```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Load dataset from CSV
data = pd.read_csv('p_iris.csv')
```

```
# Use only the first two features for training and visualization
X = data.iloc[:, :2].values # First two features
y = data.iloc[:, -1].values # Target variable (last column)
```

```
# Encode target labels (species) to numeric values
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

```
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.3, random_state=42)
```

```
# 1. SVM Model
svm_rbf = SVC(kernel='rbf', gamma='auto', probability=True)
svm_rbf.fit(X_train, y_train)
y_pred_svm = svm_rbf.predict(X_test)
```

```
# 2. Random Forest Model (Bagging)
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```
# 3. AdaBoost Model (Boosting)
ada = AdaBoostClassifier(base_estimator=SVC(kernel='linear', probability=True), n_estimators=50, random_state=42)
ada.fit(X_train, y_train)
y_pred_ada = ada.predict(X_test)
```

```
c:\Users\Om Shete\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\ensemble\_base.py:156: FutureWarning: `base_esti
  warnings.warn(
```

```
# Combine predictions using majority voting
y_pred_ensemble = np.array([y_pred_svm, y_pred_rf, y_pred_ada])
y_pred_final = np.array([np.bincount(x).argmax() for x in y_pred_ensemble.T])
```

```
# Accuracy for each model
for model_name, y_pred in zip(['SVM', 'Random Forest', 'AdaBoost', 'Ensemble'], [y_pred_svm, y_pred_rf, y_pred_ada, y_pred_final]):
    accuracy = accuracy_score(y_test, y_pred)
    print(f"{model_name} Accuracy: {accuracy:.4f}\n")
```

```
SVM Accuracy: 0.9778

Random Forest Accuracy: 1.0000

AdaBoost Accuracy: 1.0000

Ensemble Accuracy: 1.0000
```

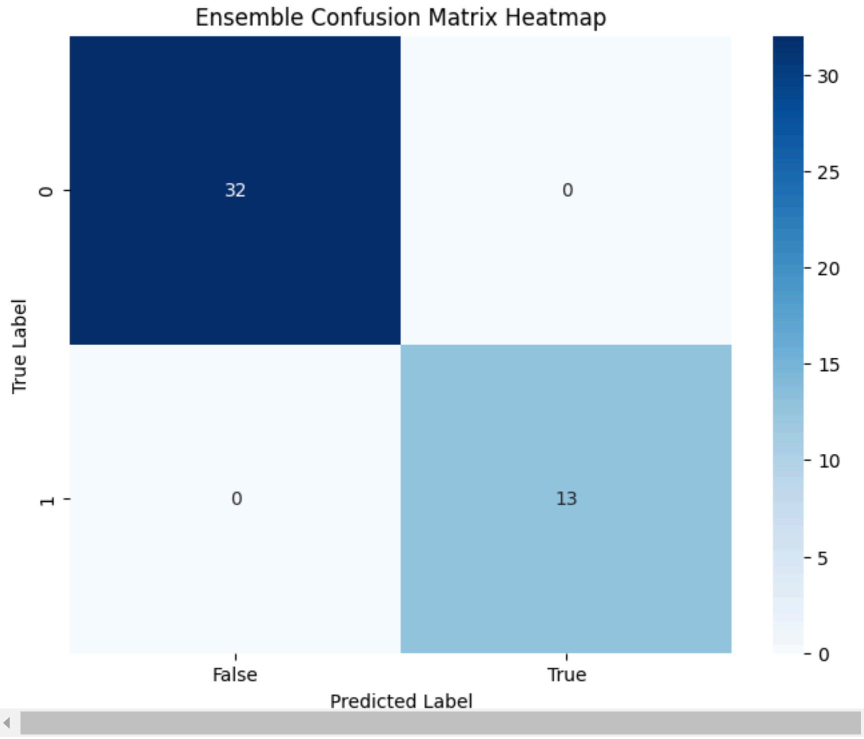
```
# Classification report for the ensemble model
report_dict = classification_report(y_test, y_pred_final, target_names=label_encoder.classes_, output_dict=True)
report_df = pd.DataFrame(report_dict).transpose()
print("Ensemble Classification Report:\n", report_df)
```

```
Ensemble Classification Report:
               precision  recall  f1-score  support
False               1.0     1.0       1.0     32.0
True                1.0     1.0       1.0     13.0
accuracy            1.0     1.0       1.0      1.0
macro avg           1.0     1.0       1.0     45.0
weighted avg        1.0     1.0       1.0     45.0
```

```
# Confusion matrix for the ensemble model
conf_matrix = confusion_matrix(y_test, y_pred_final)
print("\nEnsemble Confusion Matrix:")
print(conf_matrix)
```

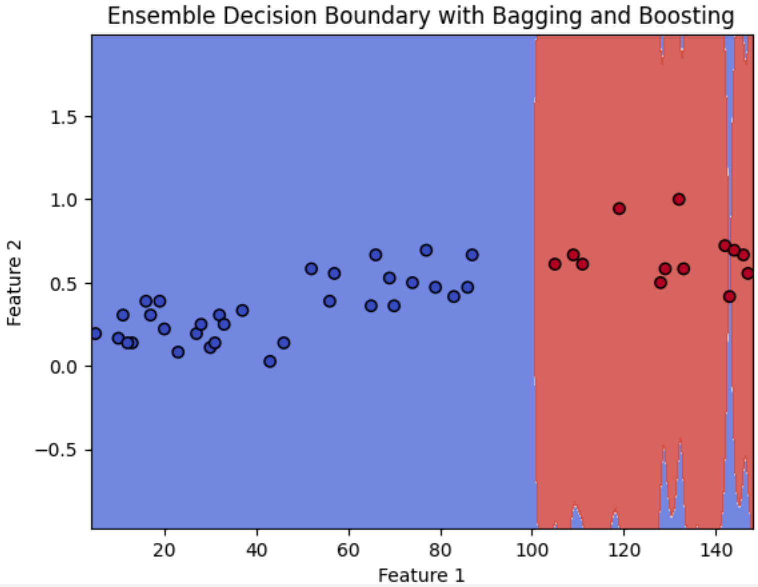
```
Ensemble Confusion Matrix:
[[32  0]
 [ 0 13]]
```

```
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classes_, yticklabels=1)
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Ensemble Confusion Matrix Heatmap')
plt.show()
```



```
def plot_decision_boundary(X, y, model):
    h = .02 # step size in the mesh
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
    Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.contourf(xx, yy, Z, alpha=0.8, cmap=plt.cm.coolwarm)
    plt.scatter(X[:, 0], X[:, 1], c=y, edgecolors='k', marker='o', cmap=plt.cm.coolwarm)
    plt.xlabel('Feature 1')
    plt.ylabel('Feature 2')
    plt.title('Ensemble Decision Boundary with Bagging and Boosting')
    plt.show()
```

```
# Since we can't train an ensemble model directly, we just plot the decision boundary using the SVM model
plot_decision_boundary(X_test, y_test, svm_rbf)
```


Ensemble Decision Boundary with Bagging and Boosting
Feature 2
1.5
1.0
0.5
0.0
−0.5
20
40
60
80
100
120
140
Feature 1