**Vivekanand Education Society’s**

**Institute of Technology**

**(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)**

**Department of Information Technology**

AIDS - 2 Lab

Experiment - 10

Aim: Supervised learning algorithm Random Forest

| Roll No. | 53 |
| --- | --- |
| Name | SAKSHI SANTOSH PATIL |
| Class | D20B |
| Subject | AIDS - 2 |
| Grade: |  |

**EXPERIMENT - 10**

**AIM:** Supervised learning algorithm Random Forest

**THEORY**:

**Supervised learning** is a type of machine learning where an algorithm learns from labeled training data to make predictions or decisions without human intervention. It is called "supervised" because it involves a "teacher" who provides the algorithm with the correct answers during training, allowing the algorithm to learn the relationship between input data and output labels.

**Random Forest** is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap. We need to approach the Random Forest regression technique like any other machine learning technique.

**Advantages Random Forest Regression**

* It is easy to use and less sensitive to the training data compared to the decision tree.
* It is more accurate than the decision tree algorithm.
* It is effective in handling large datasets that have many attributes.
* It can handle missing data, outliers, and noisy features.

**Disadvantages Random Forest Regression**

* The model can also be difficult to interpret.
* This algorithm may require some domain expertise to choose the appropriate parameters like the number of decision trees, the maximum depth of each tree, and the number of features to consider at each split.
* It is computationally expensive, especially for large datasets.
* It may suffer from overfitting if the model is too complex or the number of decision trees is too high.

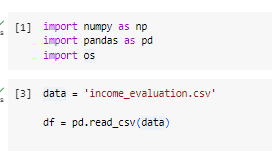
\

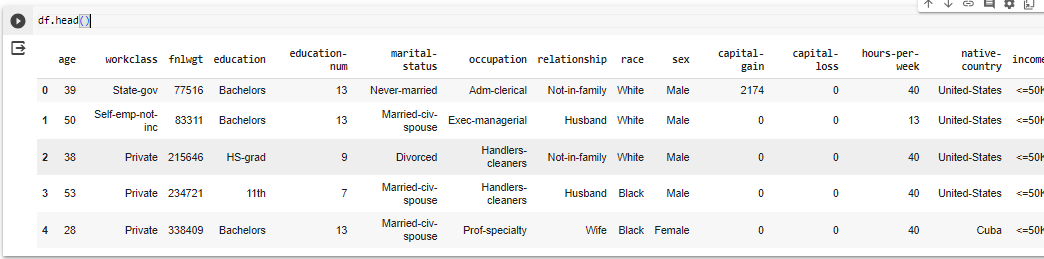
**IMPLEMENTATION**:

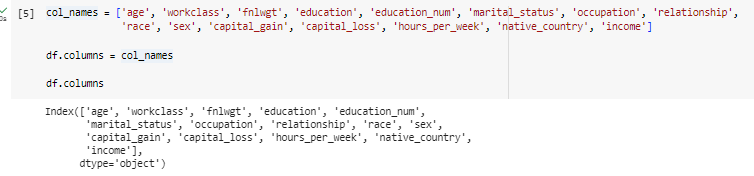
**TO-DO:** To make predictions where the prediction task is to determine whether a person makes over 50K a year. Implementing Random Forest Classification with Python and Scikit-Learn.

