

Lab Number: 1

Title

Preprocessing of Primary and Secondary Datasets Containing Dirty Data

Objective

To understand and practically perform data preprocessing on a dirty dataset using Weka Explorer.

IDE/Tools Used

Weka 3.8.6

Theory

Primary Dataset: The original, raw, operational/transactional data that comes directly from the source system (OLTP). It contains all details in one big flat table with lots of redundancy, mixed data types, and no optimization for analysis.

Secondary Dataset: Data that has been extracted, cleaned, transformed, and loaded (ETL) into a data warehouse for analytical processing (OLAP). It is split into fact tables + dimension tables (Star/Snowflake/Galaxy schemas).

Dirty Data: A dirty dataset refers to a collection of data that contains inaccuracies, inconsistencies, and errors, which can compromise its usefulness and reliability for analysis, reporting, or decision-making

Types of Dirty Data

- Missing Data
- Duplicate Records
- Inconsistent Values
- Noise
- Outliers

Data Preprocessing: Data preprocessing is the process of cleaning, transforming, and organizing raw data into a structured format that is ready for analysis or use in machine learning models.

- **Cleaning:** Involves handling missing values, removing duplicates, and correcting errors to make the data accurate and consistent.
- **Transformation:** Involves converting data into a suitable format. Examples include standardizing numerical features, normalizing data, or encoding categorical variables.
- **Integration:** Combines data from multiple sources into a single, unified dataset for analysis.

Implementation

A. For Primary Dataset

For primary dataset, a customer churn data was generated.

Steps used to clean the data:

1. Open the dataset in the pre-processor of WEKA

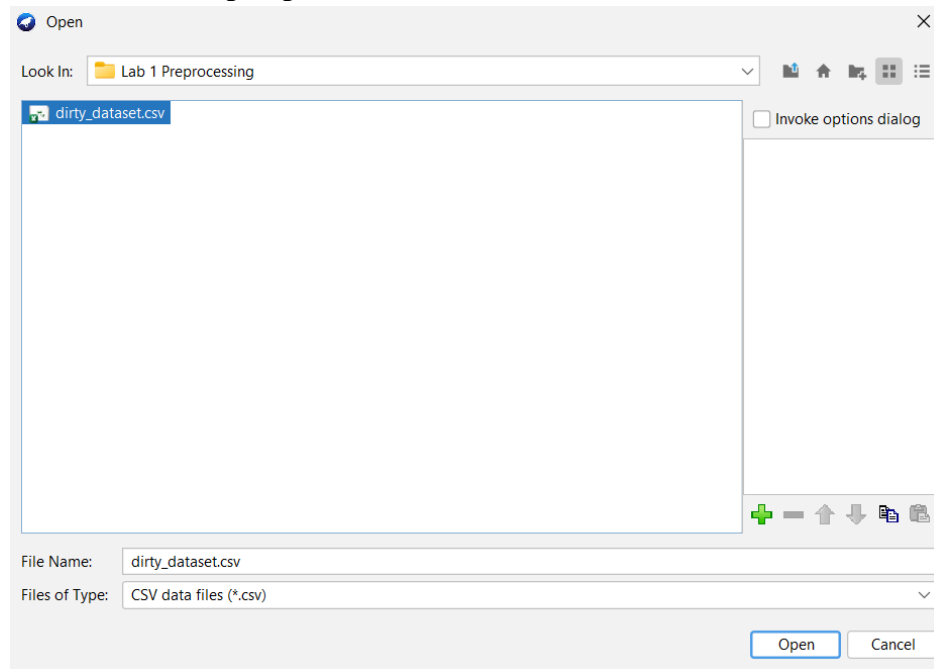


Figure 1: Opening the dataset in weka

2. Visualize the data

Viewer									
Relation: dirty_dataset									
No.	1: CustomerID String	2: Age Numeric	3: Gender Nominal	4: Income Numeric	5: Region Nominal	6: Spend Numeric	7: SignupDate Nominal	8: LastPurchase Nominal	9: Churn Nominal
1	1	25.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
2	2		Female		Europe	850.0	2023-02-30		No
3	3	45.0	Male	120000.0	Asia	5000.0	2023-03-10	2025-01-15	Yes
4	4	32.0	F	75000.0	South A...	3200.0	2023-04-05	2024-11-20	No
5	5	28.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
6	6	35.0		62000.0	Europe		2023-06-01	2024-10-10	No
7	7	999.0	Male	55000.0	Africa	300.0	2023-07-12	2023-07-12	Yes
8	8	41.0	Male	58000.0	North A...	1800.0	2023-08-20		No
9	9	29.0	Female	48000.0	Asia	1100.0	2023-09-05	2024-09-05	Yes
10	10	33.0	Male		Oceania	2200.0	2023-10-01	2024-12-10	No
11	CUST11	31.0	Female	70000.0	Europe	1500.0	2023-11-11	2024-11-11	Yes
12	12	27.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
13	13	62.0	Male	85000.0	Moon	999999.0	2025-12-01	2025-12-01	No

Figure 2: Visualization of the dataset

3. Remove unwanted columns

In this case CustomerID was removed using the unsupervised.attribute.Remove filter

