LAB Assignment No 4

Topic: Random Forest and Support Vector Machine Classifier

Question 1

Classify flower species using Random Forest.

Task:

1. Load the *Iris dataset* from sklearn.datasets.
2. Split into training (70%) and testing (30%) sets.
3. Train a **Random Forest Classifier**.
4. Predict flower species on the test set.
5. Calculate and print **model accuracy**.

Code:

```
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
iris = load_iris()
X = iris.data
y = iris.target
# 3. Split into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# 4. Train a Random Forest Classifier
rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)

# 5. Predict flower species on the test set
y_pred = rf.predict(X_test)

# 6. Calculate and print model accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Model Accuracy:", accuracy)
```

Output:

```
(.venv) PS D:\AI assignments> & "D:/AI assignments/.venv/Scripts/python.exe" "d:/AI assignments/lab4_qs1.py"
Random Forest Model Accuracy: 1.0
```

Question 2

Use SVM on Breast Cancer Dataset and Classify tumors as malignant or benign.

Task:

1. Load the *Breast Cancer* dataset using sklearn.datasets.load_breast_cancer.
2. Train an **SVM classifier** (use SVC(kernel='linear')).
3. Evaluate the model using **accuracy** and **confusion matrix**.

Code:

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
cancer = load_breast_cancer()
X = cancer.data
y = cancer.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 4. Train SVM classifier with linear kernel
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

# 5. Make predictions on the test set
y_pred = svm_model.predict(X_test)

# 6. Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

# 7. Print results
print("SVM Model Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
```

output:

```
(.venv) PS D:\AI assignments> & "D:/AI assignments/.venv/Scripts/python.exe" "d:/AI assignments/lab4_qs2.py"
SVM Model Accuracy: 0.9649122807017544
Confusion Matrix:
 [[ 59   4]
 [  2 106]]
```

Question 3

Use Random Forest on CSV Dataset (Custom) : Predict student pass/fail based on study hours and scores.

Task:

1. Load a CSV file (e.g., students.csv) with columns: study_hours, attendance, marks, result.
2. Train a **Random Forest Classifier** to predict result (Pass/Fail).
3. Display **accuracy score** and **feature importance**.

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
data = pd.read_csv("students.csv")
X = data[['study_hours', 'attendance', 'marks']]
y = data['result']
y = y.map({'Fail': 0, 'Pass': 1})
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Model Accuracy:", accuracy)
feature_importance = pd.DataFrame({
   'Feature': X.columns,
   'Importance': rf_model.feature_importances_
}).sort_values(by='Importance', ascending=False)
print("\nFeature Importance:")
print(feature_importance)
```

output:

```
(.venv) PS D:\AI assignments> & "D:/AI assignments/.venv/Scripts/python.exe" "d:/AI assignments/lab4_qs3.py"
Model Accuracy: 1.0

Feature Importance:
       Feature  Importance
0  study_hours    0.363636
1   attendance    0.363636
2        marks    0.272727
```

Question 4

Use SVM on Digits Dataset and to identify the: Handwritten digit recognition.

Task:

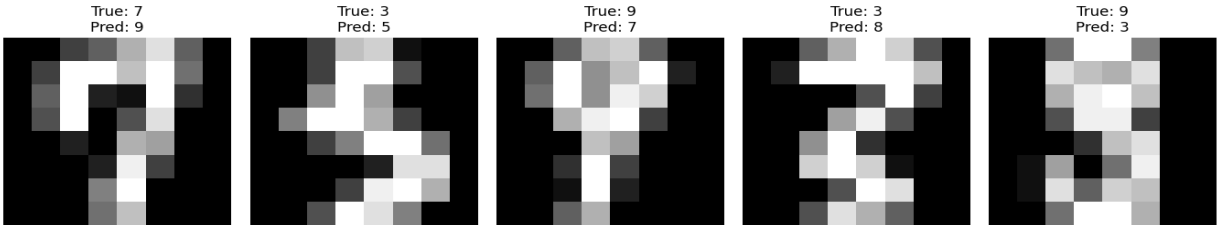
1. Load the *Digits dataset* from sklearn.datasets.load_digits.
2. Train an **SVM classifier** with an RBF kernel.
3. Test on unseen data.
4. Print **accuracy** and visualize some **misclassified samples**.

Code:

```
# Import required libraries
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import numpy as np
digits = load_digits()
X = digits.data
y = digits.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
svm_model = SVC(kernel='rbf', gamma=0.001, C=10)  # RBF kernel
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("SVM Model Accuracy:", accuracy)
misclassified_indices = np.where(y_test != y_pred)[0]
print(f"\nNumber of misclassified samples: {len(misclassified_indices)}")

plt.figure(figsize=(10, 4))
for i, index in enumerate(misclassified_indices[:5]):
  plt.subplot(1, 5, i + 1)
  plt.imshow(X_test[index].reshape(8, 8), cmap='gray')
  plt.title(f"True: {y_test[index]}\nPred: {y_pred[index]}")
  plt.axis('off')

plt.tight_layout()
plt.show()
```

output:

```
(.venv) PS D:\AI assignments> & "D:/AI assignments/.venv/Scripts/python.exe" "d:/AI assignments/lab4_qs4.py"
SVM Model Accuracy: 0.9907407407407407
```



Question 5:

Compare Random Forest vs SVM on Same Dataset (you can choose any dataset):
Compare two models on the same data.

Task:

- Use the *Wine dataset* from sklearn.datasets.load_wine.
- Train both:

  RandomForestClassifier(n_estimators=100)

- SVC(kernel='rbf')
- Print accuracy of both models.
- Conclude which performs better.

Code:

```
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
wine = load_wine()
X = wine.data
y = wine.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 3. Train Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_pred)
```

```
# 4. Train SVM with RBF kernel
svm_model = SVC(kernel='rbf', gamma='scale', random_state=42)
svm_model.fit(X_train, y_train)
svm_pred = svm_model.predict(X_test)
svm_accuracy = accuracy_score(y_test, svm_pred)

# 5. Print accuracy of both models
print("Random Forest Accuracy:", rf_accuracy)
print("SVM (RBF) Accuracy:", svm_accuracy)

# 6. Conclude which model performs better
if rf_accuracy > svm_accuracy:
    print("\n✅ Random Forest performs better on the Wine dataset.")
elif svm_accuracy > rf_accuracy:
    print("\n✅ SVM (RBF) performs better on the Wine dataset.")
else:
    print("\n🤝 Both models perform equally well on the Wine dataset.")
```

output:

```
(.venv) PS D:\AI assignments> & "D:/AI assignments/.venv/Scripts/python.exe" "d:/AI assignments/lab4_qs5.py"
Random Forest Accuracy: 1.0
SVM (RBF) Accuracy: 0.7592592592592593
```