**Dataset Link**: <https://www.kaggle.com/datasets/lodetomasi1995/income-classification>

* **Import libraries and read the dataset**



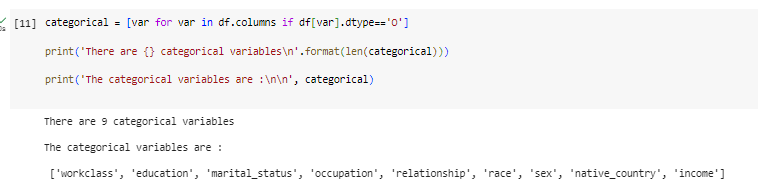


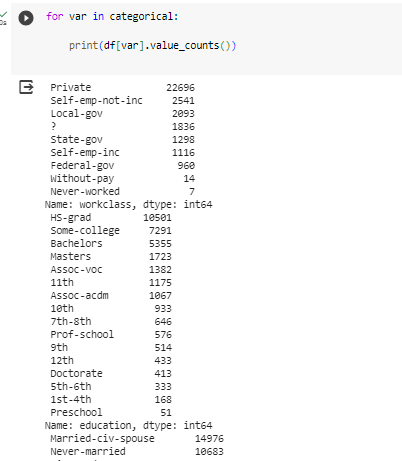


* **Checking if any null values are present or not**

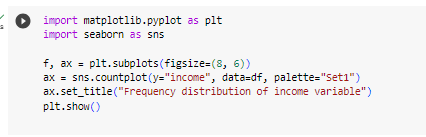


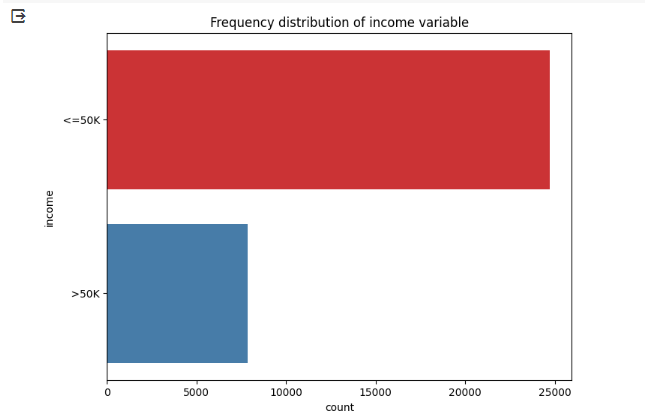
* **Extracting the categorical data**



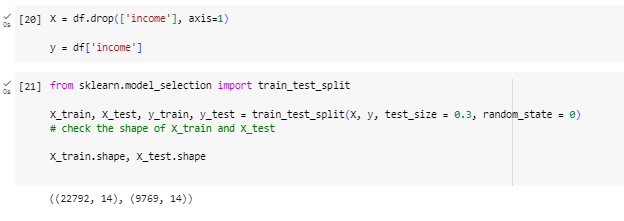


* **Plotting the frequency distribution graph**





* **Splitting the dataset into test and train**



* **Performing hot encoding**



* **This code scales your training and test data using RobustScaler to ensure consistent feature scales, which is important for certain machine learning algorithms.**

cols = X\_train.columns

from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

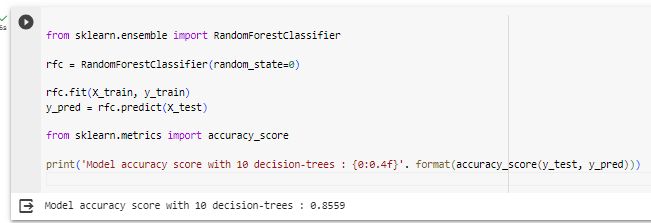
X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

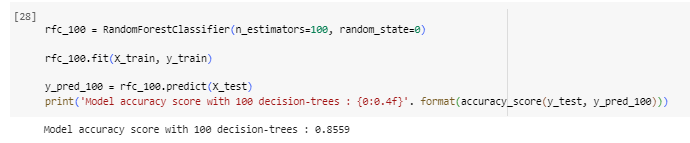
X\_train = pd.DataFrame(X\_train, columns=[cols])

X\_test = pd.DataFrame(X\_test, columns=[cols])

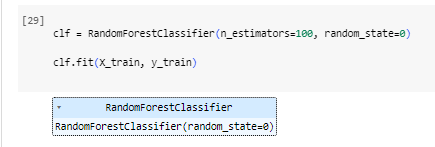
* **We now have X\_train dataset ready to be fed into the Random Forest classifier**
* **We check the accuracy of the model with 10 decision trees**

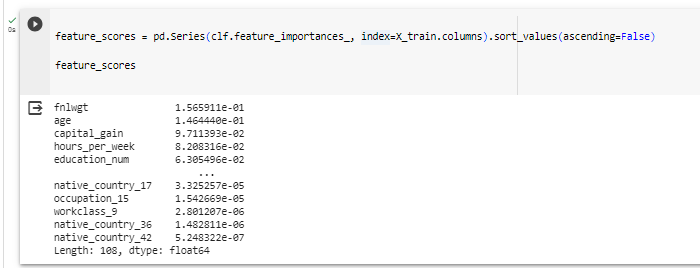


* **Checking the accuracy of the model with 100 decision trees**

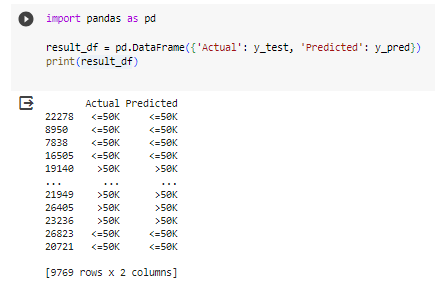


* **Random forest classifier**

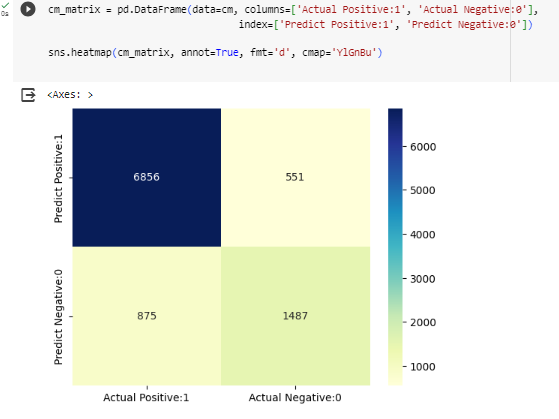




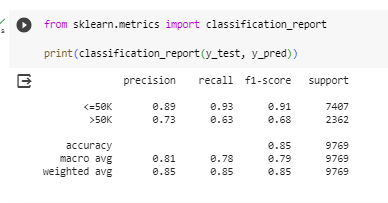
* **Predicted vs Actual Output**



* **Confusion Matrix of the model**



* **Calculating precision, recall, f1 score and support**



**CONCLUSION:**

Therefore, Random Forest is used for predicting results in machine learning. It is a powerful ensemble learning method that can be used for both classification and regression tasks. Random Forest combines the predictions of multiple decision trees to produce more accurate and robust predictions. Thus we successfully implemented random forest for income classification.