



Refer to both, code files and manual. And then ask questions and query to perform. Can you also be considerate enough to provide the answers as well ☐

Absolutely! I'll select one experiment and provide:

- Code-based viva questions
- Theory-lab-manual-based questions
- At each tier: Ask the question, *give* the model answer
And I'll include a query to perform (small code or interpretative task) with an example answer.

Let's do this **for Experiment 1: PCA on the Wine Dataset** as an example:

Experiment 1: PCA for Dimensionality Reduction (Wine Data)

BASIC

Q1: What is the primary objective of applying PCA to the wine dataset?

A: The main goal is to reduce the number of features (dimensions) while retaining most of the important information (variance) in the data. PCA transforms the original correlated variables into a smaller set of new uncorrelated variables called principal components, making it easier to visualize and analyze the data, and to distinguish between red and white wines.^[1]

Q2: Why must features be standardized before applying PCA?

A: PCA is sensitive to the scale of data. Standardizing ensures each feature has mean 0 and variance 1, so no feature dominates due to a larger original scale.

Q3: How do you interpret "explained variance ratio" in PCA?

A: It tells you how much information (variance) each principal component accounts for, relative to total variance in the data. The higher the explained variance, the more important that component is.

INTERMEDIATE

Q4: Outline the steps you follow in code to perform PCA on the wine dataset.

A:

- Standardize the features (using `StandardScaler`)
- Compute the covariance matrix of standardized data
- Calculate eigenvalues and eigenvectors of this matrix
- Order eigenvectors by decreasing eigenvalues
- Select the top-k components (based on cumulative explained variance)
- Project the data onto these principal components^[1]

Q5: In your code, after PCA, what kind of plot would you use to visualize separation between red and white wine?

A:

A scatter plot of the first two principal components, with points colored or marked by wine type (red/white). This helps visualize whether the PCA-projected data groups wines by type.

Code Query:

Suppose `wine_pca` is your PCA-transformed dataset. Write code to plot the first two principal components and color by class.

```
import matplotlib.pyplot as plt

# Assuming y contains class labels (red = 0, white = 1)
plt.scatter(wine_pca[:, 0], wine_pca[:, 1], c=y, cmap='coolwarm', alpha=0.7)
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.title('PCA: First two principal components')
plt.colorbar(label='Wine Type')
plt.show()
```

ADVANCED

Q6: What is a potential limitation of using only two principal components for visualization?

A: While 2D plots are easy to visualize, the first two components may not capture enough variance—important information might be contained in additional components. This can lead to incomplete separation or loss of relevant data structures.

Q7: If the first two principal components only explain 55% of the variance, how many should you keep, and how would you decide this in Python?

A: Often, you retain enough PCs to explain around 90-95% variance. In Python's `sklearn`, you can set `n_components` as a float and check cumulative variance:

```
from sklearn.decomposition import PCA

# Retain enough PCs to explain 95% of variance
```

```
pca = PCA(n_components=0.95)
wine_reduced = pca.fit_transform(X_scaled)
print('Number of components chosen:', pca.n_components_)
```

This will let the algorithm select as many PCs as needed to reach 95% explained variance.

Q8: Are the principal components "unique"? Why/why not?

A: Principal components are unique up to sign (direction)—they form an orthogonal basis of eigenvectors. However, each eigenvector's sign can be flipped without affecting variance explained or their orthogonality.

Would you like this format for another experiment? Just name the experiment and I'll do code/manual-derived questions, + sample answers and code snippets where needed!

✱

1. Computer-Laboratory-I-ML-Lab-Manual_print_ready.pdf