```python
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#read the data
ds = pd.read_csv(r"C:\Users\prasu\DS2\Machine_learning\classification\6. Ensamble_learning\5. RA
ds

#split the data
X = ds.iloc[:, 2:4].values
y = ds.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

#no scaling required for tree algorithms


from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=4, n_estimators=60, random_state=0,criterion='entr
classifier.fit(X_train, y_train)

#y_pred
y_pred = classifier.predict(X_test)
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
accuracy = accuracy_score(y_test, y_pred)
print('accuracy:', accuracy)
cm = confusion_matrix(y_test,y_pred)
print(cm)
clf_report = classification_report(y_test, y_pred)
print(clf_report)
variance = classifier.score(X_test, y_test)
print('variance:', variance)
bias = classifier.score(X_train, y_train)
print('bias:', bias)

# Save outputs to files
with open("output_random_forest_classification.txt", "w") as f:
    f.write(f"Accuracy: {accuracy}\n")
    f.write(f"Variance: {variance}\n")
    f.write(f"Bias: {bias}\n")

cm_df = pd.DataFrame(cm, index=['Actual Negative', 'Actual Positive'], columns=['Predicted Negat
cm_df.to_csv('confusion_matrix.csv', index=True)

with open("classification_report.txt", "w") as f:
    f.write(clf_report)
```