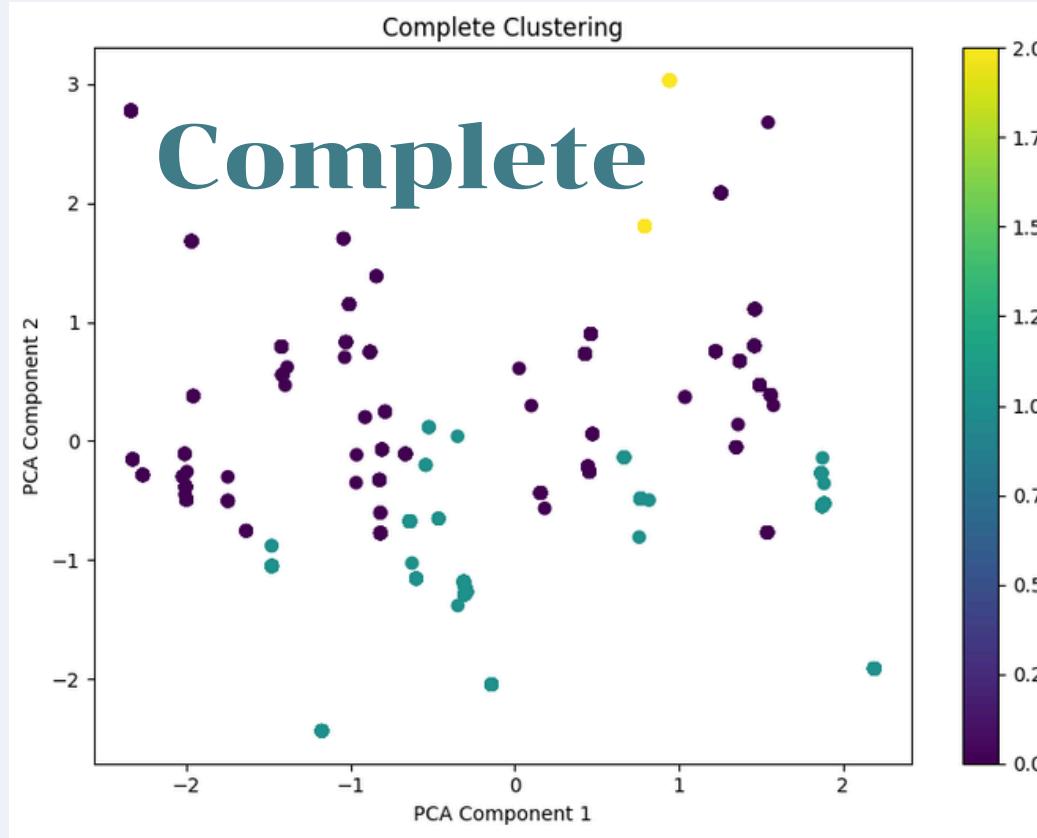
```
● ● ●  
1 Linkage='complete'  
2 Linkage='average'  
3 Linkage='single'
```



# Hierarchical Clustering



# Hierarchical Clustering



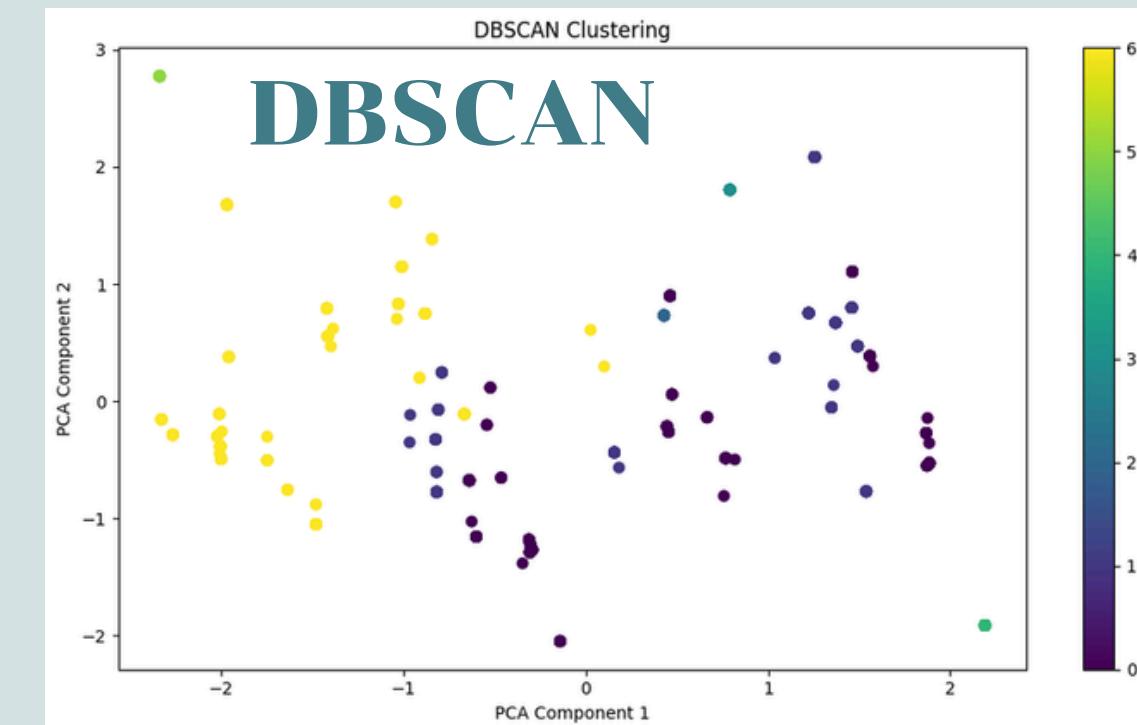
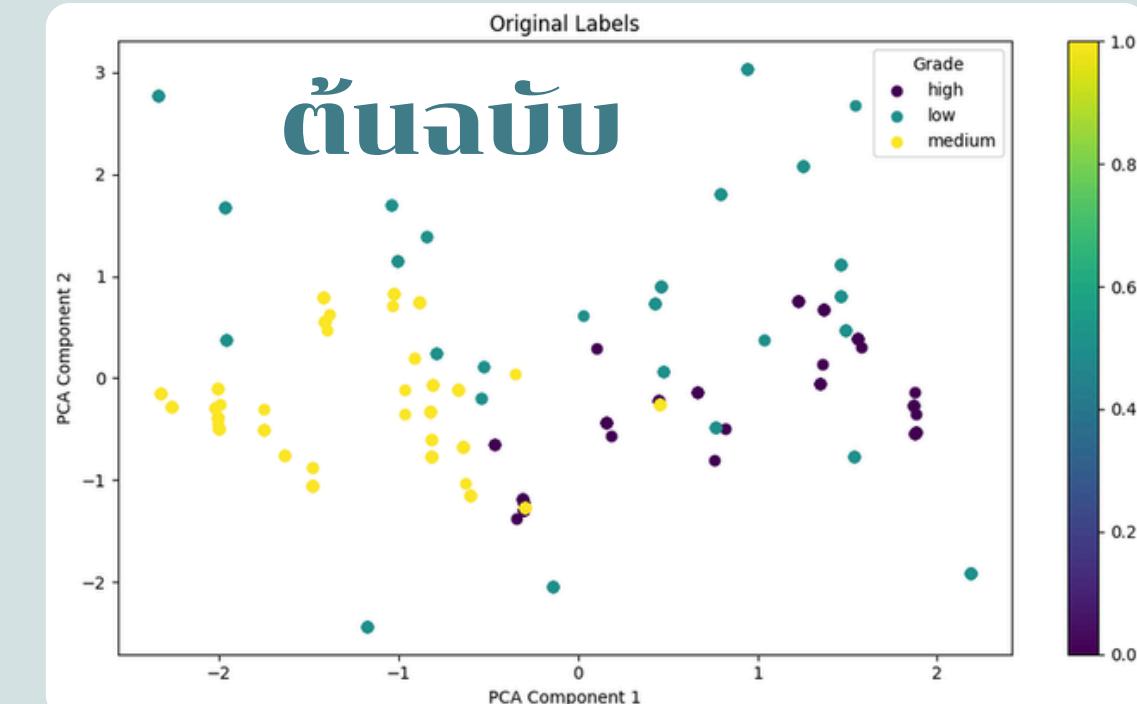
# DBSCAN Clustering



```
1 data = pd.read_csv('milknew.csv')
2
3 X = data.drop('Grade', axis=1)
4 y = data['Grade']
5
6 scaler = StandardScaler()
7 X_scaled = scaler.fit_transform(X)
8
9 pca = PCA(n_components=6)