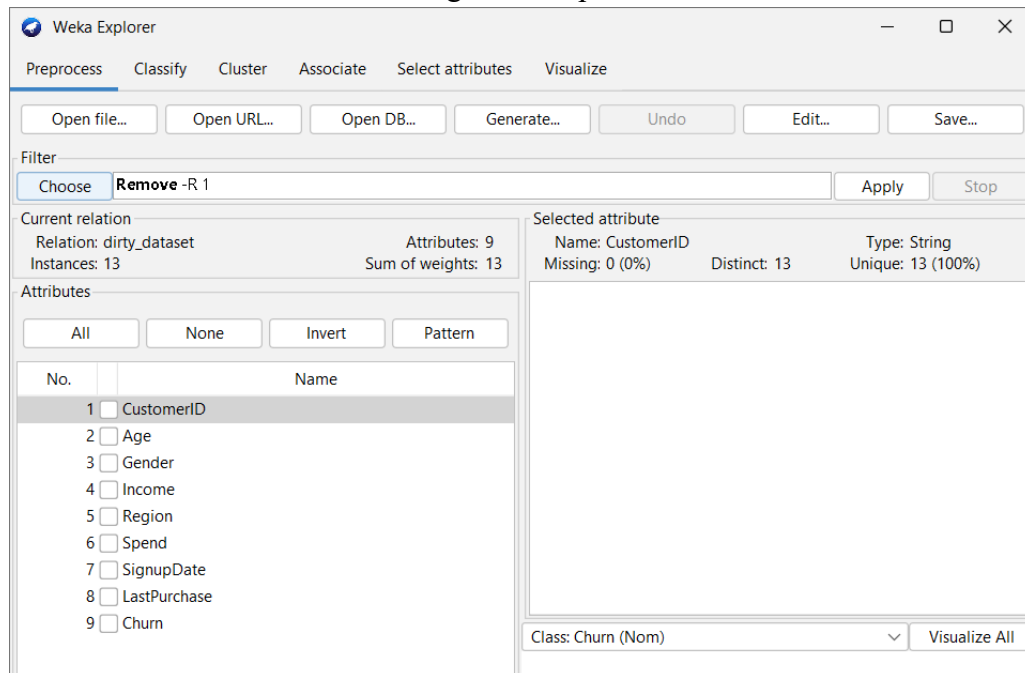


Figure 3: Applying the remove filter

Viewer								
Relation: dirty_dataset-weka.filters.unsupervised.attribute.Remove-R1								
No.	1: Age Numeric	2: Gender Nominal	3: Income Numeric	4: Region Nominal	5: Spend Numeric	6: SignupDate Nominal	7: LastPurchase Nominal	8: Churn Nominal
1	25.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
2		Female		Europe	850.0	2023-02-30		No
3	45.0	Male	120000.0	Asia	5000.0	2023-03-10	2025-01-15	Yes
4	32.0	F	75000.0	South A...	3200.0	2023-04-05	2024-11-20	No
5	28.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
6	35.0		62000.0	Europe		2023-06-01	2024-10-10	No
7	999.0	Male	55000.0	Africa	300.0	2023-07-12	2023-07-12	Yes
8	41.0	Male	58000.0	North A...	1800.0	2023-08-20		No
9	29.0	Female	48000.0	Asia	1100.0	2023-09-05	2024-09-05	Yes
10	33.0	Male		Oceania	2200.0	2023-10-01	2024-12-10	No
11	31.0	Female	70000.0	Europe	1500.0	2023-11-11	2024-11-11	Yes
12	27.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
13	62.0	Male	85000.0	Moon	999999.0	2025-12-01	2025-12-01	No

Figure 4: Dataset after removing the 1st column

4. Remove any duplicate values

In this case this was done using the `unsupervised.instance.RemoveDuplicates` filter.

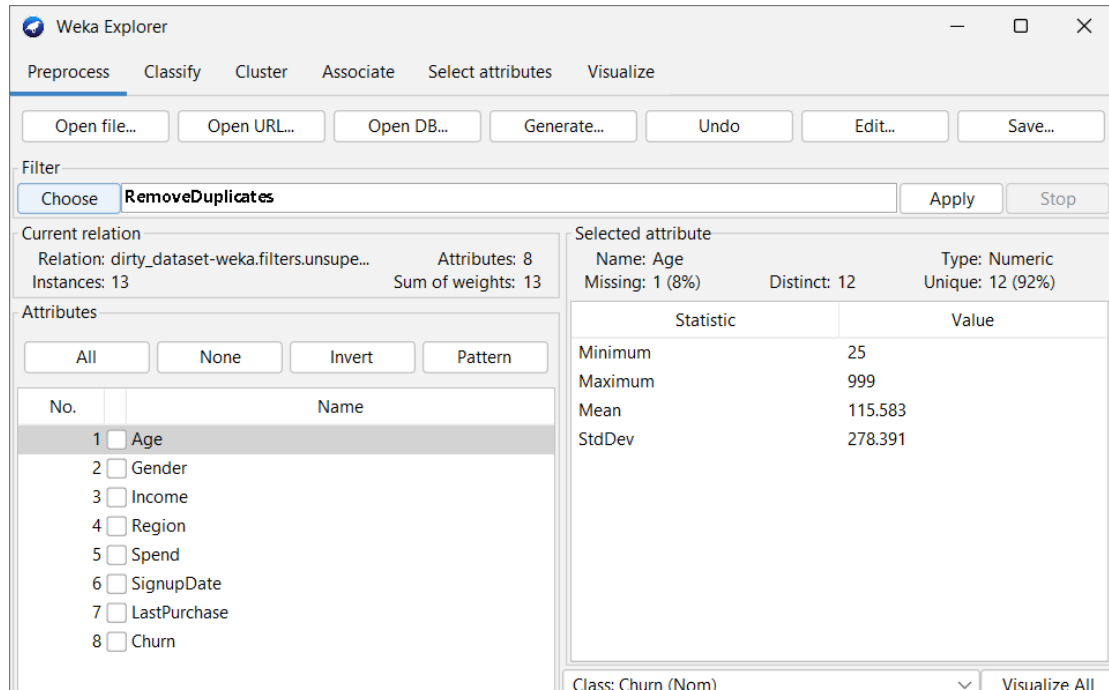


Figure 5: Applying the RemoveDuplicate Filter

5. Replace any missing values

In this case it was done using the `unsupervised.attribute.ReplaceMissingValues` filter.

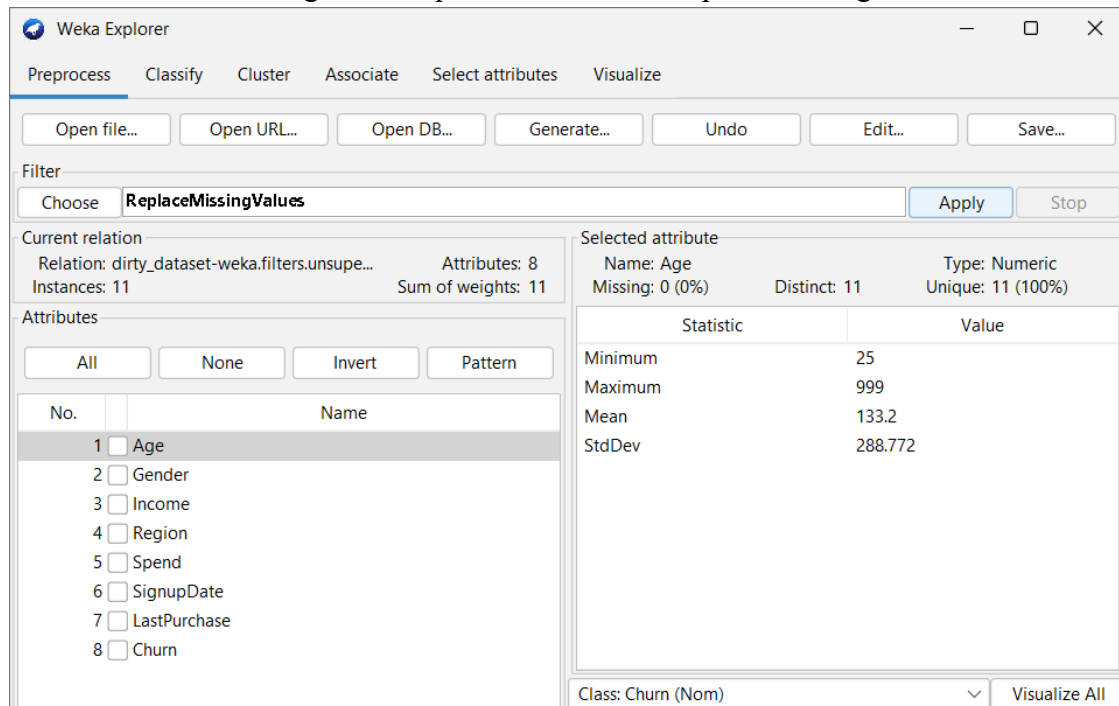


Figure 6: Applying the ReplaceMissingValues filter

6. Convert string into nominal values

This is done using the `unsupervised.attribute.StringToNominal` filter.

The screenshot shows the Weka Explorer interface. The 'Preprocess' tab is selected. The 'Filter' section shows 'StringToNominal' chosen. The 'Current relation' is 'dirty_dataset-weka.filters.unsupe...' with 8 attributes and 11 instances. The 'Attributes' list on the left shows 'Age' selected. The 'Selected attribute' section for 'Age' shows it is numeric with 11 distinct values. A statistics table for 'Age' is displayed, showing Minimum (25), Maximum (999), Mean (133.2), and StdDev (288.772). The 'Class' is set to 'Churn (Str)' and the 'Visualize All' button is visible.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose **StringToNominal** Apply Stop

Current relation
Relation: dirty_dataset-weka.filters.unsupe... Attributes: 8
Instances: 11 Sum of weights: 11

Attributes

All None Invert Pattern

No.	Name
1	<input checked="" type="checkbox"/> Age
2	<input type="checkbox"/> Gender
3	<input type="checkbox"/> Income
4	<input type="checkbox"/> Region
5	<input type="checkbox"/> Spend
6	<input type="checkbox"/> SignupDate
7	<input type="checkbox"/> LastPurchase
8	<input type="checkbox"/> Churn

Selected attribute
Name: Age Type: Numeric
Missing: 0 (0%) Distinct: 11 Unique: 11 (100%)

Statistic	Value
Minimum	25
Maximum	999
Mean	133.2
StdDev	288.772

Class: Churn (Str) Visualize All

10

Figure 7: Applying the StringToNominal Filter

7. Removing Outliers

To remove Outliers, we perform the following steps:

7.1. Interquartile Range

Choose Interquartile Range filter from unsupervised.attribute.InterquartileRange and select the following settings. This will give outlier and extreme values in the dataset.

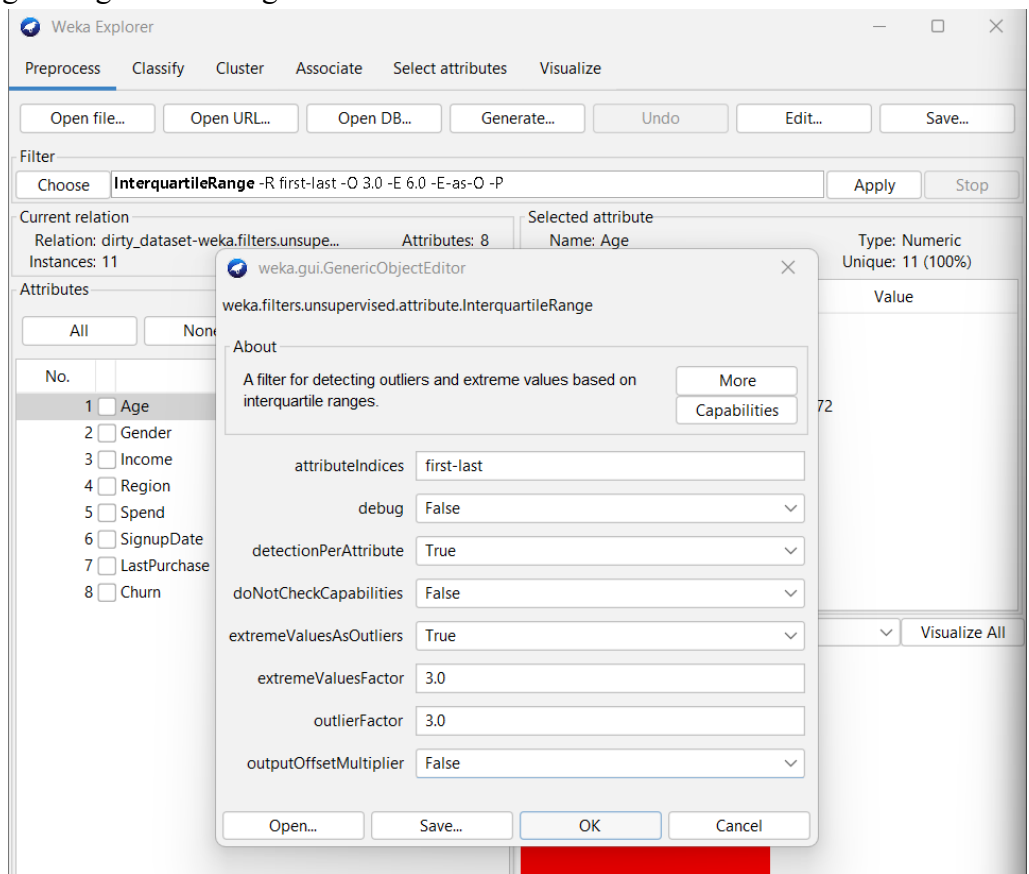


Figure 8: Selecting appropriate setting for InterquartileRange

No.	1: Age	2: Gender	3: Income	4: Region	5: Spend	6: SignupDate	7: LastPurchase	8: Age_Outlier	9: Age_ExtremeValue	10: Income_Outlier	11: Income_ExtremeValue	12: S
	Numeric	Nominal	Numeric	Nominal	Numeric	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal
1	25.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	no	no	no	no	no
2	133.2	Female	68666.666...	Europe	850.0	2023-02-30	2024-12-01	no	no	no	no	no
3	45.0	Male	120000.0	Asia	5000.0	2023-03-10	2025-01-15	no	no	no	no	no
4	32.0	F	75000.0	South A...	3200.0	2023-04-05	2024-11-20	no	no	no	no	no
5	41.0	Male	58000.0	North A...	1800.0	2023-08-20	2024-12-01	no	no	no	no	no
6	29.0	Female	48000.0	Asia	1100.0	2023-09-05	2024-09-05	no	no	no	no	no
7	33.0	Male	68666.666...	Oceania	2200.0	2023-10-01	2024-12-10	no	no	no	no	no
8	31.0	Female	70000.0	Europe	1500.0	2023-11-11	2024-11-11	no	no	no	no	no
9	999.0	Male	55000.0	Africa	300.0	2023-0...	Right click (or left+alt) for context menu	es	no	no	no	no
10	35.0	Male	62000.0	Europe	101714.9	2023-06-01	2024-10-10	no	no	no	no	yes
11	62.0	Male	85000.0	Moon	999999.0	2025-12-01	2025-12-01	no	no	no	no	yes

Figure 9: Table showing newly added columns after applying InterquartileRange Filter

7.2. Remove rows with outliers

To remove the outlier, we use `unsupervised.instance.RemoveWithValues` filter and apply the following preferences and repeat for attribute indices Age_Outlier, Income_Outlier, and Spend_Outlier (i.e. 9, 11, 13). Here `splitPoint` is 0.5 because, “No” = 0 and “Yes” = 1, so anything beside “No” will be deleted.

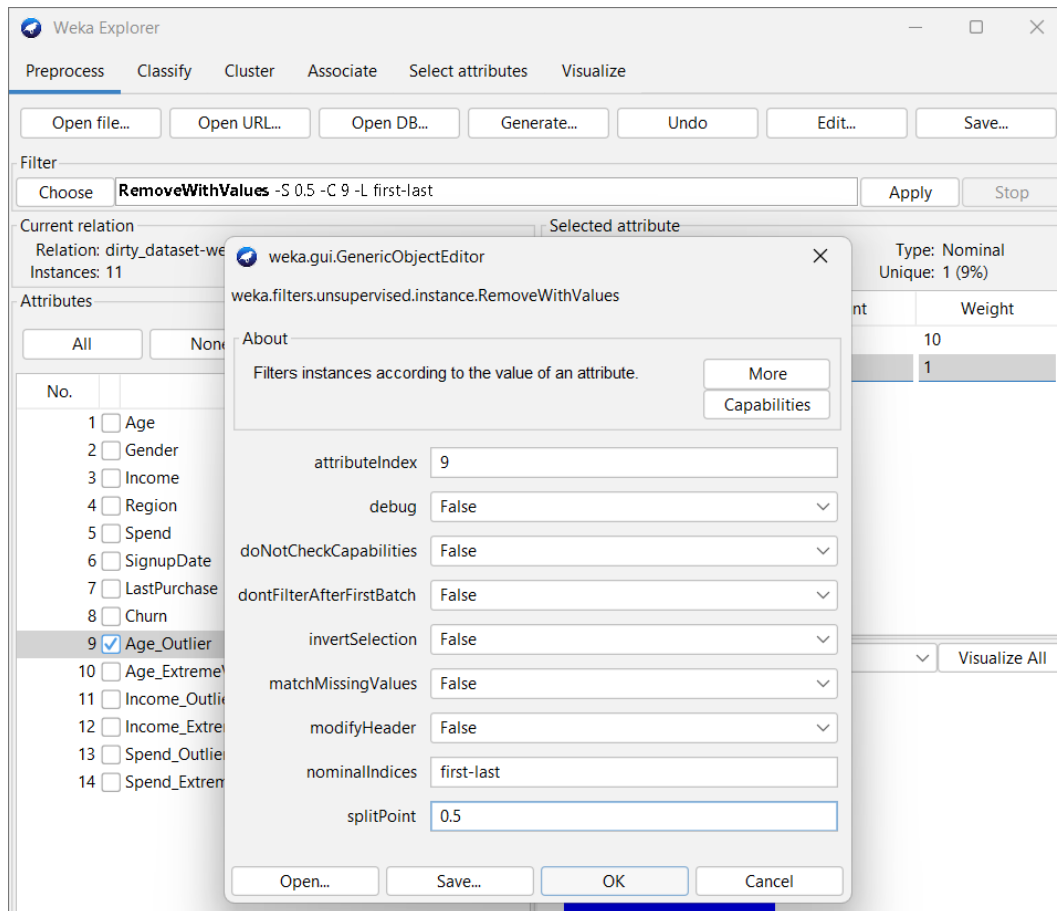


Figure 10: Removing row with outliers for column 9

7.3. Remove the columns created

Finally remove the columns from 9 to 14 by unsupervised.attribute.Remove filter.

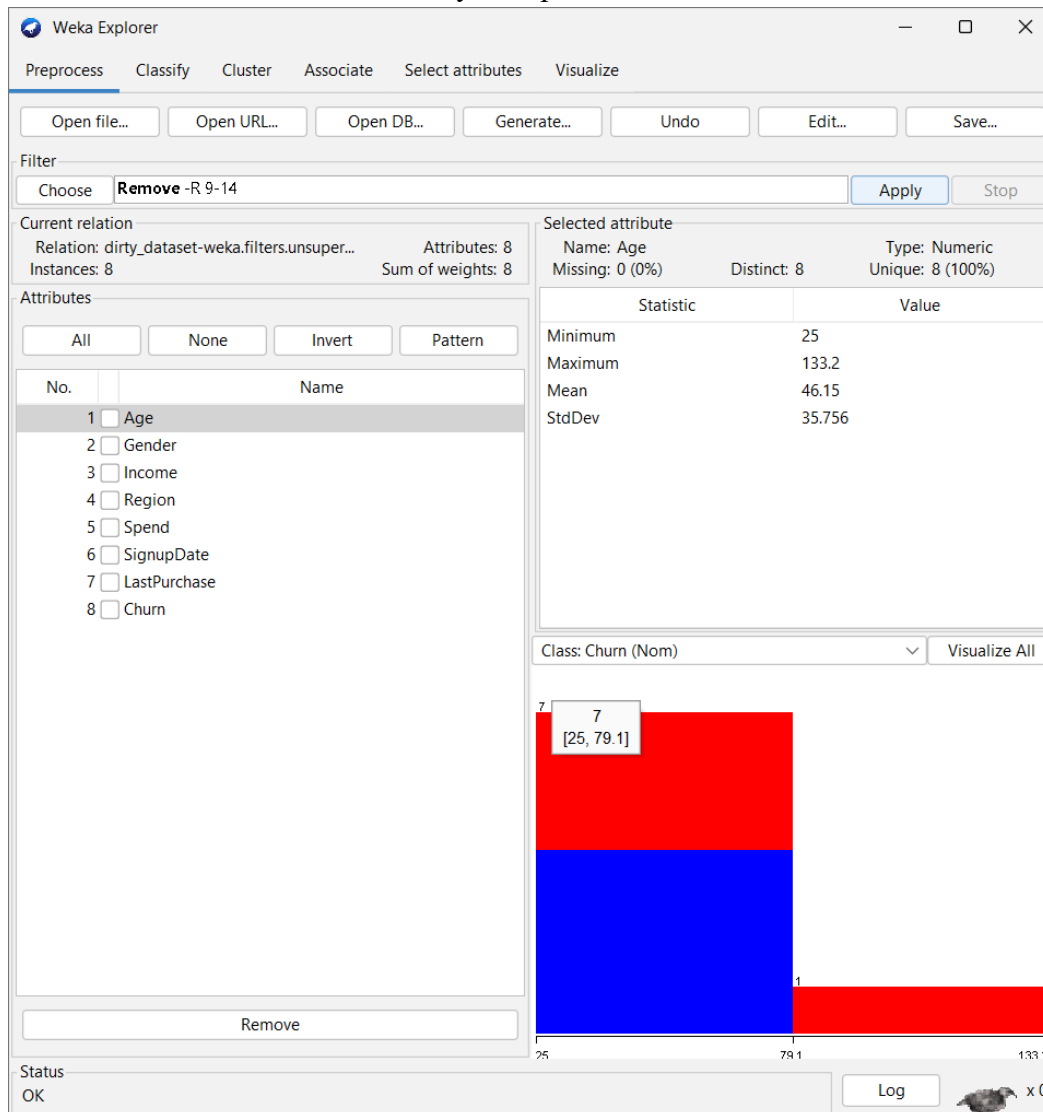


Figure 11: Removing the columns that were added previously after deleting rows with outliers

8. Finalize

Data Cleaning Process is done. Visualize and save the clean data.

Viewer

Relation: dirty_dataset-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.instance.Re

No.	1: Age Numeric	2: Gender Nominal	3: Income Numeric	4: Region Nominal	5: Spend Numeric	6: SignupDate Nominal	7: LastPurchase Nominal	8: Churn Nominal
1	25.0	Male	45000.0	North A...	1200.0	2023-01-15	2024-12-01	Yes
2	133.2	Female	68666.666...	Europe	850.0	2023-02-30	2024-12-01	No
3	45.0	Male	120000.0	Asia	5000.0	2023-03-10	2025-01-15	Yes
4	32.0	F	75000.0	South A...	3200.0	2023-04-05	2024-11-20	No
5	41.0	Male	58000.0	North A...	1800.0	2023-08-20	2024-12-01	No
6	29.0	Female	48000.0	Asia	1100.0	2023-09-05	2024-09-05	Yes
7	33.0	Male	68666.666...	Oceania	2200.0	2023-10-01	2024-12-10	No
8	31.0	Female	70000.0	Europe	1500.0	2023-11-11	2024-11-11	Yes

Figure 12: Final resulting dataset after preprocessing

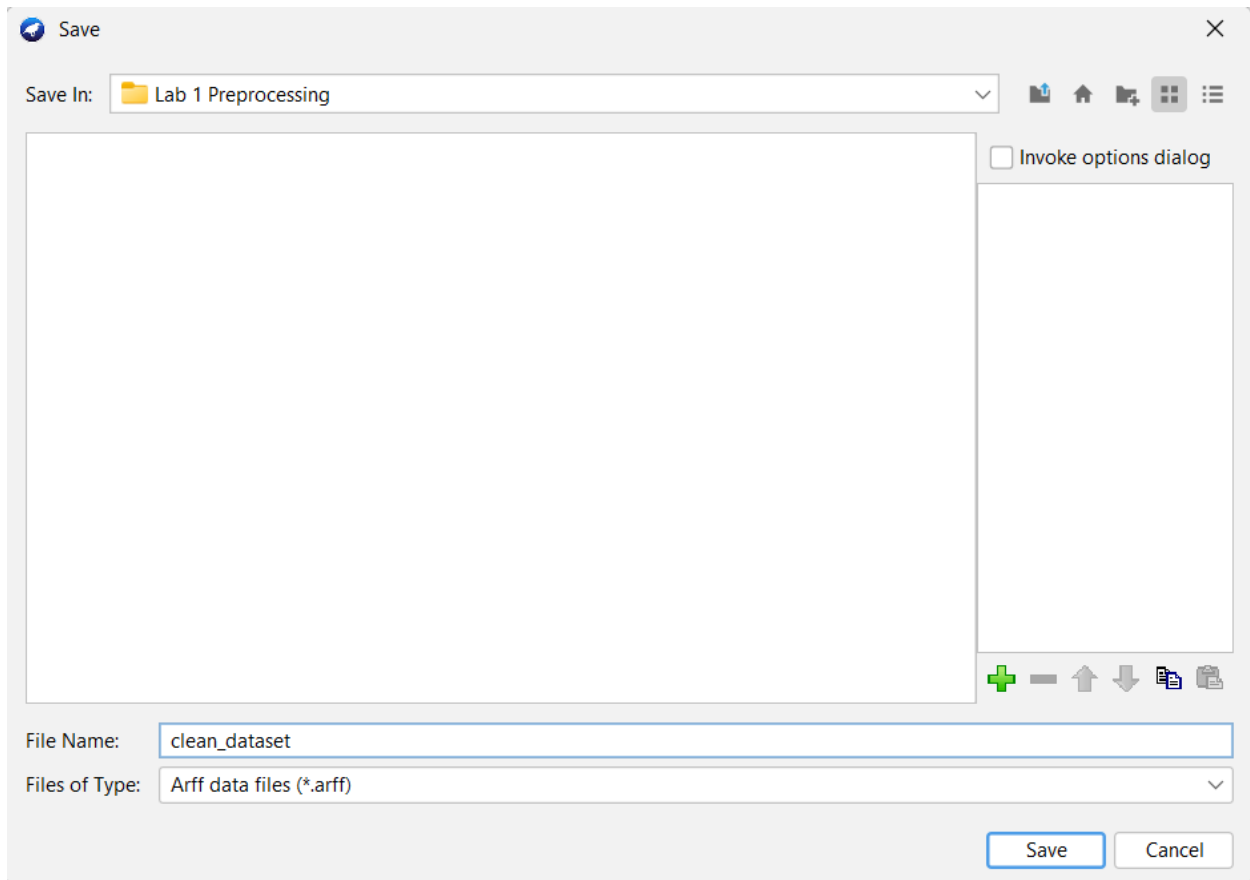


Figure 13: Saving the dataset as clean_dataset in .arff format

B. For Secondary Dataset

For secondary dataset, default data provided by the Weka, labor.arff was selected.

Steps used to clean the data:

1. Open the dataset in the pre-processor of WEKA

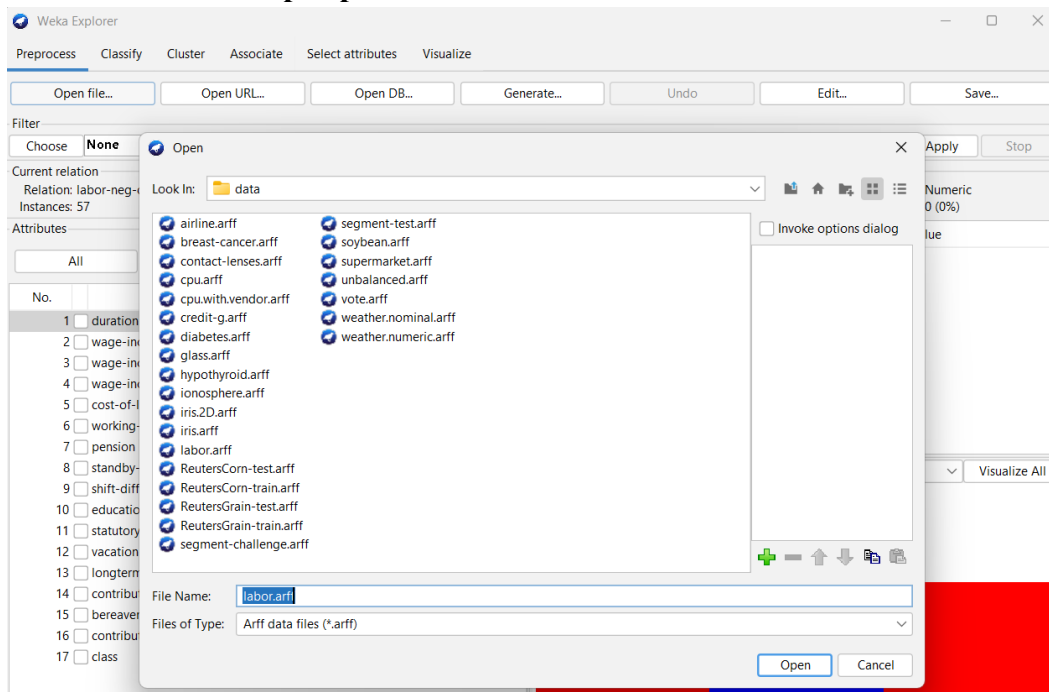


Figure 14: Opening the secondary dataset available in WEKA data

2. Visualize the data

No.	1: duration	2: wage-increase-first-year	3: wage-increase-second-year	4: wage-increase-third-year	5: cost-of-living-adjustment	6: working-hours	7: pension	8: standby-pay
35	3.0	2.0	2.5	2.1	tc	40.0	none	2.0
36	2.0	2.0	2.0	2.0	none	40.0	none	2.0
37	1.0	2.0			tc	40.0	ret_allw	4.0
38	1.0	2.8			none	38.0	empl_contr	2.0
39	3.0	2.0	2.5	2.0		37.0	empl_contr	2.0
40	2.0	4.5	4.0		none	40.0		2.0
41	1.0	4.0			none		none	2.0
42	2.0	2.0	3.0		none	38.0	empl_contr	2.0
43	2.0	2.5			tc	39.0	empl_contr	2.0
44	2.0	2.5	3.0		tcf	40.0	none	2.0
45	2.0	4.0	4.0		none	40.0	none	2.0
46	2.0	4.5	4.0			40.0		2.0
47	2.0	4.5	4.0		none	40.0		2.0
48	2.0	4.6	4.6		tcf	38.0		2.0
49	2.0	5.0	4.5		none	38.0		14.0
50	2.0	5.7	4.5		none	40.0	ret_allw	2.0
51	2.0	7.0	5.3					2.0
52	3.0	2.0	3.0		tcf		empl_contr	2.0
53	3.0	3.5	4.0	4.5	tcf	35.0		2.0
54	3.0	4.0	3.5		none	40.0	empl_contr	2.0
55	3.0	5.0	4.4		none	38.0	empl_contr	10.0
56	3.0	5.0	5.0	5.0		40.0		2.0
57	3.0	6.0	6.0	4.0		35.0		2.0

Figure 15: Visualizing the data

3. Remove unwanted columns

This step was not necessary as all columns were needed.

4. Remove any duplicate values

In this case this was done using the `unsupervised.instance.RemoveDuplicates` filter.

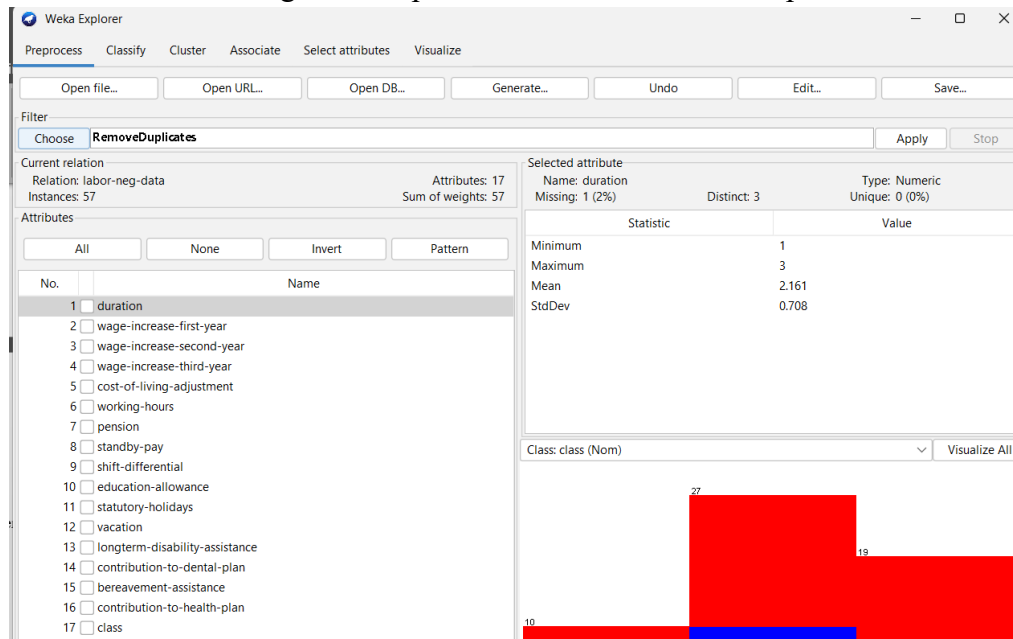


Figure 16: Applying the RemoveDuplicate filter

5. Replace any missing values

In this case it was done using the `unsupervised.attribute.ReplaceMissingValues` filter.

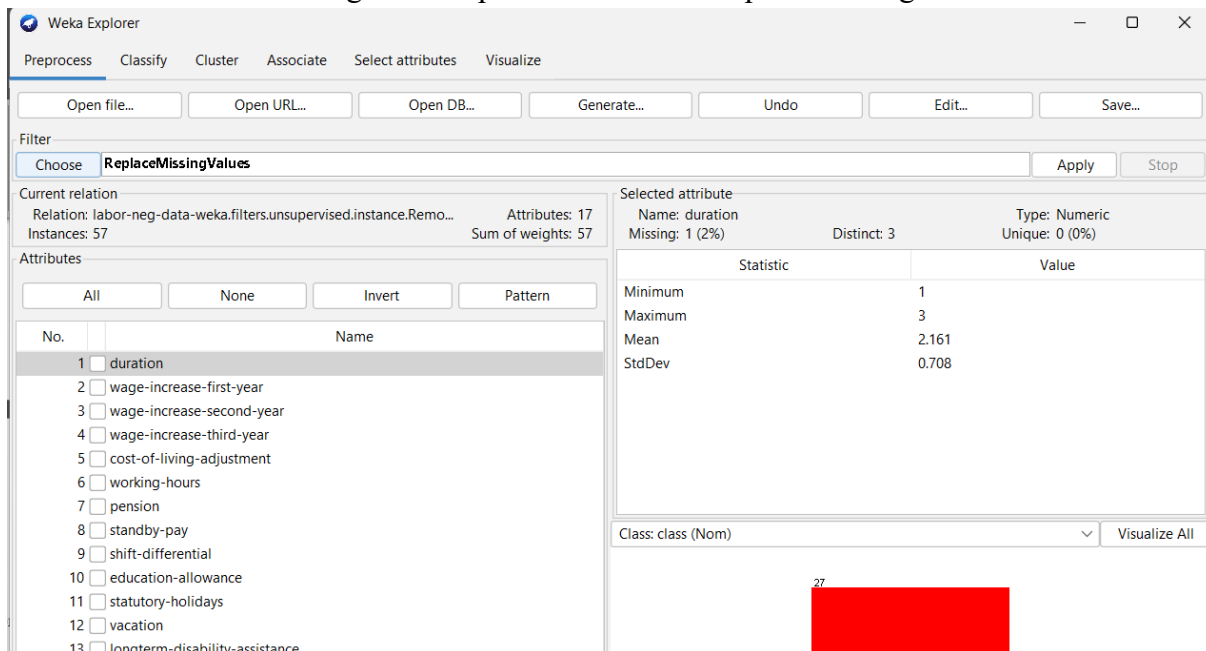


Figure 17: Applying the ReplaceMissingValues filter

6. Convert string into nominal values

This is done using the `unsupervised.attribute.StringToNominal` filter.

The screenshot shows the Weka Explorer interface. The 'Filter' tab is active, and the 'StringToNominal' filter is selected, with the option '-R first-last' chosen. The 'Current relation' is 'labor-neg-data-weka.filters.unsupervised.instance.Remo...', with 17 attributes and 57 instances. The 'Attributes' list on the left shows 17 attributes, with 'duration' selected. The 'Selected attribute' panel on the right shows statistics for 'duration': Minimum 1, Maximum 3, Mean 2.161, and StdDev 0.701. The 'Class' is 'class (Nom)'. A visualization of the data is shown at the bottom, with a red bar for 'duration' and a blue bar for 'class'.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter: Choose **StringToNominal** -R first-last [Apply] [Stop]

Current relation: labor-neg-data-weka.filters.unsupervised.instance.Remo... Attributes: 17 Instances: 57 Sum of weights: 57

Attributes: All | None | Invert | Pattern

No.	Name
1	<input checked="" type="checkbox"/> duration
2	<input type="checkbox"/> wage-increase-first-year
3	<input type="checkbox"/> wage-increase-second-year
4	<input type="checkbox"/> wage-increase-third-year
5	<input type="checkbox"/> cost-of-living-adjustment
6	<input type="checkbox"/> working-hours
7	<input type="checkbox"/> pension
8	<input type="checkbox"/> standby-pay
9	<input type="checkbox"/> shift-differential
10	<input type="checkbox"/> education-allowance
11	<input type="checkbox"/> statutory-holidays
12	<input type="checkbox"/> vacation
13	<input type="checkbox"/> longterm-disability-assistance
14	<input type="checkbox"/> contribution-to-dental-plan
15	<input type="checkbox"/> bereavement-assistance
16	<input type="checkbox"/> contribution-to-health-plan
17	<input type="checkbox"/> class

Selected attribute: Name: duration Missing: 0 (0%) Distinct: 4 Type: Numeric Unique: 1 (2%)

Statistic	Value
Minimum	1
Maximum	3
Mean	2.161
StdDev	0.701

Class: class (Nom) [Visualize All]

Visualization: A bar chart showing the distribution of the 'duration' attribute. The x-axis represents the duration values (1, 2, 3) and the y-axis represents the frequency. The bar for duration=1 is red and has a height of 28. The bar for duration=2 is red and has a height of 19. The bar for duration=3 is blue and has a height of 10.

Figure 18: Conversion of String to Nominal Values

7. Removing Outliers

To remove Outliers, we perform the following steps:

7.1. Interquartile Range

Choose Interquartile Range filter from unsupervised.attribute.InterquartileRange and select the following settings. This will give outlier and extreme values in the dataset.

