Experiment No. 4
Apply Random Forest Algorithm on Adult Census Income Dataset and analyze the performance of the model
Date of Performance:
Date of Submission:

Aim: Apply Random Forest Algorithm on Adult Census Income Dataset and analyze the performance of the model.

Objective: Able to perform various feature engineering tasks, apply Random Forest Algorithm on the given dataset and maximize the accuracy, Precision, Recall, F1 score.

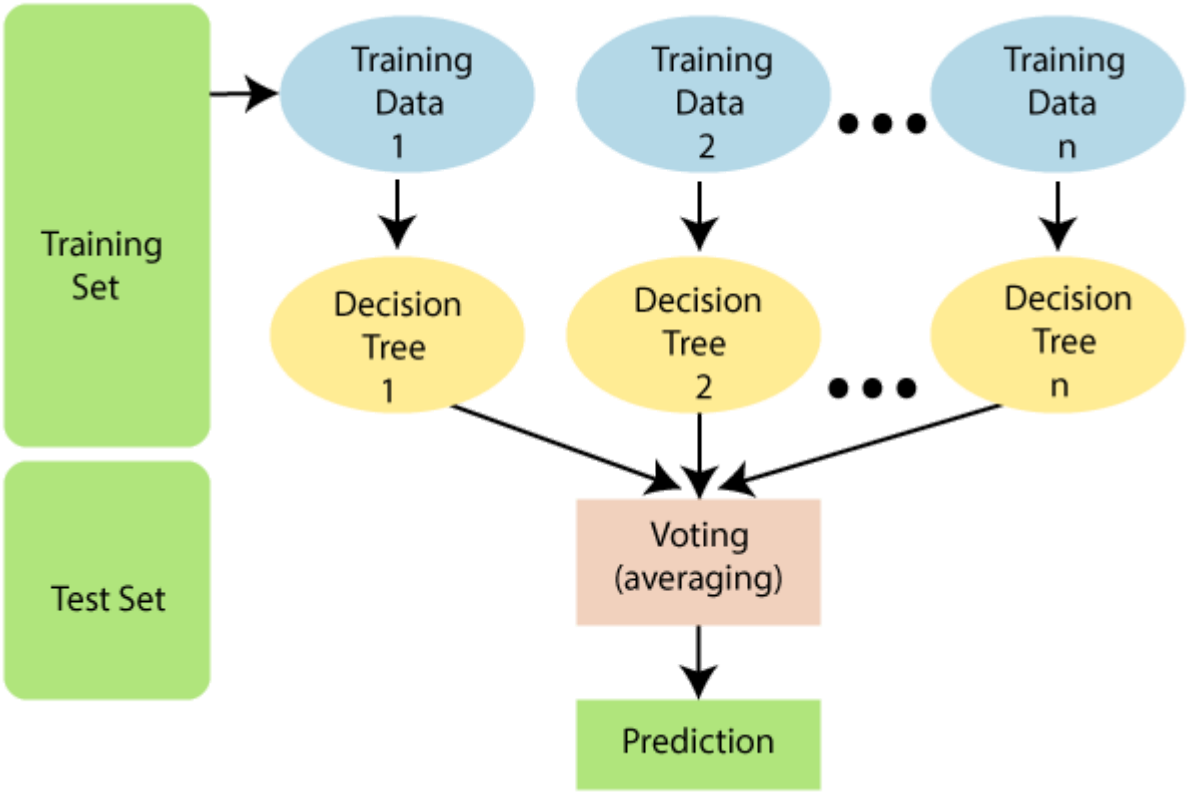
Theory:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Dataset:

Predict whether income exceeds $50K/yr based on census data. Also known as "Adult" dataset.

Attribute Information:

Listing of attributes:

>50K, <=50K.

