

```
# Import necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.datasets import load_iris

# Load the Iris dataset from scikit-learn

iris = load_iris()

# Convert the dataset into a pandas DataFrame for easier manipulation
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['species'] = iris.target
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})

# Display the first few rows of the dataframe
print(df.head())

# Extract features and target
X = df[iris.feature_names] # Features: Sepal and Petal measurements
y = df['species'] # Target: Species of Iris flower

# Split the dataset into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the features to have mean=0 and variance=1
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize and train the model (Random Forest Classifier)
model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train_scaled, y_train)

# Predict the labels for the test set
y_pred = model.predict(X_test_scaled)


# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')


# Detailed classification report
print('\nClassification Report:')
print(classification_report(y_test, y_pred))

# Import necessary libraries
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.datasets import load_iris


# Load the Iris dataset from scikit-learn
iris = load_iris()

# Convert the dataset into a pandas DataFrame for easier manipulation
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['species'] = iris.target
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})


# Display the first few rows of the dataframe
print(df.head())


# Extract features and target
```

```
X = df[iris.feature_names] # Features: Sepal and Petal measurements
y = df['species'] # Target: Species of Iris flower

# Split the dataset into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the features to have mean=0 and variance=1
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize and train the model (Random Forest Classifier)
model = RandomForestClassifier(random_state=42)
model.fit(X_train_scaled, y_train)

# Predict the labels for the test set
y_pred = model.predict(X_test_scaled)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')

# Detailed classification report
print('\nClassification Report:')
print(classification_report(y_test, y_pred))
```