Lab Assignment 12

Fundamentals of Machine Learning

Name: M Jeyapathy

USN: 22BTARD016

Q. Explain random forest with a sample project.

Random Forest is an ensemble learning technique that combines the predictions of multiple individual models, typically decision trees, to improve overall predictive accuracy and robustness.

It operates by constructing a multitude of decision trees during the training phase and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. The "random" in Random Forest stems from the fact that each tree is trained on a random subset of the data, and for each split in the tree, a random subset of features is considered.

This randomness helps to decorrelate the individual trees, making the ensemble more resistant to overfitting and capable of capturing complex relationships in the data.

Key Components:

1. **Decision Trees:**
    - The basic building blocks of a Random Forest are decision trees. Each decision tree is constructed by selecting a random subset of features and a random subset of training data.
2. **Bootstrapping:**

- For each tree, a random subset of the training data is sampled with replacement (bootstrapping). This creates diverse datasets for training individual trees.

3. **Random Feature Selection:**
    - At each node of a decision tree, a random subset of features is considered for splitting. This helps in decorrelating the trees.
4. **Voting or Averaging:**
    - In the case of classification, each tree "votes" for a class, and the class with the majority of votes is chosen as the final prediction. For regression, the predictions from each tree are averaged.

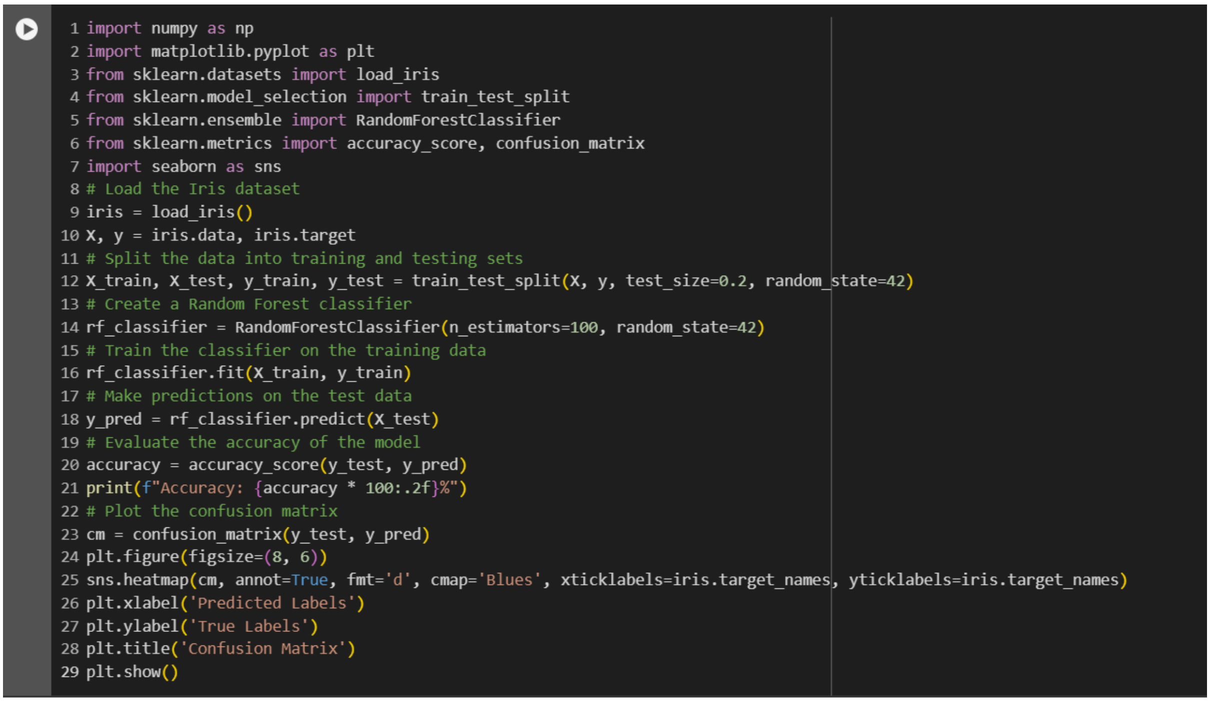
Applications:

- **Classification:**
  Image classification, spam detection, medical diagnosis, etc.
- **Regression:**
  Predicting stock prices, housing prices, etc.
- **Feature Selection:**
  Identifying the most important features in a dataset.

