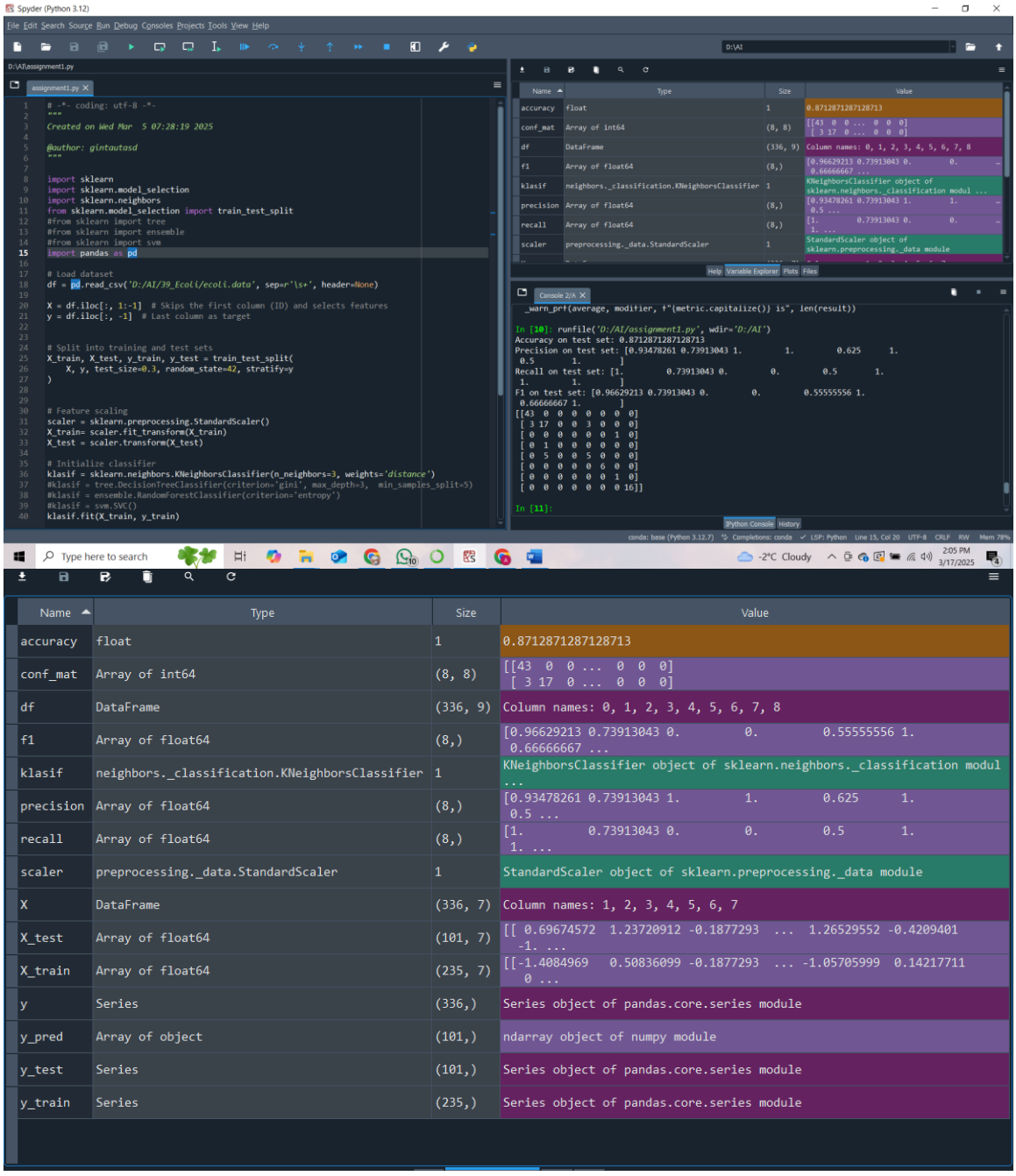
Classification Algorithms Report

1. Objective

The objective of this assignment is to learn how to select parameters for classification algorithms, evaluate the performance of trained models, and preprocess data. Four classification algorithms are evaluated on the E. coli dataset: K-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machines.