10 X_pca = pca.fit_transform(X_scaled)
11
12 dbSCAN = DBSCAN(eps=1.8, min_samples=21)
13 dbSCAN_labels = dbSCAN.fit_predict(X_pca)
```



```
1 dbSCAN = DBSCAN(eps=1.8, min_samples=21)
```



# DBSCAN Clustering

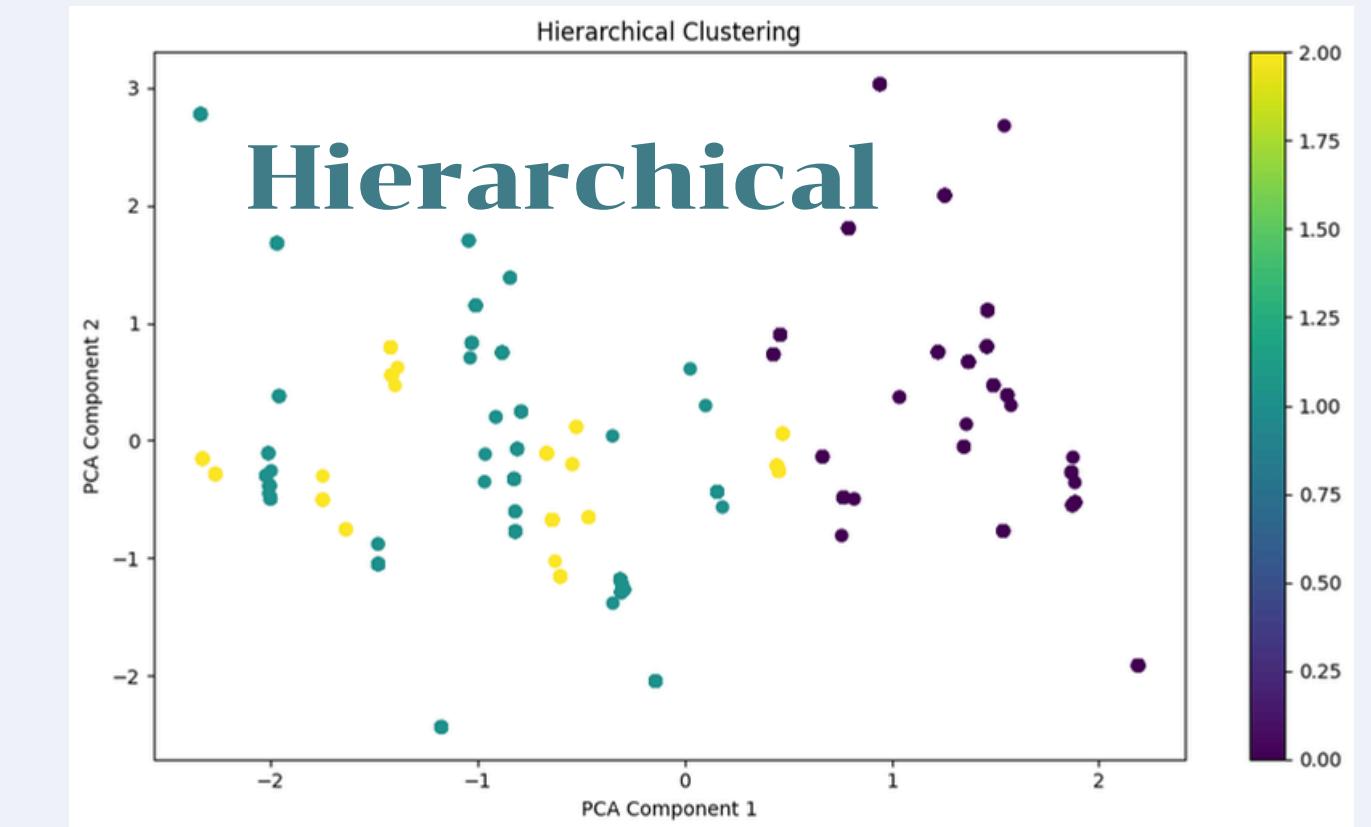
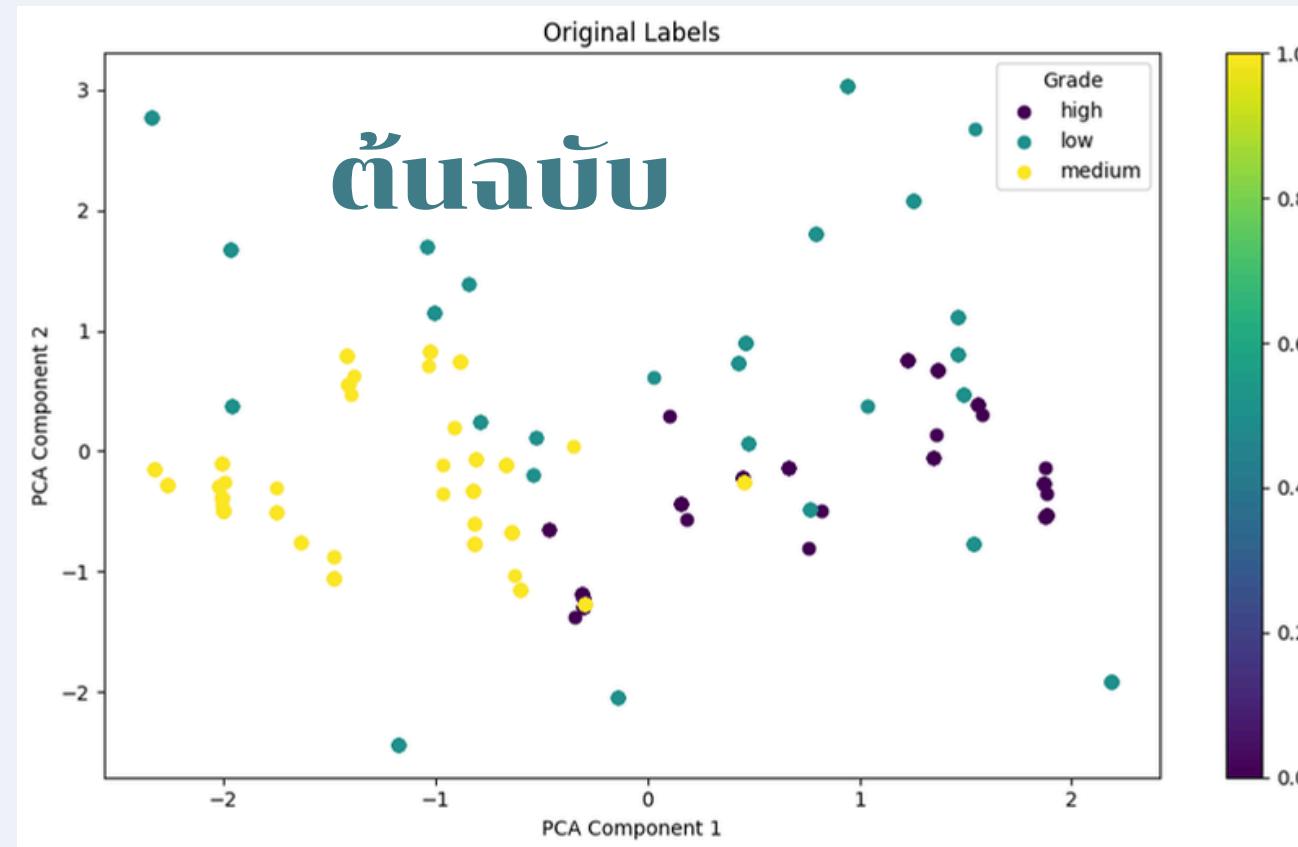
# តើអាមេរិក



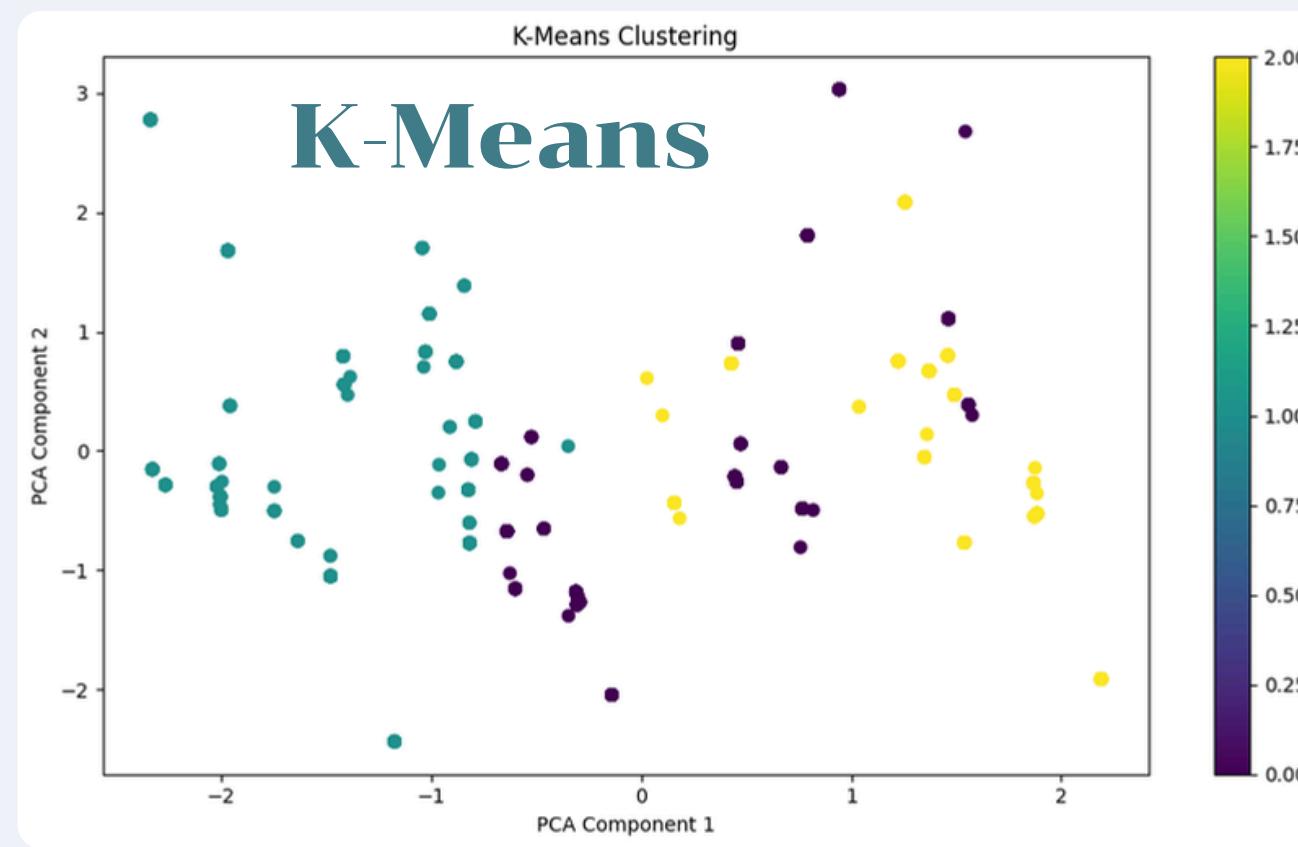
# DBSCAN

```
1 DBSCAN Labels:  
2 0 6 0 1 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1  
