

```
In [1]: from sklearn.datasets import load_wine
        from sklearn.model_selection import train_test_split, RandomizedSearchCV, ShuffleSplit
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.metrics import accuracy_score
```

```
In [2]: # Step 1: Load the wine dataset
        wine = load_wine()
        X, y = wine.data, wine.target
```

```
In [3]: # Step 2: Split the dataset into train and test dataset
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [4]: # Step 3: Use random search CV to hyperparameter tune the Decision Tree
param_dist = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2', None]
}

tree = DecisionTreeClassifier()
random_search = RandomizedSearchCV(tree, param_distributions=param_dist, n_iter=100)
random_search.fit(X_train, y_train)
```

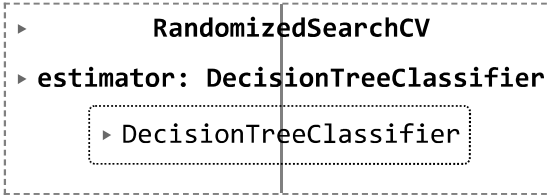
```
C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:425: FitFailedWarning:
105 fits failed out of a total of 500.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
```

Below are more details about the failures:

```
-----
---
105 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py", line 729, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\base.py", line 1145, in wrapper
    estimator._validate_params()
  File "C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\base.py", line 638, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\utils\_param_validation.py", line 96, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of DecisionTreeClassifier must be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got 'auto' instead.
```

```
warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:979: UserWarning: One or more of the test scores are non-finite: [0.91453202 nan nan 0.90812808 0.90123153 0.90123153 0.88029557 0.87364532 0.86576355 nan 0.90098522 0.89408867 0.8455665 0.87339901 nan 0.91527094 0.90788177 nan 0.8726601 0.91477833 0.90788177 0.90172414 0.92216749 0.83793103 nan 0.87364532 0.87364532 0.91502463 0.89408867 0.92881773 0.88842365 nan 0.90862069 0.83103448 0.94310345 nan 0.88768473 0.92241379 0.89384236 0.91477833 0.90788177 0.86674877 nan nan 0.89408867 nan 0.92931034 nan 0.91502463 0.87315271 0.86650246 0.88793103 0.90788177 nan 0.85295567 0.85246305 0.90098522 0.89433498 0.89384236 0.89359606 0.90812808 0.92167488 nan 0.83891626 nan 0.87389163 0.91527094 nan nan nan 0.92192118 0.88004926 0.91527094 0.92241379 0.91502463 0.94334975 0.90049261 0.8726601 0.90812808 nan 0.9362069 0.91551724 0.80270936 0.86674877 0.85221675 0.85221675 0.90812808 0.89384236 0.85320197 0.90123153 0.90837438 nan 0.88694581 0.89507389 0.90837438 0.89507389 0.91527094 0.8591133 nan 0.86625616]
warnings.warn(
```

```
Out[4]:
```



```

  ▶ RandomizedSearchCV
  ▶ estimator: DecisionTreeClassifier
    ▶ DecisionTreeClassifier

```

```
In [5]: print("Best Hyperparameters for Decision Tree:", random_search.best_params_)

Best Hyperparameters for Decision Tree: {'min_samples_split': 2, 'min_samples_
_leaf': 1, 'max_features': 'log2', 'max_depth': 30, 'criterion': 'entropy'}
```

```
In [6]: # Evaluate Decision Tree
tree_best = random_search.best_estimator_
y_pred_tree = tree_best.predict(X_test)
accuracy_tree = accuracy_score(y_test, y_pred_tree)
print("Accuracy of Decision Tree:", accuracy_tree)

Accuracy of Decision Tree: 0.8888888888888888
```

```
In [7]: # Step 4: Grow a random forest
# Create 10 subsets of the training dataset
cv = ShuffleSplit(n_splits=10, test_size=0.2, random_state=42)
```

```
In [8]: # Train 1 decision tree on each subset
trees = []
for train_index, _ in cv.split(X_train):
    tree = DecisionTreeClassifier(**random_search.best_params_)
    tree.fit(X_train[train_index], y_train[train_index])
    trees.append(tree)
```

```
In [9]: # Evaluate all the trees on the test dataset
accuracies = []
for tree in trees:
    y_pred_tree = tree.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred_tree)
    accuracies.append(accuracy)
```

```
In [10]: # Calculate average accuracy of the trees in the random forest
average_accuracy_rf = sum(accuracies) / len(accuracies)
print("Average Accuracy of Random Forest:", average_accuracy_rf)
```

Average Accuracy of Random Forest: 0.9138888888888888

```
In [ ]:
```

