```
!pip install scikit-optimize
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
import warnings
from skopt import BayesSearchCV
from skopt.space import Real, Categorical, Integer

# Importing Data
data = pd.read_csv('wineq.csv')
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Algorithm Selection
models = [
    ('SVM', SVC()),
    ('Random Forest', RandomForestClassifier()),
    ('Logistic Regression', LogisticRegression()),
    ('Decision Tree', DecisionTreeClassifier())
]

# Hyperparameter Optimization using Bayesian Optimization
param_grids = {
    'SVM': {
        'C': Real(1e-4, 1e+5, prior='log-uniform'),
        'kernel': Categorical(['linear', 'rbf', 'poly', 'sigmoid'])
    },
    'Random Forest': {
        'n_estimators': Integer(100, 200),
        'max_depth': Integer(10, 30),
        'min_samples_split': Integer(1, 10),
        'min_samples_leaf': Integer(1, 10)
    },
    'Logistic Regression': {
        'C': Real(1e-6, 1e+6, prior='log-uniform'),
        'penalty': Categorical(['l1', 'l2']),
        'solver': Categorical(['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'])
    },
    'Decision Tree': {
```

```
        'max_depth': Integer(1, 100),
        'min_samples_split': Integer(2, 15),
        'min_samples_leaf': Integer(1, 7)
    }
}

final_scores = {}
best_params = {}

for name, model in models:
    try:
        opt = BayesSearchCV(
            model,
            param_grids[name],
            n_iter=10,
            cv=3,
            scoring='accuracy',
            n_jobs=-1,
            random_state=0
        )
        opt.fit(X_train, y_train)
        best_model = opt.best_estimator_
        accuracy = best_model.score(X_test, y_test)
        final_scores[name] = accuracy
        best_params[name] = opt.best_params_
        print(f"{name} - Tuned Hyperparameters: {opt.best_params_}")

    except Exception as e:
        print(f"Bayesian Optimization for {name} raised an exception: {e}")

best_model_name = max(final_scores, key=final_scores.get)
for name, accuracy in final_scores.items():
    print(f"{name} Accuracy: {accuracy}")

print("Best Model:", best_model_name, final_scores[best_model_name])
```

```
Collecting scikit-optimize
  Downloading scikit_optimize-0.9.0-py2.py3-none-any.whl (100 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100.3/100.3 kB 1.3 MB/s eta 0:00:00
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.10/dist-packages (from scikit-optimize) (1.3.2)
Collecting pyaml>=16.9 (from scikit-optimize)
  Downloading pyaml-23.9.7-py3-none-any.whl (23 kB)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.10/dist-packages (from scikit-optimize) (1.23.5)
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.10/dist-packages (from scikit-optimize) (1.11.3)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from scikit-optimize) (1.2.2)
```

```
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from pyaml>=16.9->scikit-optimize) (6.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->scikit-optimize) (3.2.0)
Installing collected packages: pyaml, scikit-optimize
Successfully installed pyaml-23.9.7 scikit-optimize-0.9.0
SVM - Tuned Hyperparameters: OrderedDict([('C', 50603.554533845774), ('kernel', 'rbf')])
Random Forest - Tuned Hyperparameters: OrderedDict([('max_depth', 21), ('min_samples_leaf', 2), ('min_samples_split', 4), ('n_estimators', 119)])
Bayesian Optimization for Logistic Regression raised an exception: Solver sag supports only 'l2' or 'none' penalties, got l1 penalty.
Decision Tree - Tuned Hyperparameters: OrderedDict([('max_depth', 24), ('min_samples_leaf', 2), ('min_samples_split', 11)])
SVM Accuracy: 0.63125
Random Forest Accuracy: 0.71875
Decision Tree Accuracy: 0.615625
Best Model: Random Forest 0.71875
```