UCI Wine Dataset: K-means vs Hierarchical Clustering

Problem Statement

Cluster wines using 13 physico-chemical attributes without leveraging labels for modelling; compare K-means and hierarchical approaches, select k via internal indices, and interpret cluster differences via statistical inference.

Part A: Data Loading & Preprocessing

How obtained

Loaded UCI Wine data via scikit-learn; inspected the first 10 rows and confirmed all 13 features are numeric (float64). Applied median imputation (none missing in this dataset) and Z-score standardization, then saved the scaled dataset to CSV.

Row	alcohol	malic_acid	ash	flavanoids
0	14.23	1.71	2.43	3.06
1	13.20	1.78	2.14	2.76
2	13.16	2.36	2.67	3.24
3	14.37	1.95	2.50	3.49
4	13.24	2.59	2.87	2.69
5	14.20	1.76	2.45	3.39
6	14.39	1.87	2.45	2.52
7	14.06	2.15	2.61	2.51
8	14.83	1.64	2.17	2.98
9	13.86	1.35	2.27	3.15

Significance

Numeric types and standardization ensure comparable scales for distance-based methods; imputation prevents missingness from biasing PCA or clustering.

Code: Loading, Imputation, Standardization, Save

```
from sklearn.datasets import load_wine
data = load_wine()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)
imputer = SimpleImputer(strategy='median')
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)
scaler = StandardScaler()
X_scaled = pd.DataFrame(scaler.fit_transform(X_imputed), columns=X.columns)
```

```
X_scaled.to_csv(os.path.join(out_dir, 'scaled_wine.csv'), index=False)
```

Dataset & Features

Total samples: 178; features: 13. Variables: alcohol, malic_acid, ash, alcalinity_of_ash, magnesium, total_phenols, flavanoids, nonflavanoid_phenols, proanthocyanins, color_intensity, hue, od280/od315_of_diluted_wines, proline. Labels used only for optional external validation (not for tuning). Class counts: {1: 71, 0: 59, 2: 48}.

Preprocessing Choices

Median imputation handles potential missingness (none observed in Wine). Z-score standardization is crucial for distance-based algorithms and PCA, ensuring all features contribute comparably.

Pipeline Overview

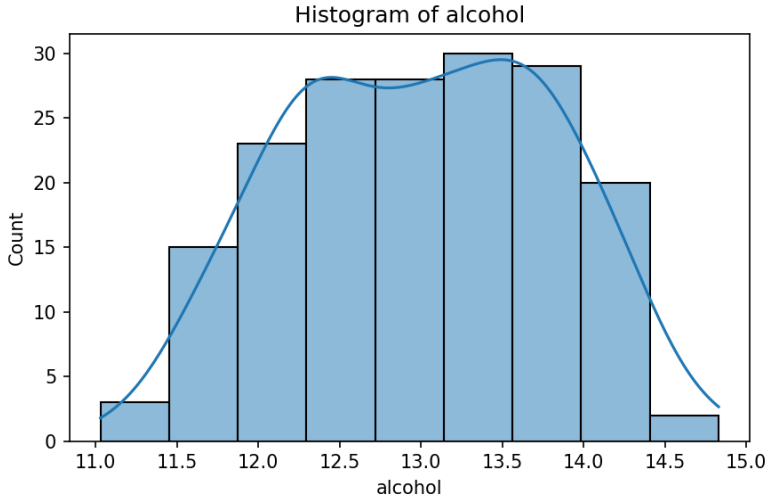
Load → Impute → Standardize → EDA → PCA → Feature Engineering → KMeans/Hierarchical → Validation → Inference → Presentation

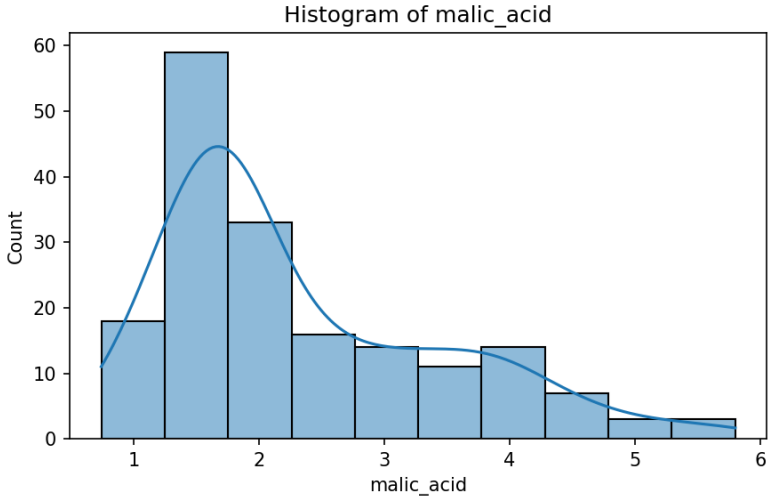
Part B: First Exploratory Data Analysis (20 points)