CODE:

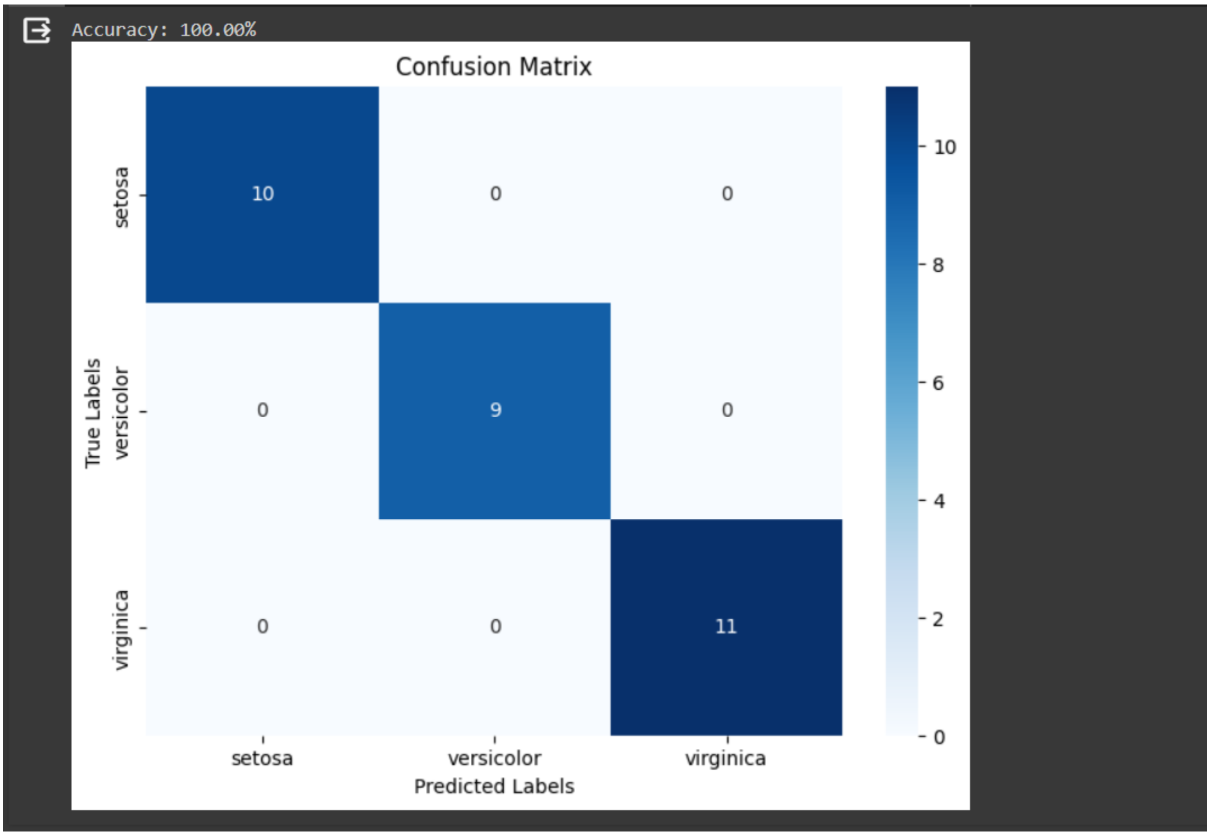
```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the classifier on the training data
rf_classifier.fit(X_train, y_train)
# Make predictions on the test data
y_pred = rf_classifier.predict(X_test)
# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
# Plot the confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=iris.target_names, yticklabels=iris.target_names)
```

```
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()
```



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the classifier on the training data
rf_classifier.fit(X_train, y_train)
# Make predictions on the test data
y_pred = rf_classifier.predict(X_test)
# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
# Plot the confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=iris.target_names, yticklabels=iris.target_names)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()
```

Output:



```
Accuracy: 100.00%
```

Github Link:

https://github.com/Jeyapathy/Machine-Learning