3 1 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1  
4 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0  
5 0 0 1 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 1  
6 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 0 1 0 0 0  
7 0 0 0 0 0 0 -1 2 0 1 3 1 1 -1 0 4 0 1 5 6 0 6 6 -1 0 2 0 1 3 1 1 -1 4 0 1 0 4 0 1 5 6 -1 0 6 6 2 2 0 1 3  
8 1 -1 1 4 1 4 0 0 5 1 5 6 -1 0 6 6 2 2 0 1 3 1 -1 1 0 6 6 -1 0 2 0 1 3 1 1 -1 4 0 1 0 4 0 0 1 3 1 1 -1 4 0  
9 1 0 4 0 1 5 6 -1 0 4 1 4 0 0 5 1 5 0 6 6 2 2 0 1 3 1 -1 1 0 6 6 -1 0 2 0 1 3 1 1 -1 4 0 1 0 4 0 0 1 3 1  
10 1 -1 4 0 1 0 4 0 1 5 6 -1 0 4 1 4 0 0 4 0 1 5 6 -1 0 6 6 2 2 0 1 3 1 -1 1 4 1 4 0 0 5 1 5 6 2 2 0 1 3 1  
11 -1 1 4 1 4 0 0 5 1 5 6 -1 0 6 6 2 2 0 1 3 1 -1 1 0 6 -1 0 6 2 0 3 4 1 6 0 4 0 1 0 5 6 0 4 -1 0 6 6 2 2 0  
12 1 3 1 -1 1 0 6 -1 0 6 2 0 3 4 1 6 0 4 0 1 0 5 6 4 0 1 0 4 0 0 1 3 1 1 -1 4 0 1 0 4 0 1 5 6 -1 0 4 1 4 0  