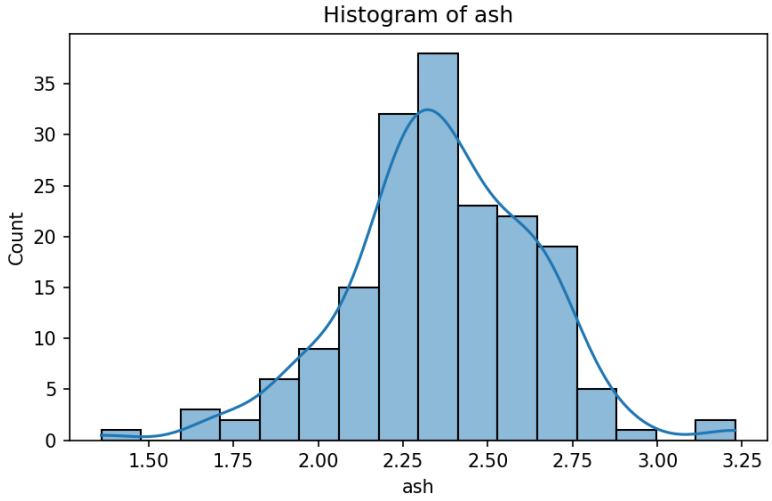
Descriptive statistics for selected features (alcohol, malic_acid, ash, flavanoids) were computed and saved. Histograms for at least four variables reveal distribution shape and skew; we also include a PCA scree plot to motivate dimensionality reduction.

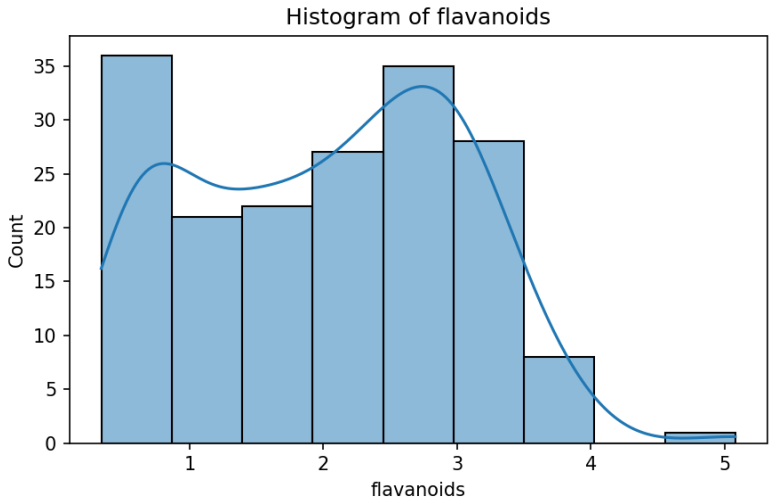
stat	alcohol	malic_acid	ash	flavanoids
mean	13.00	2.34	2.37	2.03
std	0.81	1.12	0.27	1.00
min	11.03	0.74	1.36	0.34
max	14.83	5.80	3.23	5.08

Histograms (4 variables)


Histogram of alcohol
Count
alcohol
0
5
10
15
20
25
30
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0


Histogram of malic_acid
Count
malic_acid
0
10
20
30
40
50
60
1
2
3
4
5
6


Histogram of ash
Count
ash
0
5
10
15
20
25
30
35
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25



