Name :

Adult Census Income

Report Document will contain :

\*Introduction

\*Dataset Preprocessing

\*Classifier Evaluation

\*Visualization Analysis

\*Classifier Comparison

\*Conclusion

**Introduction**

**1. Dataset Description:** The "Adult Census Income" dataset is a collection of census data extracted from the 1994 Census Bureau database by Ronny Kohavi and Barry Becker. The dataset includes various demographic and employment-related features of individuals and is used to predict whether a person makes over $50K a year. The data was processed to ensure reasonably clean records by applying specific conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)).

The dataset is available on Kaggle and includes 15 columns and 48,842 rows. The columns are as follows:

1. **age**: The age of the individual.
2. **workclass**: The class of work the individual belongs to.
3. **fnlwgt**: Final weight, an estimate of the population that the individual represents.
4. **education**: The highest level of education achieved by the individual.
5. **education-num**: The number of years of education.
6. **marital-status**: The marital status of the individual.
7. **occupation**: The occupation of the individual.
8. **relationship**: The relationship status of the individual within a family.
9. **race**: The race of the individual.
10. **sex**: The gender of the individual.
11. **capital-gain**: Income from capital gains.
12. **capital-loss**: Income from capital losses.
13. **hours-per-week**: The number of hours worked per week.
14. **native-country**: The country of origin.
15. **income**: The income class (<=50K or >50K).

**Dataset Link:** [Adult Census Income Dataset on Kaggle](https://www.kaggle.com/datasets/uciml/adult-census-income" \t "_new)

**Project Description: Predicting Income Using the Adult Census Income Dataset**

**Introduction:** The "Adult Census Income" dataset is a comprehensive collection of demographic and employment information used to predict whether an individual's annual income exceeds $50K. This dataset, prepared by Ronny Kohavi and Barry Becker, has been widely used in data mining and machine learning applications.

**Objective:** The objective of this project is to build a machine learning model that can accurately predict whether an individual's income exceeds $50K based on their demographic and employment attributes.

**Problem Statement:** Predicting whether an individual's income exceeds $50K using the "Adult Census Income" dataset is a binary classification problem. The goal is to classify individuals into one of two income classes (<=50K or >50K) based on various features such as age, education, occupation, and hours worked per week.

To solve this problem, we will follow these steps:

**Data Preprocessing:**

* 1. Handle missing values.
  2. Normalize or standardize numerical features.
  3. Encode categorical attributes.
  4. Perform feature selection if necessary.