13 0 4 0 1 0 1 3 1 -1 1 0 6 6 -1 0 2 0 1 3 1 1 -1 4 0 1 0 4 0 0 1 3 1 1 -1 4 0 1 0 4 0 1 5 6 -1 0 1 4 0 0 4  
14 0 1 5 1 6 -1 0 6 6 0 6 2 0 3 4 1 6 0 0 1 0 6 4 1 0 4 0 0 3 1 1 0 6 6 -1 0 2 0 1 3 1 1 0 1 5 6 -1 0 4 1  
15 3 4 1 6 0 4 0 1 5 6 4 1 0 4 0 4 0 0 4 0 1 5 1 6 -1 0 6 6 0 6 2 0 3 4 1 6 6 0 6 0 1 0 6 6 6 6 6 6 6  
16 6 6 6 0 1 0 6 6 0 1 0 6 6 6 1 6 6 6 6 6 0 0 6 6 0 6 1 1 0 1 0 6 6 6 1 6 6 6 6 6 0 0 6 6 6 6 6 6 6  
17 0 1 0 6 6 0 6 6 0 1 0 6 6 0 1 0 6 6 6 1 6 0 6 6 0 6 1 1 0 1 0 6 6 6 6 0 0 6 6 6 6 6 6 6 0 1 0 6  