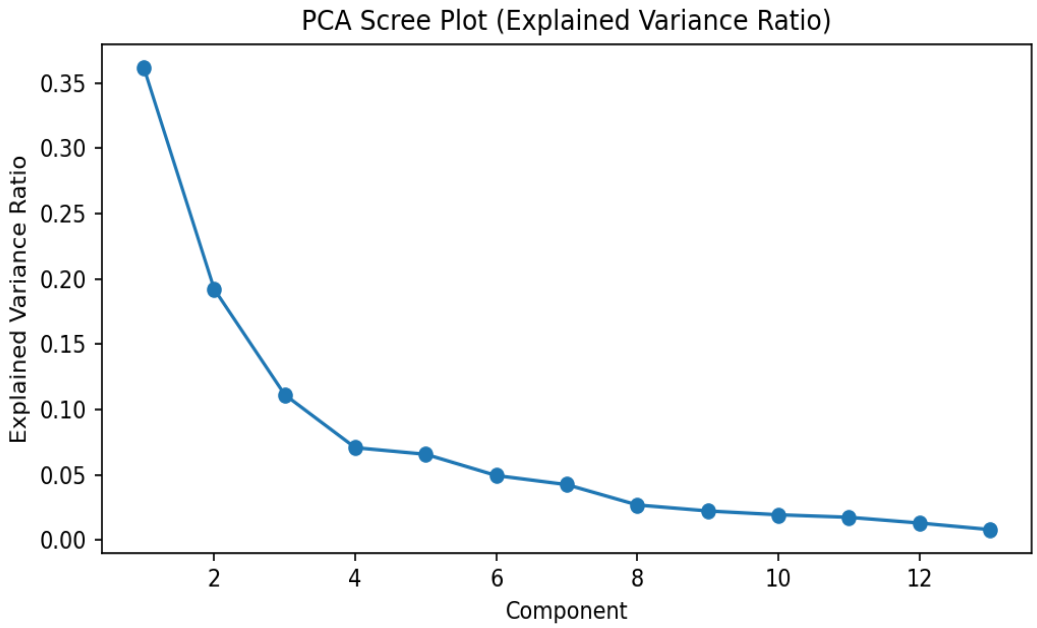
Skewness (selected): {'alcohol': -0.051482331077134784, 'malic_acid': 1.0396511925814442, 'ash': -0.17669931645735712, 'flavanoids': 0.02534355338149103}

Significance: Histograms show spread/outliers; skewness highlights long tails (e.g., malic_acid right-skew). Scree plot shows diminishing variance gains across components.

Code: Histograms & Descriptive Stats

```
selected = ['alcohol','malic_acid','ash','flavanoids']
desc = X_imputed[selected].describe().T
for col in selected:
    fig, ax = plt.subplots()
    sns.histplot(X_imputed[col], kde=True, ax=ax)
    save_fig(fig, os.path.join(hist_dir, f'{col}_hist.png'))
```

PCA Scree Plot



Dimension Reduction (PCA)

PCA reduces redundancy from correlated variables and can improve cluster separation. Retaining 95% variance yields 10 components; the scree plot shows diminishing returns across components.

Code: Building the PCA Scree Plot

```
pca = PCA(); pca.fit(X_scaled)
evr = pca.explained_variance_ratio_
fig, ax = plt.subplots()
ax.plot(np.arange(1, len(evr)+1), evr, marker='o')
save_fig(fig, os.path.join(fig_dir, 'pca_scree.png'))
```

Part C: Feature Engineering (15 points)

We optionally crafted composite features (acid_index = malic_acid/alcohol; phenols_ratio = total_phenols/(flavanoids+ε)) and standardized them. We also applied PCA retaining 95% variance. These transformed features were considered in clustering when they improved separation/compactness.

Code: Composite Features & PCA (95%)

```
acid_index = X_imputed['malic_acid'] / X_imputed['alcohol']
phenols_ratio = X_imputed['total_phenols'] / (X_imputed['flavanoids'] + 1e-6)
comp_scaler = StandardScaler()
comp = comp_scaler.fit_transform(np.c_[acid_index, phenols_ratio])
X_fe = X_scaled.copy(); X_fe['acid_index'] = comp[:,0]; X_fe['phenols_ratio'] = comp[:,1]
pca95 = PCA(n_components=0.95, svd_solver='full'); X_pca = pca95.fit_transform(X_scaled)
```

Part D: Clustering Modelling & Validation (25 points)