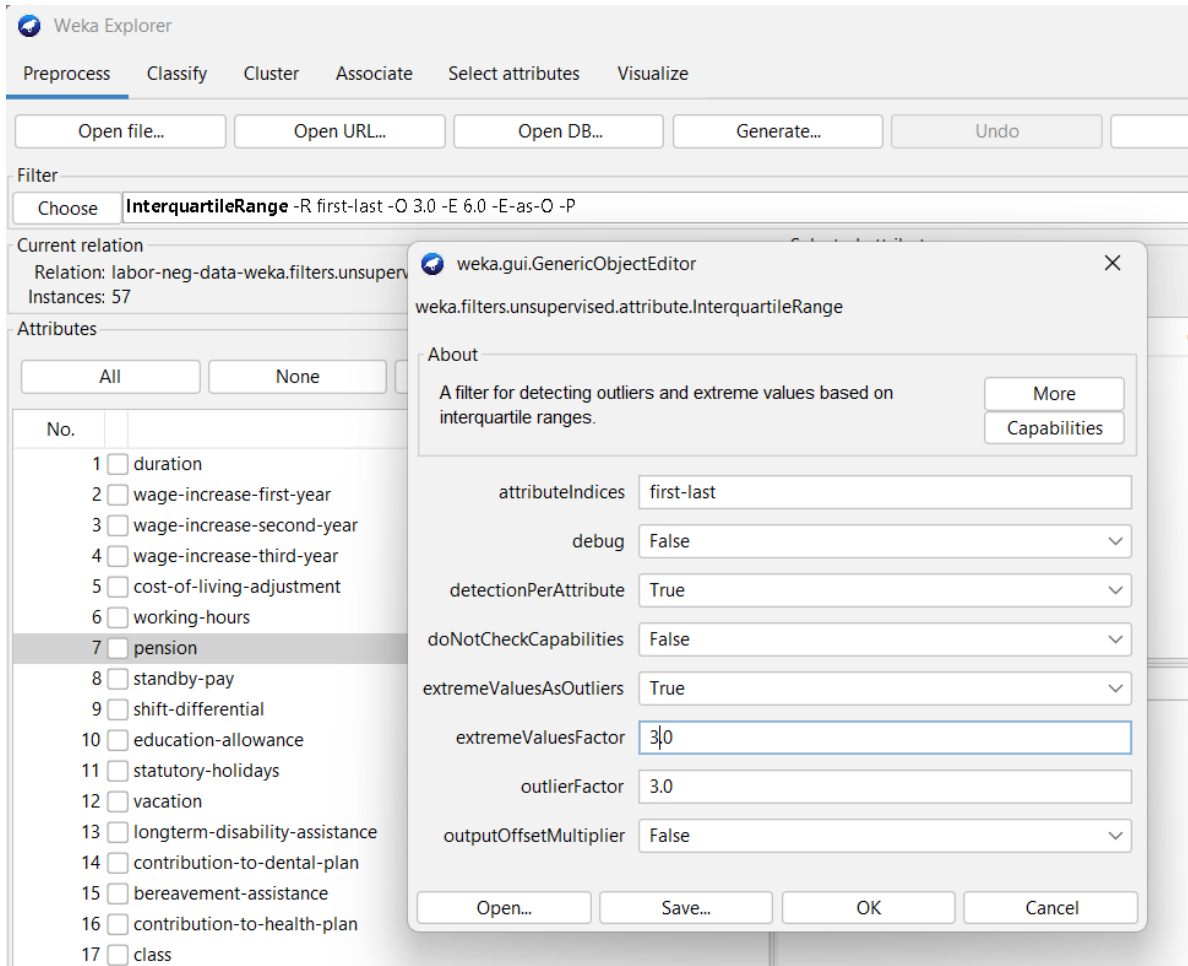


Figure 19: Selecting appropriate setting for InterquartileRange

7.2. Remove rows with outliers

To remove the rows with outliers, we use `unsupervised.instance.RemoveWithValues` filter and apply the following preferences and repeat for attribute indices 18 to 32. Here, `splitPoint` is 0.5 because, “No” = 0 and “Yes” = 1, so anything besides “No” will be deleted.

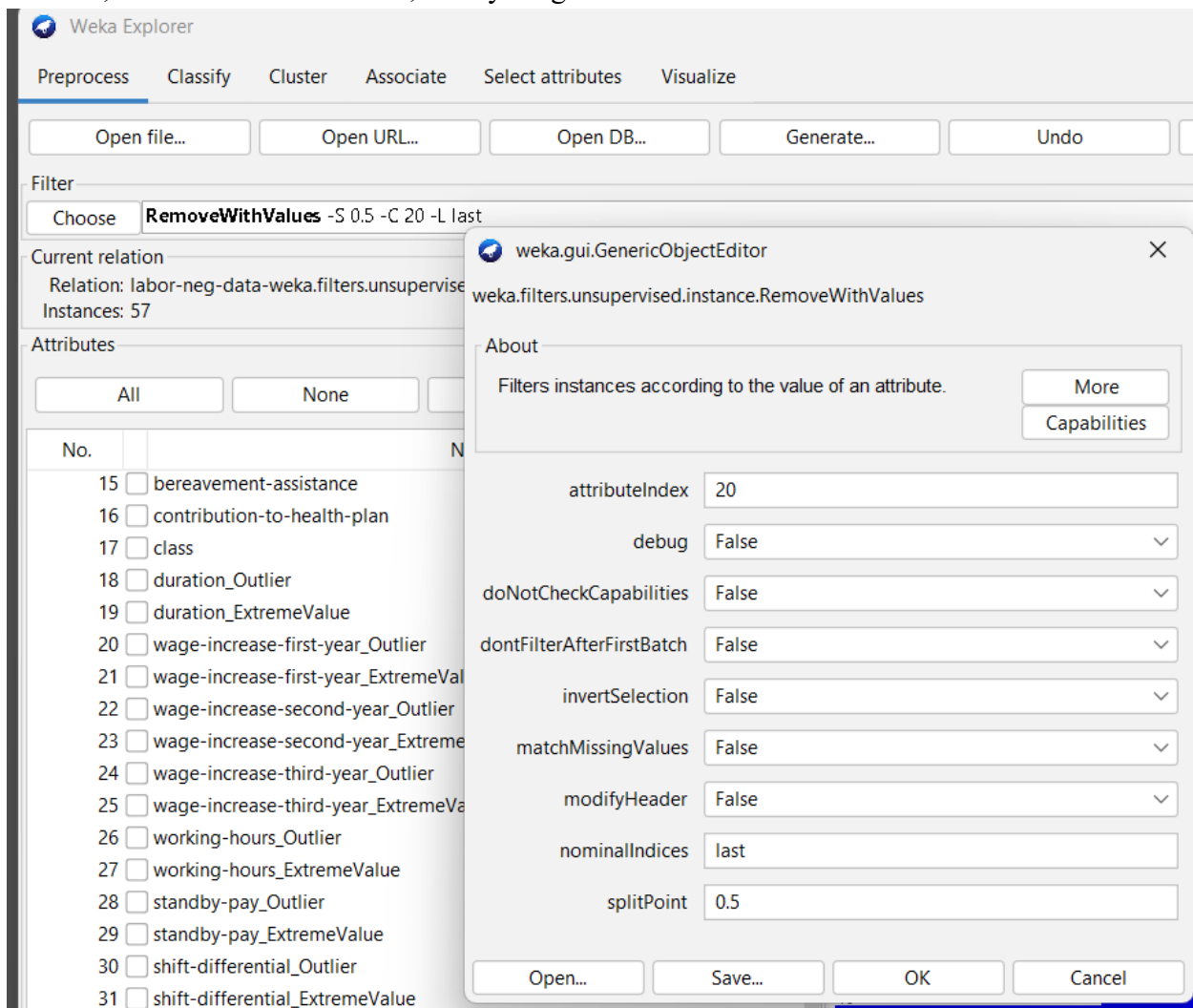


Figure 20: Removing rows containing outliers

7.3. Remove the columns created

Finally remove the columns from 9 to 14 by unsupervised.attribute.Remove filter.

