

```

41 acc2 = accuracy_score(y_test, model.predict(X_test_mm))
42 print(f"3. MinMaxScaler → Accuracy: {acc2:.4f}")
43
44 # 4. Feature Extraction using PCA (reduce to 2 components)
45 pca = PCA(n_components=2)
46 X_train_pca = pca.fit_transform(X_train_std)
47 X_test_pca = pca.transform(X_test_std)
48
49 model.fit(X_train_pca, y_train)
50 acc3 = accuracy_score(y_test, model.predict(X_test_pca))
51 print(f"4. PCA (2 components) → Accuracy: {acc3:.4f}")
52 print(f" Explained Variance: {pca.explained_variance_ratio_.sum():.4f}")
53
54 # 5. Feature Selection - SelectKBest (top 2 features)
55 selector_k = SelectKBest(score_func=f_classif, k=2)
56 X_train_k = selector_k.fit_transform(X_train_std, y_train)
57 X_test_k = selector_k.transform(X_test_std)
58
59 model.fit(X_train_k, y_train)
60 acc4 = accuracy_score(y_test, model.predict(X_test_k))
61 print(f"5. SelectKBest (k=2) → Accuracy: {acc4:.4f}")
62 print(" Selected features:", [feature_names[i] for i in
    ↪ selector_k.get_support(indices=True)])
63
64 # 6. Recursive Feature Elimination (RFE)
65 rfe = RFE(estimator=LogisticRegression(max_iter=200),
    ↪ n_features_to_select=2)
66 X_train_rfe = rfe.fit_transform(X_train_std, y_train)
67 X_test_rfe = rfe.transform(X_test_std)
68
69 model.fit(X_train_rfe, y_train)
70 acc5 = accuracy_score(y_test, model.predict(X_test_rfe))
71 print(f"6. RFE (2 features) → Accuracy: {acc5:.4f}")
72 print(" Selected features:", [feature_names[i] for i in
    ↪ range(len(rfe.support_)) if rfe.support_[i]])
73
74 # Final Comparison Plot
75 results = pd.DataFrame({
76     'Method': ['No Scaling', 'StandardScaler', 'MinMaxScaler', 'PCA',
    ↪ 'SelectKBest', 'RFE'],
77     'Accuracy': [acc0, acc1, acc2, acc3, acc4, acc5]
78 })
79
80 plt.figure(figsize=(10,6))
81 sns.barplot(x='Accuracy', y='Method', data=results, palette='viridis')
82 plt.title('Feature Scaling & Selection Comparison on Iris')
83 plt.xlim(0.8, 1.0)
84

```