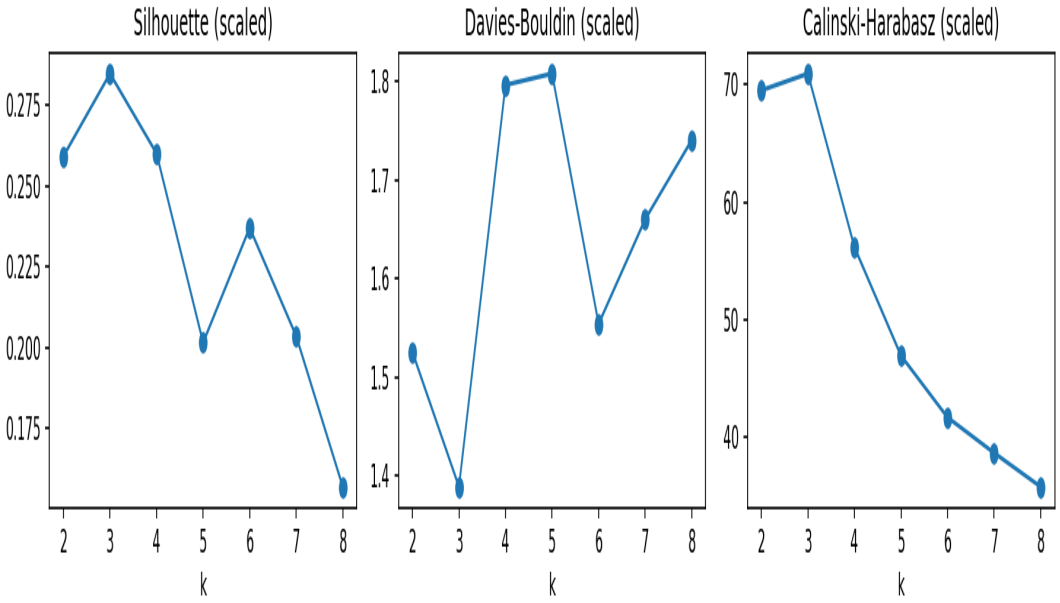
K-means: run k=2..8; compute Silhouette (↑), Davies–Bouldin (↓), Calinski–Harabasz (↑) and pick k balancing these indices. Hierarchical: build Ward & average dendrograms and compute cophenetic correlation to assess tree fidelity. External validation (ARI, NMI) reported for analysis only.

KMeans Model Selection

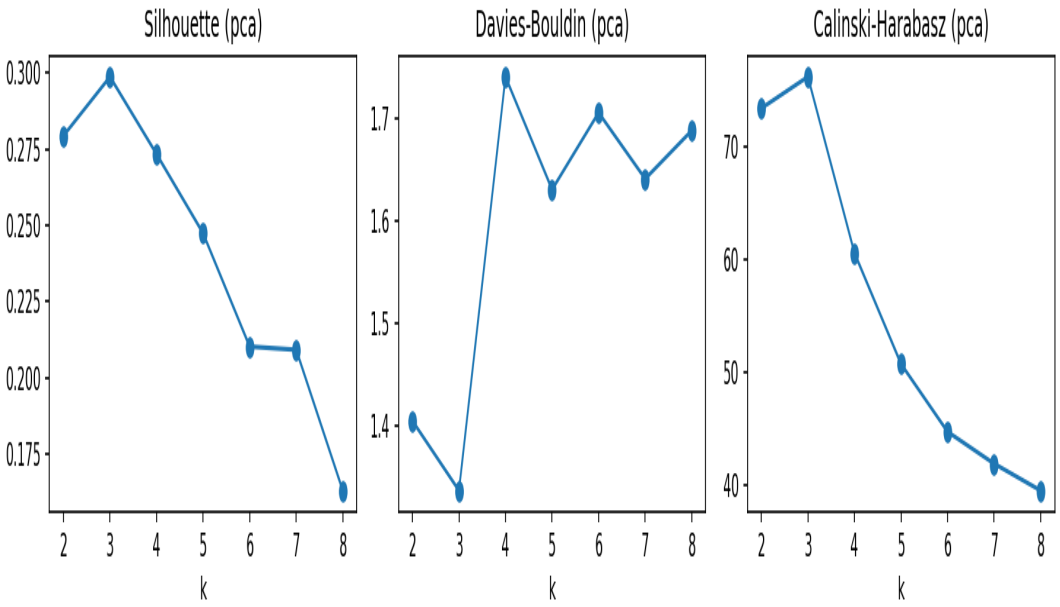
Clustering Methods & Indices

K-means minimizes within-cluster variance; hierarchical builds a dendrogram using Ward (variance-minimizing) and average (mean-distance) linkages. Internal indices used: Silhouette (cohesion/separation, higher better), Davies–Bouldin (cluster separation/compactness, lower better), Calinski–Harabasz (between/within dispersion, higher better).