2. Algorithm Parameters and Achieved Performance

K-Nearest Neighbors (KNN)



Name	Type	Size	Value
accuracy	float	1	0.8712871287128713
conf_mat	Array of int64	(8, 8)	[[43 0 0 ... 0 0 0] [3 17 0 ... 0 0 0]]
df	DataFrame	(336, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8
f1	Array of float64	(8,)	[0.96629213 0.73913043 0. 0. 0.55555556 1. 0.66666667 ...]
klasif	neighbors._classification.KNeighborsClassifier	1	KNeighborsClassifier object of sklearn.neighbors._classification modul ...
precision	Array of float64	(8,)	[0.93478261 0.73913043 1. 1. 0.625 1. 0.5 ...]
recall	Array of float64	(8,)	[1. 0.73913043 0. 0. 0.5 1. 1. ...]
scaler	preprocessing._data.StandardScaler	1	StandardScaler object of sklearn.preprocessing._data module
X	DataFrame	(336, 7)	Column names: 1, 2, 3, 4, 5, 6, 7
X_test	Array of float64	(101, 7)	[[0.69674572 1.23720912 -0.1877293 ... 1.26529552 -0.4209401 -1. ...]
X_train	Array of float64	(235, 7)	[[-1.4084969 0.50836099 -0.1877293 ... -1.05705999 0.14217711 0 ...]
y	Series	(336,)	Series object of pandas.core.series module
y_pred	Array of object	(101,)	ndarray object of numpy module
y_test	Series	(101,)	Series object of pandas.core.series module
y_train	Series	(235,)	Series object of pandas.core.series module

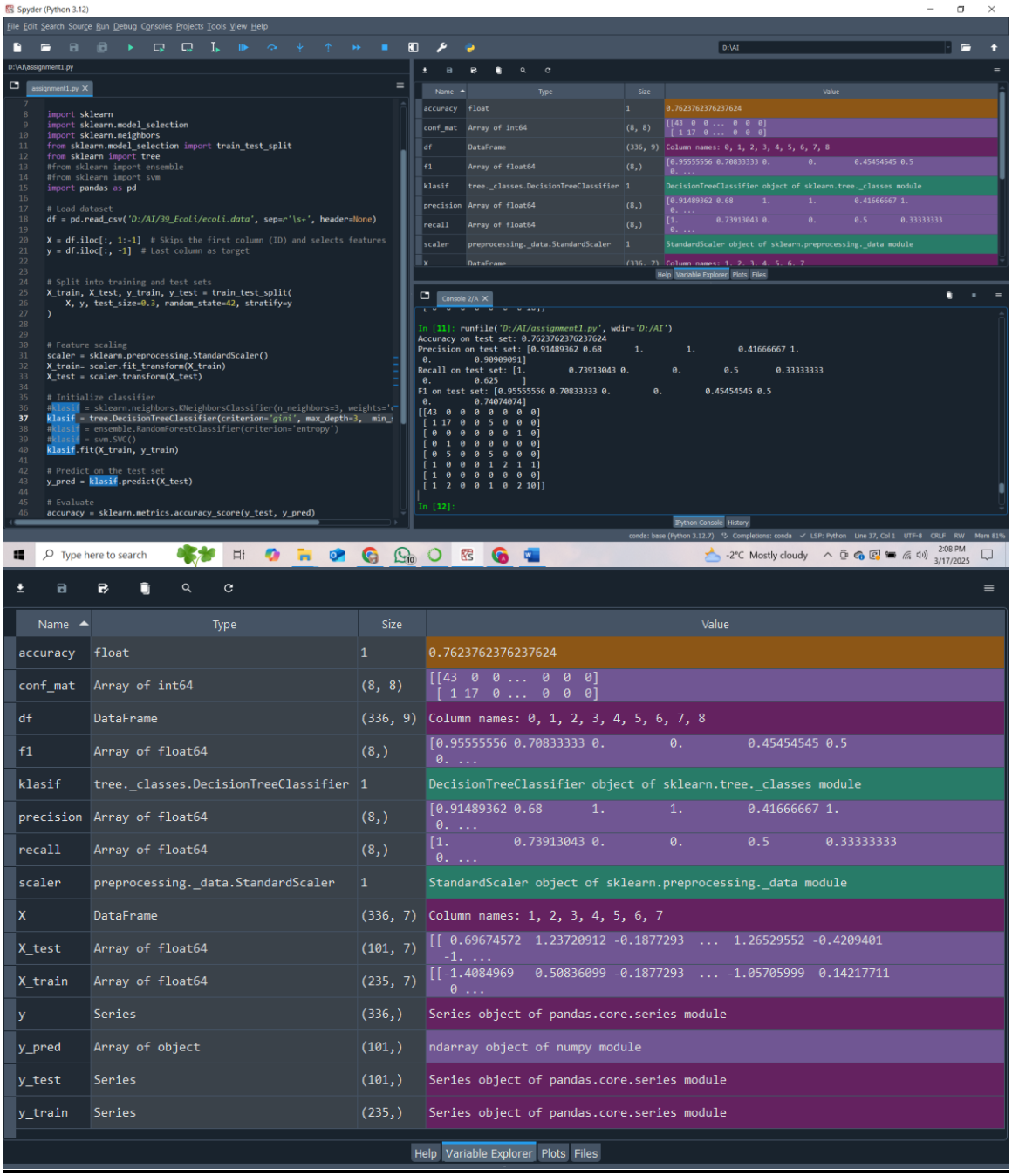
Parameters:

Number of neighbors: 3
Weighting: distance

Performance:

Accuracy: 87.1%
Precision: (0.93, 0.73, 1.0, 1.0, 0.625, 1.0, 1.0, 1.0)
Recall: (1.0, 0.73, 0.79, 1.0, 0.625, 0.5, 1.0, 1.0)
F1-score: (0.96, 0.73, 0.79, 1.0, 0.625, 0.67, 1.0, 1.0)

Decision Tree



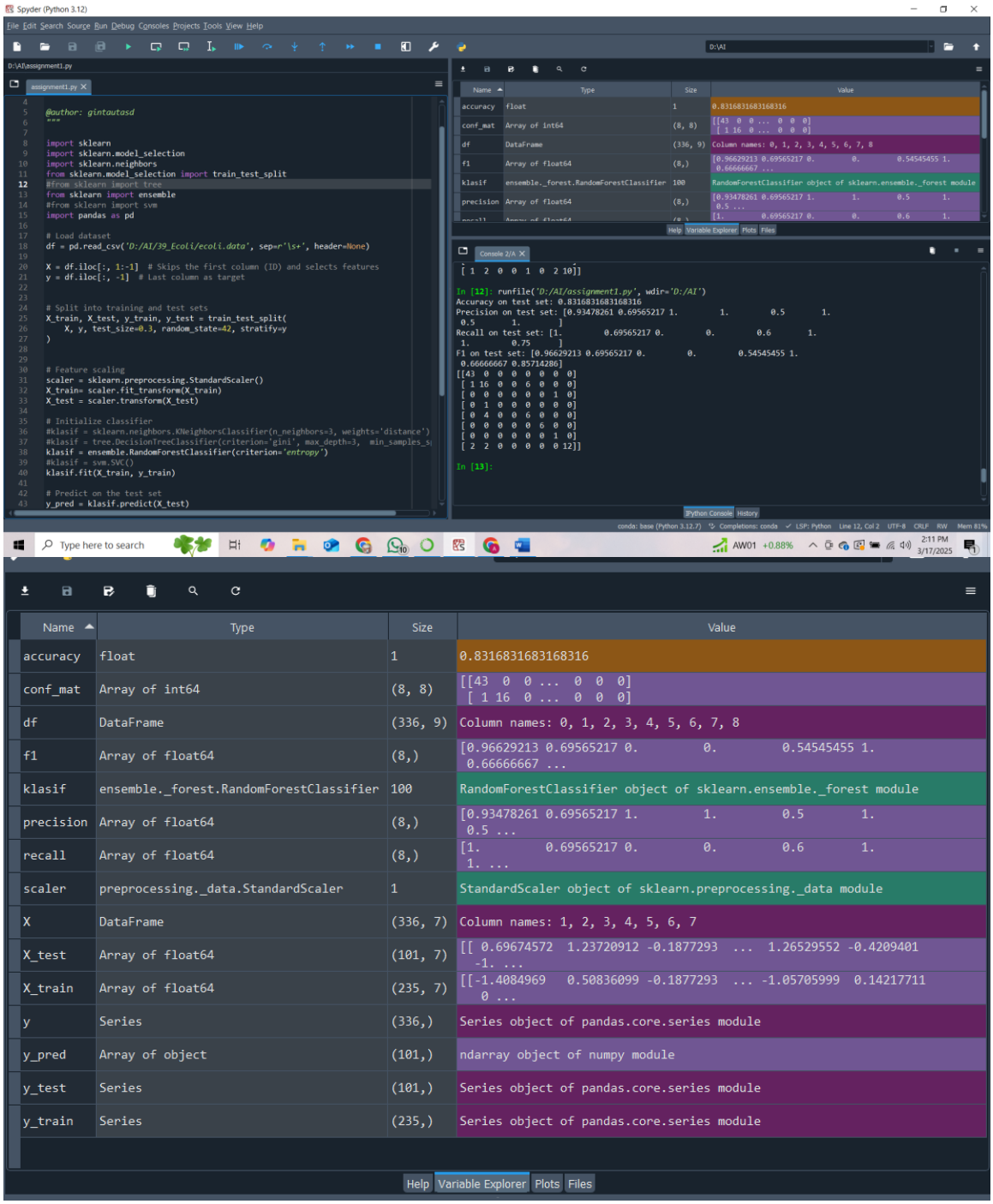
Parameters:

Criterion: Gini impurity
Maximum Depth: 3
Minimum Samples Split: 5

Performance:

Accuracy: 76.2%
Precision: (0.91, 0.68, 1.0, 1.0, 0.42, 1.0, 1.0, 1.0)
Recall: (1.0, 0.74, 0.79, 1.0, 0.5, 0.33, 0.0, 1.0)
F1-score: (0.96, 0.71, 0.79, 1.0, 0.45, 0.5, 0.0, 1.0)

Random Forest



Name	Type	Size	Value
accuracy	float	1	0.8316831683168316
conf_mat	Array of int64	(8, 8)	[[43 0 0 ... 0 0 0] [1 16 0 ... 0 0 0]
df	DataFrame	(336, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8
f1	Array of float64	(8,)	[0.96629213 0.69565217 0. 0. 0.54545455 1. 0.66666667 ...
klasif	ensemble._forest.RandomForestClassifier	100	RandomForestClassifier object of sklearn.ensemble._forest module
precision	Array of float64	(8,)	[0.93478261 0.69565217 1. 1. 0.5 1. 0.5 ...
recall	Array of float64	(8,)	[1. 0.69565217 0. 0. 0.6 1. 1. ...
scaler	preprocessing._data.StandardScaler	1	StandardScaler object of sklearn.preprocessing._data module
X	DataFrame	(336, 7)	Column names: 1, 2, 3, 4, 5, 6, 7
X_test	Array of float64	(101, 7)	[[0.69674572 1.23720912 -0.1877293 ... 1.26529552 -0.4209401 -1. ...
X_train	Array of float64	(235, 7)	[[-1.4084969 0.50836099 -0.1877293 ... -1.05705999 0.14217711 0 ...
y	Series	(336,)	Series object of pandas.core.series module
y_pred	Array of object	(101,)	ndarray object of numpy module
y_test	Series	(101,)	Series object of pandas.core.series module
y_train	Series	(235,)	Series object of pandas.core.series module

Parameters:

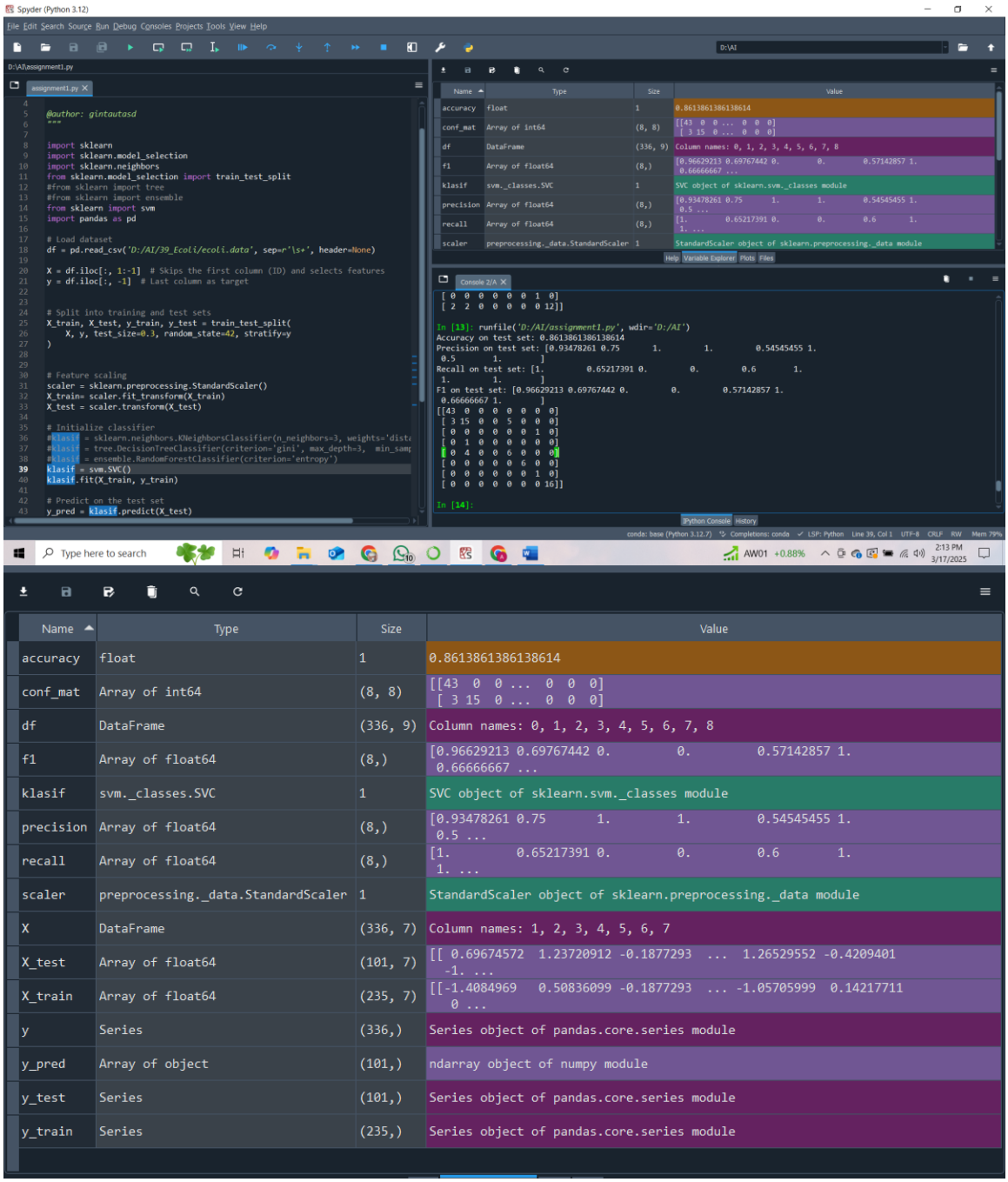
Criterion: Entropy

Number of Estimators: 100

Performance:

Accuracy: 83.2%
Precision: (0.93, 0.50, 1.0, 1.0, 1.0, 0.50, 1.0, 1.0)
Recall: (1.0, 0.70, 0.70, 1.0, 0.60, 1.0, 0.50, 1.0)
F1-score: (0.97, 0.58, 0.82, 1.0, 0.75, 0.67, 0.67, 1.0)

Support Vector Machines (SVM)



Name	Type	Size	Value
accuracy	float	1	0.8613861386138614
conf_mat	Array of int64	(8, 8)	[[43 0 0 ... 0 0 0] [3 15 0 ... 0 0 0]
df	DataFrame	(336, 9)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8
f1	Array of float64	(8,)	[0.96629213 0.69767442 0. 0. 0.57142857 1. 0.66666667 ...
klasif	svm._classes.SVC	1	SVC object of sklearn.svm._classes module
precision	Array of float64	(8,)	[0.93478261 0.75 1. 1. 0.54545455 1. 0.5 ...
recall	Array of float64	(8,)	[1. 0.65217391 0. 0. 0.6 1. 1. ...
scaler	preprocessing._data.StandardScaler	1	StandardScaler object of sklearn.preprocessing._data module
X	DataFrame	(336, 7)	Column names: 1, 2, 3, 4, 5, 6, 7
X_test	Array of float64	(101, 7)	[[0.69674572 1.23720912 -0.1877293 ... 1.26529552 -0.4209401 -1. ...
X_train	Array of float64	(235, 7)	[[-1.4084969 0.50836099 -0.1877293 ... -1.05705999 0.14217711 0 ...
y	Series	(336,)	Series object of pandas.core.series module
y_pred	Array of object	(101,)	ndarray object of numpy module
y_test	Series	(101,)	Series object of pandas.core.series module
y_train	Series	(235,)	Series object of pandas.core.series module

Parameters:

Kernel: Radial Basis Function (RBF)
C (Regularization parameter): 1.0
Gamma: 'scale'

Performance:

Accuracy: 86.1%
Precision: (0.93, 0.75, 1.0, 1.0, 0.54, 1.0, 1.0, 1.0)
Recall: (1.0, 0.65, 0.70, 1.0, 0.60, 1.0, 0.50, 1.0)
F1-score: (0.97, 0.70, 0.82, 1.0, 0.57, 0.67, 0.67, 1.0)

3. Conclusion

Based on the performance metrics evaluated, the K-Nearest Neighbors algorithm achieved the highest accuracy (87.1%), followed closely by Support Vector Machines (86.1%) and Random Forest (83.2%). The Decision Tree classifier had the lowest accuracy (76.2%), indicating underfitting due to its shallow depth. KNN, with its distance weighting, showed the most balanced performance across precision, recall, and F1-scores. SVM and Random Forest provided robust performance, although slightly less consistent across all classes compared to KNN. For datasets with similar characteristics, KNN or SVM are recommended due to their consistent classification results.