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

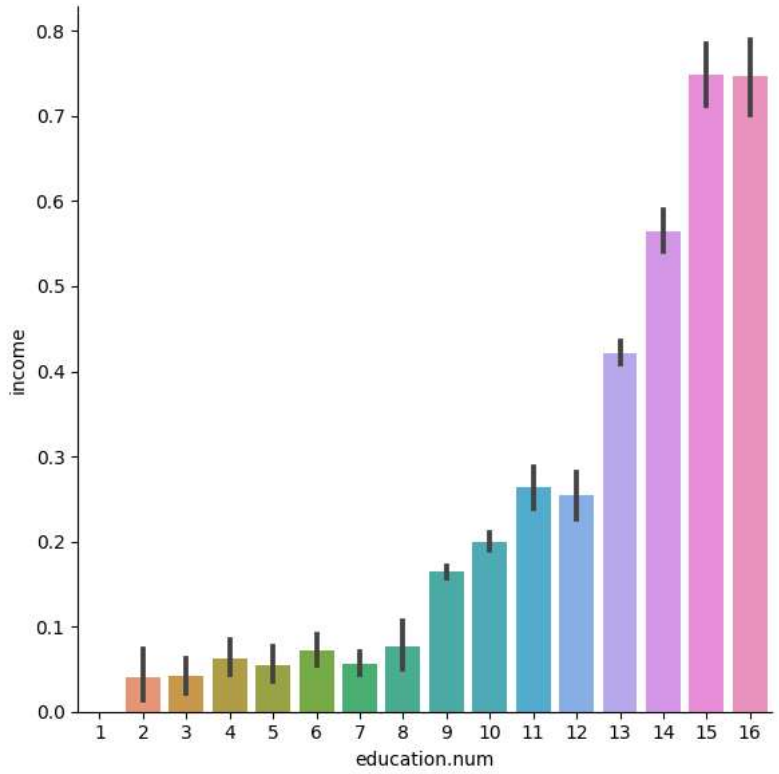
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad &Tobago, Peru, Hong, Holand-Netherlands.

Code:

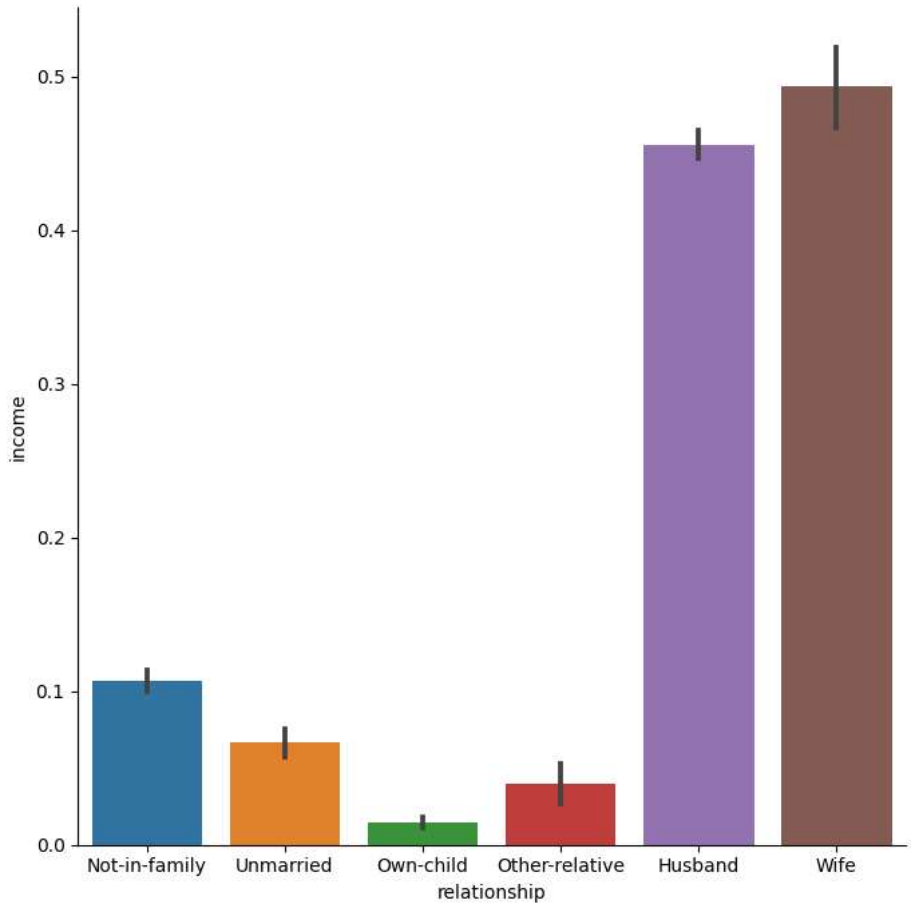
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split,cross_val_score,KFold,GridSearchCV
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
import scikitplot as skplt
import warnings

warnings.filterwarnings("ignore")

dataset = pd.read_csv('adult.csv')
dataset.head()
```

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female	0
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0