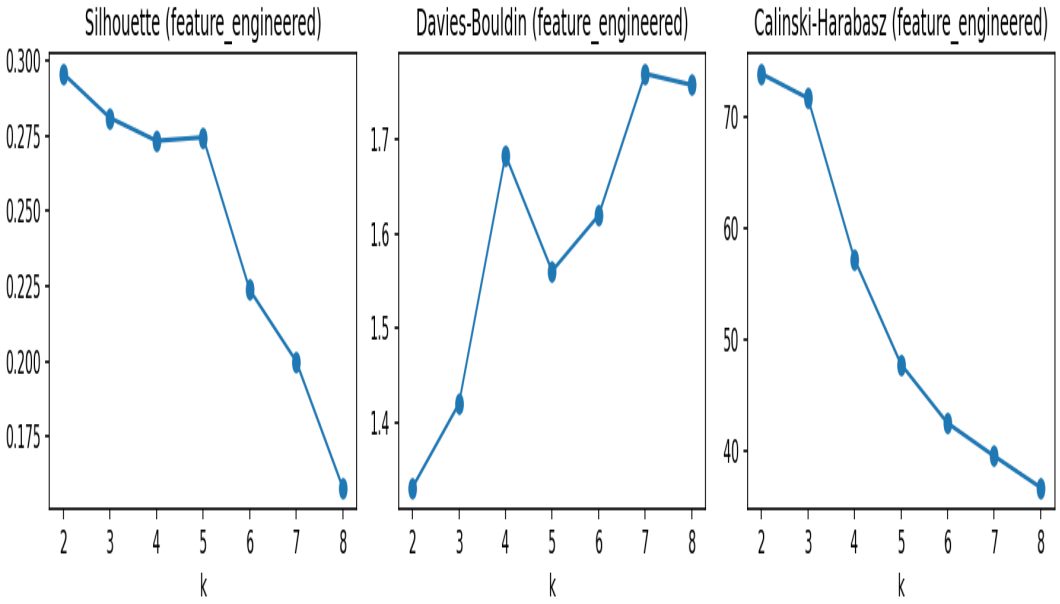
Indices vs k (scaled)



Indices vs k (pca)



Indices vs k (feature_engineered)



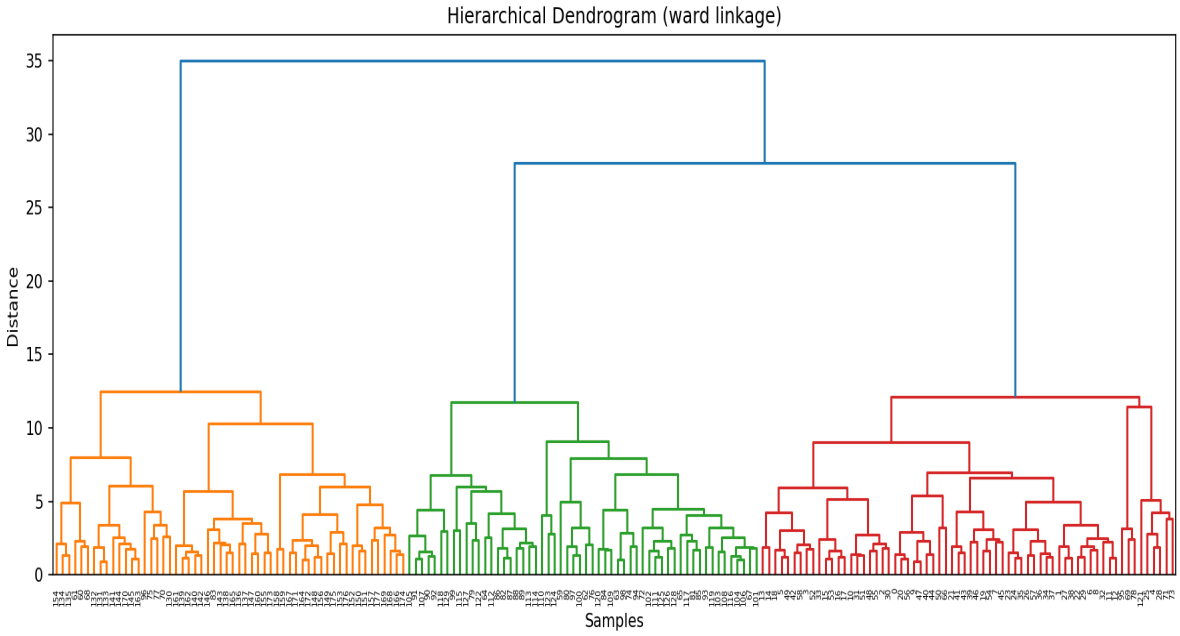
Selected configuration: set=pca, k=3. Metrics: Silhouette=0.299, Davies–Bouldin=1.336, Calinski–Harabasz=76.2.