The screenshot shows the Weka Explorer interface with the 'Preprocess' tab selected. The 'Filter' section shows 'Remove -R 18-33' applied. The 'Current relation' is 'labor-neg-data-weka.filters.unsupervised.instance.Remove...'. The 'Attributes' list shows 17 attributes, with 'duration' selected. The 'Selected attribute' panel shows statistics for 'duration': Minimum 1, Maximum 3, Mean 1.944, StdDev 0.498. The 'Class: class (Nom)' is selected, and a bar chart visualization shows the distribution of the class across the attribute values.

Statistic	Value
Minimum	1
Maximum	3
Mean	1.944
StdDev	0.498

Class	Value
1	5
2	24
3	3

Figure 21: Removal of previously created unwanted columns

8. Finalize

Data Cleaning Process is done. Visualize and save the clean data.

No.	1: duration Numeric	2: wage-increase-first-year Numeric	3: wage-increase-second-year Numeric	4: wage-increase-third-year Numeric	5: cost-of-living-adjustment Nominal	6: working-hours Numeric	7: pension Nominal	8: standby-charge Numeric
1	1.0		5.0	3.971739	3.913333 none		40.0 empl_contr	7.44
2	2.0		4.5	5.8	3.913333 none		35.0 ret_allw	7.44
3	2.160714		3.803571	3.971739	3.913333 none		38.0 empl_contr	7.44
4	2.0		2.0	2.5	3.913333 none		35.0 empl_contr	7.44
5	1.0		5.7	3.971739	3.913333 none		40.0 empl_contr	7.44
6	2.0		6.4	6.4	3.913333 none		38.0 empl_contr	7.44
7	2.0		3.5	4.0	3.913333 none		40.0 empl_contr	7.44
8	2.0		4.5	4.0	3.913333 none		37.0 empl_contr	7.44
9	1.0		2.8	3.971739	3.913333 none		35.0 empl_contr	7.44
10	1.0		2.0	3.971739	3.913333 none		38.0 none	7.44
11	2.0		4.3	4.4	3.913333 none		38.0 empl_contr	7.44
12	2.0		2.5	3.0	3.913333 none		40.0 none	7.44
13	2.0		4.5	4.0	3.913333 none		40.0 empl_contr	7.44
14	2.0		4.5	4.5	3.913333 tcf	38.039216	empl_contr	7.44
15	2.0		3.0	3.0	3.913333 none		33.0 empl_contr	7.44
16	2.0		5.0	4.0	3.913333 none		37.0 empl_contr	7.44
17	3.0		2.0	2.5	3.913333 none		35.0 none	7.44
18	2.0		2.5	2.5	3.913333 none		38.0 empl_contr	7.44
19	2.0		4.0	5.0	3.913333 none		40.0 none	7.44
20	2.0		2.0	2.0	3.913333 none		40.0 none	7.44
21	2.0		4.5	4.0	3.913333 none		40.0 empl_contr	7.44
22	1.0		4.0	3.971739	3.913333 none	38.039216	none	7.44
23	2.0		2.0	3.0	3.913333 none		38.0 empl_contr	7.44
24	2.0		2.5	2.5	3.913333 none		38.0 empl_contr	7.44

Figure 22: Final view of the data

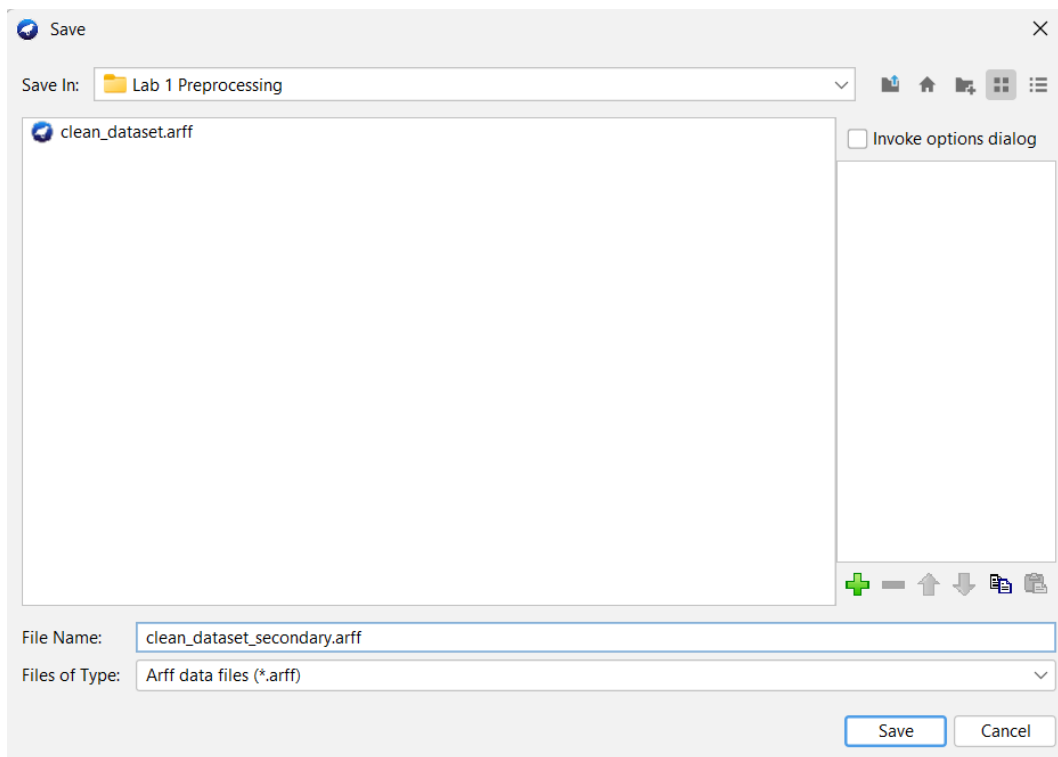


Figure 23: Saving the file as clean dataset in .arff format

Discussion

The preprocessing performed in this lab showed how essential data cleaning is before any data mining or analytical task. The primary dataset contained several typical forms of dirty data such as missing values, duplicated entries, string attributes, and outliers, which required multiple WEKA filters to handle. The removal of unnecessary attributes (like CustomerID) helped reduce noise, while ReplaceMissingValues ensured completeness. Converting strings to nominal values was necessary to prepare the dataset for algorithms that only accept categorical attributes.

Outliers were an important focus of discussion, as WEKA's InterquartileRange filter first tagged them and subsequent RemoveWithValues operations eliminated them. This illustrated how outliers can be systematically detected rather than manually assumed. The secondary dataset required fewer cleaning steps, which highlighted the difference between raw operational data and structured warehouse-ready data. The process demonstrated that preprocessing is often the most time-consuming but impactful stage of data preparation.

Conclusion

This lab successfully demonstrated the full workflow of data preprocessing using WEKA. The primary dataset required extensive cleaning, while the secondary dataset required minimal adjustments, confirming the intended differences between the two. After applying filters for missing values, duplicates, inconsistent formats, and outliers, both datasets were transformed into clean, analysis-ready .arff files. The lab reinforced the importance of preprocessing for accuracy, consistency, and meaningful downstream analysis.