```
dataset = dataset[(dataset != '?').all(axis=1)]
dataset['income']=dataset['income'].map({'<=50K': 0, '>50K': 1})

sns.catplot(x='education.num',y='income',data=dataset,kind='bar',height=6)
plt.show()
```



```
sns.catplot(x='relationship', y='income', data=dataset, kind='bar', height=7)
plt.show()
```




```
dataset['marital.status']=dataset['marital.status'].map({'Married-civ-spouse':'Married', 'Divorced':'Single', 'Never-married':'Single', 'Sepa
'Widowed':'Single', 'Married-spouse-absent':'Married', 'Married-AF-spouse':'Married'})


for column in dataset:
    enc=LabelEncoder()
    if dataset.dtypes[column]==np.object:
         dataset[column]=enc.fit_transform(dataset[column])


plt.figure(figsize=(14,10))
sns.heatmap(dataset.corr(),annot=True,fmt='.2f')
plt.show()
```


```
dataset=dataset.drop(['relationship','education'],axis=1)
```

```
dataset=dataset.drop(['occupation','fnlwgt','native.country'],axis=1)
```

```
print(dataset.head())
```

```
   age  workclass  education.num  marital.status  race  sex  capital.gain  \
1   82          2              9               1     4    0             0
3   54          2              4               1     4    0             0
4   41          2             10               1     4    0             0
5   34          2              9               1     4    0             0
6   38          2              6               1     4    1             0

   capital.loss  hours.per.week  income
1          4356              18       0
3          3900              40       0
4          3900              40       0
5          3770              45       0
6          3770              40       0
```

```
X=dataset.iloc[:,0:-1]
y=dataset.iloc[:,-1]
print(X.head())
print(y.head())
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.33,shuffle=False)
```

```
   age  workclass  education.num  marital.status  race  sex  capital.gain  \
1   82          2              9               1     4    0             0
3   54          2              4               1     4    0             0
4   41          2             10               1     4    0             0
5   34          2              9               1     4    0             0
6   38          2              6               1     4    1             0

   capital.loss  hours.per.week
1          4356              18
3          3900              40
4          3900              40
5          3770              45
6          3770              40
1    0
3    0
4    0
5    0
6    0
Name: income, dtype: int64
```

```
clf=GaussianNB()
cv_res=cross_val_score(clf,x_train,y_train,cv=10)
print(cv_res.mean()*100)
```

```
76.68213951528749
```

```
clf=DecisionTreeClassifier()
cv_res=cross_val_score(clf,x_train,y_train,cv=10)
print(cv_res.mean()*100)
```

```
74.23325135581347
```

```
clf=RandomForestClassifier(n_estimators=100)
cv_res=cross_val_score(clf,x_train,y_train,cv=10)
print(cv_res.mean()*100)
```

```
76.73219315993944
```

```
clf=RandomForestClassifier(n_estimators=50,max_features=5,min_samples_leaf=50)
clf.fit(x_train,y_train)
pred=clf.predict(x_test)
print("Accuracy: %f " % (100*accuracy_score(y_test, pred)))
```

```
Accuracy: 84.558971
```

```
classification_rep = classification_report(y_test, pred)
print("Classification Report:")
print(classification_rep)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.94      0.91      7942
           1       0.68      0.45      0.54      2012

    accuracy                           0.85      9954
   macro avg       0.77      0.70      0.73      9954
weighted avg       0.83      0.85      0.83      9954
```

```
confusion_mat = confusion_matrix(y_test, pred)
print("Confusion Matrix:")
print(confusion_mat)
```

```
Confusion Matrix:
[[7503  439]
 [1098  914]]
```

Conclusion:-

1. The correlation heatmap is one of the important metrics used to understand how the attributes are related to each other. In this experiment we can see from the heatmap "education" and "education.num" are highly correlated, same can be said about the "marital.status" and "relationship" ,thus,we can drop "relationship" and "education".

2. We have achieved an accuracy of 84%.
   The following was the confusion matrix obtained
   Confusion Matrix:
   [[7524 418]
   [1114 898]]
   We achieved precision of 0.87 on 0 class and 0.68 on 1 class
   We got recall of 0.95 on 0 class and 0.45 on 1 class.
   We got F1 score of 0.91 on 0 class and 0.54 on 1 class.

3. Using both Decision tree and Random Forest we have achieved accuracy of 84% in both algorithms. However the other metrics like precision, recall and F1 score was lower in Random Forest on 1 class.