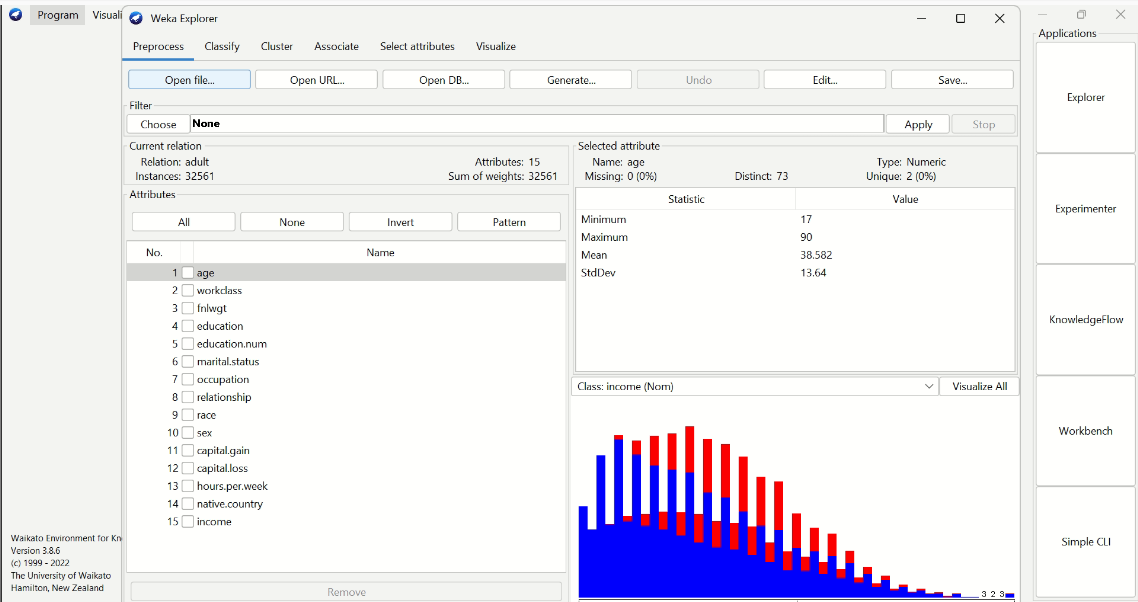
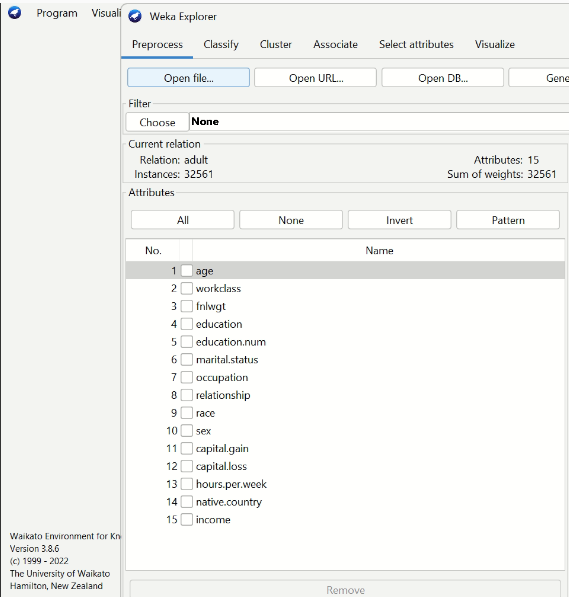
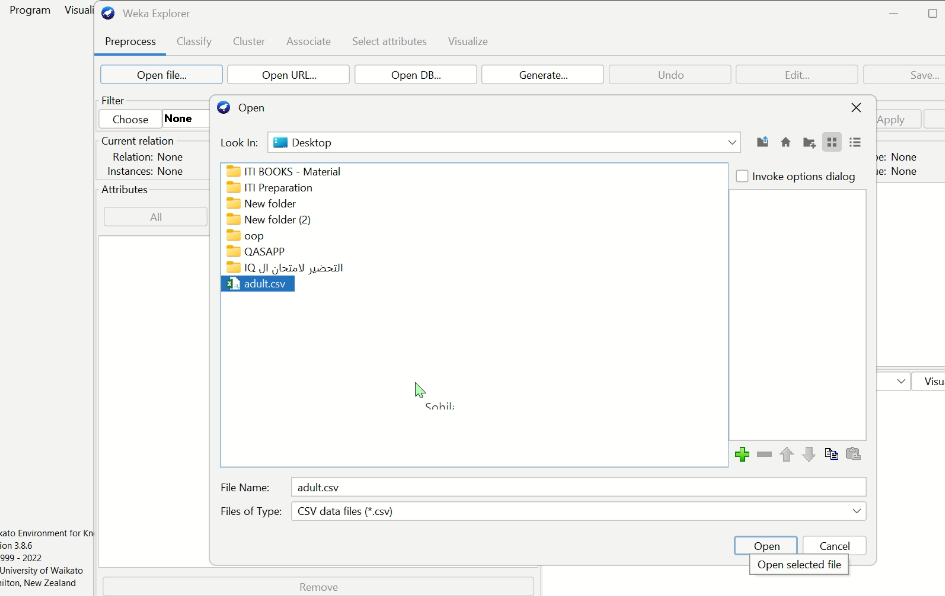
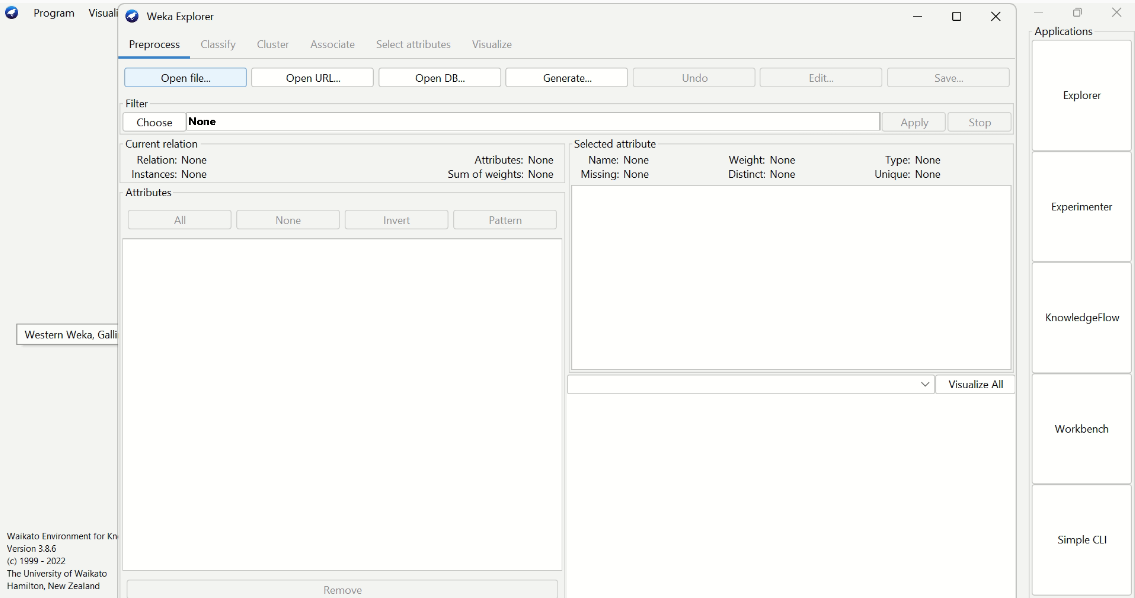
**Model Selection:**

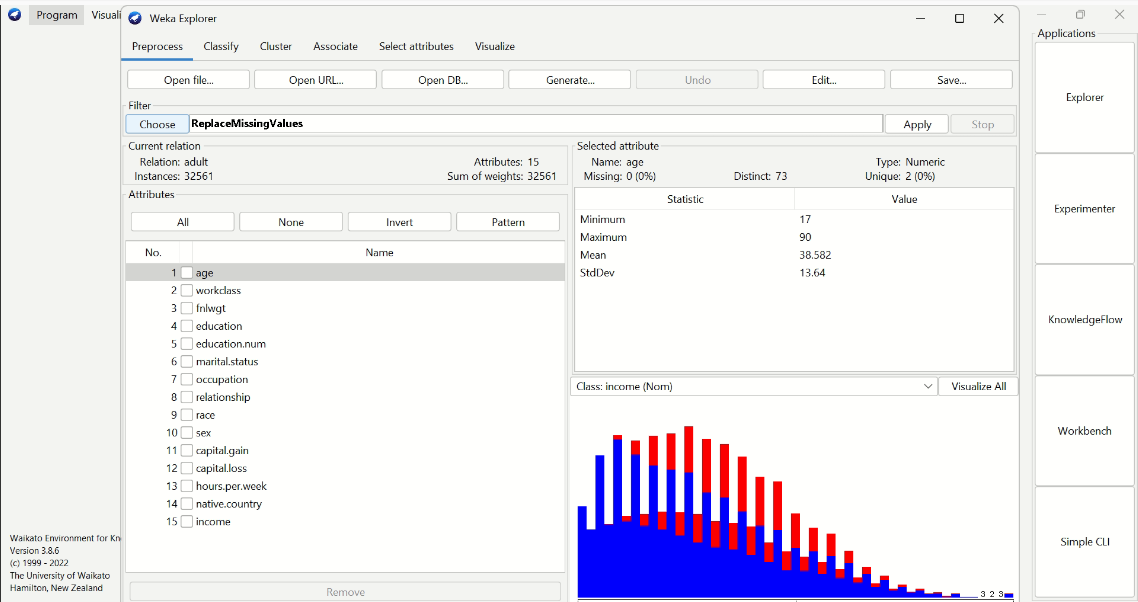
* 1. Train and evaluate different classification models such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN).
  2. Choose the best-performing model based on evaluation metrics such as Accuracy, Precision, Recall, and F1-Score.