18 6 0 6 6 0 1 0 6 6 0 1 0 6 6 6 1 6 0 6 6 0 6 1 6 0 1 0 6 6 6 1 6 6 6 6 6 0 0 6 6 0 6 1 1 0 1 0 6 6 6  
19 0 0 6 6 0 6 1 1 0 1 0 6 6 6 1 6 6 6 6 6 0 0 6 6 6 6 6 6 6 0 1 6 6 6 1 6 0 1 6 6 6 6 6 0 0 6 6  
20 6 6 6 6 6 6 6 6 0 1 6 6 6 1 6 6 6 0 6 6 0 1 0 6 6 0 1 0 6 6 6 1 6 0 6 6 0 6 1 6 0 1 6 0 0 6 6 6 6  
21 6 6 6 0 1 0 6 6 0 6 6 0 1 0 6 6 0 1 0 6 6 6 1 6 0 6 6 6 0 1 6 0 1 0 6 6 1 6 6 6 6 6 6 6 6 0 1 6 1  
22 6 1 6 6 0 6 1 6 0 6 6 6 6 6 6 0 -1 0 1 0 6 6 6 6 0 6 6 6 0 1 6 6 6 6 1 6 6 6 6 6 0 6 0 1 6 0 1 0 6 6  
23 1 6 6 6 6 6 6 6 6
```

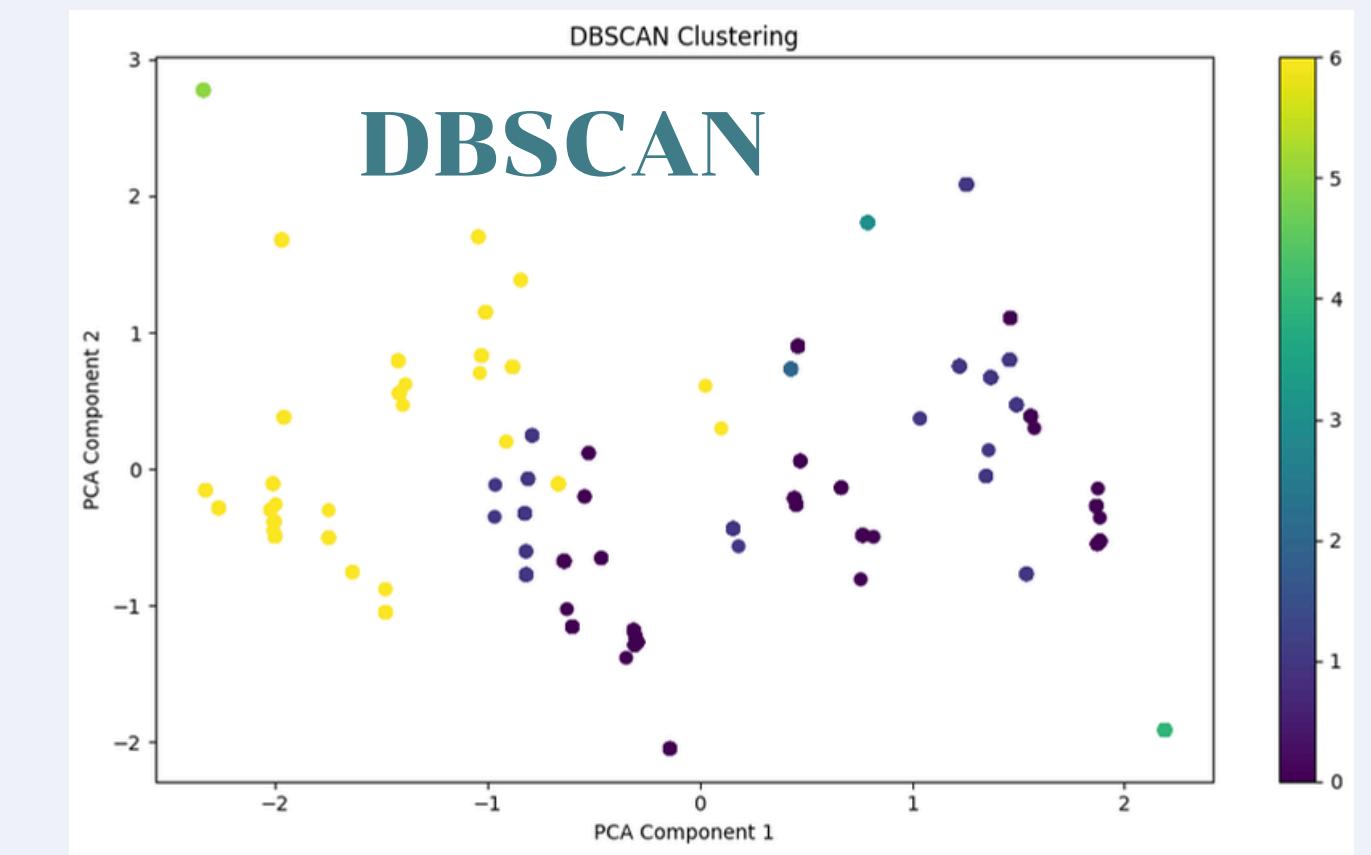
# Summary



```
hierarchical = AgglomerativeClustering(n_clusters=3, metric='euclidean', Linkage='ward')
```



```
kmeans = KMeans(n_clusters=3, random_state=42)
```



```
dbscan = DBSCAN(eps=1.8, min_samples=21)
```





# Unsupervised Machine Learning: **Milk Quality Clustering**

🌐 <https://www.kaggle.com/datasets/cpluzshrijayan/milkquality>