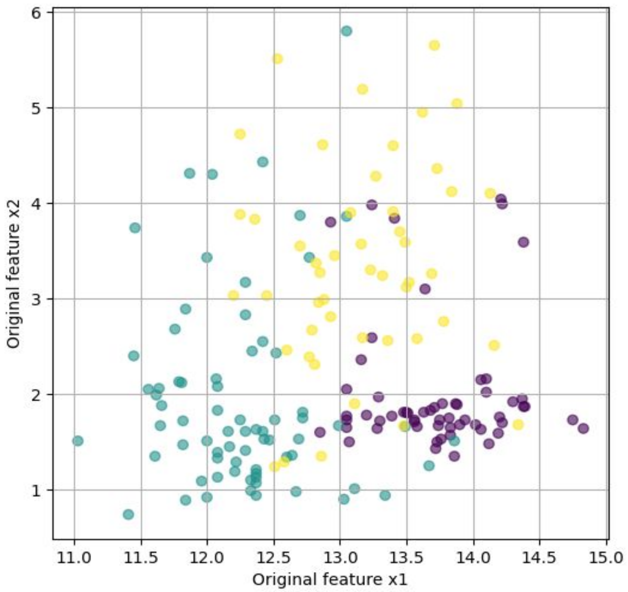
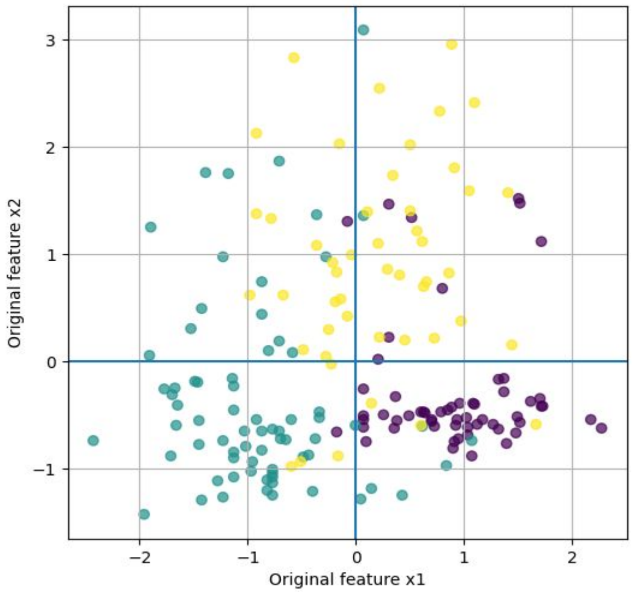
Preparing the Data for PCA & Finding Principal Directions



Overview of the dataset:
Each point represents a wine sample.
The dataset contains 177 samples, each described by 13 features.

$$X = X_{raw} - 1\mu$$

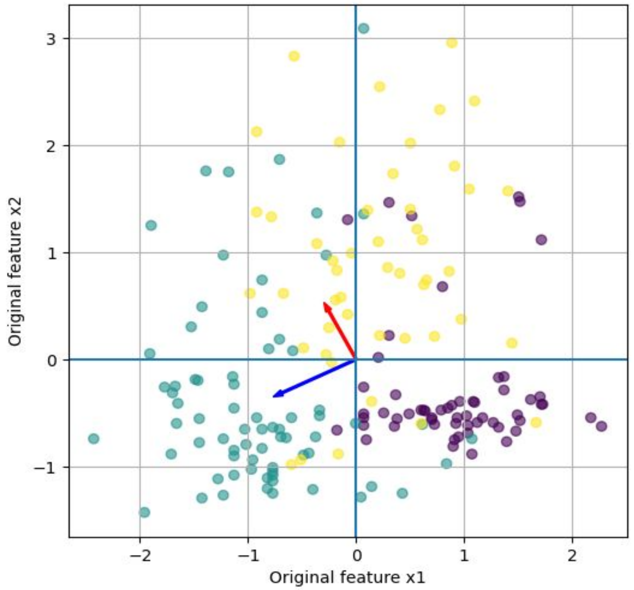
$$X_{std,ij} = \frac{X_{ij}}{\sigma_j}$$



Centring aligns data around zero so PCA captures true variance. Standardisation scales features equally, preventing large values from dominating.