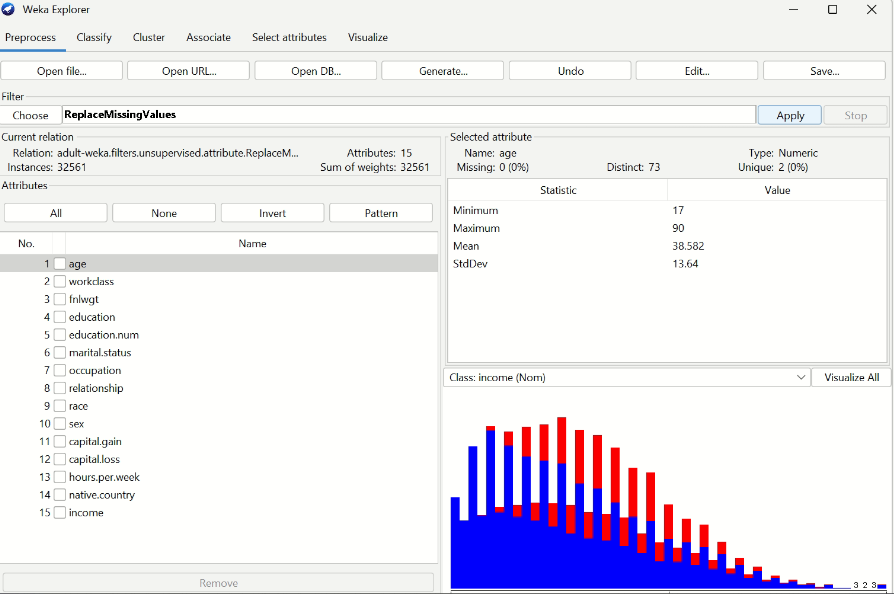
Data preprocessing steps:

Here are some data preprocessing steps that can be performed on the "House Price India" dataset using Weka:

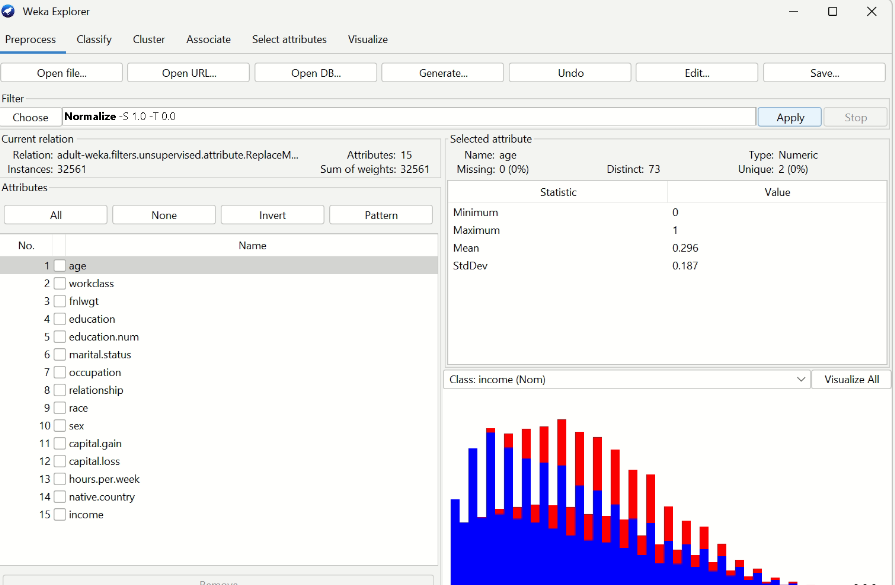
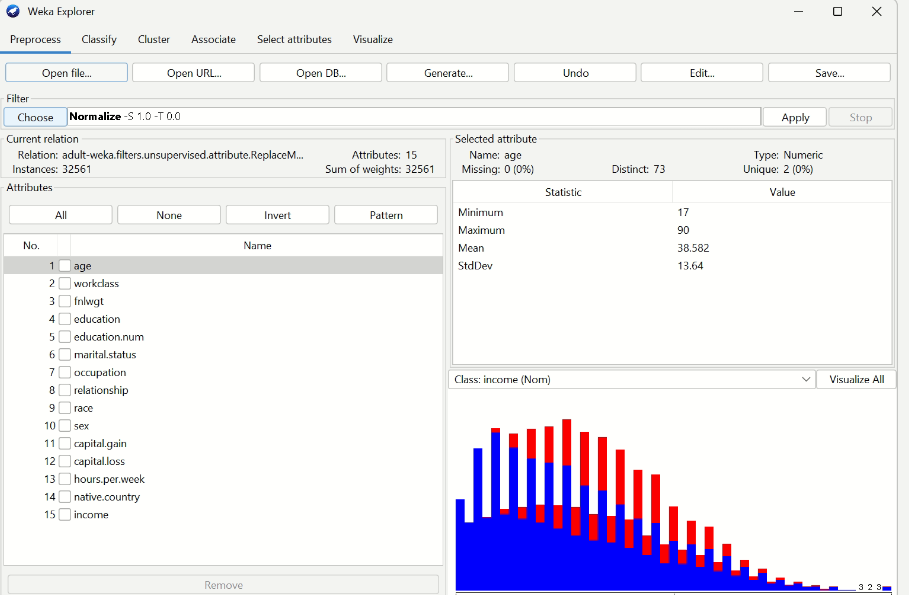
1-Handling missing values: The dataset may contain missing values, which can be handled by replacing them with suitable values such as the mean or median of the column, or by removing the rows or columns with missing values altogether. In Weka, you can use the "ReplaceMissingValues" filter to replace missing values with the mean or median of the column