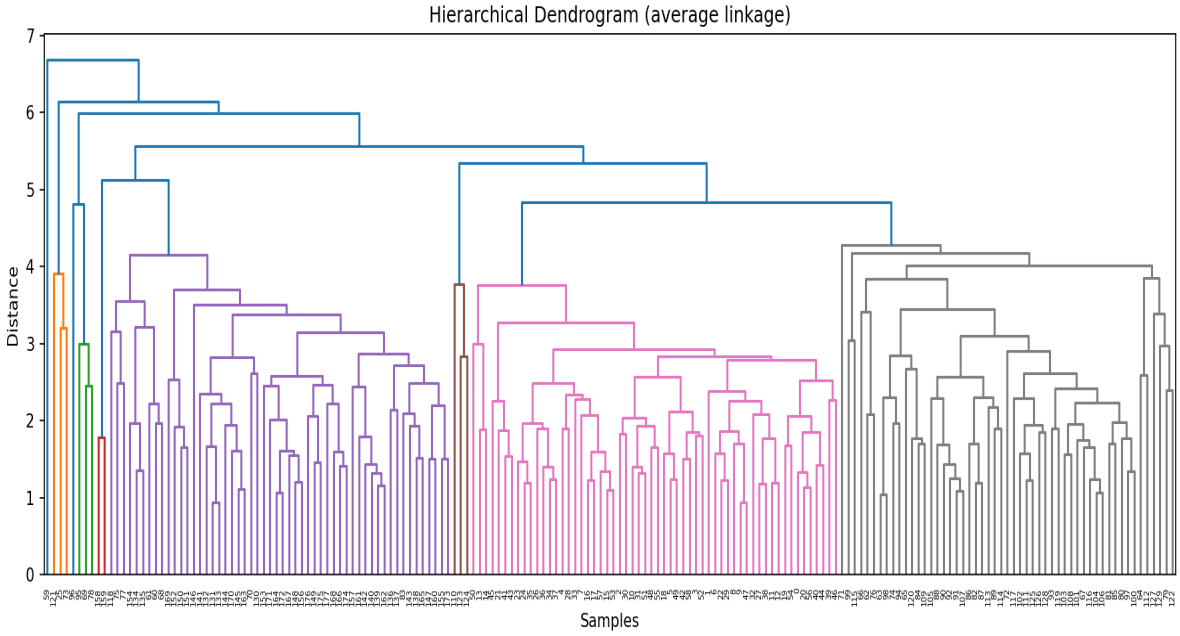
Code: Computing Indices vs k

```
for k in range(2, 9):
```

```
km = KMeans(n_clusters=k, n_init=10, random_state=42)
labels = km.fit_predict(features)
sil = silhouette_score(features, labels)
db = davies_bouldin_score(features, labels)
ch = calinski_harabasz_score(features, labels)  # plot vs k
```

Hierarchical Dendrograms





Cophenetic correlation evaluates dendrogram fidelity to original distances. Ward: 0.664, Average: 0.761.

