## Program 8

Implement Random forest ensemble method on a given dataset

## Screenshot:

NA PARTIES AND ADDRESS OF THE PARTIES AND ADDRES
Random Forest Argorithm
(2) Input
Training dataset D with n egs & m datues
> No. of tree T to unale
417 3. 1
(2) For each free + from 1 to I
(i) Randonly select bootshap sample (with deployment) of size in done Dataset D. This subject is used to have orce t
Dataset D. This subject is used to hown tree to
air Fae each unde in Me
- Randonly select & feature ( where K, M) from John no
I make the test transfer among & features to get note
based on armson (like evini holy or info gain)
-> Kepeat work her reacher stopping cond in lig war dea
or mon samples per leaf)
(in) For each test sample 2:
Pass I through each of Talwision trees to get production
-> > > > > 16 its clay itran prob, early here voke for claye
> It is segremen just, each one provides a greated value
CVAME 2 AM
(iv) out
-) for classification: final production is majoring book from all The
- the regretion: Luca prediction is any of predictions from
ou Thong
and the second of the second
(V) 2hd
710

SNO.	capa	Twochkney	Communication	Riachial Ukrowied	e Job
1	39	Yes	4000	-youd	49
2	<9	No	Nodelate	Avglood	Yes
3	39	No	Moderay	mg	No
4	3,9	No	Moderan	My	16
5	>9	yes	Modulak	good	Yes
1 1 1 1 1 1		and the second			
(1)		[CGPA]			the state of the s
( )	20				
	< YU	1 Tube	= Cneys		
		Y"5/	(no		
		Lyen		A property to heart	
		V	70		
10. Th	beaching	eneral			
(2) 1 H		20			
	/		77		
/		Dear the - a Ken. La	61		
<del>\(\frac{1}{19}\)</del>	>	Reachical Knowled			
719	>	good tro			

## Code:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

 $from\ sklearn.ensemble\ import\ Random Forest Classifier$ 

from sklearn.metrics import accuracy\_score

df = pd.read\_csv('/content/iris (3).csv')

# Prepare the data

X = df.drop('species', axis=1)

y = df['species']

```
# Build and evaluate the Random Forest classifier with default n_estimators
rf_classifier_default = RandomForestClassifier(random_state=42)
rf_classifier_default.fit(X_train, y_train)
y_pred_default = rf_classifier_default.predict(X_test)
default_accuracy = accuracy_score(y_test, y_pred_default)
print(f"Accuracy with default n_estimators (10): {default_accuracy}")
# Fine-tune the model by varying the number of trees
best_accuracy = 0
best_n_estimators = 0
for n_estimators in range(1, 101): # Try different number of trees
  rf_classifier = RandomForestClassifier(n_estimators=n_estimators, random_state=42)
  rf_classifier.fit(X_train, y_train)
  y_pred = rf_classifier.predict(X_test)
  accuracy = accuracy_score(y_test, y_pred)
  if accuracy > best_accuracy:
    best_accuracy = accuracy
    best_n_estimators = n_estimators
print(f"\nBest Accuracy: {best_accuracy} achieved with n_estimators = {best_n_estimators}".
```

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)