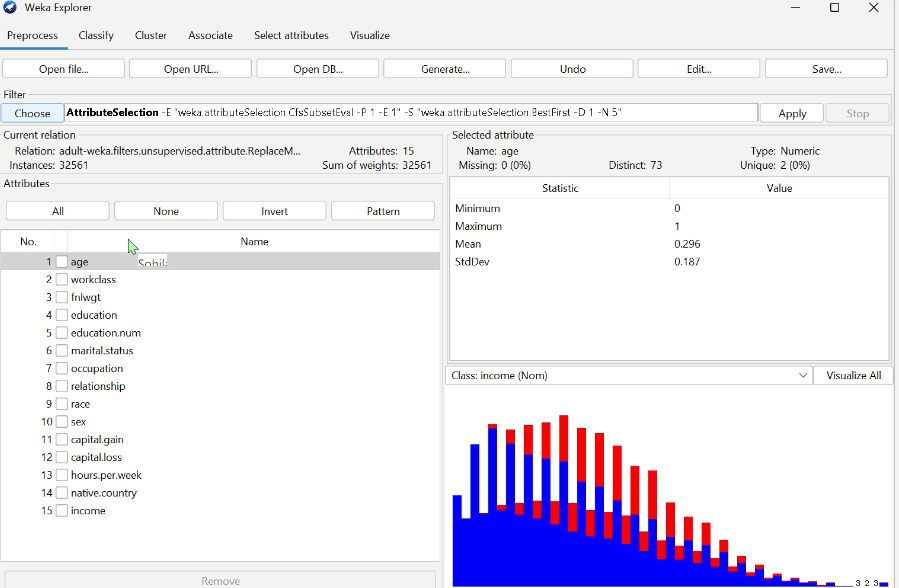


1. Normalize the data

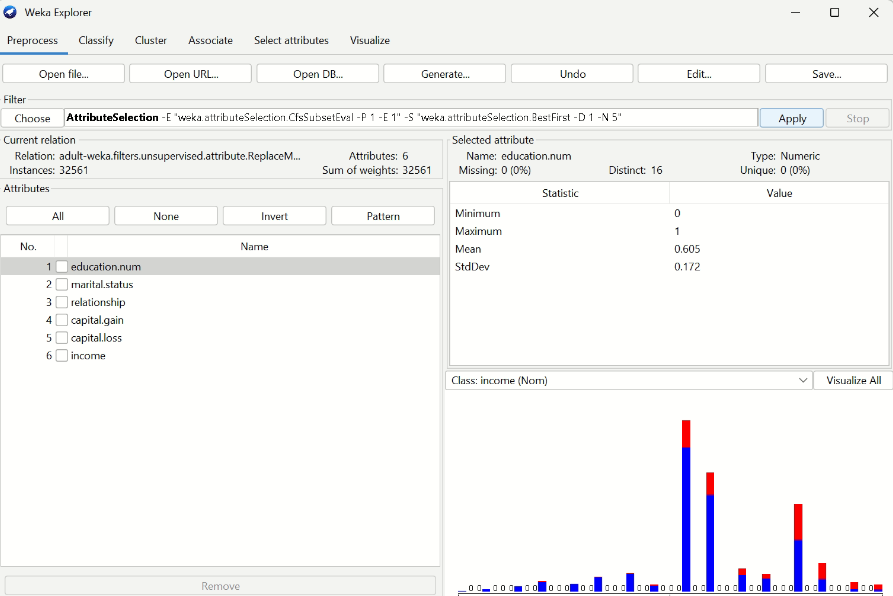


3-select important feature

1. Click on the "Preprocess" tab in the top menu.
2. Click on the "Filter" drop-down menu and select "AttributeSelection".
3. In the "AttributeSelection" window, select the search method you want to use to select important features, such as "Ranker" or "CfsSubsetEval".
4. Set any other options for the search method you chose, such as the number of top-ranked features to keep.
5. Click "Apply" to run the filter and select the important features.
6. Save the new dataset with the important features selected by the filter.