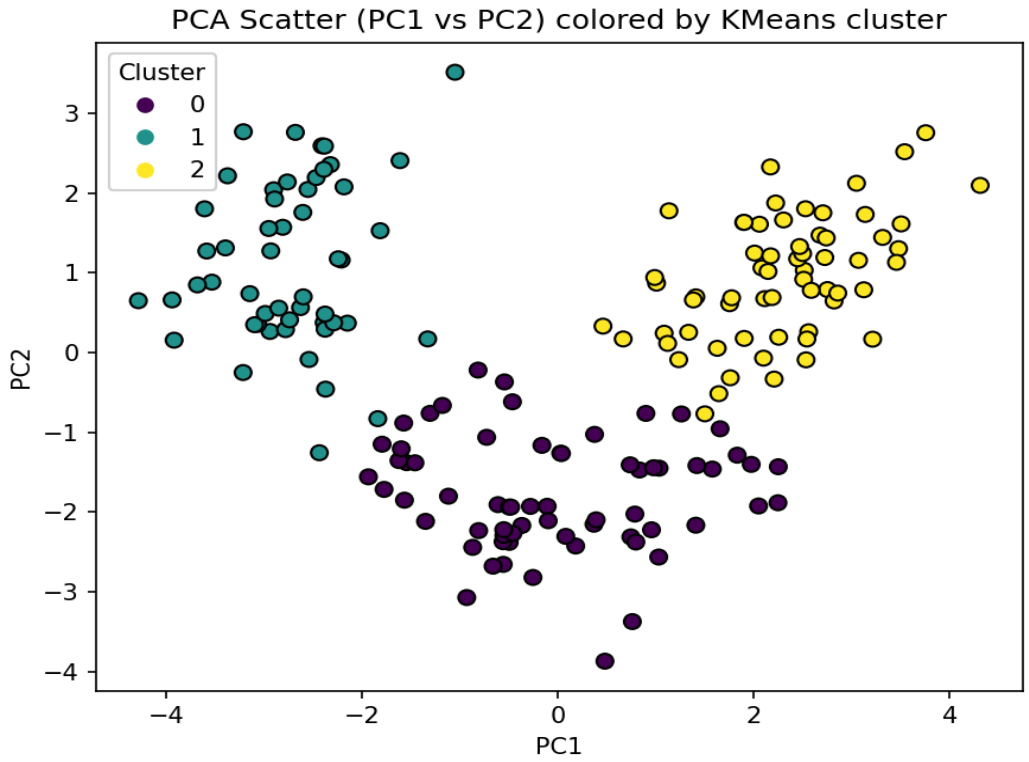
Code: Building Dendrograms

```
Z_ward = linkage(features_for_kmeans, method='ward', metric='euclidean')
fig, ax = plt.subplots()
dendrogram(Z_ward, ax=ax)
save_fig(fig, os.path.join(fig_dir, 'dendrogram_ward.png'))
```

Code: Cophenetic Correlation

```
Y = pdist(features_for_kmeans)
coph_ward, _ = cophenet(Z_ward, Y)
coph_avg, _ = cophenet(Z_avg, Y)
json.dump({'ward': coph_ward, 'average': coph_avg}, open(os.path.join(out_dir,'cophenetic.json'),'w
```

PCA Scatter by KMeans Clusters



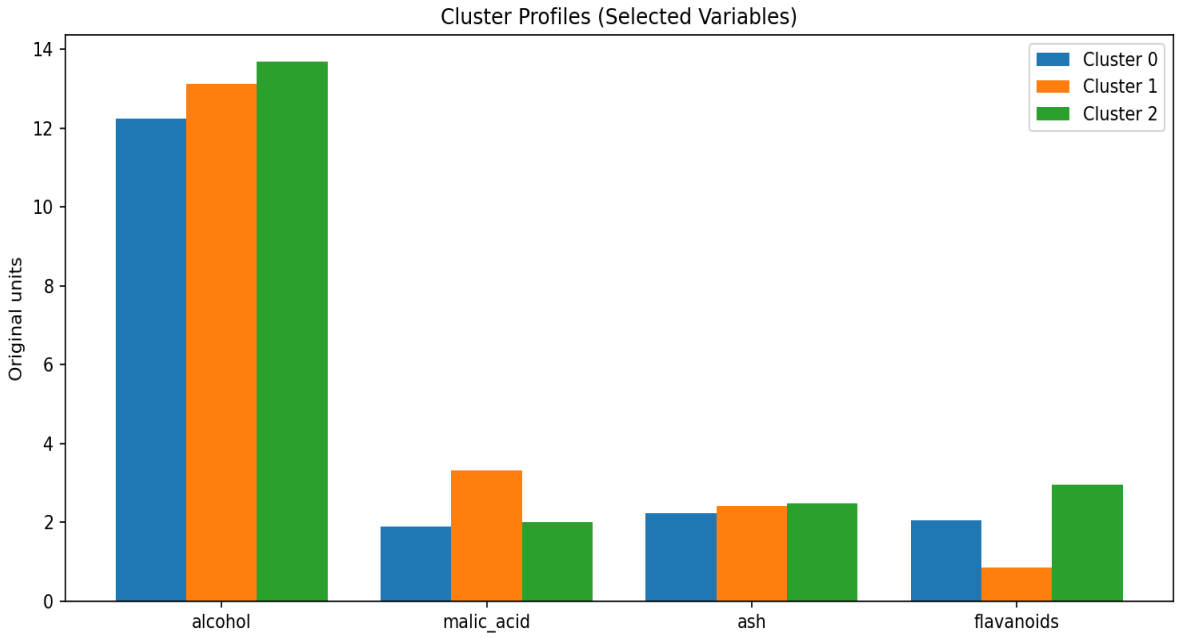
Code: PCA Scatter by Cluster

```
pca2 = PCA(n_components=2)
X_pca2 = pca2.fit_transform(X_scaled)
fig, ax = plt.subplots()
ax.scatter(X_pca2[:, 0], X_pca2[:, 1], c=kmeans_labels, cmap='viridis')
save_fig(fig, os.path.join(fig_dir, 'pca_scatter_kmeans.png'))
```

Cluster Centroids (Original Units)

Cluster	alcohol	malic_acid	ash	flavanoids
0	12.25	1.90	2.23	2.06
1	13.13	3.31	2.41	0.86
2	13.68	1.99	2.47	2.96

Cluster Profiles (Selected Variables)



Code: Cluster Profiles Bar Chart

```
key_vars = ['alcohol', 'malic_acid', 'ash', 'flavanoids']
fig, ax = plt.subplots(figsize=(10, 5))
idx = np.arange(len(key_vars)); width = 0.8 / chosen_k
for c in range(chosen_k):
    ax.bar(idx + c * width, centroids_original.loc[c, key_vars], width)
save_fig(fig, os.path.join(fig_dir, 'cluster_profiles_bar.png'))
```

Part E: Second EDA & Statistical Inference (15 points)

Provide centroids in original units and PCA scatter coloured by cluster to interpret clusters. Use ANOVA/Kruskal-Wallis on alcohol and flavanoids; report p-values and effect sizes (eta², ε²). If labels used for validation, note any discrepancies.

Statistical Inference

Alcohol: ANOVA p=0.0000, eta²=0.564; Kruskal p=0.0000, ε²=0.571

Flavanoids: ANOVA p=0.0000, eta²=0.756; Kruskal p=0.0000, ε²=0.750

Effect sizes (eta², ε²) contextualize practical differences beyond p-values.

Code: ANOVA & Kruskal-Wallis

```
df = X_imputed.copy(); df['cluster'] = kmeans_labels
groups = [g['alcohol'].values for _, g in df.groupby('cluster')]
f, p = stats.f_oneway(*groups)
h, pkw = stats.kruskal(*groups)
# eta² and ε² computed from sums of squares and H-statistic
```

External Validation (Optional)

Adjusted Rand Index (ARI)=0.897, Normalized Mutual Info (NMI)=0.876. Labels were not used in tuning; external metrics are reported for analysis only.

Part F: PCA Influence & Suitability (Trade-offs)

How PCA influenced clustering decisions: PCA features yielded the best balance of internal indices (e.g., silhouette) at k=3, indicating reduced collinearity/noise improved separability.

Hierarchical vs K-means suitability: hierarchical exposes multi-scale structure but depends on linkage choice and can show chaining; K-means is efficient and interpretable for roughly convex clusters. Choice depends on goals and data geometry.

Reflection & Rationale

Standardization ensures fair distances; PCA reduces noise and collinearity, often improving cluster compactness/separation. K-means is efficient and interpretable for convex clusters; hierarchical reveals multiscale structure and is evaluated via cophenetic correlation. In this run, PCA features with k=3 yielded the best internal indices, suggesting reduction clarified the grouping signal.

Trade-offs: K-means sensitivity to scaling and initialization vs. hierarchical dependence on linkage choice and potential chaining. Future work: stability analysis, alternative algorithms (GMM, DBSCAN), domain-informed feature engineering.

Applications

Quality control: group similar wines; product segmentation; inventory planning and sourcing; marketing tailored to cluster profiles.

Benefits & Limitations

Benefits: fast, interpretable clustering; dendrograms reveal structure; PCA reduces noise. Limitations: K-means assumes convex clusters and needs standardization; hierarchical is sensitive to linkage; PCA components may be less interpretable.