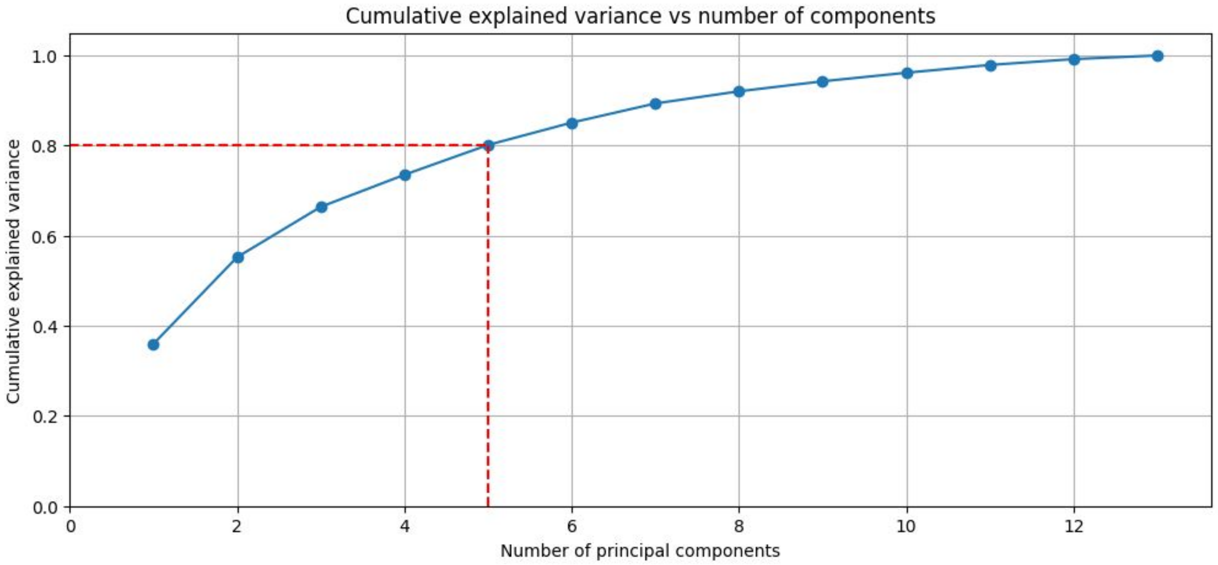
$$C\, v_i = \lambda_i\, v_i$$

$$C = \frac{1}{n\text{-}1} X_{std}^T X_{std}$$



Eigenvectors plotted in the centered and standardized data space. PCA has not yet been applied.

Choosing Components and Projection



$$\mathrm{VarRatio}_i = \frac{\lambda_i}{\Sigma_{j=1}^{d} \lambda_j}$$

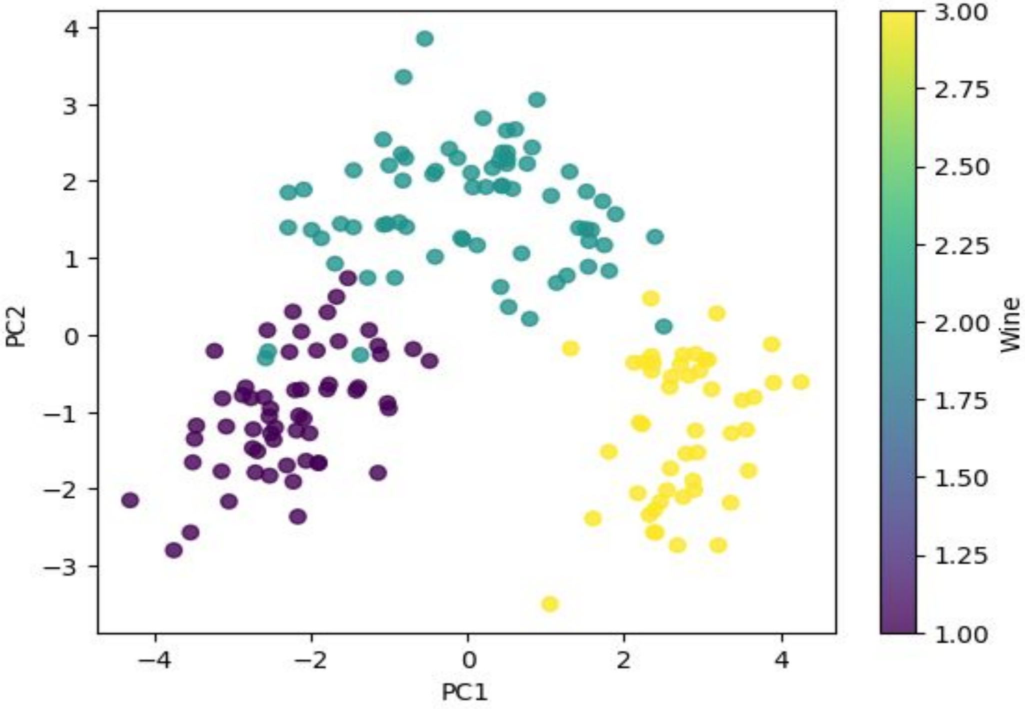
$$\mathrm{CumVar}_k = \sum_{i=1}^{k} \mathrm{VarRatio}_i$$

This quantity is used to determine the smallest number of principal components k that retain a desired proportion of the total variance.

```
k  = np.argmax(cumulative_variance >= 0.8) + 1
```

After selecting k principal components, the projected data is:

$$X_{\text{proj}} = X_{\text{std}} V_k$$



What I Learnt

- PCA is an eigen-decomposition of the covariance matrix, not a black-box algorithm.
- Variance explained is directly linked to eigenvalues, not eigenvector.
- Dimensionality reduction is a linear projection that preserves variance, not labels.
- Standardisation changes the covariance structure, so PCA results depend strongly on preprocessing.

What were the Challenges?

- Implementing PCA without relying on closed-form eigenvalue formulas, since $\det(A-\lambda I)=0$ is infeasible for high-dimensional data.
- Understanding the distinction between principal directions (eigenvectors) and projected data (scores in PC space.
- Choosing the number of components using cumulative explained variance instead of an arbitrary k.
- Interpreting principal components after standardization, since they are defined in a unitless feature space.