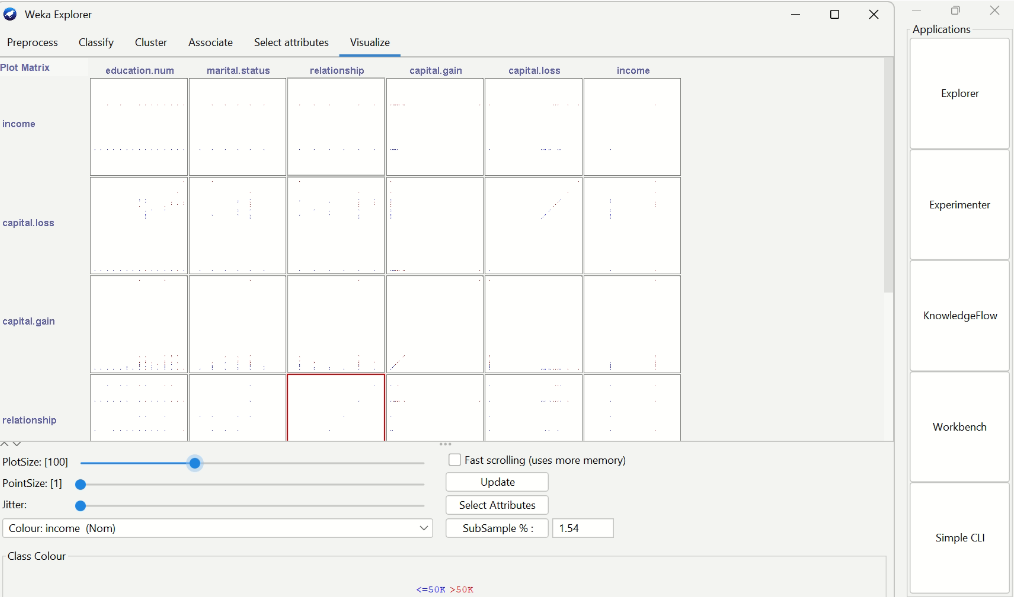
Before

After



Then

\*choose visualize all from Process tab



Classifier evaluation

\* **Model Selection steps:**

The 4 algorithms that we will review are:

1. Random Forest
2. Decision Tree

3-kNN(

■ Supplied test set

■ Percentage split (e.g., 70% training, 30% testing)

■ 10-fold cross-validation))

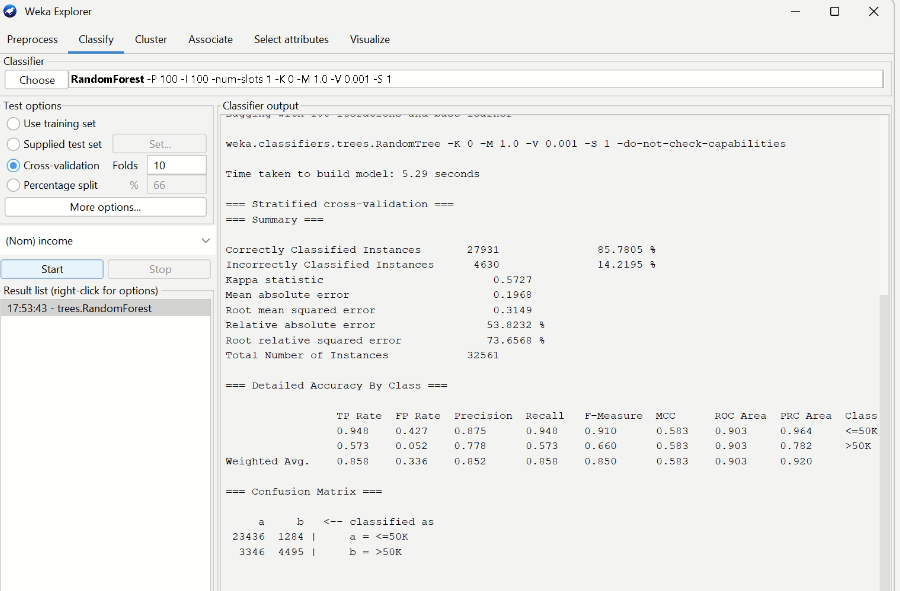
1-Random Forest

Choose Random Forest algorithm:

1-Click the “Choose” button and select “Random

Forest” under the “trees” group.

2-Click on the name of the algorithm to review the algorithm configuration.



3-Click “OK” to close the algorithm configuration.

4-Click the “Start” button to run the algorithm on the Adult Census Income

You can see that with the default configuration that linear regression achieves an RMSE of 0.3149

## 2-Decision Tree

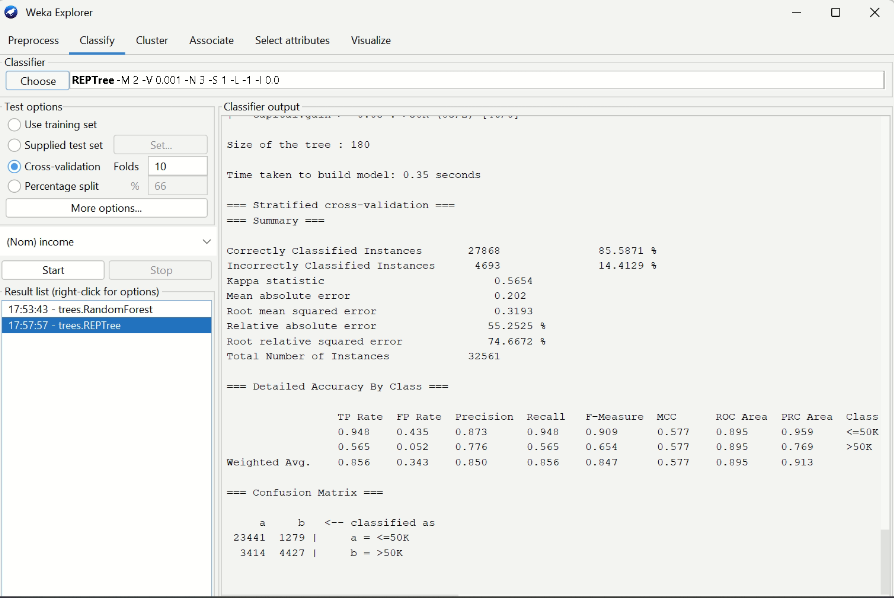
Choose the decision tree algorithm:

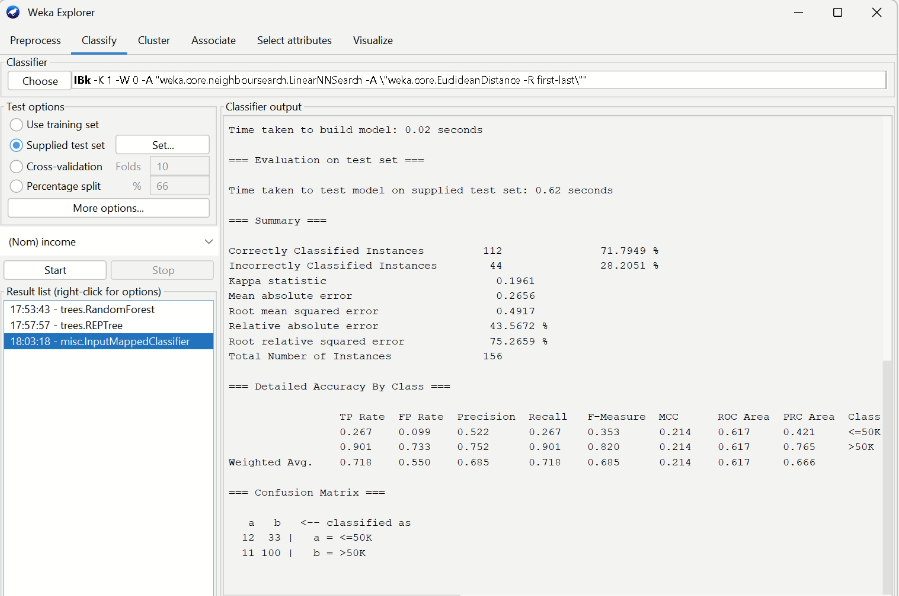
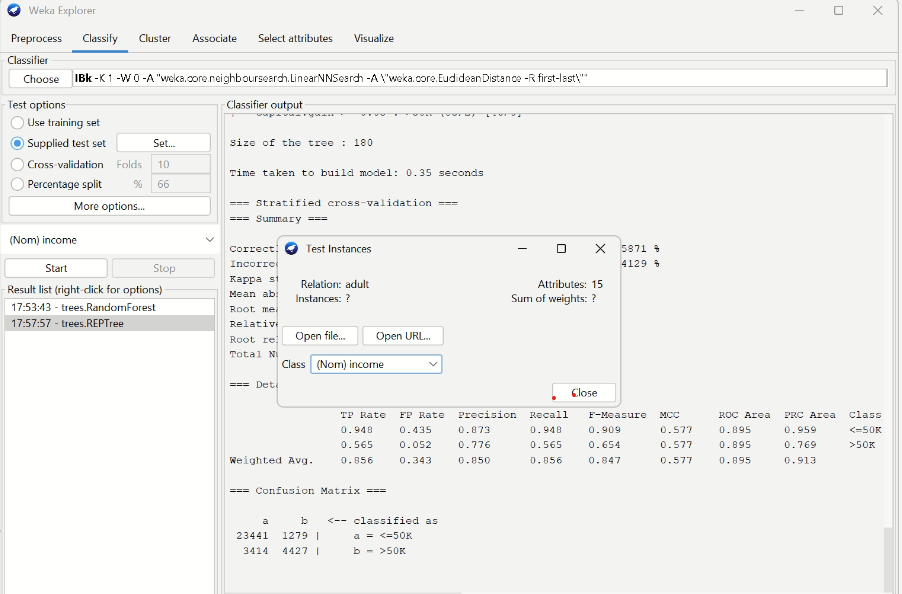
1. Click the “Choose” button and select “REPTree” under the “trees” group.
2. Click on the name of the algorithm to review the algorithm configuration.

3-Click “OK” to close the algorithm configuration.

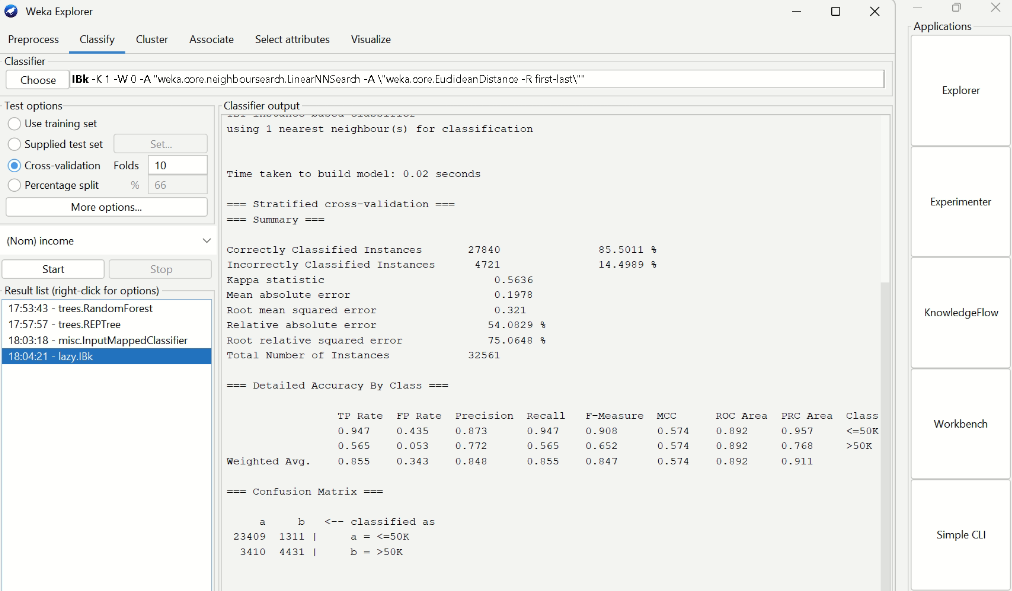
4-Click the “Start” button to run the algorithm on the Adult Census Income

You can see that with the default configuration that linear regression achieves an RMSE of 0.3193

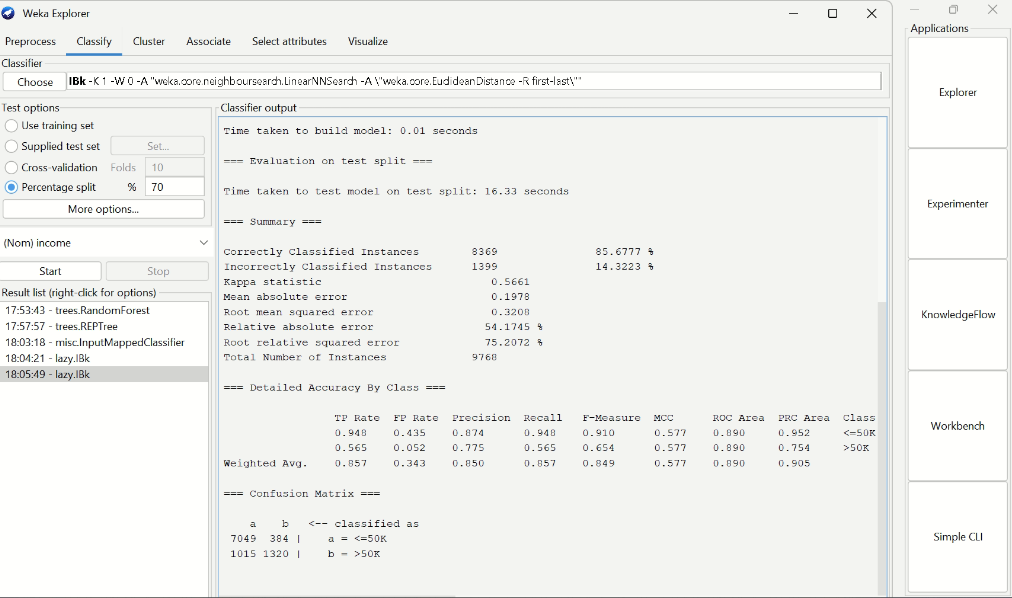


1. KNN(Supplied test set)

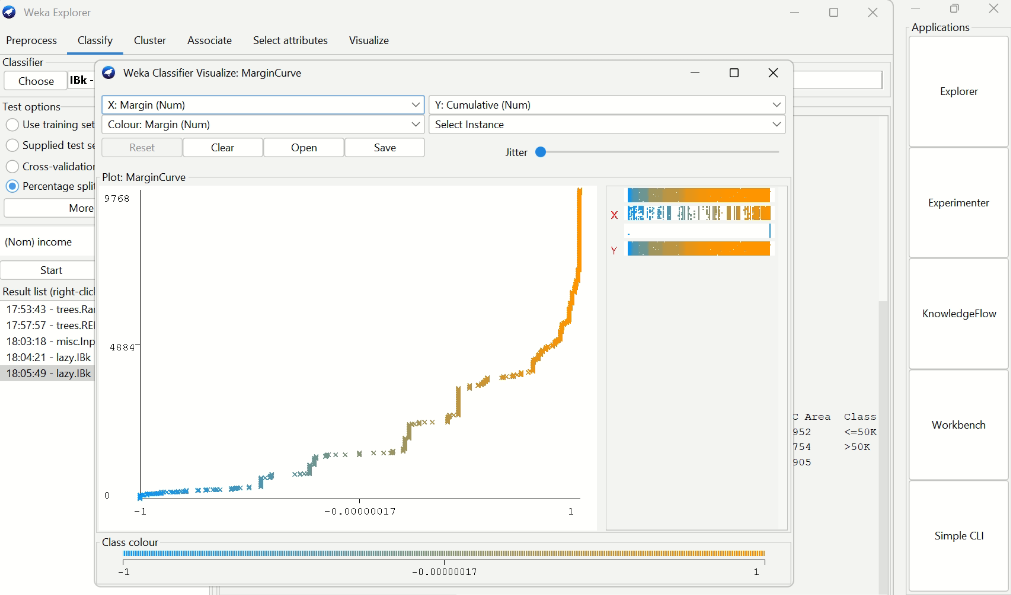
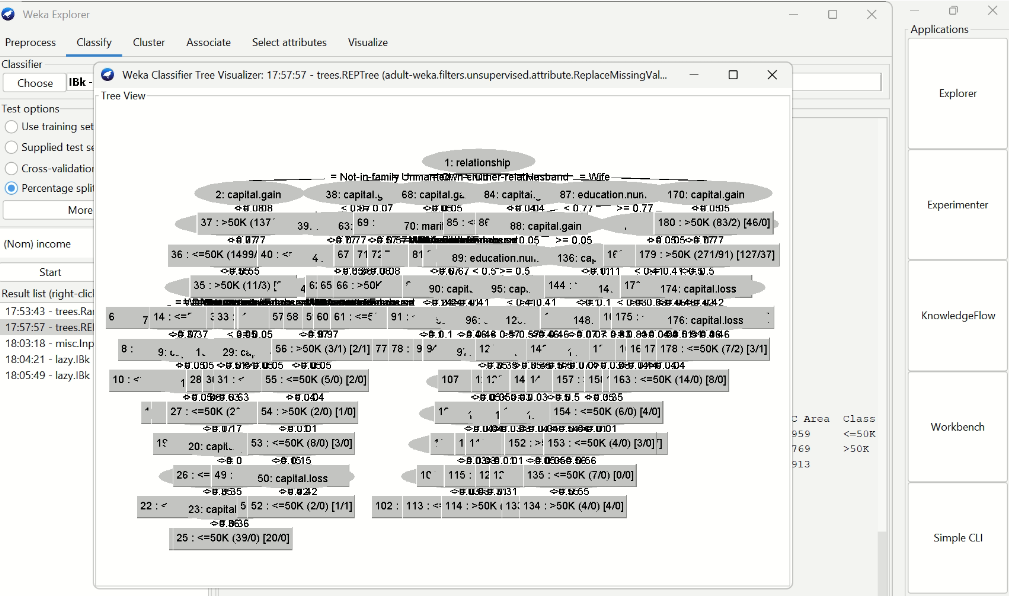
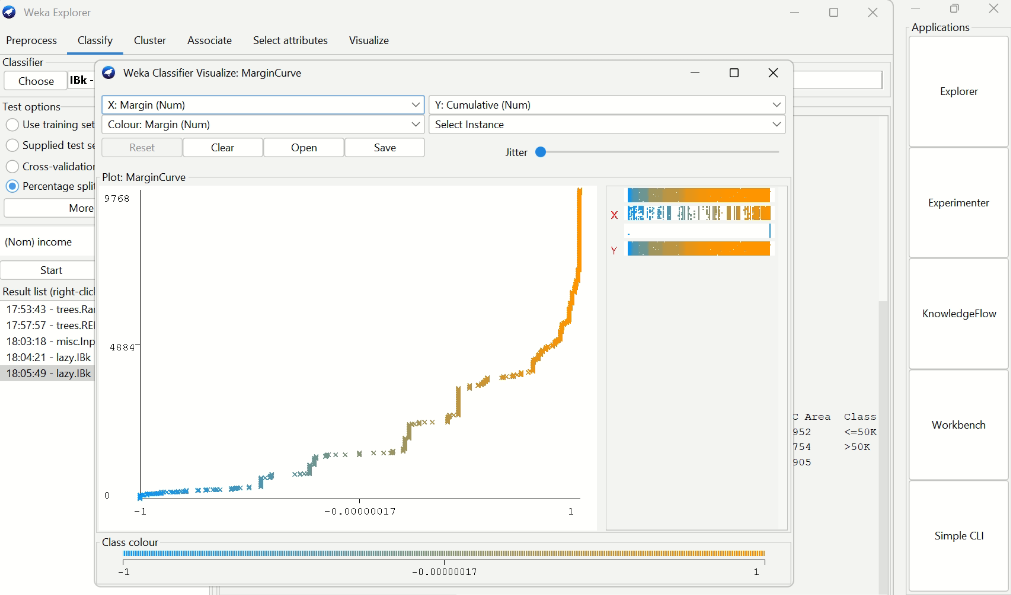
KNN(10-fold cross-validation)

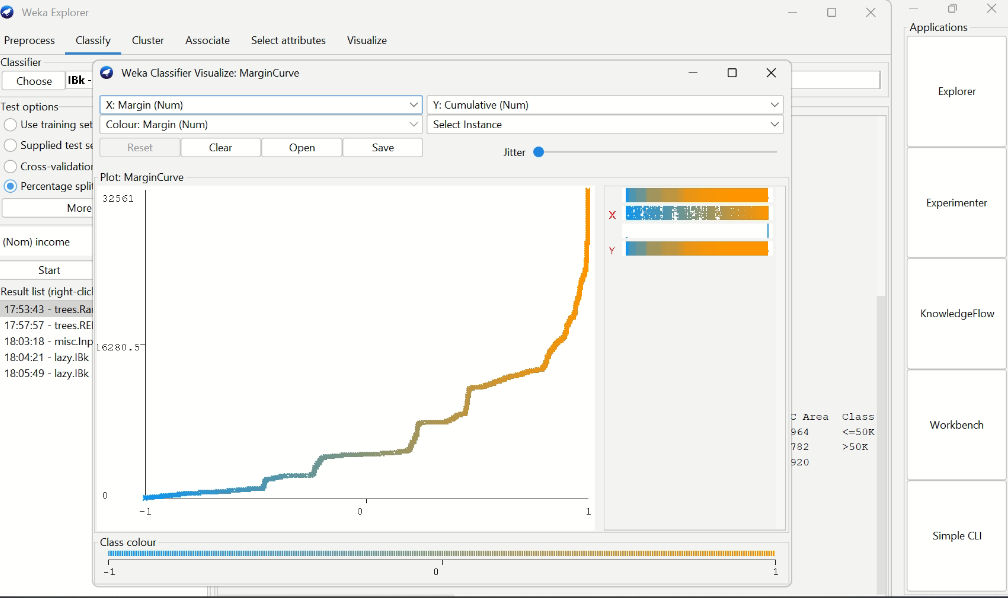


Percentage split ( 70% training, 30% testing)



Visualization Analysis





Classifier Comparison

The comparison of the results between different algorithms for the used mining task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1-Random Forest | 2-Decision Tree | 3-kNN( Supplied test set ) | 4- kNN(Percentage split (e.g., 70% training, 30% testing) | 5-kNN( 10-fold cross-validation)) |
| 0.3149 | 0.3193 | 0.4917 | 0.321 | 0.3208 |

So the best algorithm can we applied in this data is kNN(■ Supplied test set ))