EXPERIMENT 2 : Supervised Learning Models

Support Vector Machine (SVM) and Random Forest for Binary & Multiclass Classification

AIM:

To build classification models using Support Vector Machines (SVM) and Random Forest, apply them to a dataset, and evaluate the models using performance metrics like accuracy and confusion matrix. To build classification models using Support Vector Machines (SVM) and Random Forest, apply them to a dataset, and evaluate the models using performance metrics like accuracy and confusion matrix. To build classification models using Support Vector Machines (SVM) and Random Forest, apply them to a dataset, and evaluate the models using performance metrics like accuracy and confusion matrix.

SOURCE CODE:

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
iris = load_iris()
X = iris.data
y = iris.target
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
svm_model = SVC(kernel='linear')
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict(X_test)
print("SVM Accuracy:", accuracy_score(y_test, y_pred_svm))
print("SVM Confusion Matrix:\n", confusion_matrix(y_test, y_pred_svm))
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
y_pred_rf = rf_model.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
print("Random Forest Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
sns.heatmap(confusion_matrix(y_test, y_pred_svm), annot=True, cmap='Blues')
plt.title("SVM Confusion Matrix")
plt.subplot(1,2,2)
sns.heatmap(confusion_matrix(y_test, y_pred_rf), annot=True, cmap='Greens')
plt.title("Random Forest Confusion Matrix")
plt.show()
```

OUTPUT: