Profiling for Authentication and Authorization

Sai Siva Saketh Kantimahanthi (01099378)
Likhitha S Madenahalli (01090053)

Abstract

From my perspective Data Mining is a process of understanding the given huge chunks of data by extracting knowledge from the given data and finding similar patterns to understand several behaviors. Although the given data cannot be directly used to find patterns and extract information, we must alter the given raw data through some procedures like pre-processing, modeling, clustering, post-processing and then visualizing that data to understand the hidden patterns. Using Association and Clustering techniques we prove that the data we have obtained is having some relevance and to prove that data mining works for the procedure.

This report presents the data of 19 users from a department which has login information and file access information. The data is given in excel format and there are close to 1000 records in it. We had to preprocess the data given in excel filtering out the unnecessary data that was given according to the requirement we want i.e., we only considered the data which is related to user login pattern and then tried to find some patterns manually. We have also used OpenRefine to check the patterns that we found are correct. We have used OpenRefine since it works better with data like spreadsheet file formats but still, it is more like a database.

We used Notepad++ to create Attribute-Relation file format files from the pre-processed data that we obtained. There are six different cases that we have created according to the requirements and ran the files in Weka tool to crosscheck our manually obtained results. Clustering techniques were used to check if the patterns we found are correct and to check how much our results are correct. Clustering techniques were used to check the relevance of the obtained data. We used SimpleKMeans clustering technique and visualized the data through graphs.

Contents

Page number

List of Tables and Figures

Introduction

We have the data of Login and Access for 19 users from a department. The goal is to build a user profile for all the users of the department based on the given information. The user profile must be based on the login and logoff times, library programs/utilities executed, files that are accessed and file created and the printer usage. Based on the given input parameters a user must have a login pattern from his sessions that the user has logged in and access pattern from the files that were accessed.

Provided information

We were provided with three types of information in the form of an excel sheet. The attributes of the types of data are listed below for each type respectively.

Type 1:

- Type of record
- User
- Machine
- Date
- Login time
- Logout time
- Average number of user processes at any time
- Maximum number of user processes
- Total keyboard characters typed
- CPU use by user processes

Type 2

- Type of record
- User
- Machine
- Date
- Start time
- Program
- Execution Time
- File: R- Read, RW – Read write, W – Write
- Printer

Type 3:

- Type of record
- User
- Machine
- Date
- Start time
- Email Program
- Email Address
- Received and Sent
- Bytes

Irregularities

The given data is initially all mixed up but through OpenRefine, we have initially uploaded the excel sheet to OpenRefine and then applied filters according to the above-mentioned types to get results. The users have a login date, login time and logout times different for different users. Processing such data is harder with users having multiple login information unless we normalize the given data by setting up a common time making sure that multiple records of similar times are falling in the same bucket.

Few users are working on different days of a week in a month in the given data. When we looked at the calendar, most of the dates were found that they are weekdays, and few were weekends. So, we have altered the given data as Weekdays and Weekends. Similarly, most of the number formats that were given in the excel sheet were given in the form of text and they must be converted to numbers. After all such conversions then we filter out the types 2 and 3 to identify the user login patterns.

Resource usage pattern the programs are denoted with different program names starting in LP and UP. We found some patterns with the program access and denoted with different notation with many cases like it. Like program access, the Files are also having different data. The files are also normalized to match several records which can be seen in further explanation. There are six printers as well in printer's column. The count is considered in our case.

Under the third type of data, the email information is given in the data. There is a problem with the with the data. The email which is mentioned in the data is not sure that the email is sent, or the email received. And it is taken into consideration that the bytes are sent and received but I'm unclear on this. The attachments are not to be normalized as they are just 3 values and we considered them as they are.

Login Pattern

For the login pattern, we have considered the Type 1 record data. When looked at the Machine usage per user, few users are strictly using the same machine every day and hence the same machine is taken for their record. But for the users who are using multiple machines were notated as ‘MM’ which is Multiple Machines. The column Dates are modified as Weekdays and Weekends based on the dates that are given.

The Login times for an instance for U01 has always logged in between 8:00:00 to 8:30:00 every day. All such records were considered as 8 in the below Table 1. Similarly, all the other users who have the same records were changed so it is easy to execute. But for the users who are having multiple login information are marked as ‘MLT’ i.e., Multiple Login Times. For Logout time we used the same process where users who logged out around 18:00:00 every day got 18 as unique input and ‘MLOT’ i.e., Multiple Logout Times.

The other tabs Average number of user processes, Maximum user processes, Keyboard characters used, and CPU Usage are taken as their average value in this scenario as we did not consider them into the login patterns.

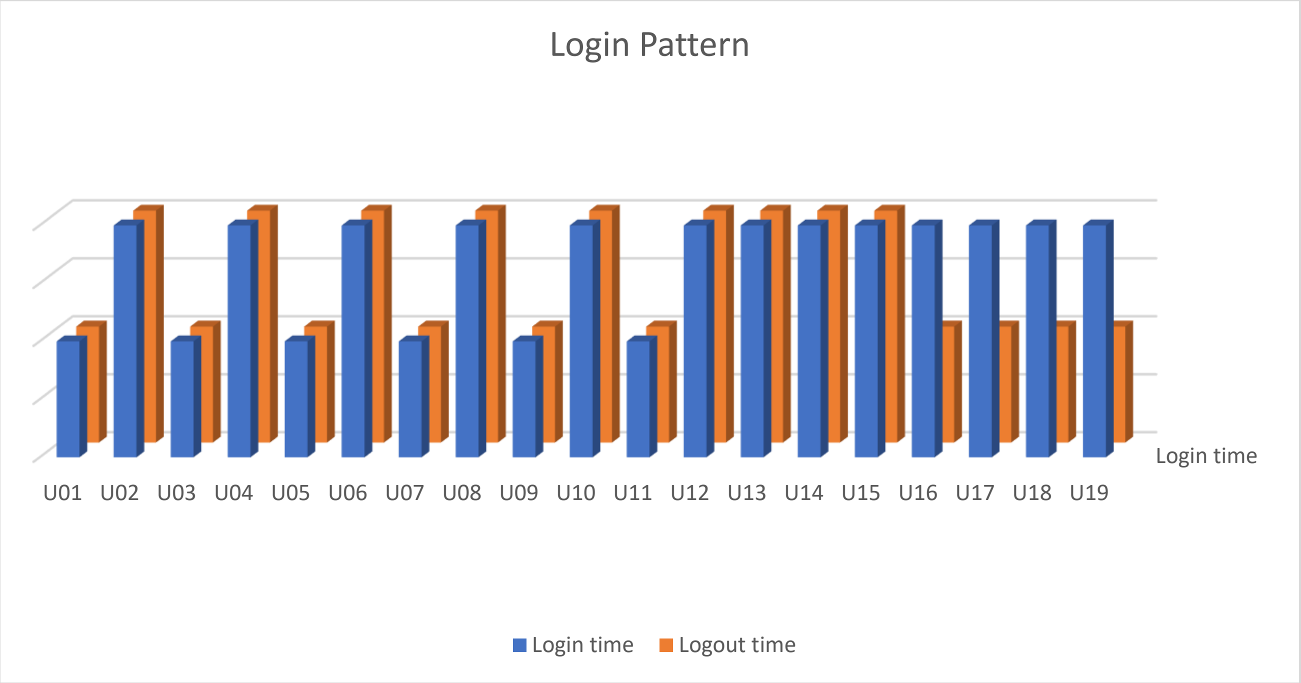
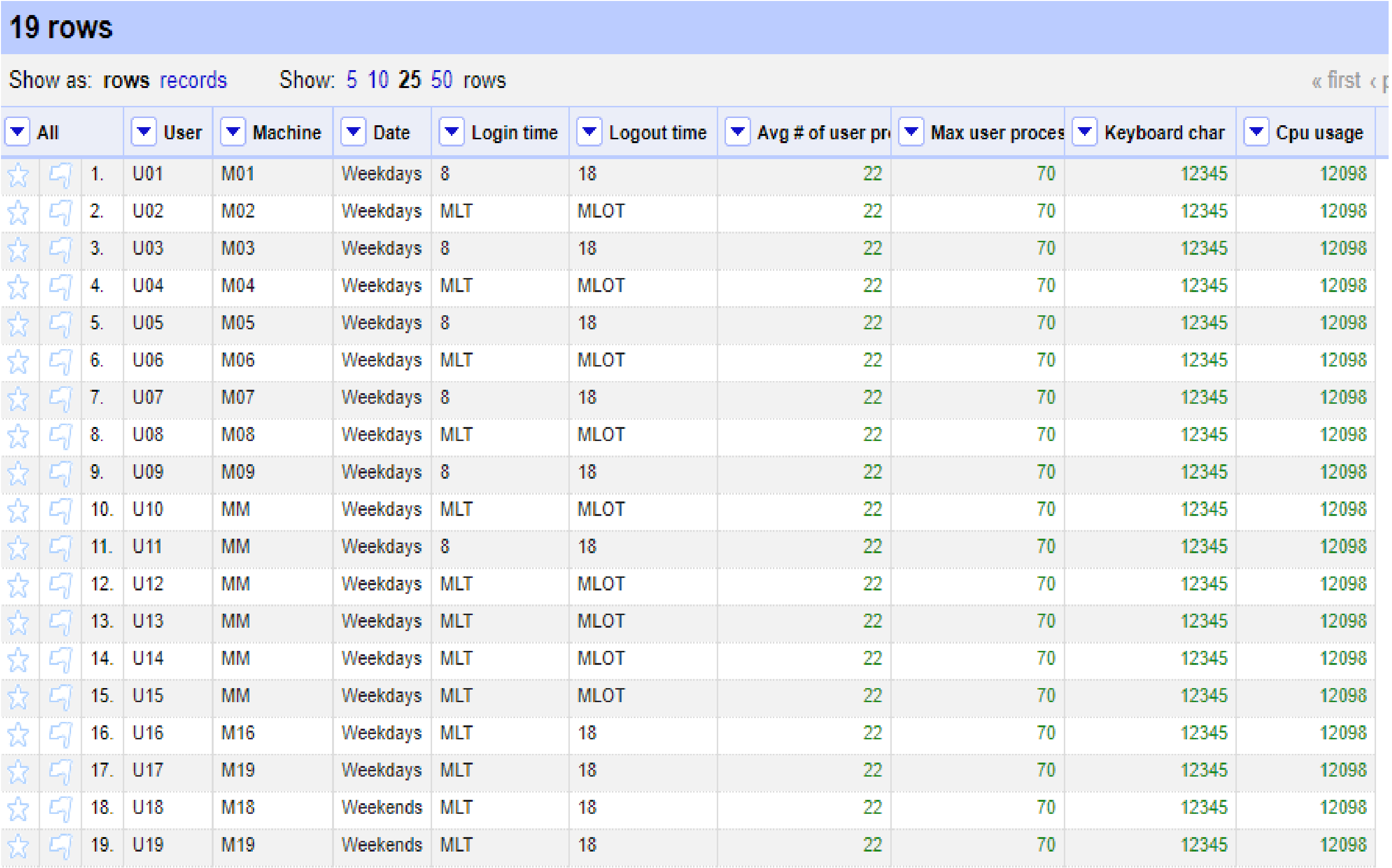


Figure 1



19 rows

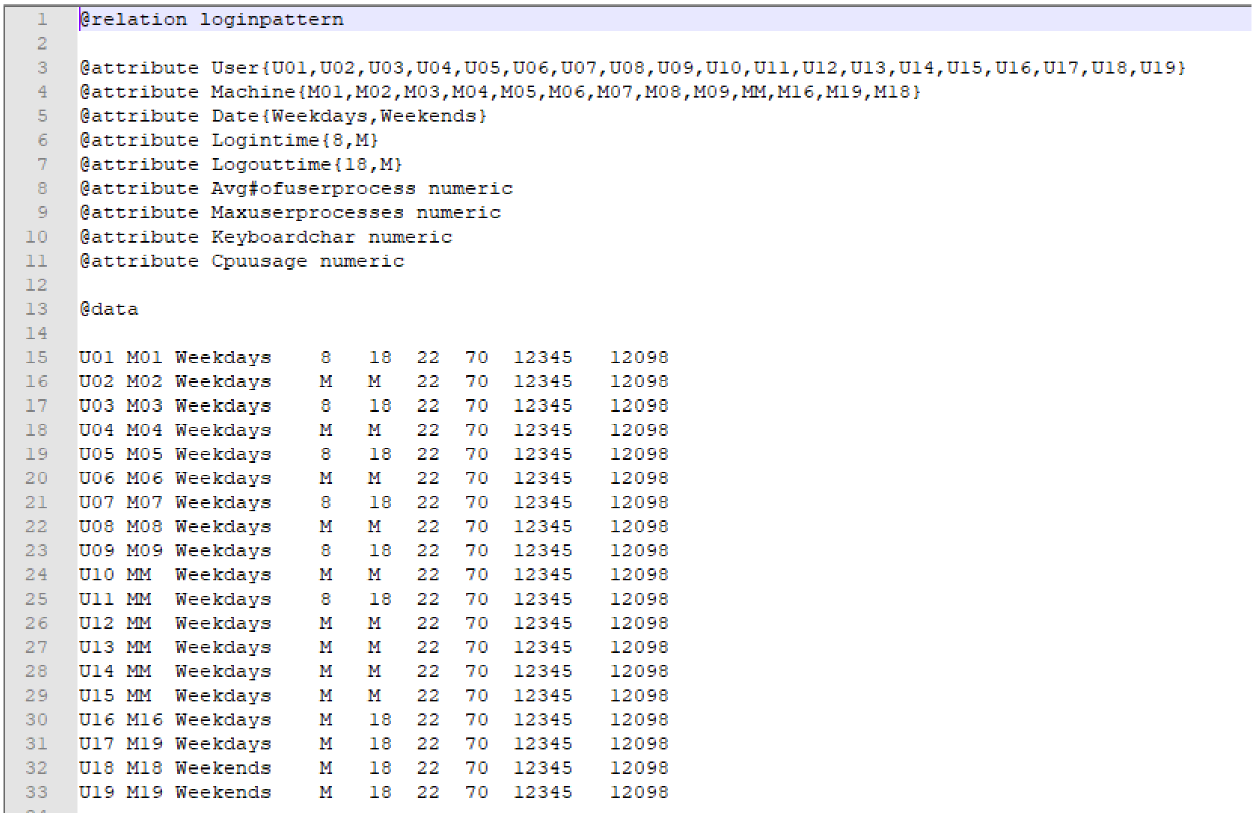
Show as: **rows** records   Show: 5 10 **25** 50 rows   « first ‹ p

▼ All			▼ User	▼ Machine	▼ Date	▼ Login time	▼ Logout time	▼ Avg # of user pr	▼ Max user proces	▼ Keyboard char	▼ Cpu usage
☆	⚐	1.	U01	M01	Weekdays	8	18	22	70	12345	12098
☆	⚐	2.	U02	M02	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	3.	U03	M03	Weekdays	8	18	22	70	12345	12098
☆	⚐	4.	U04	M04	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	5.	U05	M05	Weekdays	8	18	22	70	12345	12098
☆	⚐	6.	U06	M06	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	7.	U07	M07	Weekdays	8	18	22	70	12345	12098
☆	⚐	8.	U08	M08	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	9.	U09	M09	Weekdays	8	18	22	70	12345	12098
☆	⚐	10.	U10	MM	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	11.	U11	MM	Weekdays	8	18	22	70	12345	12098
☆	⚐	12.	U12	MM	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	13.	U13	MM	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	14.	U14	MM	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	15.	U15	MM	Weekdays	MLT	MLOT	22	70	12345	12098
☆	⚐	16.	U16	M16	Weekdays	MLT	18	22	70	12345	12098
☆	⚐	17.	U17	M19	Weekdays	MLT	18	22	70	12345	12098
☆	⚐	18.	U18	M18	Weekends	MLT	18	22	70	12345	12098
☆	⚐	19.	U19	M19	Weekends	MLT	18	22	70	12345	12098

Table 1

To compare the obtained results, we have created an Attribute-Relation file format. Below Figure 2 is the LoginPattern.arff file that we have created to check in the Weka tool.

The data in Figure 2 below is from the Notepad++, that was used to create attribute-relation file format. It is the result of the preprocessing that has been discussed previously. Once the arff file format is ready then we imported the file to Weka for further analysis.

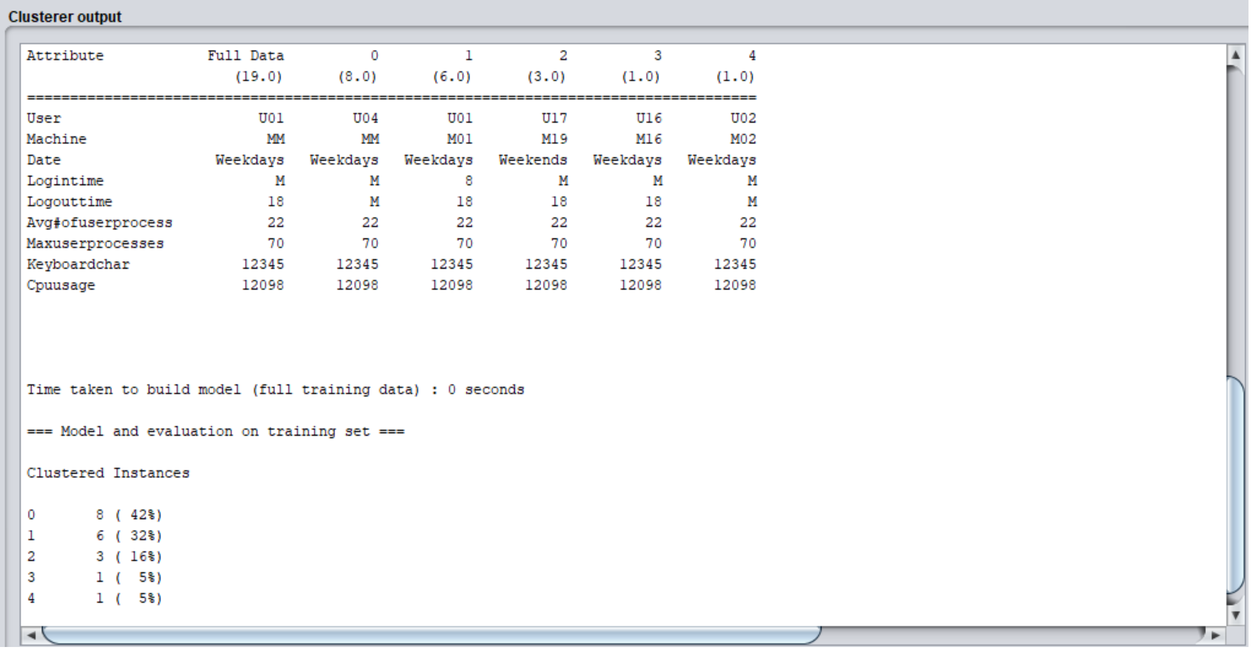


```
@relation loginpattern

@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U15,U16,U17,U18,U19}
@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,MM,M16,M19,M18}
@attribute Date{Weekdays,Weekends}
@attribute Logintime{8,M}
@attribute Logouttime{18,M}
@attribute Avg#ofuserprocess numeric
@attribute Maxuserprocesses numeric
@attribute Keyboardchar numeric
@attribute Cpuusage numeric

@data

U01 M01 Weekdays    8   18   22   70   12345    12098
U02 M02 Weekdays    M   M    22   70   12345    12098
U03 M03 Weekdays    8   18   22   70   12345    12098
U04 M04 Weekdays    M   M    22   70   12345    12098
U05 M05 Weekdays    8   18   22   70   12345    12098
U06 M06 Weekdays    M   M    22   70   12345    12098
U07 M07 Weekdays    8   18   22   70   12345    12098
U08 M08 Weekdays    M   M    22   70   12345    12098
U09 M09 Weekdays    8   18   22   70   12345    12098
U10 MM  Weekdays    M   M    22   70   12345    12098
U11 MM  Weekdays    8   18   22   70   12345    12098
U12 MM  Weekdays    M   M    22   70   12345    12098
U13 MM  Weekdays    M   M    22   70   12345    12098
U14 MM  Weekdays    M   M    22   70   12345    12098
U15 MM  Weekdays    M   M    22   70   12345    12098
U16 M16 Weekdays    M   18   22   70   12345    12098
U17 M19 Weekdays    M   18   22   70   12345    12098
U18 M18 Weekends    M   18   22   70   12345    12098
U19 M19 Weekends    M   18   22   70   12345    12098
```

Figure 2

Once the file is loaded into Weka, we have applied SimpleKMeans clustering technique on the data that has been loaded to check whether the users are having similar pattern during analyzing their login pattern. Figure 3 gives the Cluster output of SimpleKMeans algorithm with 5 clusters as input given. The clustered instances are mentioned in figure 3.

Figure 4 is the output visualization for SimpleKMeans for login pattern. The X-axis has the instance number and users are on the Y-axis. The 5 clusters are denoted with 5 different colors in the plotted graph and are marked accordingly.

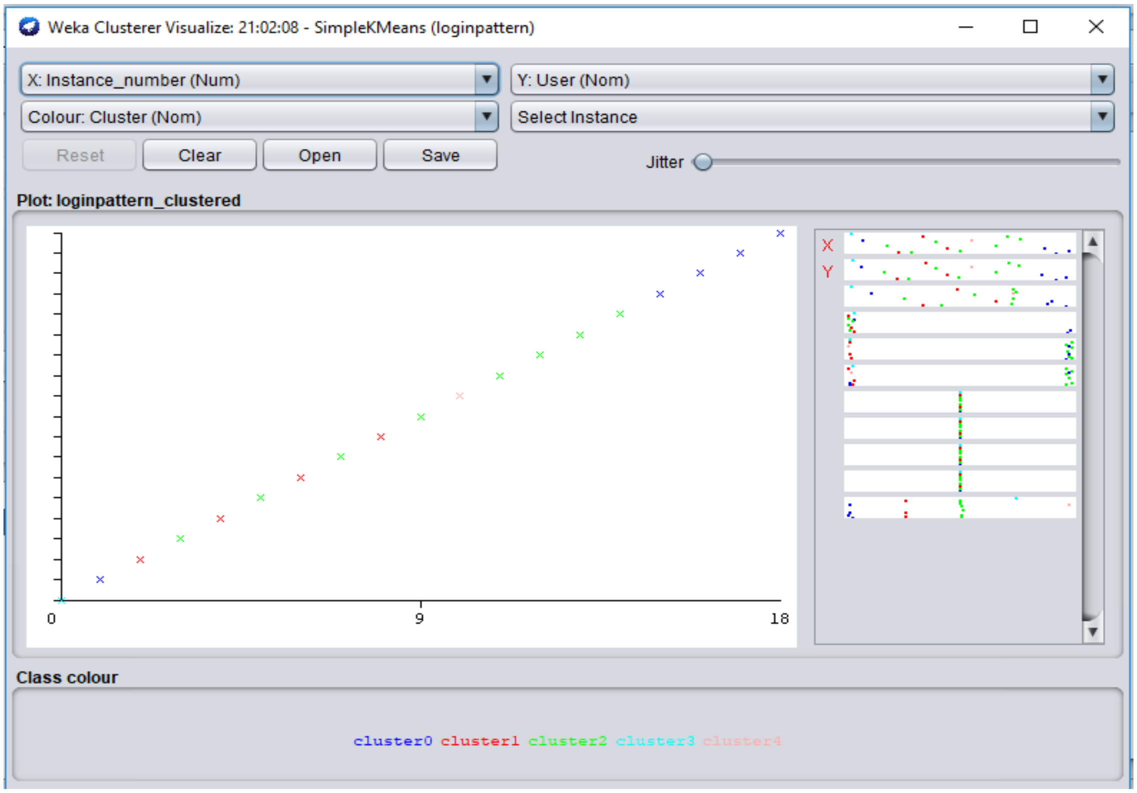


Clusterer output

```
Attribute            Full Data          0          1          2          3          4
                        (19.0)      (8.0)      (6.0)      (3.0)      (1.0)      (1.0)
=====================================================================================
User                       U01        U04        U01        U17        U16        U02
Machine                     MM         MM        M01        M19        M16        M02
Date                  Weekdays   Weekdays   Weekdays   Weekends   Weekdays   Weekdays
Logintime                    M          M          8          M          M          M
Logouttime                  18          M         18         18         18          M
Avg#ofuserprocess           22         22         22         22         22         22
Maxuserprocesses            70         70         70         70         70         70
Keyboardchar             12345      12345      12345      12345      12345      12345
Cpuusage                 12098      12098      12098      12098      12098      12098




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       8 ( 42%)
1       6 ( 32%)
2       3 ( 16%)
3       1 (  5%)
4       1 (  5%)
```

Figure 3

The below figure indicates the clusters that are formed with the respective users in that cluster.



Weka Clusterer Visualize: 21:02:08 - SimpleKMeans (loginpattern)

X: Instance_number (Num)

Y: User (Nom)

Colour: Cluster (Nom)

Select Instance

Plot: loginpattern_clustered

Class colour

cluster0 cluster1 cluster2 cluster3 cluster4

Figure 4

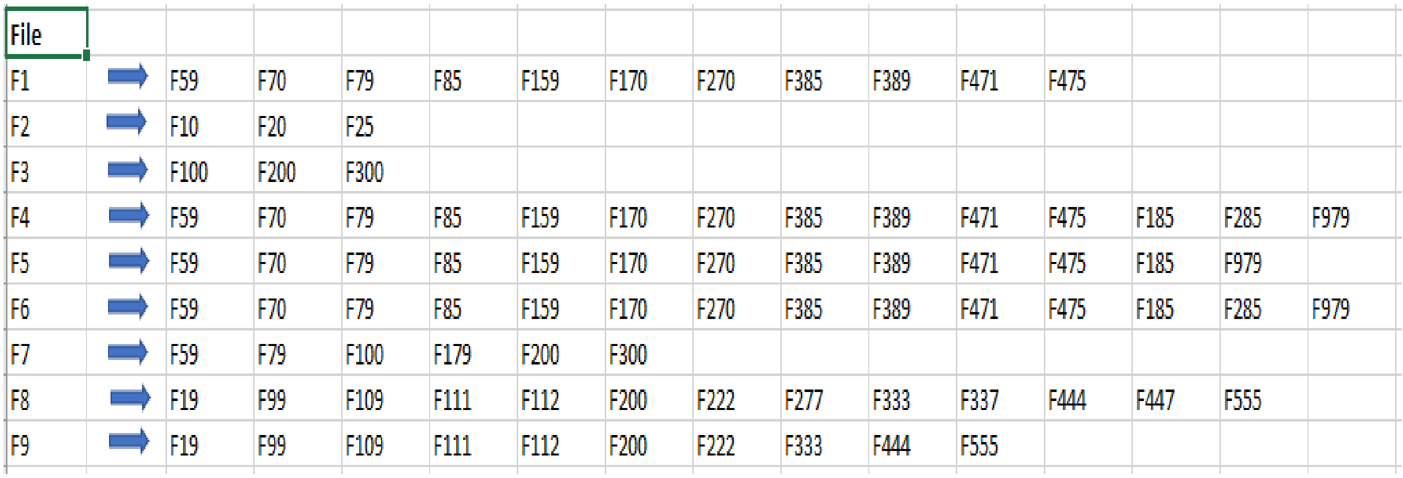
Program Access Pattern

For Program access pattern, type 2 data is considered, and rest of the rows are filtered out. Since there are multiple records for each user, we have analyzed and simplified the individual's columns based on the patterns. For instance, U01 is always accessing Machine M01, so it is taken that U01 will always access M01.

Now looking at the Date, we have considered the dates and found that they are either Weekdays and Weekends. Based on it, the dates are denoted as Weekdays and Weekends. Most of the users are working on weekdays except few who are working on weekends.

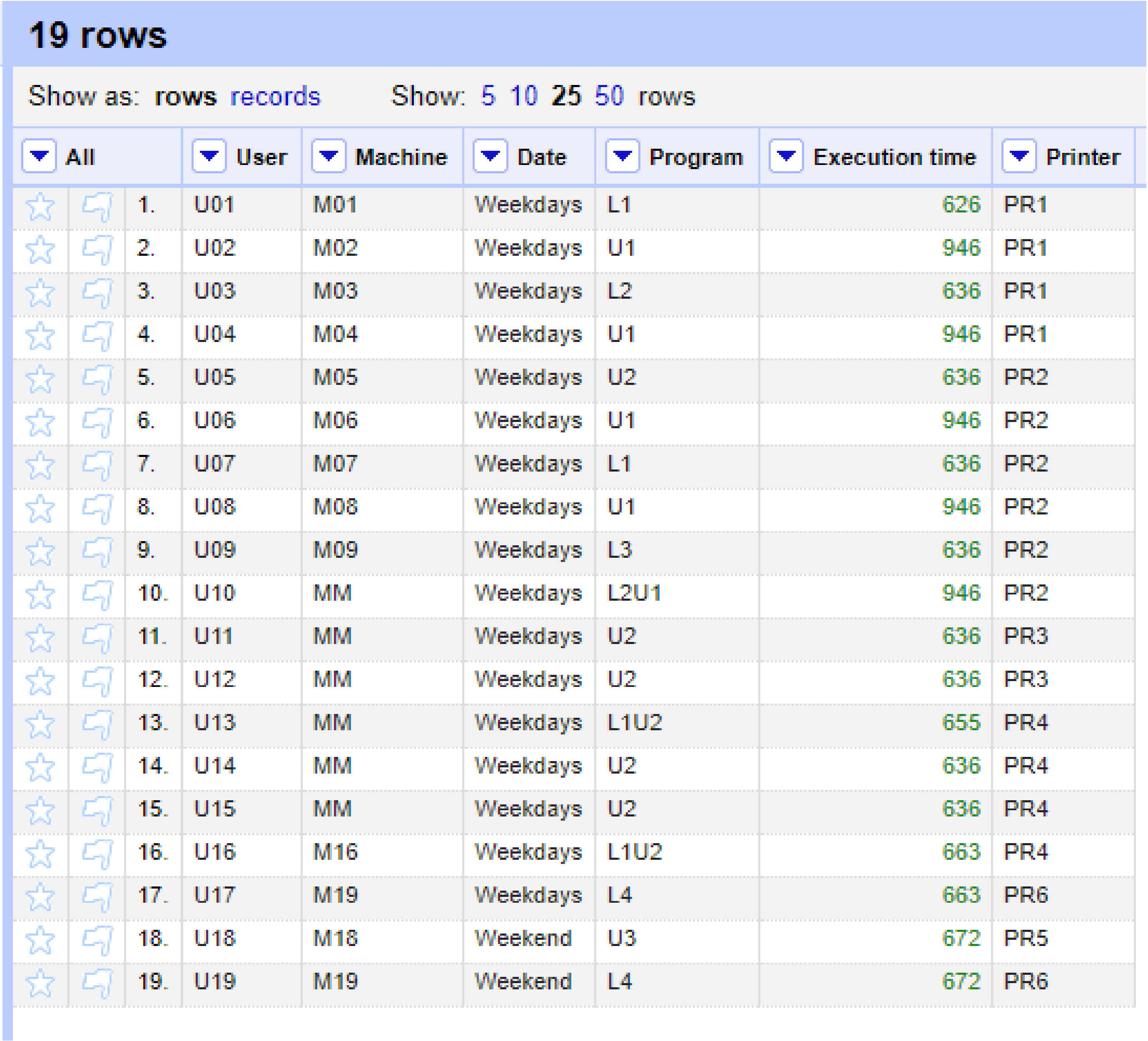
While processing the program id, there are some patterns that are given in the program column. When the programs that are used are filtered according to the user id, there are some patterns where multiple users are using the same set of files, and all such files are marked with different notation. Please find the Table 2 for more information of how the similar files are named.

For the execution time, the average time of execution is considered for all the users. By considering all the information from above mentioned, a table has been prepared with all the modified information. The Table 3 below shows the data.



File															
F1	➡	F59	F70	F79	F85	F159	F170	F270	F385	F389	F471	F475			
F2	➡	F10	F20	F25											
F3	➡	F100	F200	F300											
F4	➡	F59	F70	F79	F85	F159	F170	F270	F385	F389	F471	F475	F185	F285	F979
F5	➡	F59	F70	F79	F85	F159	F170	F270	F385	F389	F471	F475	F185	F979	
F6	➡	F59	F70	F79	F85	F159	F170	F270	F385	F389	F471	F475	F185	F285	F979
F7	➡	F59	F79	F100	F179	F200	F300								
F8	➡	F19	F99	F109	F111	F112	F200	F222	F277	F333	F337	F444	F447	F555	
F9	➡	F19	F99	F109	F111	F112	F200	F222	F333	F444	F555				

Table 2



19 rows

Show as: rows records   Show: 5 10 25 50 rows

All	User	Machine	Date	Program	Execution time	Printer
☆ ⚐ 1.	U01	M01	Weekdays	L1	626	PR1
☆ ⚐ 2.	U02	M02	Weekdays	U1	946	PR1
☆ ⚐ 3.	U03	M03	Weekdays	L2	636	PR1
☆ ⚐ 4.	U04	M04	Weekdays	U1	946	PR1
☆ ⚐ 5.	U05	M05	Weekdays	U2	636	PR2
☆ ⚐ 6.	U06	M06	Weekdays	U1	946	PR2
☆ ⚐ 7.	U07	M07	Weekdays	L1	636	PR2
☆ ⚐ 8.	U08	M08	Weekdays	U1	946	PR2
☆ ⚐ 9.	U09	M09	Weekdays	L3	636	PR2
☆ ⚐ 10.	U10	MM	Weekdays	L2U1	946	PR2
☆ ⚐ 11.	U11	MM	Weekdays	U2	636	PR3
☆ ⚐ 12.	U12	MM	Weekdays	U2	636	PR3
☆ ⚐ 13.	U13	MM	Weekdays	L1U2	655	PR4
☆ ⚐ 14.	U14	MM	Weekdays	U2	636	PR4
☆ ⚐ 15.	U15	MM	Weekdays	U2	636	PR4
☆ ⚐ 16.	U16	M16	Weekdays	L1U2	663	PR4
☆ ⚐ 17.	U17	M19	Weekdays	L4	663	PR6
☆ ⚐ 18.	U18	M18	Weekend	U3	672	PR5
☆ ⚐ 19.	U19	M19	Weekend	L4	672	PR6

Table 3

A graph has been plotted on the data from programs used and their respective execution time. Figure 5 below is that graph plotted between the execution time on the y-axis and the programs on the x-axis.

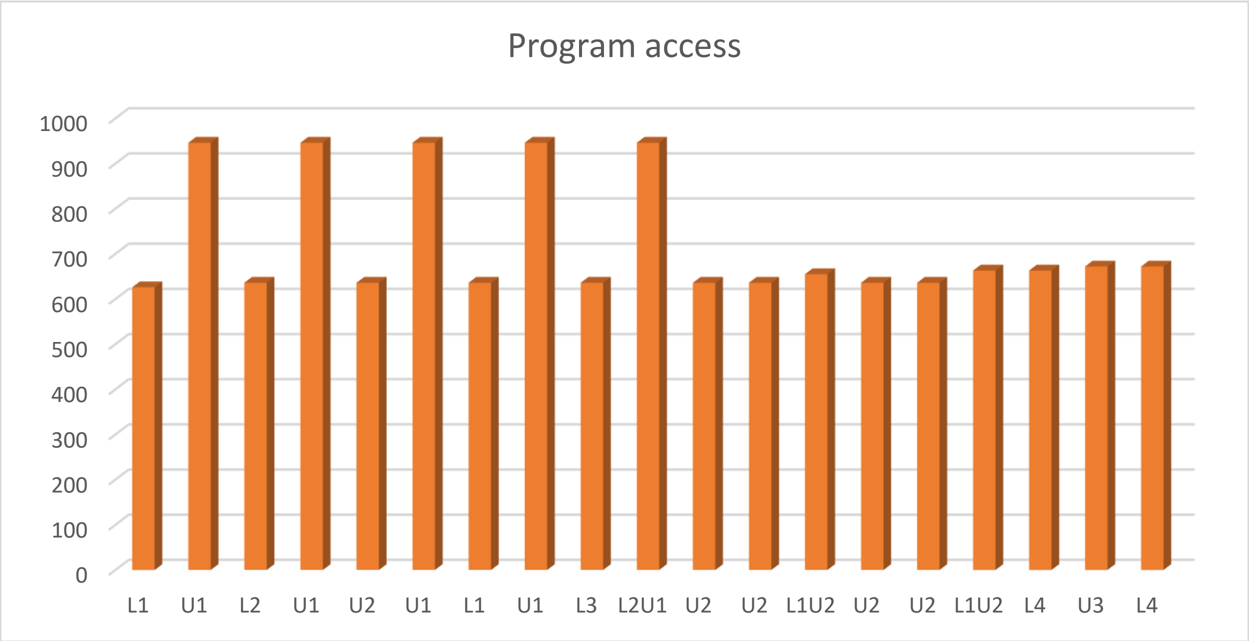
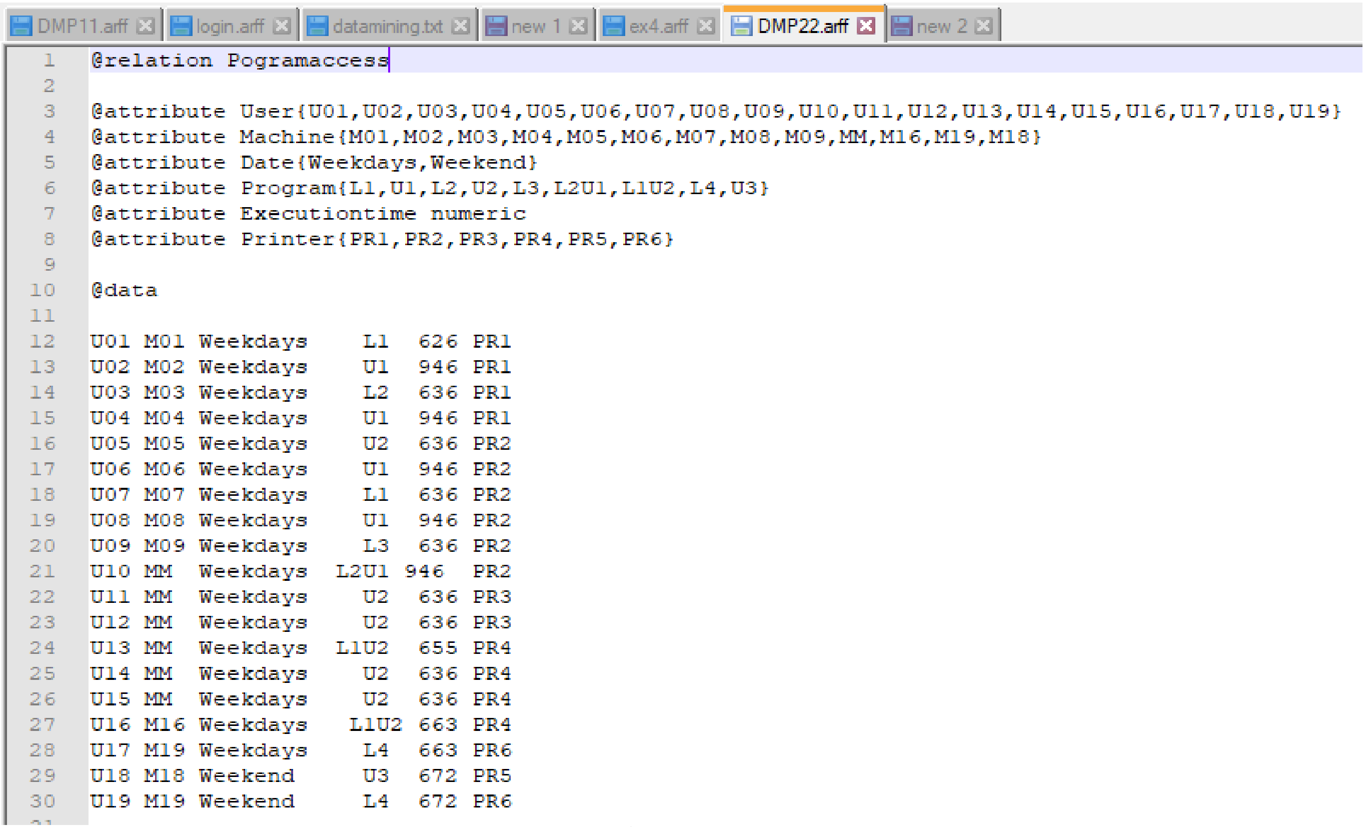


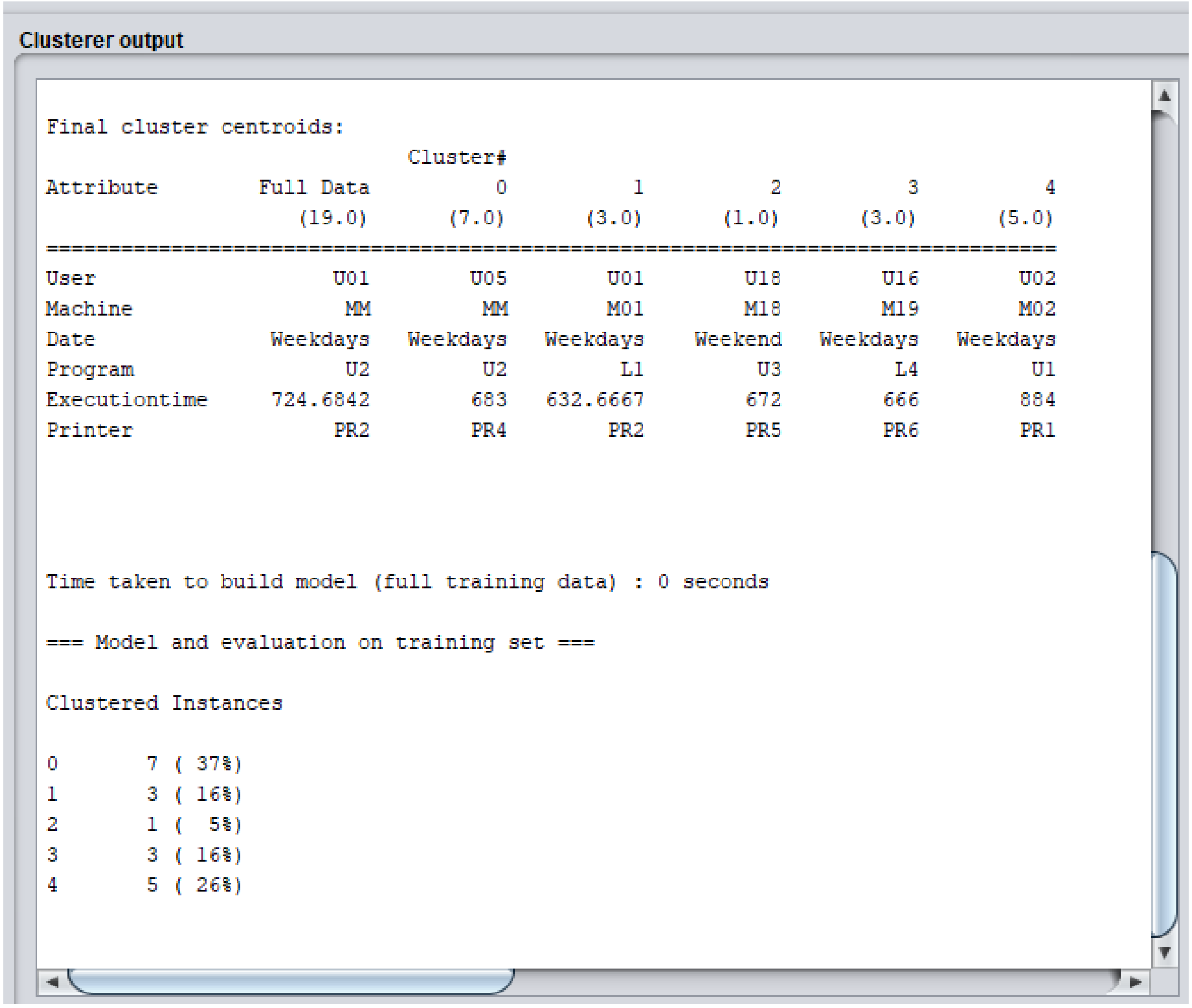
Figure 5



```
@relation Pogramaccess

@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U15,U16,U17,U18,U19}
@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,MM,M16,M19,M18}
@attribute Date{Weekdays,Weekend}
@attribute Program{L1,U1,L2,U2,L3,L2U1,L1U2,L4,U3}
@attribute Executiontime numeric
@attribute Printer{PR1,PR2,PR3,PR4,PR5,PR6}

@data

U01 M01 Weekdays    L1  626 PR1
U02 M02 Weekdays    U1  946 PR1
U03 M03 Weekdays    L2  636 PR1
U04 M04 Weekdays    U1  946 PR1
U05 M05 Weekdays    U2  636 PR2
U06 M06 Weekdays    U1  946 PR2
U07 M07 Weekdays    L1  636 PR2
U08 M08 Weekdays    U1  946 PR2
U09 M09 Weekdays    L3  636 PR2
U10 MM  Weekdays  L2U1 946  PR2
U11 MM  Weekdays    U2  636 PR3
U12 MM  Weekdays    U2  636 PR3
U13 MM  Weekdays  L1U2  655 PR4
U14 MM  Weekdays    U2  636 PR4
U15 MM  Weekdays    U2  636 PR4
U16 M16 Weekdays   L1U2 663 PR4
U17 M19 Weekdays    L4  663 PR6
U18 M18 Weekend     U3  672 PR5
U19 M19 Weekend     L4  672 PR6
```

Figure 6

After preprocessing all the data which is shown in Table 3, the attribute relation file format is created which is shown in the above figure, Figure 6. The data from the Figure 6 is from notepad++ that is converted to Programaccess.arff file and then it is run in Weka tool for further analysis.



```
Final cluster centroids:
                             Cluster#
Attribute        Full Data          0          1          2          3          4
                    (19.0)      (7.0)      (3.0)      (1.0)      (3.0)      (5.0)
=================================================================================
User                   U01        U05        U01        U18        U16        U02
Machine                 MM         MM        M01        M18        M19        M02
Date              Weekdays   Weekdays   Weekdays    Weekend   Weekdays   Weekdays
Program                 U2         U2         L1         U3         L4         U1
Executiontime     724.6842        683   632.6667        672        666        884
Printer                PR2        PR4        PR2        PR5        PR6        PR1




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       7 ( 37%)
1       3 ( 16%)
2       1 (  5%)
3       3 ( 16%)
4       5 ( 26%)
```

Figure 7

Figure 7 is the output that is obtained from applying SimpleKMeans clustering on the Program access pattern file. The output shows the clustered instances and the values of the clusters for the respective attributes.

Figure 8 is the visualization of the SimpleKMeans fir Program access pattern file that has been loaded. The X-axis denotes the instance number and Y-axis denotes the User Id. Five clusters are marked with five different colors. The clusters are marked on the visualization graph clustered as points.

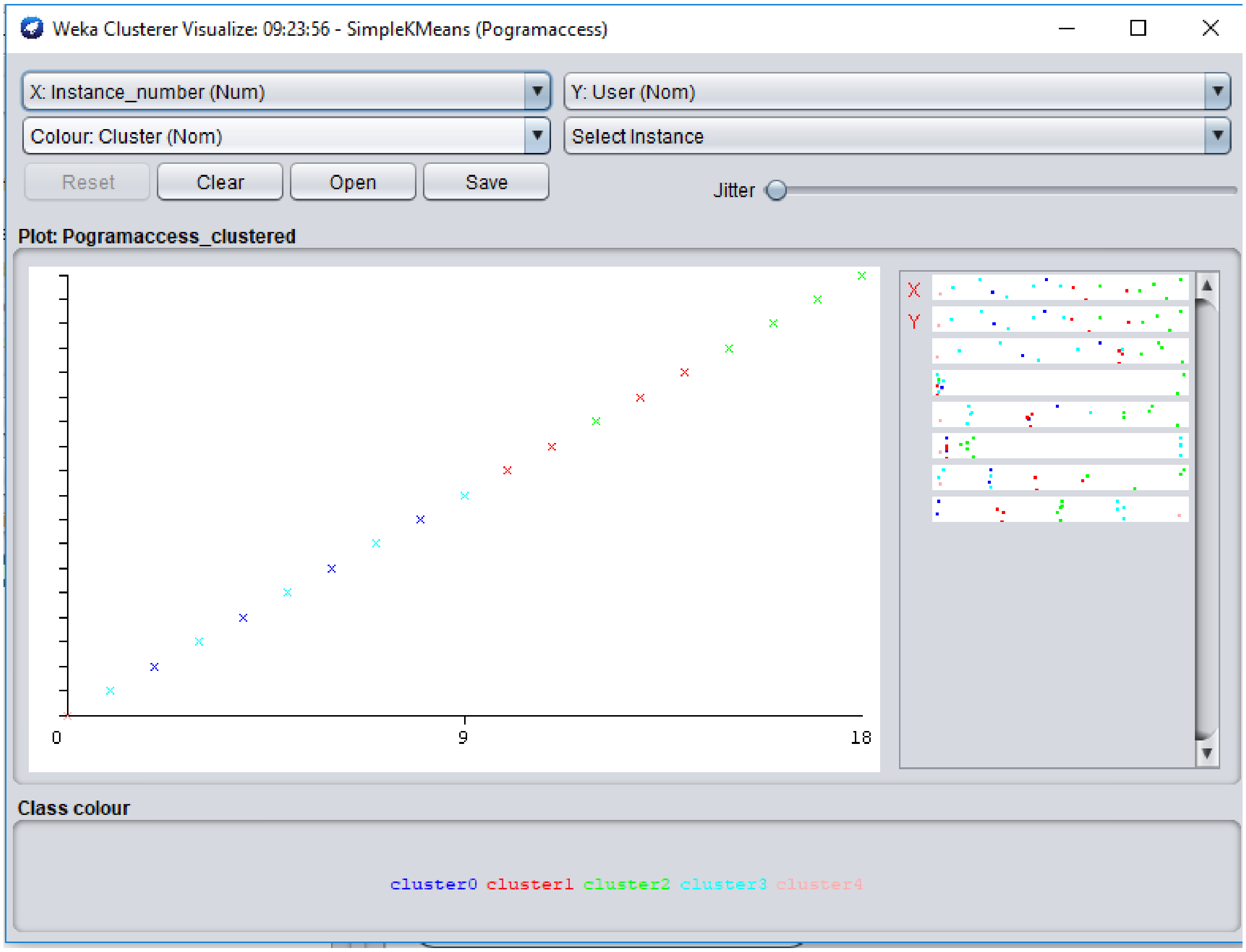

Weka Clusterer Visualize: 09:23:56 - SimpleKMeans (Pogramaccess)
X: Instance_number (Num)
Y: User (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: Pogramaccess_clustered
0
9
18
X
Y
Class colour
cluster0 cluster1 cluster2 cluster3 cluster4

Figure 8

File Access Pattern

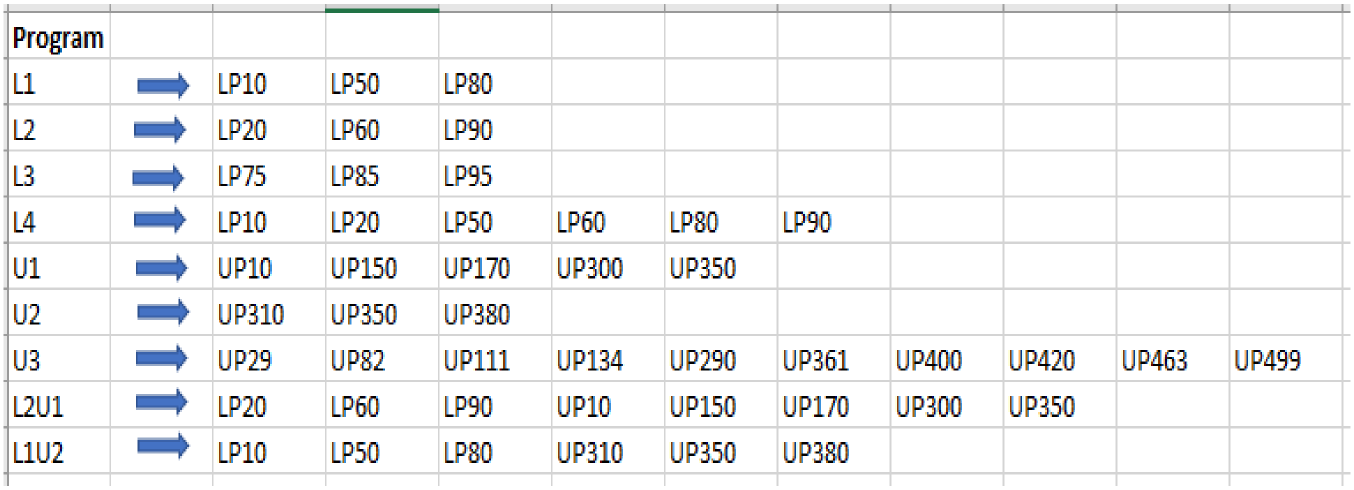
The file access pattern is like the Program Access pattern where initially the user id, Machine id, and Data columns are preprocessed similarly. The Users are having multiple records for the same User id and to find a pattern for the same user we observed that some users are using the same machine every day, but few users are using multiple machines. Hence the users using the same machine are marked with that machine id and the users with multiple machines are marked as MM i.e., Multiple Machines.

While considering the Date column, most of the user's login dates are falling on the Weekdays and few users are logged only on Weekends. The Users who logged on weekdays and weekends are marked accordingly.

We initially found difficulty while looking and the Files column but then we wanted to find if there's any pattern in the way the users are using the file and wanted to check the File ids for the individual users. First, the file id's according to the user is filtered and wrote down to check if there is any pattern in them. There are few patterns that were found with users using a set of files together.

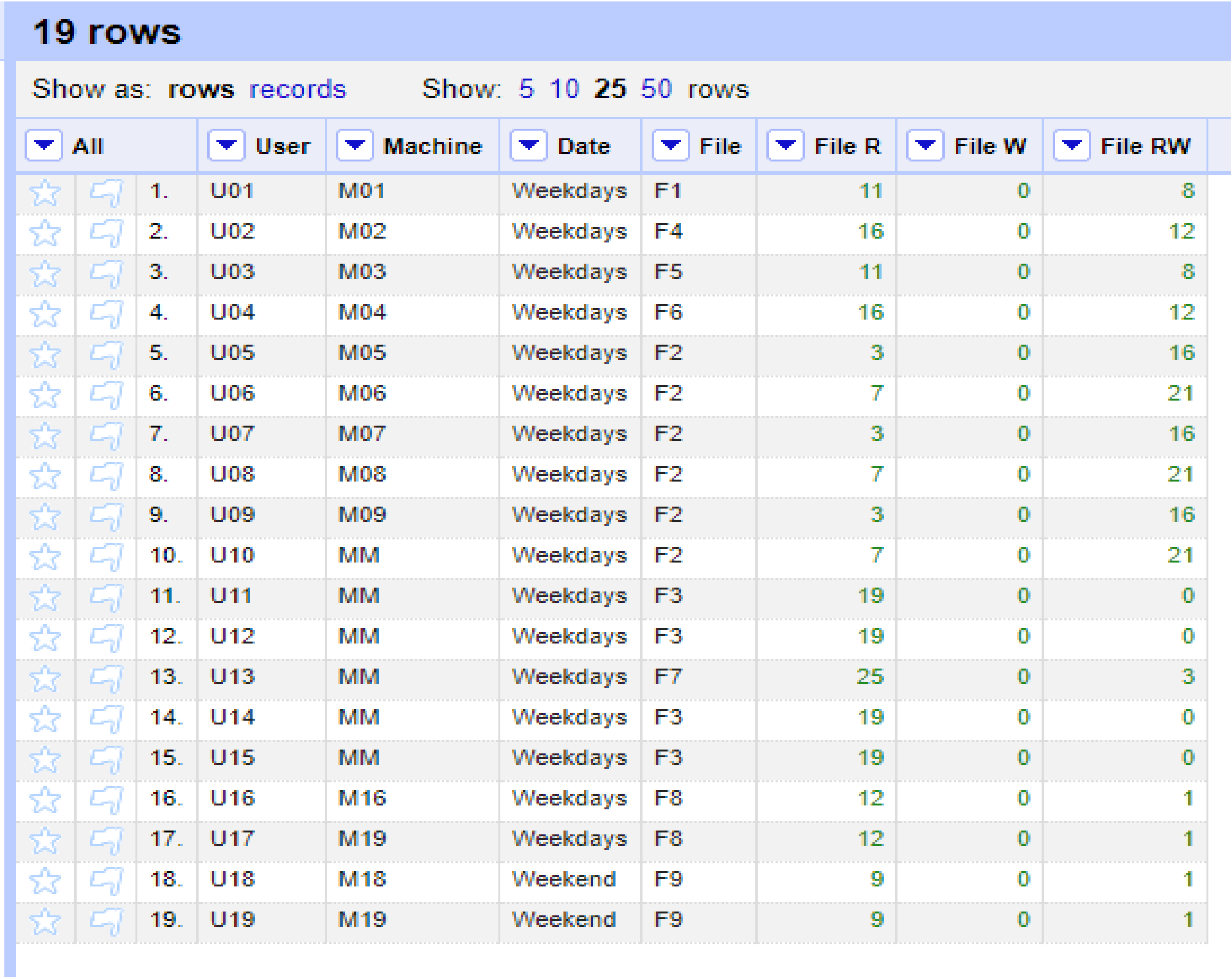
The table 4 below represents how the pattern among the files are found and are named after a single file name like F1, F2 etc. as shown in Table 4. Similarly, for individual users, under the File Read & Write, individual users had either Read, Read/Write or Write are calculated.

For example, U01 has 11 Reads in the data and there are not writes and is marked as o. But had RW of 8 counts and marked accordingly. This is how all the users are marked in table 5 shown below.



Program											
L1	➡	LP10	LP50	LP80							
L2	➡	LP20	LP60	LP90							
L3	➡	LP75	LP85	LP95							
L4	➡	LP10	LP20	LP50	LP60	LP80	LP90				
U1	➡	UP10	UP150	UP170	UP300	UP350					
U2	➡	UP310	UP350	UP380							
U3	➡	UP29	UP82	UP111	UP134	UP290	UP361	UP400	UP420	UP463	UP499
L2U1	➡	LP20	LP60	LP90	UP10	UP150	UP170	UP300	UP350		
L1U2	➡	LP10	LP50	LP80	UP310	UP350	UP380				

Table 4



19 rows

Show as: rows records   Show: 5 10 25 50 rows

All			User	Machine	Date	File	File R	File W	File RW
☆	⚐	1.	U01	M01	Weekdays	F1	11	0	8
☆	⚐	2.	U02	M02	Weekdays	F4	16	0	12
☆	⚐	3.	U03	M03	Weekdays	F5	11	0	8
☆	⚐	4.	U04	M04	Weekdays	F6	16	0	12
☆	⚐	5.	U05	M05	Weekdays	F2	3	0	16
☆	⚐	6.	U06	M06	Weekdays	F2	7	0	21
☆	⚐	7.	U07	M07	Weekdays	F2	3	0	16
☆	⚐	8.	U08	M08	Weekdays	F2	7	0	21
☆	⚐	9.	U09	M09	Weekdays	F2	3	0	16
☆	⚐	10.	U10	MM	Weekdays	F2	7	0	21
☆	⚐	11.	U11	MM	Weekdays	F3	19	0	0
☆	⚐	12.	U12	MM	Weekdays	F3	19	0	0
☆	⚐	13.	U13	MM	Weekdays	F7	25	0	3
☆	⚐	14.	U14	MM	Weekdays	F3	19	0	0
☆	⚐	15.	U15	MM	Weekdays	F3	19	0	0
☆	⚐	16.	U16	M16	Weekdays	F8	12	0	1
☆	⚐	17.	U17	M19	Weekdays	F8	12	0	1
☆	⚐	18.	U18	M18	Weekend	F9	9	0	1
☆	⚐	19.	U19	M19	Weekend	F9	9	0	1

Table 5

Table 5 has the information that is processed according to the users and the file accessed and the related information about Read, Write and Read/Write information. With the obtained information, a graph has been plotted for Read, Write and Read/Write and the number of files accessed.

In figure 9, the X-axis shows the files that are accessed by the users from left to right starting from U01 to U19. And the Y-axis denotes the number of files accessed. Since there are no Writes in the given data only Read and Read/Write is shown in the 3-D graph representation.

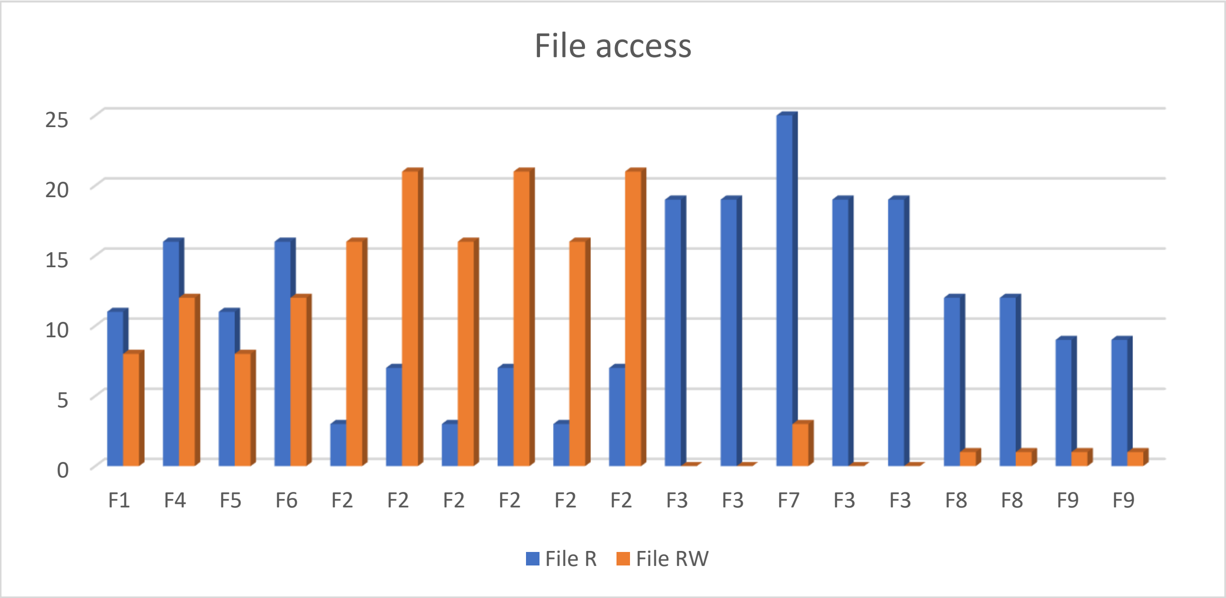
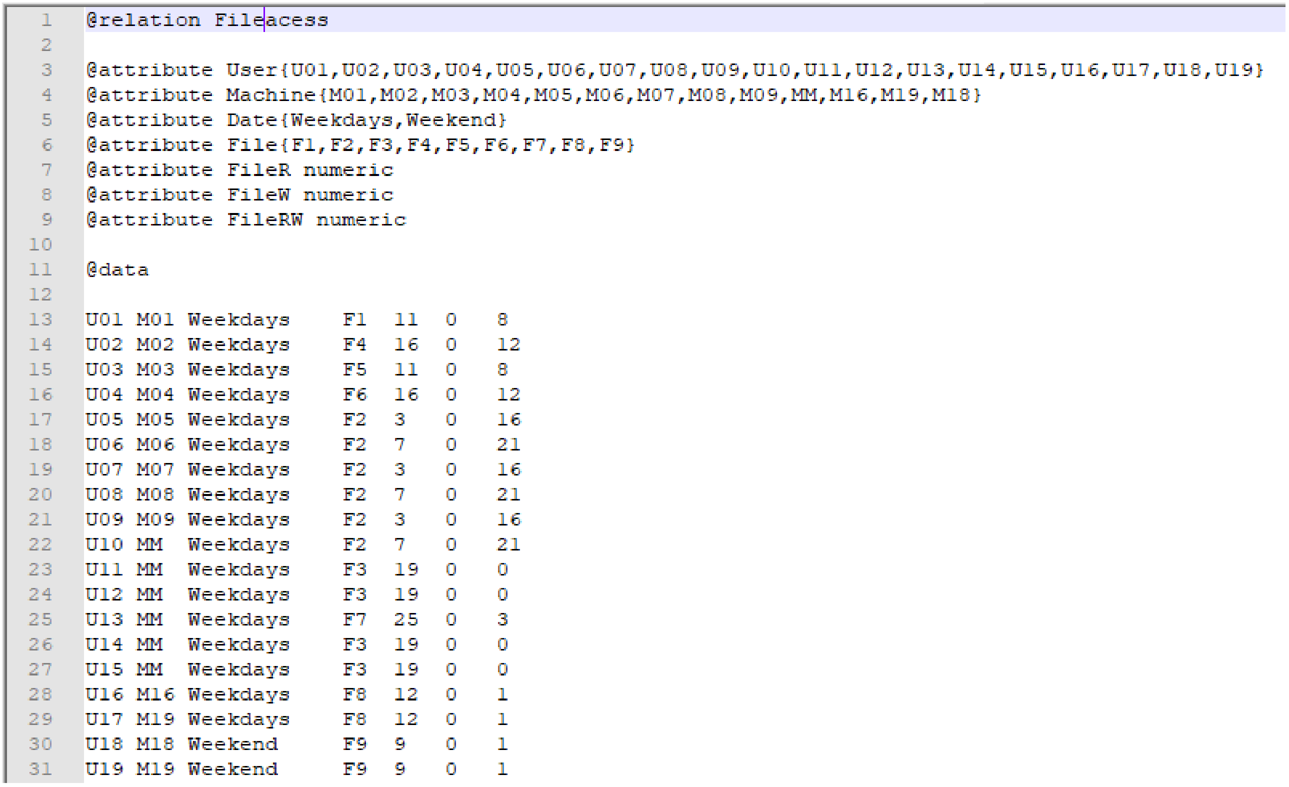


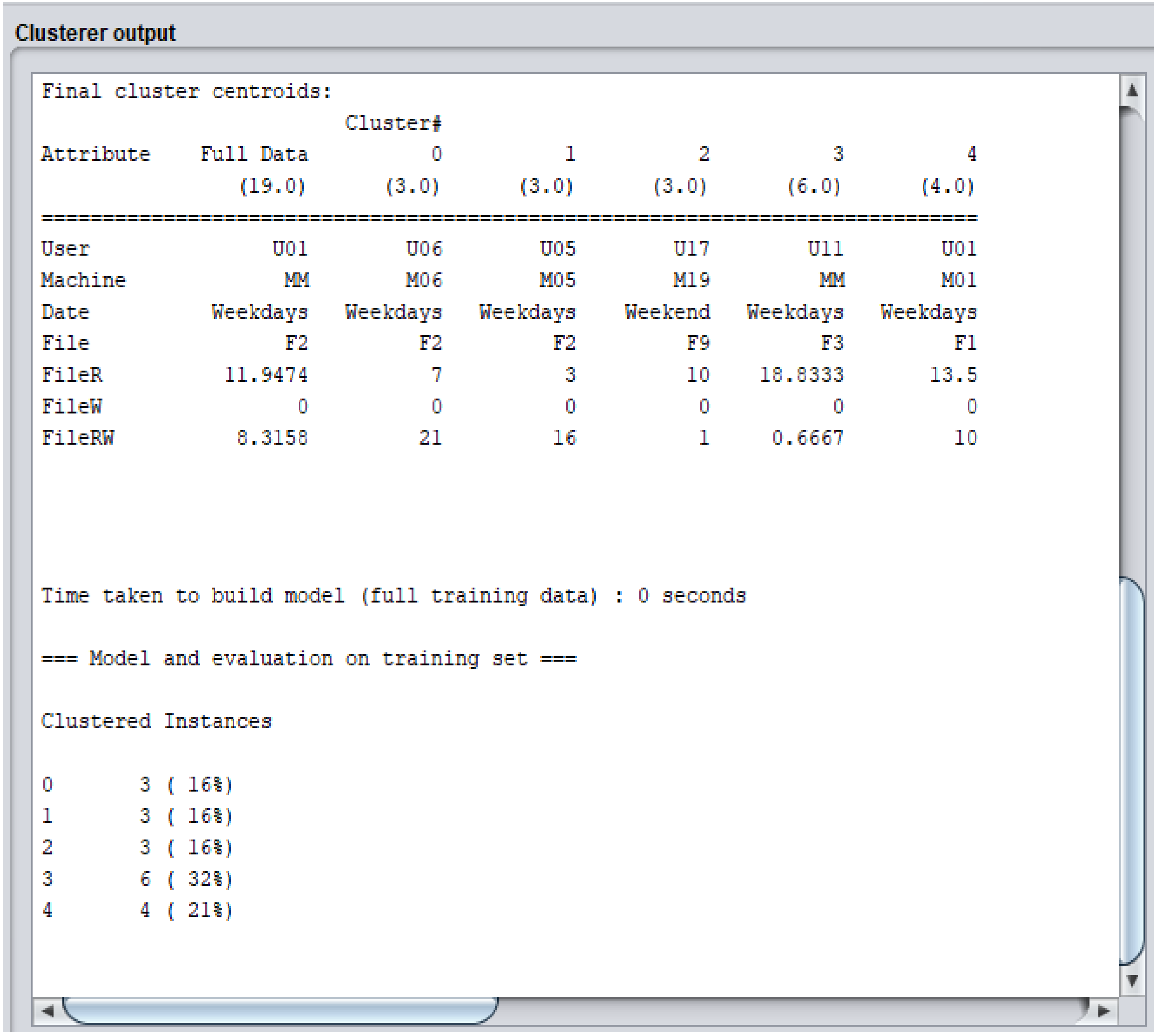
Figure 9



```
@relation Fileacess

@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U15,U16,U17,U18,U19}
@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,MM,M16,M19,M18}
@attribute Date{Weekdays,Weekend}
@attribute File{F1,F2,F3,F4,F5,F6,F7,F8,F9}
@attribute FileR numeric
@attribute FileW numeric
@attribute FileRW numeric

@data

U01 M01 Weekdays    F1  11  0   8
U02 M02 Weekdays    F4  16  0   12
U03 M03 Weekdays    F5  11  0   8
U04 M04 Weekdays    F6  16  0   12
U05 M05 Weekdays    F2  3   0   16
U06 M06 Weekdays    F2  7   0   21
U07 M07 Weekdays    F2  3   0   16
U08 M08 Weekdays    F2  7   0   21
U09 M09 Weekdays    F2  3   0   16
U10 MM  Weekdays    F2  7   0   21
U11 MM  Weekdays    F3  19  0   0
U12 MM  Weekdays    F3  19  0   0
U13 MM  Weekdays    F7  25  0   3
U14 MM  Weekdays    F3  19  0   0
U15 MM  Weekdays    F3  19  0   0
U16 M16 Weekdays    F8  12  0   1
U17 M19 Weekdays    F8  12  0   1
U18 M18 Weekend     F9  9   0   1
U19 M19 Weekend     F9  9   0   1
```

Figure 10

The data from the Table 5, is now converted to the attribute-relation file format. Figure 10 is taken from Notepad++ which is converting the given data to Fileaccess.arff file. The file is then loaded into Weka tool to apply the SimpleKMeans clustering algorithm. By doing so, we checked that the patterns we found are having a minimal error rate.

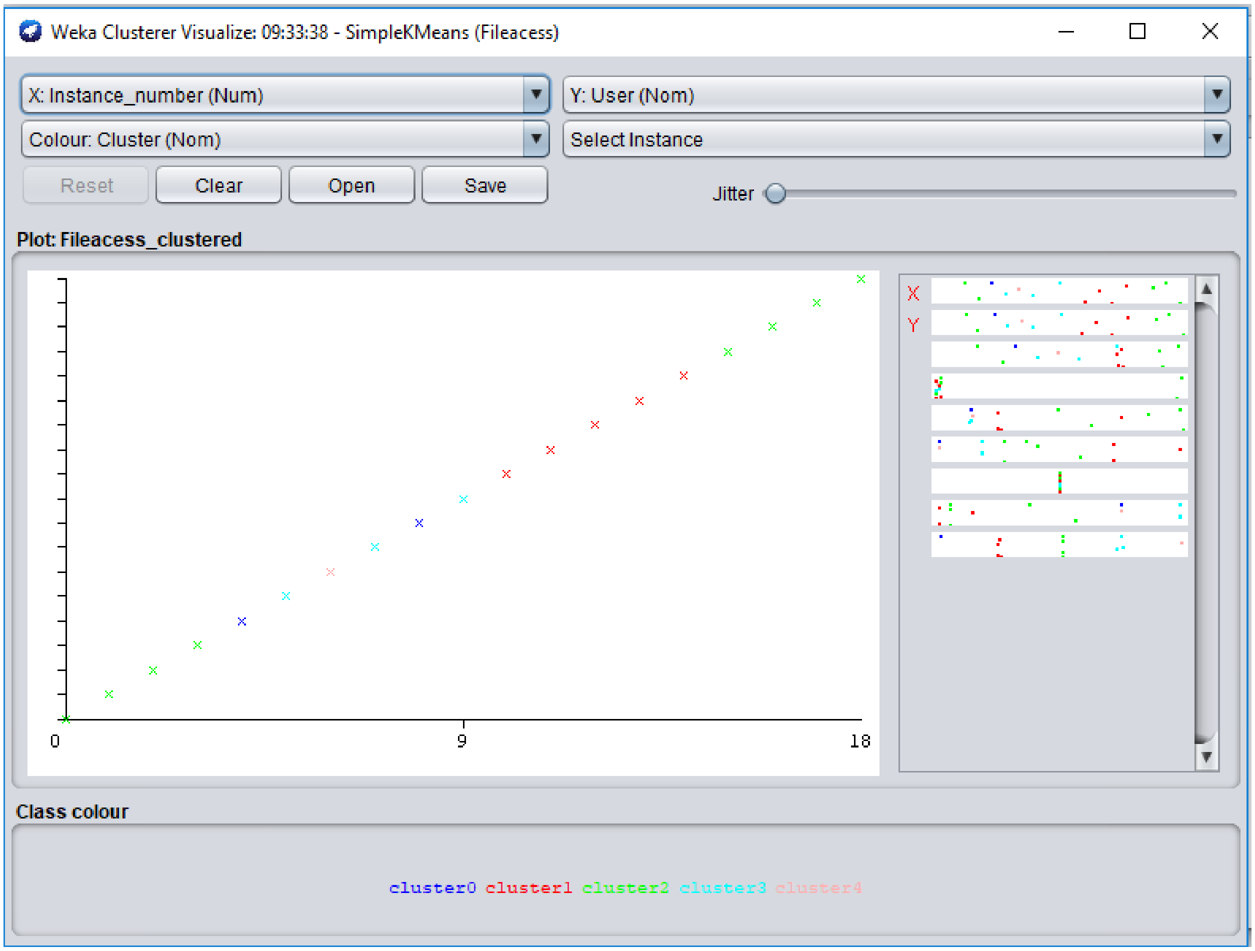


```
Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                (19.0)      (3.0)      (3.0)      (3.0)      (6.0)      (4.0)
==============================================================================
User               U01        U06        U05        U17        U11        U01
Machine             MM        M06        M05        M19         MM        M01
Date          Weekdays   Weekdays   Weekdays    Weekend   Weekdays   Weekdays
File                F2         F2         F2         F9         F3         F1
FileR          11.9474          7          3         10    18.8333       13.5
FileW                0          0          0          0          0          0
FileRW          8.3158         21         16          1     0.6667         10




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       3 ( 16%)
1       3 ( 16%)
2       3 ( 16%)
3       6 ( 32%)
4       4 ( 21%)
```

Figure 11

Figure 11 is the output for SimpleKMeans clustering of File access Pattern which has clustered instances and the final cluster centroids. The output shows 5 clusters of data.

Figure 12 shows the visualization of SimpleKMeans algorithm for File access pattern. The denotes the instance number and Users on Y-axis. The five clusters are denoted with five different colors in the graph and are marked on the graph.



Weka Clusterer Visualize: 09:33:38 - SimpleKMeans (Fileacess)

X: Instance_number (Num)

Y: User (Nom)

Colour: Cluster (Nom)

Select Instance

Plot: Fileacess_clustered

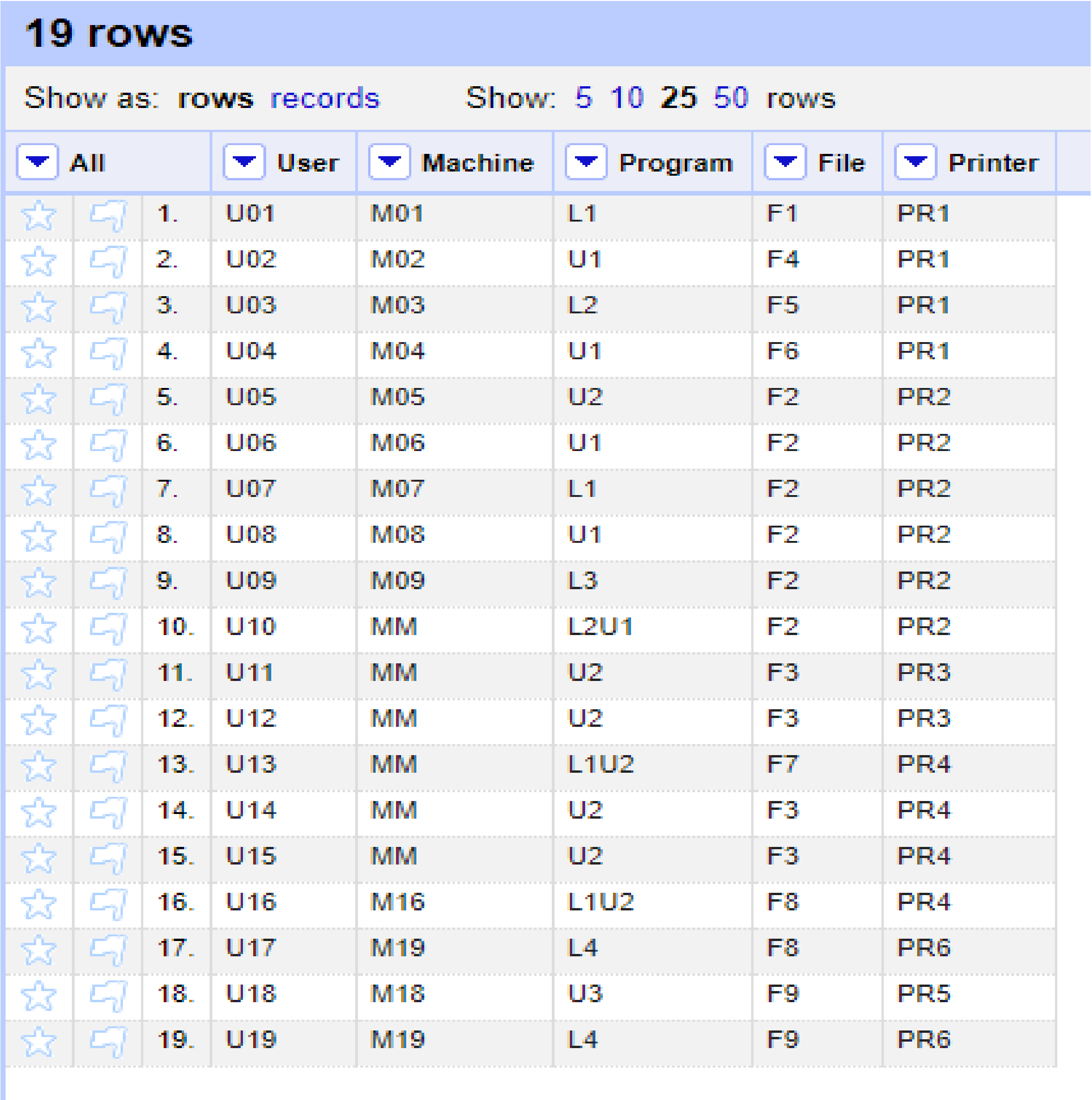
Class colour

cluster0 cluster1 cluster2 cluster3 cluster4

Figure 12

Printer Usage Pattern

In the case of printer usage pattern, User is considered as standard and based on usage of printers by a user. In this scenario, the User U01 is always using PR1. Similarly, there are other users who always use the same printer. The other columns like Machine, Program, and file are considered just as previous observations. The below Table 6 shows the way how each user information associated with the printer is tabulated.



19 rows

Show as: **rows** records Show: 5 10 **25** 50 rows

All			User	Machine	Program	File	Printer
☆		1.	U01	M01	L1	F1	PR1
☆		2.	U02	M02	U1	F4	PR1
☆		3.	U03	M03	L2	F5	PR1
☆		4.	U04	M04	U1	F6	PR1
☆		5.	U05	M05	U2	F2	PR2
☆		6.	U06	M06	U1	F2	PR2
☆		7.	U07	M07	L1	F2	PR2
☆		8.	U08	M08	U1	F2	PR2
☆		9.	U09	M09	L3	F2	PR2
☆		10.	U10	MM	L2U1	F2	PR2
☆		11.	U11	MM	U2	F3	PR3
☆		12.	U12	MM	U2	F3	PR3
☆		13.	U13	MM	L1U2	F7	PR4
☆		14.	U14	MM	U2	F3	PR4
☆		15.	U15	MM	U2	F3	PR4
☆		16.	U16	M16	L1U2	F8	PR4
☆		17.	U17	M19	L4	F8	PR6
☆		18.	U18	M18	U3	F9	PR5
☆		19.	U19	M19	L4	F9	PR6

Table 6

A 3-Dimensional graph has been plotted with users and files they accessed on X-axis and usage of printers on Y-axis. Figure 13 above shows the graphical representation of printer usage patterns.

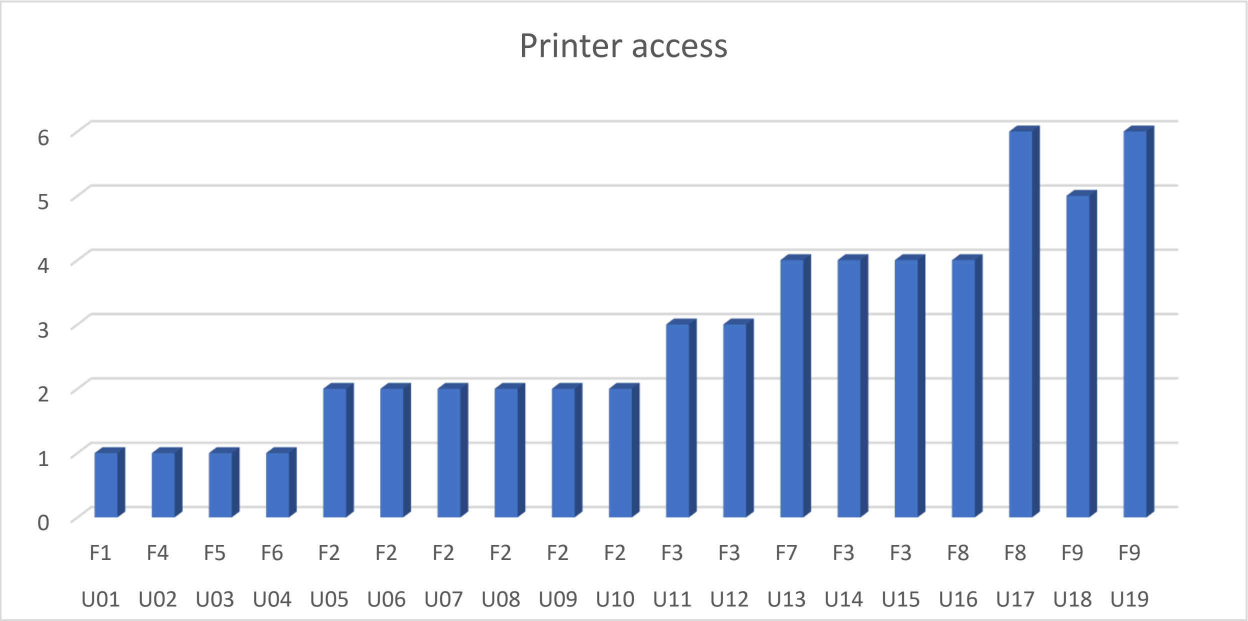
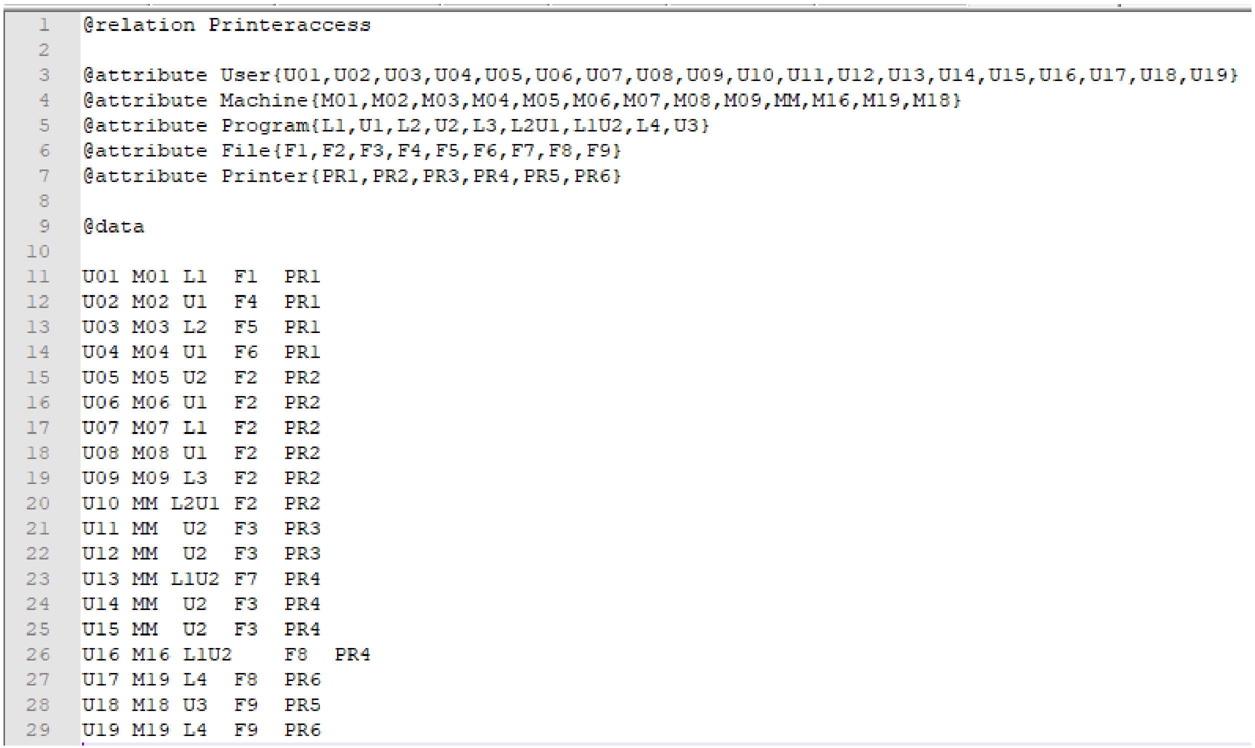


Figure 13

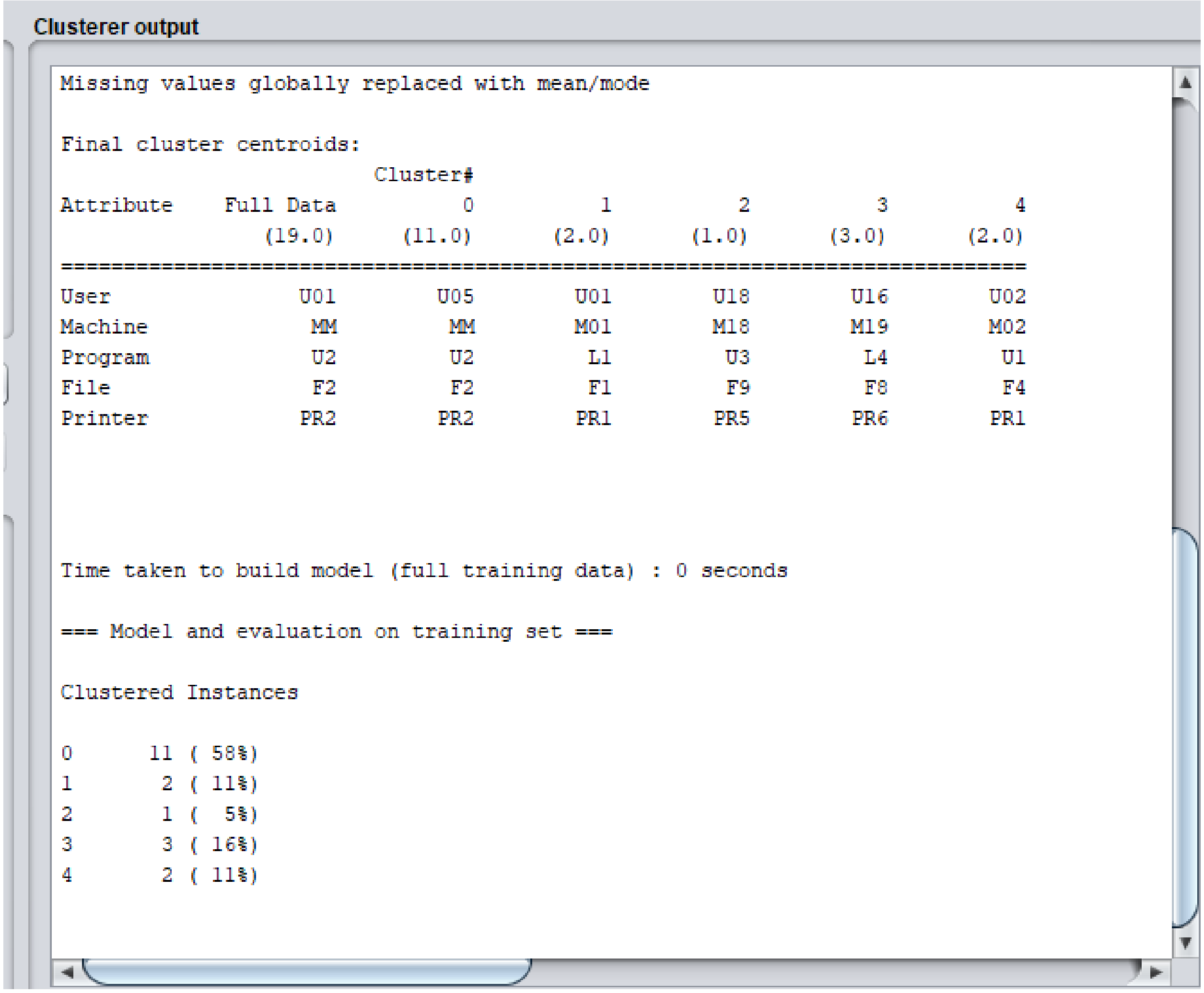


```
@relation Printeraccess

@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U15,U16,U17,U18,U19}
@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,MM,M16,M19,M18}
@attribute Program{L1,U1,L2,U2,L3,L2U1,L1U2,L4,U3}
@attribute File{F1,F2,F3,F4,F5,F6,F7,F8,F9}
@attribute Printer{PR1,PR2,PR3,PR4,PR5,PR6}

@data

U01 M01 L1   F1   PR1
U02 M02 U1   F4   PR1
U03 M03 L2   F5   PR1
U04 M04 U1   F6   PR1
U05 M05 U2   F2   PR2
U06 M06 U1   F2   PR2
U07 M07 L1   F2   PR2
U08 M08 U1   F2   PR2
U09 M09 L3   F2   PR2
U10 MM  L2U1 F2   PR2
U11 MM   U2  F3   PR3
U12 MM   U2  F3   PR3
U13 MM  L1U2 F7   PR4
U14 MM   U2  F3   PR4
U15 MM   U2  F3   PR4
U16 M16 L1U2     F8   PR4
U17 M19 L4   F8   PR6
U18 M18 U3   F9   PR5
U19 M19 L4   F9   PR6
```

Figure 14

The obtained data now is converted into attribute-relation file format using notepad++. Figure 14 indicates the notepad++ conversion of Printer access file to Printeraccess.arff file. Once the data file is loaded into the Weka tool to compare our manual work, SimpleKMeans clustering technique is applied to data.

Figure 15 shows the output of the Printeraccess.arff file that is loaded into the Weka tool. It shows the Final cluster centroids and five clusters are divided. The cluster instances for printer access are also shown in the figure below.



```
Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                (19.0)     (11.0)      (2.0)      (1.0)      (3.0)      (2.0)
=============================================================================
User               U01        U05        U01        U18        U16        U02
Machine             MM         MM        M01        M18        M19        M02
Program             U2         U2         L1         U3         L4         U1
File                F2         F2         F1         F9         F8         F4
Printer            PR2        PR2        PR1        PR5        PR6        PR1




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      11 ( 58%)
1       2 ( 11%)
2       1 (  5%)
3       3 ( 16%)
4       2 ( 11%)
```

Figure 15

The clusters obtained from SimpleKMeans is visualized in Figure 16 in the form of a graph. The graph has instance number on the X-axis and User on the Y-axis. The five clusters are marked with five different colors.

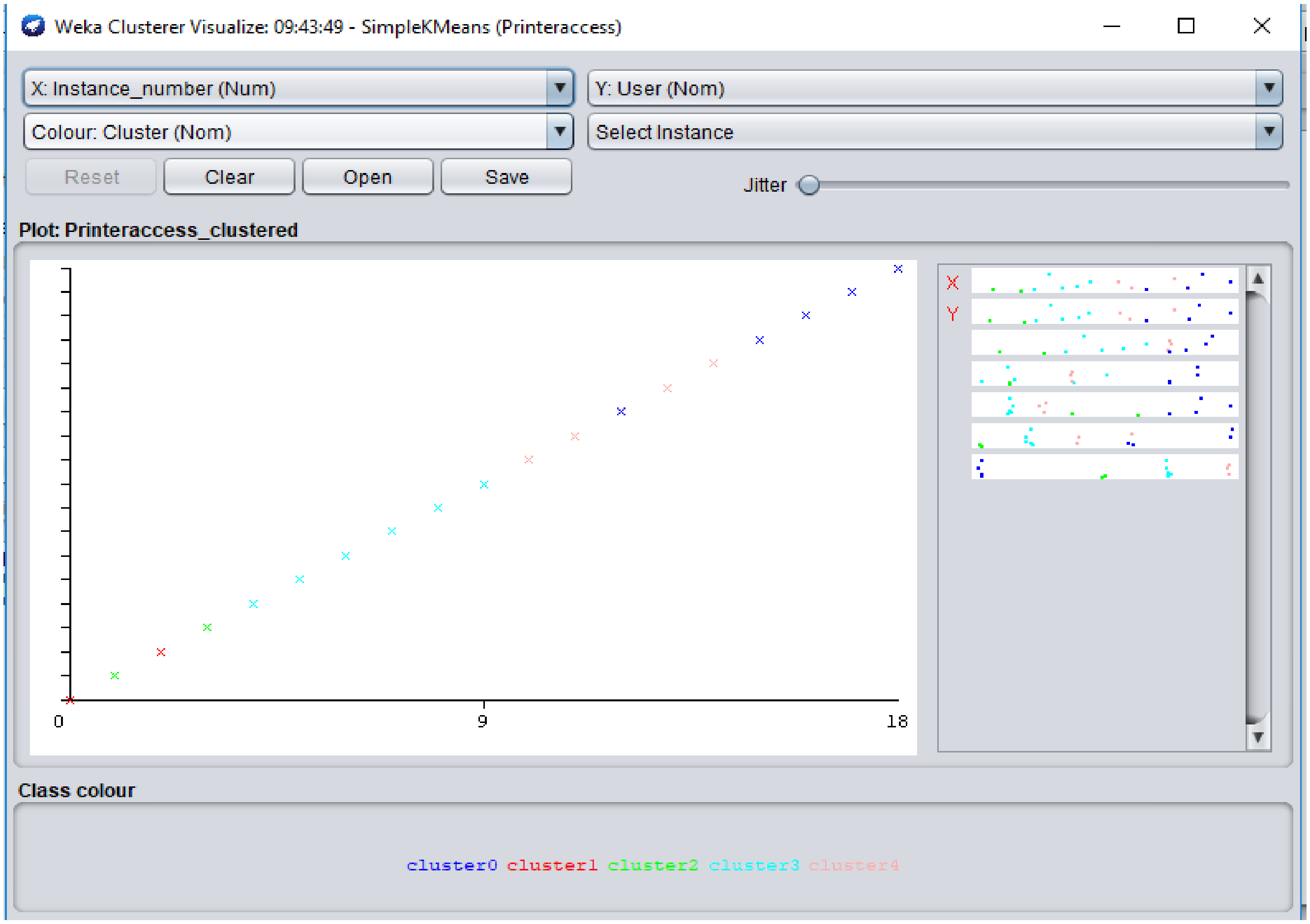

Weka Clusterer Visualize: 09:43:49 - SimpleKMeans (Printeraccess)
X: Instance_number (Num)
Y: User (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: Printeraccess_clustered
0
9
18
X
Y
Class colour
cluster0 cluster1 cluster2 cluster3 cluster4

Figure 16

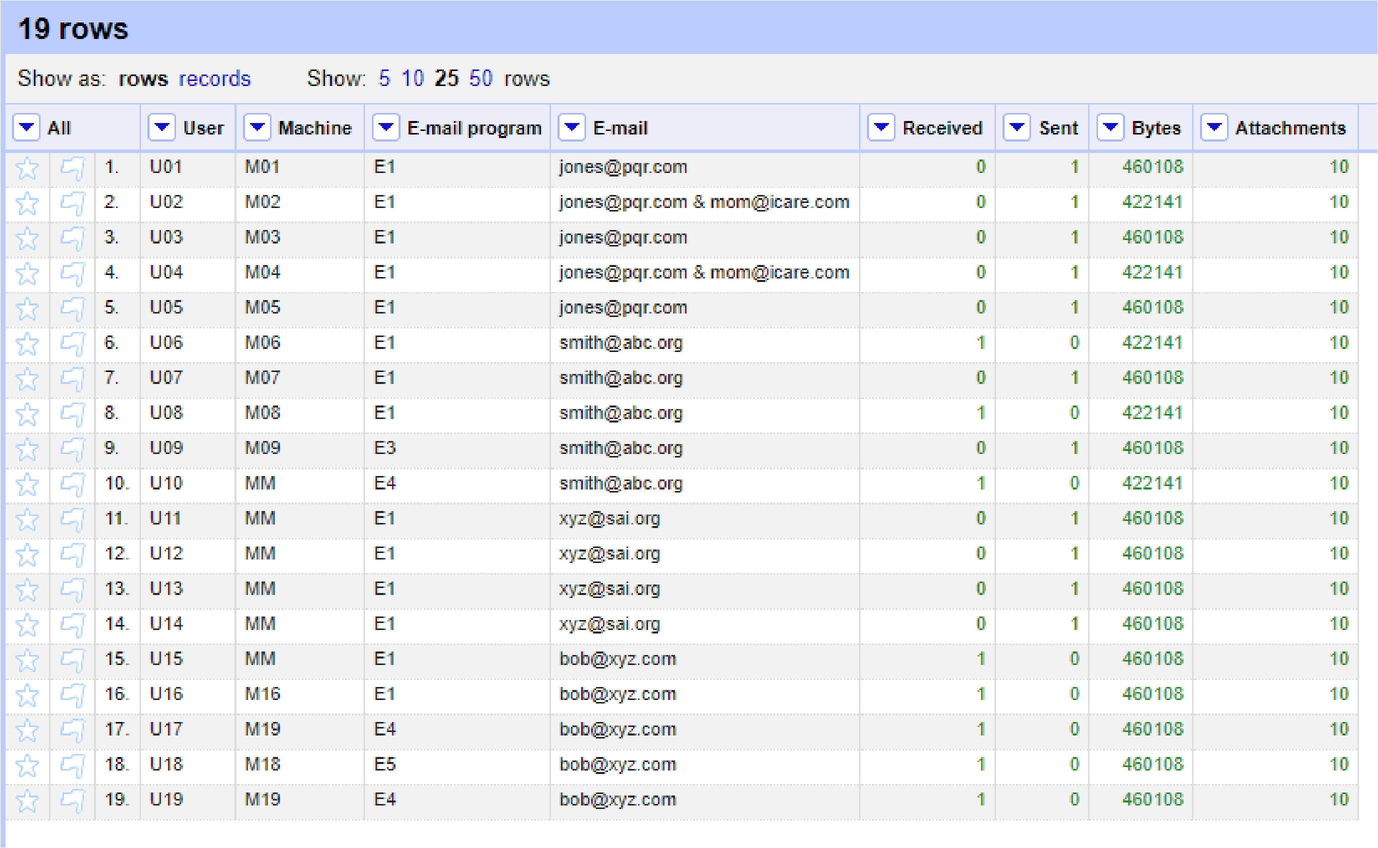
E-mail Pattern

For E-mail pattern, the type 3 records of data are considered. Like the other cases or preprocessing the data, the email patterns are based on the user perspective. The Machine login is considered as same if every time the user logged into the same machine and if there are multiple machines that a user logs in.

The E-mail program is of three type E1, E2 and E3 and the users who are using same the type of email programs are given the same value. Email address is like this as most of the users are using the same email address for communication, but few are using multiple email addresses. And such users are marked with both the email addresses.

Under the sent or received space if a user sends the data or receives the data, there are two separate columns for Received and Sent emails. If a user receives the data, he will be marked as 1 and if he doesn’t it is marked as 0. If a user sends the data, the user will be marked as 1 and if the user doesn’t it is marked as 0.

The Attachments column is the count of each user and then the count is noted in the column which is shown in Table 7. The Table 7 below shows the processed data by considering all mentioned changes.



19 rows

Show as: rows records   Show: 5 10 25 50 rows

All	User	Machine	E-mail program	E-mail	Received	Sent	Bytes	Attachments
1.	U01	M01	E1	jones@pqr.com	0	1	460108	10
2.	U02	M02	E1	jones@pqr.com & mom@icare.com	0	1	422141	10
3.	U03	M03	E1	jones@pqr.com	0	1	460108	10
4.	U04	M04	E1	jones@pqr.com & mom@icare.com	0	1	422141	10
5.	U05	M05	E1	jones@pqr.com	0	1	460108	10
6.	U06	M06	E1	smith@abc.org	1	0	422141	10
7.	U07	M07	E1	smith@abc.org	0	1	460108	10
8.	U08	M08	E1	smith@abc.org	1	0	422141	10
9.	U09	M09	E3	smith@abc.org	0	1	460108	10
10.	U10	MM	E4	smith@abc.org	1	0	422141	10
11.	U11	MM	E1	xyz@sai.org	0	1	460108	10
12.	U12	MM	E1	xyz@sai.org	0	1	460108	10
13.	U13	MM	E1	xyz@sai.org	0	1	460108	10
14.	U14	MM	E1	xyz@sai.org	0	1	460108	10
15.	U15	MM	E1	bob@xyz.com	1	0	460108	10
16.	U16	M16	E1	bob@xyz.com	1	0	460108	10
17.	U17	M19	E4	bob@xyz.com	1	0	460108	10
18.	U18	M18	E5	bob@xyz.com	1	0	460108	10
19.	U19	M19	E4	bob@xyz.com	1	0	460108	10

Table 7

A 3-Dimensional representation is shown in Figure 17 below which shows the E-mail program, the users, and their values. The X-axis has the E-mail program information and the Users with the values representing on it.

Now, to check the manually obtained values, the values from OpenRefine are to be converted into the attribute-relation file format. To do so, the values are to be taken into notepad++ and then converted into Emailaccess.arff file. Figure 18 represents the attribute relation file format and that file must be loaded on to Weka.

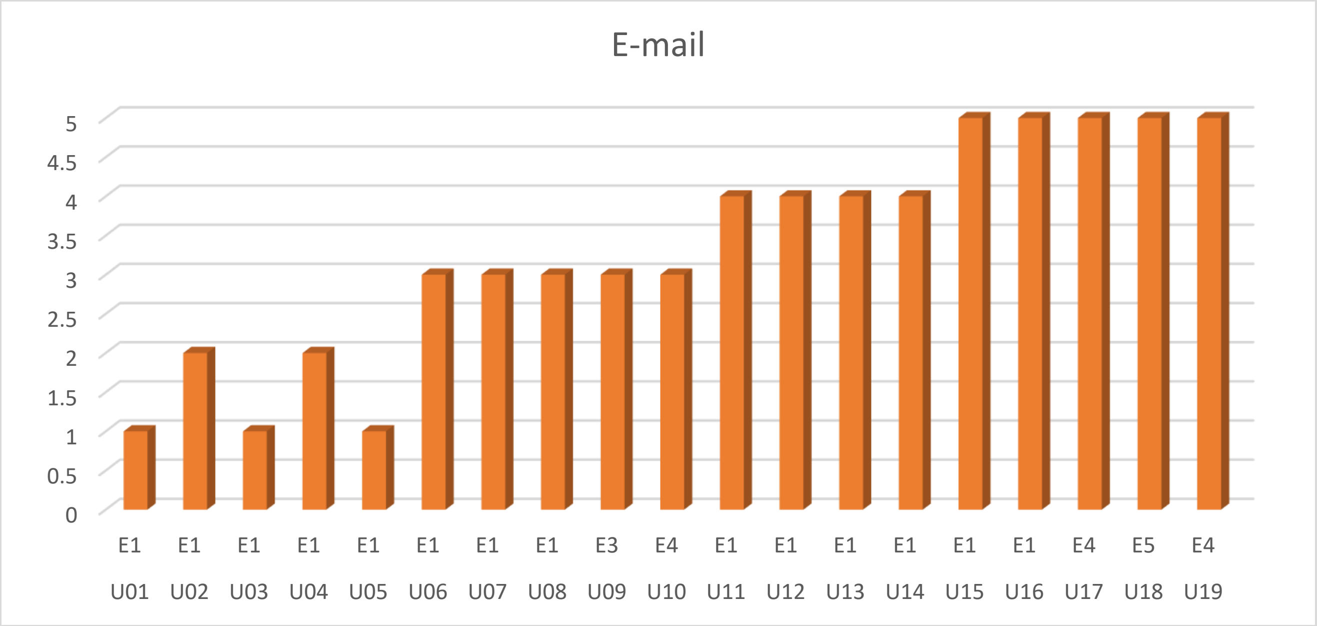
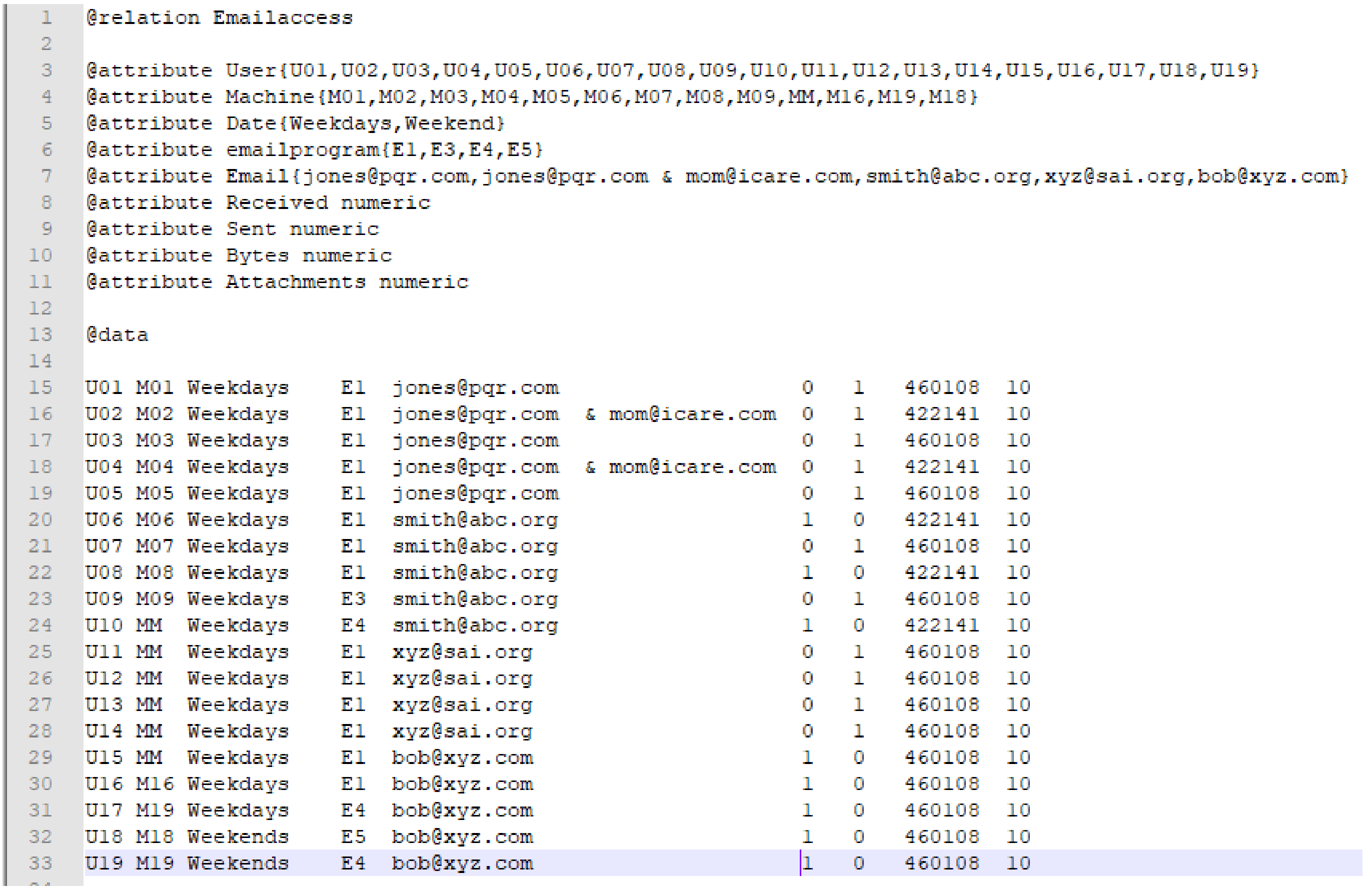
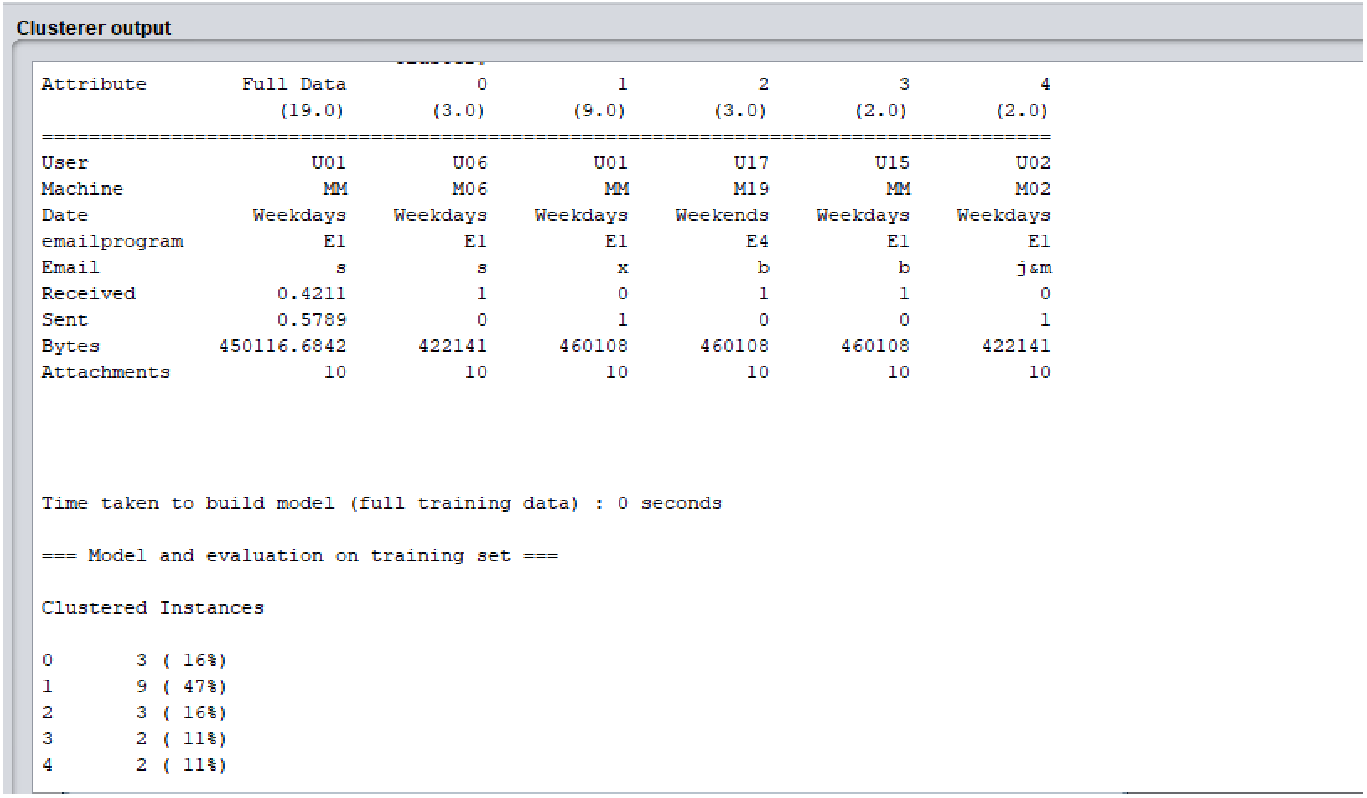


Figure 17



```
@relation Emailaccess

@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U15,U16,U17,U18,U19}
@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,MM,M16,M19,M18}
@attribute Date{Weekdays,Weekend}
@attribute emailprogram{E1,E3,E4,E5}
@attribute Email{jones@pqr.com,jones@pqr.com & mom@icare.com,smith@abc.org,xyz@sai.org,bob@xyz.com}
@attribute Received numeric
@attribute Sent numeric
@attribute Bytes numeric
@attribute Attachments numeric

@data

U01 M01 Weekdays    E1  jones@pqr.com                    0   1   460108  10
U02 M02 Weekdays    E1  jones@pqr.com   & mom@icare.com  0   1   422141  10
U03 M03 Weekdays    E1  jones@pqr.com                    0   1   460108  10
U04 M04 Weekdays    E1  jones@pqr.com   & mom@icare.com  0   1   422141  10
U05 M05 Weekdays    E1  jones@pqr.com                    0   1   460108  10
U06 M06 Weekdays    E1  smith@abc.org                    1   0   422141  10
U07 M07 Weekdays    E1  smith@abc.org                    0   1   460108  10
U08 M08 Weekdays    E1  smith@abc.org                    1   0   422141  10
U09 M09 Weekdays    E3  smith@abc.org                    0   1   460108  10
U10 MM  Weekdays    E4  smith@abc.org                    1   0   422141  10
U11 MM  Weekdays    E1  xyz@sai.org                      0   1   460108  10
U12 MM  Weekdays    E1  xyz@sai.org                      0   1   460108  10
U13 MM  Weekdays    E1  xyz@sai.org                      0   1   460108  10
U14 MM  Weekdays    E1  xyz@sai.org                      0   1   460108  10
U15 MM  Weekdays    E1  bob@xyz.com                      1   0   460108  10
U16 M16 Weekdays    E1  bob@xyz.com                      1   0   460108  10
U17 M19 Weekdays    E4  bob@xyz.com                      1   0   460108  10
U18 M18 Weekends    E5  bob@xyz.com                      1   0   460108  10
U19 M19 Weekends    E4  bob@xyz.com                      1   0   460108  10
```

Figure 18



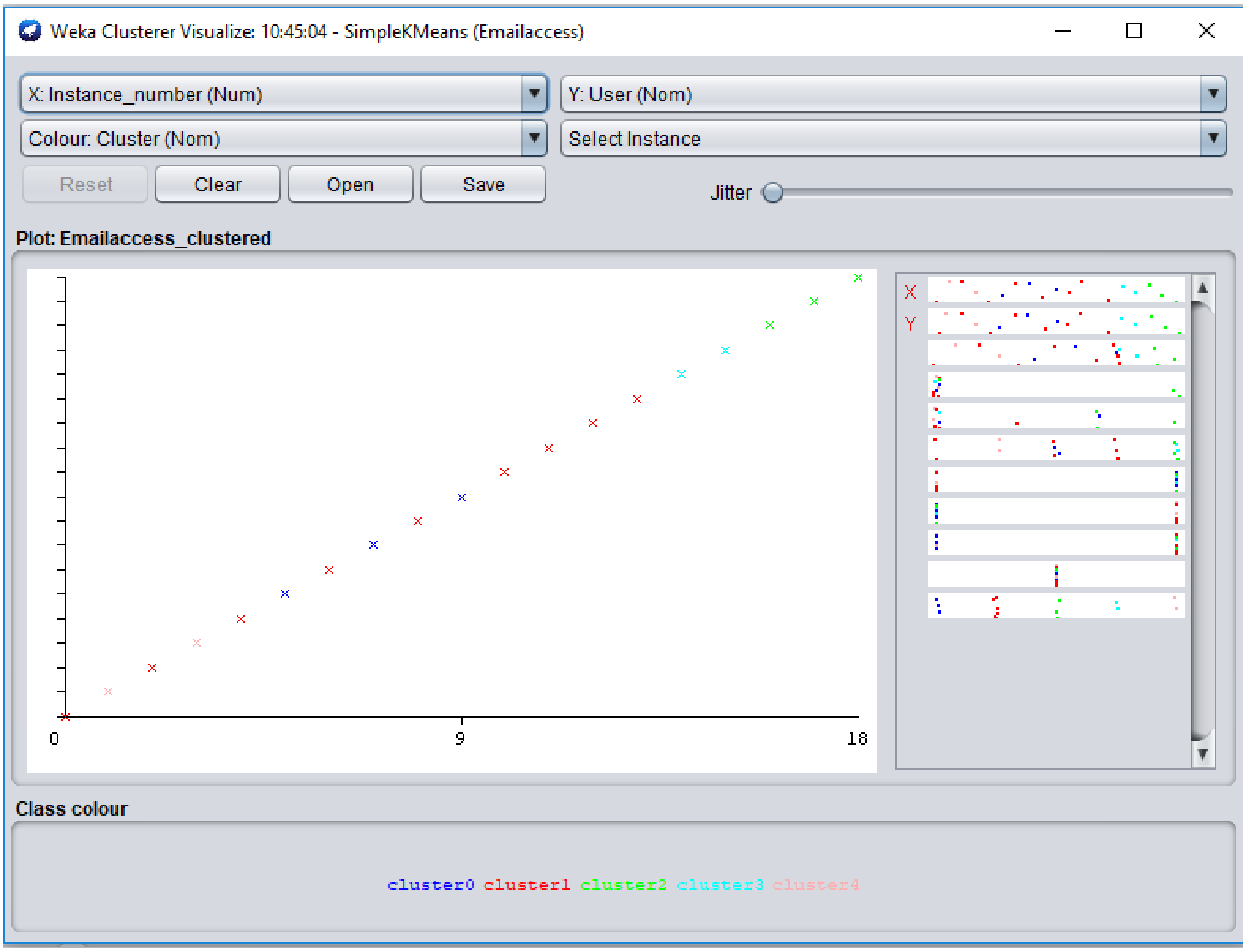
Clusterer output

```
Attribute         Full Data          0           1           2           3           4
                     (19.0)      (3.0)       (9.0)       (3.0)       (2.0)       (2.0)
=====================================================================================
User                    U01        U06         U01         U17         U15         U02
Machine                  MM        M06          MM         M19          MM         M02
Date               Weekdays   Weekdays    Weekdays    Weekends    Weekdays    Weekdays
emailprogram             E1         E1          E1          E4          E1          E1
Email                     s          s           x           b           b         j&m
Received             0.4211          1           0           1           1           0
Sent                 0.5789          0           1           0           0           1
Bytes           450116.6842     422141      460108      460108      460108      422141
Attachments              10         10          10          10          10          10




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       3 ( 16%)
1       9 ( 47%)
2       3 ( 16%)
3       2 ( 11%)
4       2 ( 11%)
```

Figure 19

Once the file is loaded into Weka to check the clusters, SimpleKMeans algorithm is applied to check the clusters and the Figure 19 indicates the output of the Email access pattern. The final cluster centroid is shown in the figure and the clustered instances with five clusters in it.

Figure 20 shows the visualization of SimpleKMeans algorithm for File access pattern. The denotes the instance number and Users on Y-axis. The five clusters are denoted with five different colors in the graph and are marked on the graph.



Weka Clusterer Visualize: 10:45:04 - SimpleKMeans (Emailaccess)

X: Instance_number (Num)

Y: User (Nom)

Colour: Cluster (Nom)

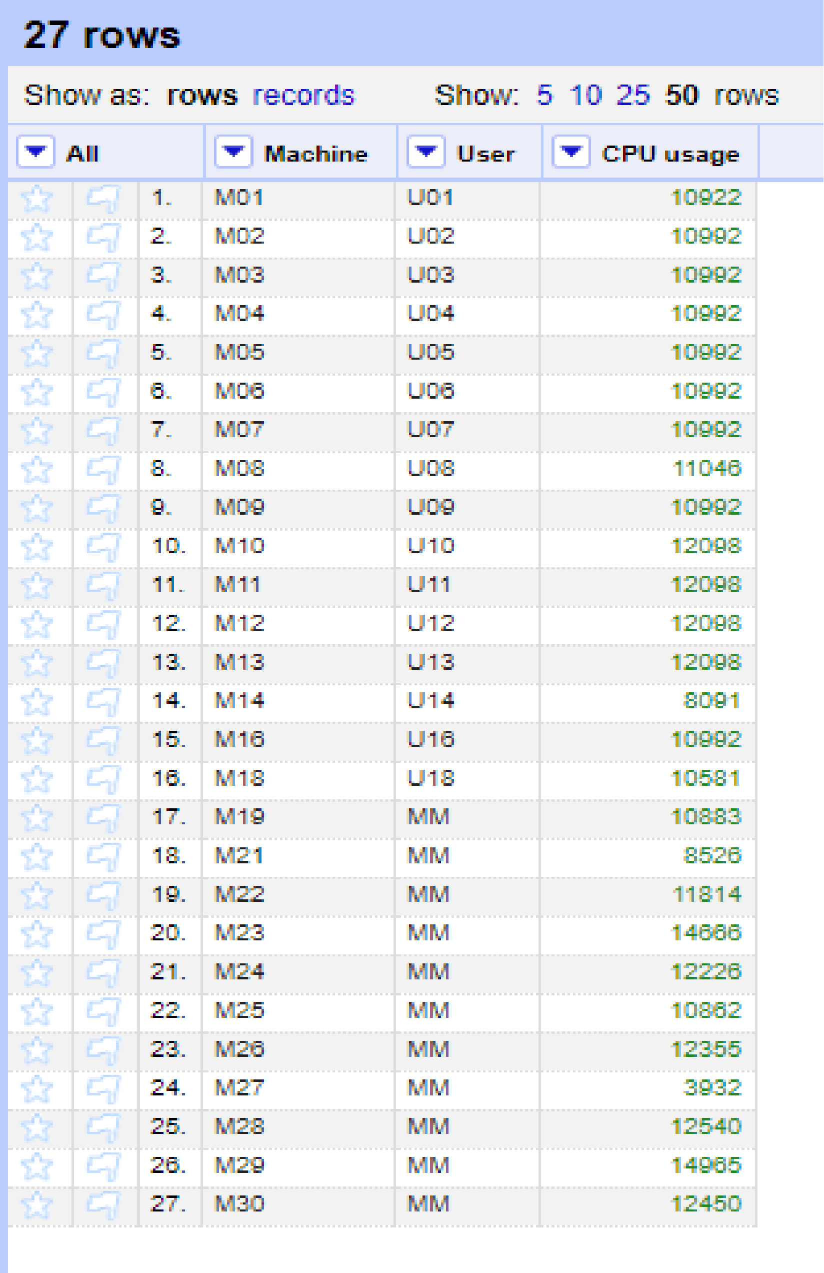
Plot: Emailaccess_clustered

Figure 20

Machine Usage Pattern

The Machine Usage Pattern, the machine id is considered as the base and users using it will be changing. By considering the Machine id the analysis will be done by observing how users are using the machine. For example, in the given data there are 30 Machines and only 19 users. Now while looking at Machine M01, only user u01 is using that machine every time in the given data. In such cases, the U01 is taken for the machine M01.

But for the machine where there are multiple users logging into it, it is marked as Multiple Machines i.e., MM which is nothing but multiple users using the same machine in this case. The Table 8 clearly shows how the mentioned notations are tabulated. Similarly, the average CPU usage is considered for all the machines and the respective values are mentioned for the individual machines.



27 rows

Show as: rows records Show: 5 10 25 50 rows

All	Machine	User	CPU usage
1.	M01	U01	10922
2.	M02	U02	10992
3.	M03	U03	10992
4.	M04	U04	10992
5.	M05	U05	10992
6.	M06	U06	10992
7.	M07	U07	10992
8.	M08	U08	11046
9.	M09	U09	10992
10.	M10	U10	12098
11.	M11	U11	12098
12.	M12	U12	12098
13.	M13	U13	12098
14.	M14	U14	8091
15.	M16	U16	10992
16.	M18	U18	10581
17.	M19	MM	10883
18.	M21	MM	8526
19.	M22	MM	11814
20.	M23	MM	14666
21.	M24	MM	12226
22.	M25	MM	10862
23.	M26	MM	12355
24.	M27	MM	3932
25.	M28	MM	12540
26.	M29	MM	14965
27.	M30	MM	12450

Table 8

From the above table, a graph has been plotted between the CPU usage, users and the machines in Figure 21 below. The Users are mentioned on the X-axis and the Machines and the CPU usage on the Y-axis.

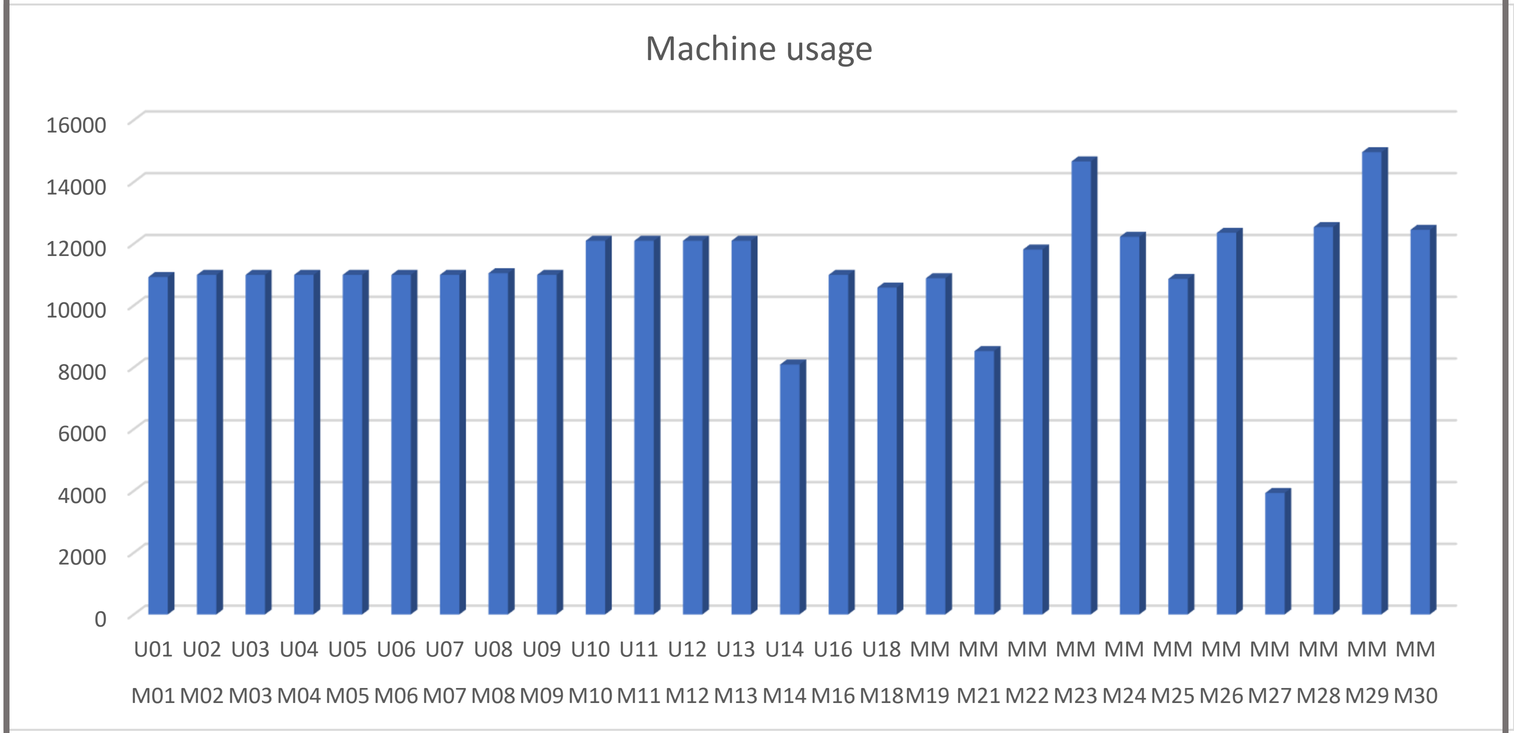
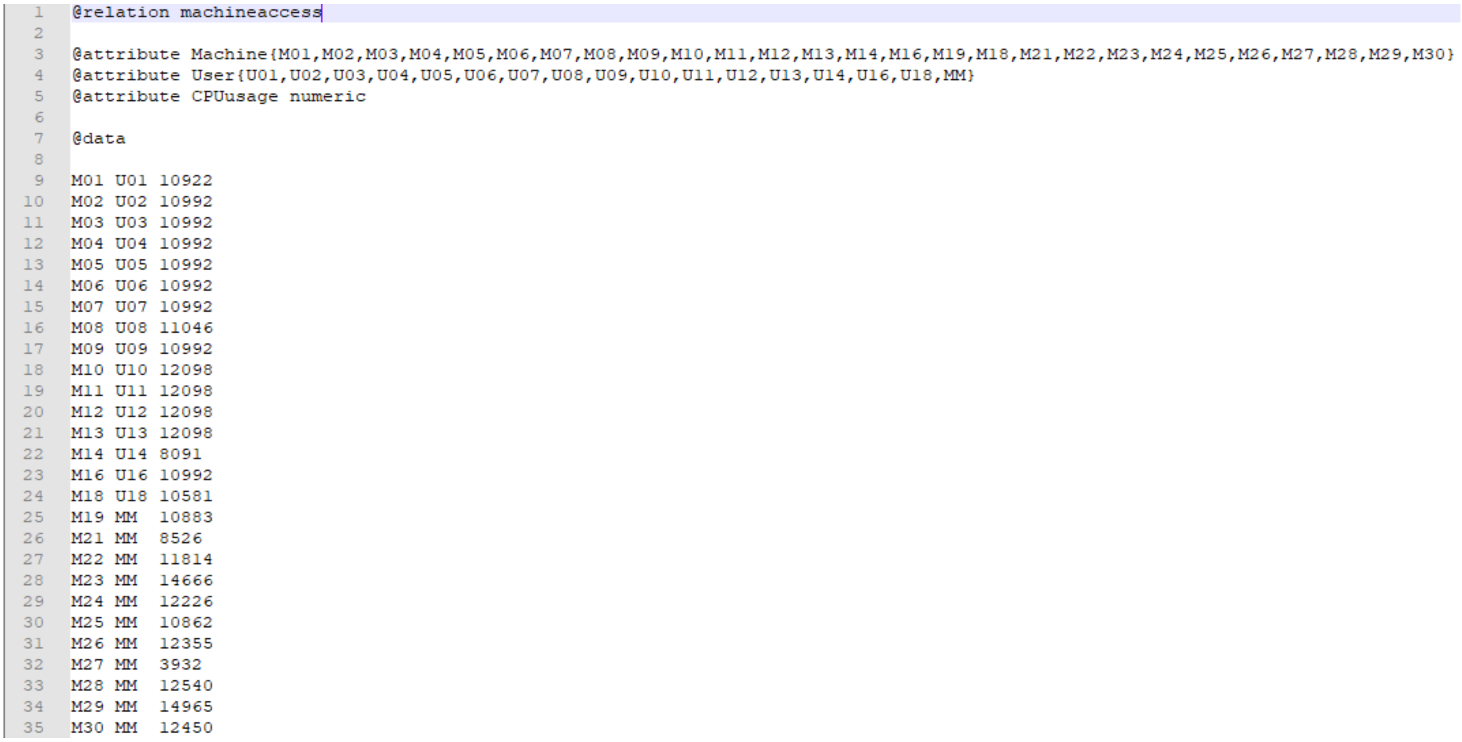


Figure 21

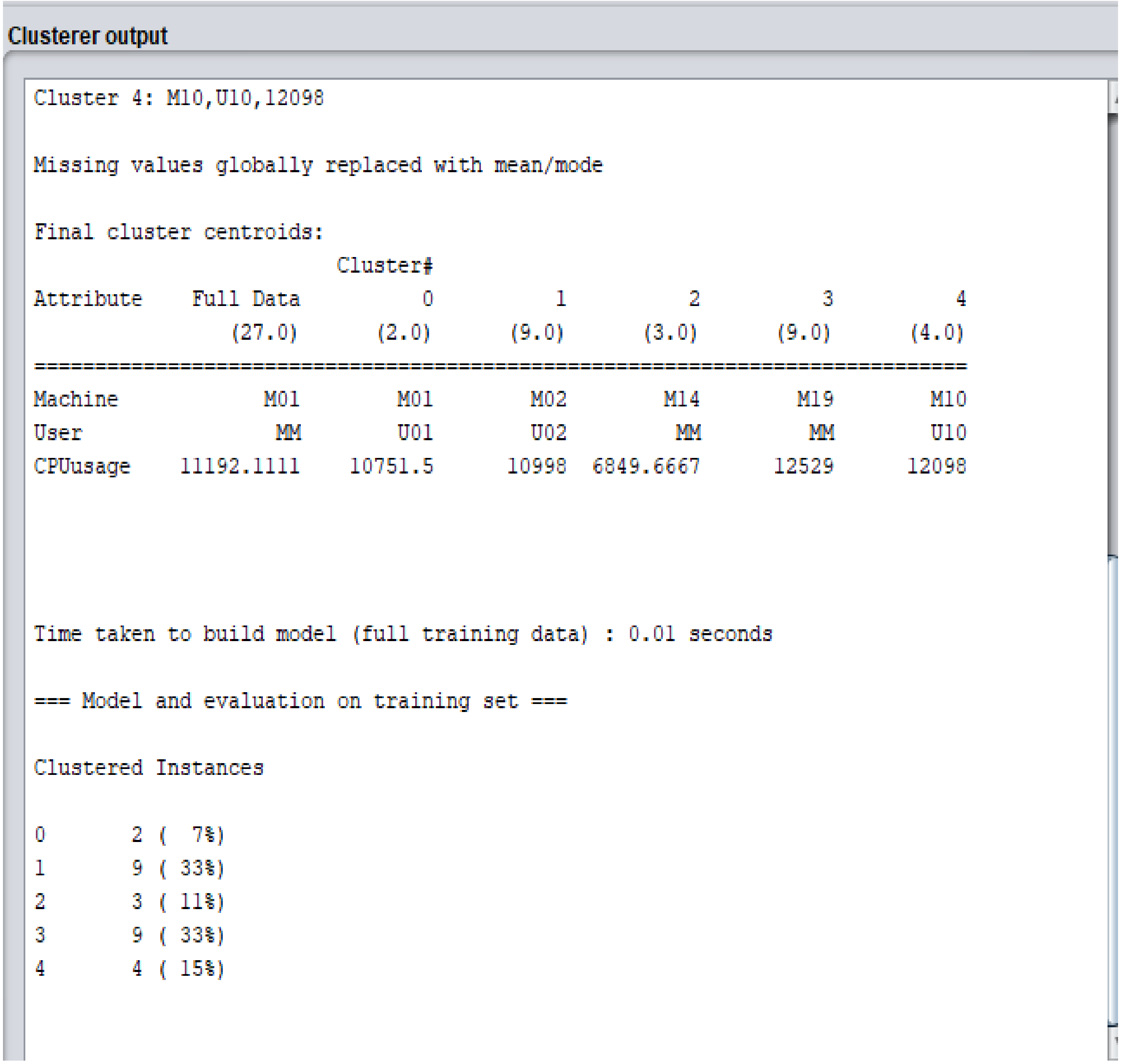


```
@relation machineaccess

@attribute Machine{M01,M02,M03,M04,M05,M06,M07,M08,M09,M10,M11,M12,M13,M14,M16,M19,M18,M21,M22,M23,M24,M25,M26,M27,M28,M29,M30}
@attribute User{U01,U02,U03,U04,U05,U06,U07,U08,U09,U10,U11,U12,U13,U14,U16,U18,MM}
@attribute CPUusage numeric

@data

M01 U01 10922
M02 U02 10992
M03 U03 10992
M04 U04 10992
M05 U05 10992
M06 U06 10992
M07 U07 10992
M08 U08 11046
M09 U09 10992
M10 U10 12098
M11 U11 12098
M12 U12 12098
M13 U13 12098
M14 U14 8091
M16 U16 10992
M18 U18 10581
M19 MM  10883
M21 MM  8526
M22 MM  11814
M23 MM  14666
M24 MM  12226
M25 MM  10862
M26 MM  12355
M27 MM  3932
M28 MM  12540
M29 MM  14965
M30 MM  12450
```

Figure 22

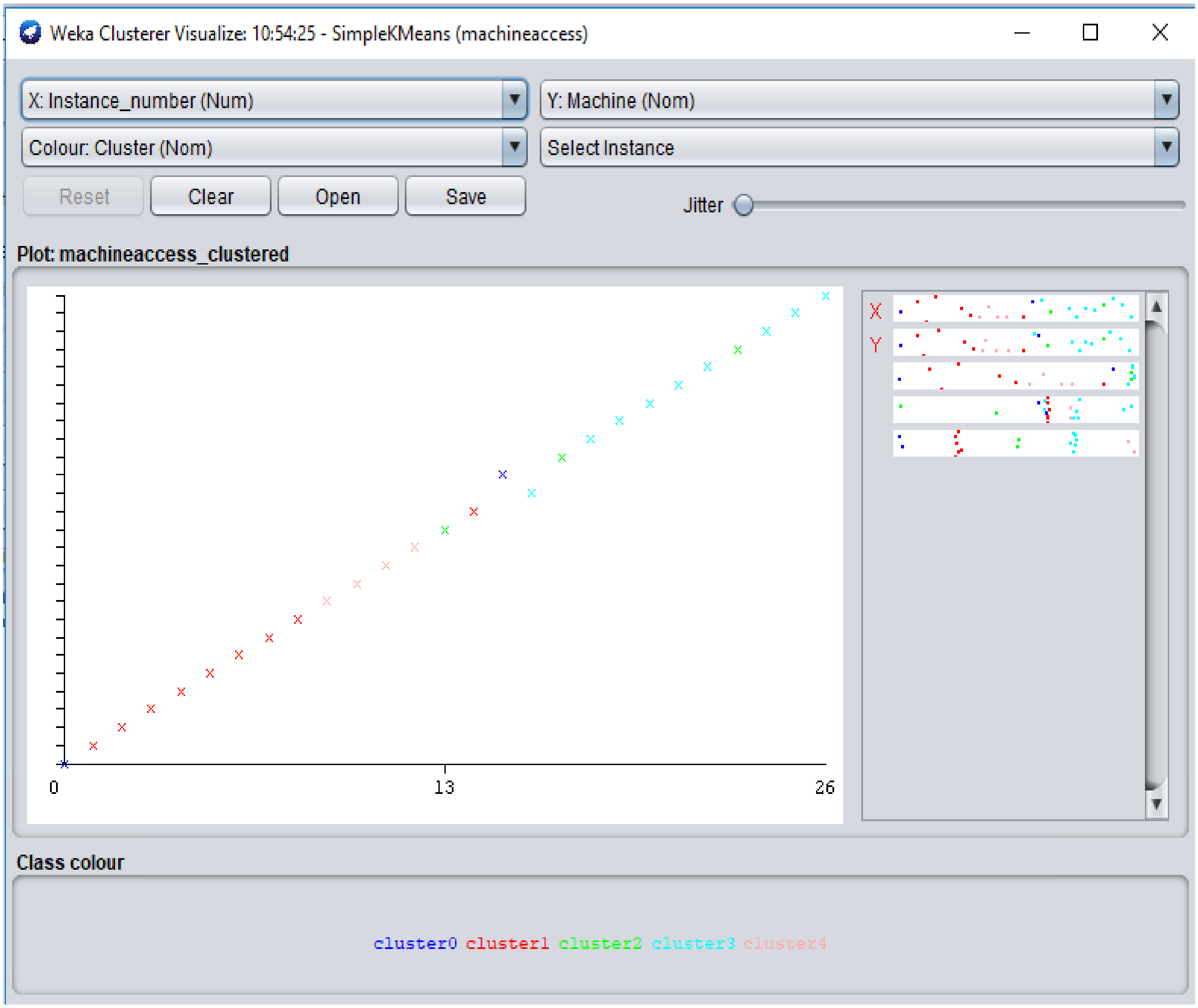
The Machine access data is then changed to the attribute-relation file format. Figure 22 is the notepad++ format for the machine access data. This file will be loaded to Weka tool for further analysis.

The Figure 23 is the output for the SimpleKMeans algorithm for the Machine acess data which has the Final cluster centroid values mentioned in the figure along with the clustered instances.



```
Cluster 4: M10,U10,12098

Missing values globally replaced with mean/mode

Final cluster centroids:
                             Cluster#
Attribute      Full Data            0           1           2           3           4
                  (27.0)        (2.0)       (9.0)       (3.0)       (9.0)       (4.0)
==================================================================================
Machine              M01          M01         M02         M14         M19         M10
User                  MM          U01         U02          MM          MM         U10
CPUusage      11192.1111      10751.5       10998   6849.6667       12529       12098




Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0        2 (  7%)
1        9 ( 33%)
2        3 ( 11%)
3        9 ( 33%)
4        4 ( 15%)
```

Figure 23



Weka Clusterer Visualize: 10:54:25 - SimpleKMeans (machineaccess)

X: Instance_number (Num)

Y: Machine (Nom)

Colour: Cluster (Nom)

Plot: machineaccess_clustered

Figure 24

The obtained results of SimpleKMeans from Weka tool are then visualized in the form of a graph with the instance number on the x-axis and the Machine id on the y-axis. Figure 24 indicates the visualization results of the SimpleKMeans algorithm.

From our understanding of typically understanding Data Mining, we have considered Data Preprocessing, application of Association rules and clustering of data. By doing all the mentioned phases, the given raw data can be pruned and will be visible to find patterns.

The procedure that is mentioned previously is associated with data preprocessing by using OpenRefine, Microsoft Excel and then by using Notepad++ we converted the data into Attribute-Related file format. We then used Weka, the Data Mining tool to apply SimpleKMeans clustering algorithm to see if the patterns that we have found manually are real and logically proven. Hence, clusters of data are then shown with bunch of users in one cluster which indicates that the patterns that were found are true.

Since we are done with pre-processing of data as shown in the above processes, we will now look at the Association rules of data and see if the accuracy with data is matching our requirements.

Association Rules

Association rules are applied after the data is preprocessed on individual cases. There are 6 different cases that are considered in this report and there are 6 different tables that are prepared for each case in OpenRefine. The Association rules are applied on all cases below.

Login pattern

- If user login at 131010
  User logout at 202040
  Then maximum number of user process is 30

  Support - 4
  Coverage – 5
  Therefore, Accuracy = 4/5 (100) = 80%

Program access

- If the user U09 runs program file
  The execution time for that file is 000340
  Then the program is LP095

  Support - 7
  Coverage – 7
  Therefore, Accuracy = 7/7(100) = 100%

File access

- If user start using a file at 115201
  The execution time for that file is 001040
  Then file used is F0270
  Support - 4
  Coverage – 4
  Therefore, Accuracy = 4/4 (100) = 100%

Printer access

- If user U19 used
  the machine M19 used
  Then Printer PR6 is used

  Support - 9
  Coverage – 10
  Therefore, Accuracy = 9/10 (100) = 90%

E-mail access

- If Machine M04 is used
  Jones(jones@pqr.com) email is used
  Then the email is Sent to the user

  Support - 11
  Coverage – 11
  Therefore, Accuracy = 11/11 (100) = 100%

Machine usage

- If Machine M24 is used
  On 090508 which is weekday
  Then the user login at 181540

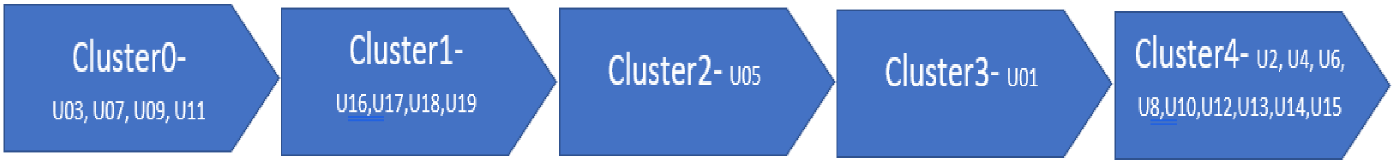
  Support - 4
  Coverage – 5
  Therefore, Accuracy = 4/5 (100) = 80%

Clustering Techniques

After we have generated attribute-relation file format for the six different cases that are preprocessed and when the file is run in Weka using the SimpleKMeans Clustering algorithm. The K-means clustering results from Weka tool are already discussed in the respective cases but the users or the clusters that are formed are depicted below for the individual case that is considered.

While working with Weka tool, each cluster is given with different colors so the cluster elements i.e., users can be differentiated.

Login Pattern



Cluster0- U03, U07, U09, U11

Cluster1- U16,U17,U18,U19

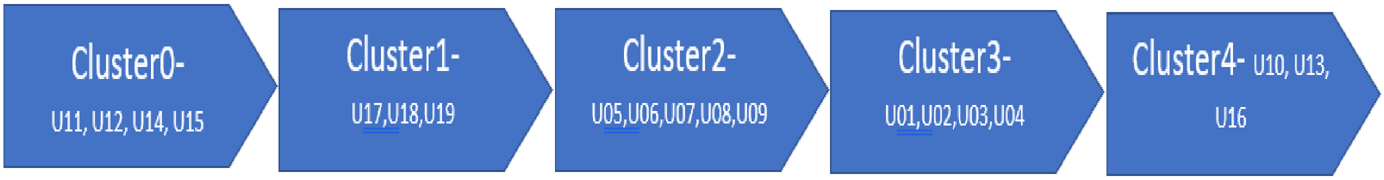
Cluster2- U05

Cluster3- U01

Cluster4- U2, U4, U6, U8,U10,U12,U13,U14,U15

In the Login pattern, the above figure shows 5 clusters that are taken in the SimpleKMeans and the respective Users are placed in the cluster to symbolically show that those users belong to that cluster.

Program Access Pattern



Cluster0- U11, U12, U14, U15

Cluster1- U17,U18,U19

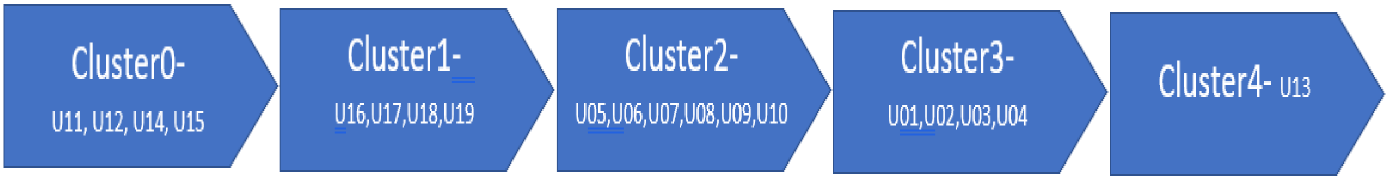
Cluster2- U05,U06,U07,U08,U09

Cluster3- U01,U02,U03,U04

Cluster4- U10, U13, U16

In Program access pattern it is shown in the above figure that there are 5 clusters that are formed because of the SimpleKMeans algorithm and the respective users are placed in the clusters. It is different for all the cases mentioned based on the data that is given as input for the SimpleKMeans algorithm.

File Access Pattern



Cluster0- U11, U12, U14, U15

Cluster1- U16,U17,U18,U19

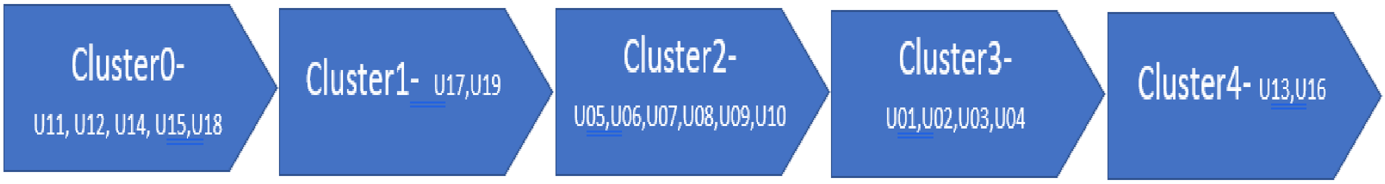
Cluster2- U05,U06,U07,U08,U09,U10

Cluster3- U01,U02,U03,U04

Cluster4- U13

This is like the other access patterns and it has 5 clusters with respective users in the clusters. The users are clustered in their respective clusters based on the data they had and the way their file access is being done.

Printer Usage Pattern



Cluster0- U11, U12, U14, U15,U18

Cluster1- U17,U19

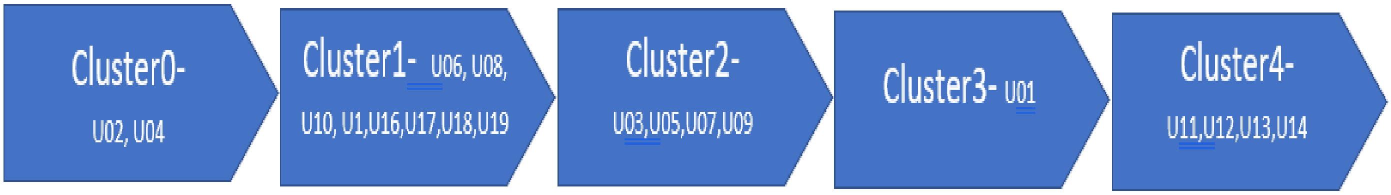
Cluster2- U05,U06,U07,U08,U09,U10

Cluster3- U01,U02,U03,U04

Cluster4- U13,U16

The printer usage pattern is no different from other cluster diagrams that are plotted. It also has 5 clusters with their respective users in them.

E-mail Access Pattern



Cluster0- U02, U04

Cluster1- U06, U08, U10, U1,U16,U17,U18,U19

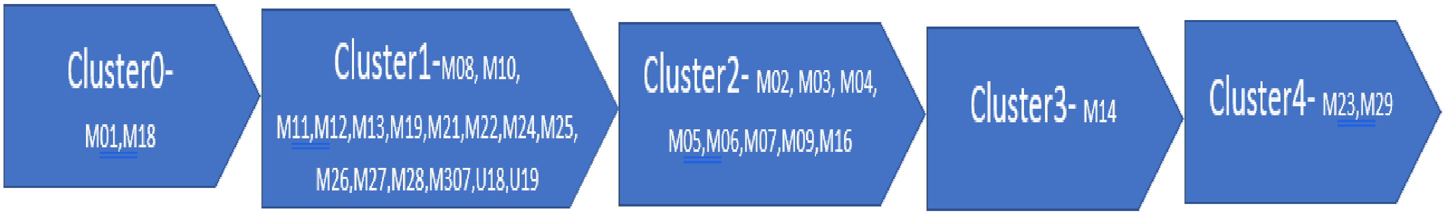
Cluster2- U03,U05,U07,U09

Cluster3- U01

Cluster4- U11,U12,U13,U14

E-mail access is like the other patterns that are mentioned with 5 clusters which has different users that belong to those clusters.

Machine Usage Pattern



Cluster0- M01,M18

Cluster1-M08, M10, M11,M12,M13,M19,M21,M22,M24,M25, M26,M27,M28,M307,U18,U19

Cluster2- M02, M03, M04, M05,M06,M07,M09,M16

Cluster3- M14

Cluster4- M23,M29

The Machine usage pattern is different from other clusters that were shown before, but the machines are clustered in the clusters instead of the users in this case.

Conclusion

From analyzing the given data about the historical login and access data for all the 19 users from a department, we have observed some key findings among the data.

Login access pattern:

Most of the users are working during the weekdays except the users U18, U19 who are working only on weekends. Only few users have login time at 8 and logout time as 18. But majority of the users has multiple login and logout timings.

Program access pattern:

Here the key finding is that a set of files are being accessed by the users. For instance, LP10, LP50, LP80 are marked as L1 in our observation and L1 is being accessed by U01, U04, U07. This is just a simple example of our finding and there are much larger sets of files that are being accesses together by multiple users.

File access pattern:

The file access pattern is like that of the program access pattern where there are set of files that are being accessed together. For example, F10, F20 and F25 are marked as F2 in our data representation. And the file F2 is used by U05, U06, U07, U08, U09 and U10. And there are more examples like this where the users are using similar set of files together.

Printer access pattern:

Since the printer has only few six types of data and most notable observation is that users who are using printer PR2 are higher and the ones who are using this printer are also using the File set F2 from out notation.

Email access pattern:

It is observed that the most used E-mail program is E1 and it is nearly used by 80% of the users. The email that the users have been using are used by the multiple users. But the users U02 and U04 are using two different emails for communication.

Machine usage pattern:

While the machine usage was considered by the machine id and it is observed that 2/3rd of the machines are strictly used by single users and the rest of the machines are used multiple users. It is also keen that the maximum CPU usage is observed in the multiple usage machines M23, M29 which has the highest CPU usage out of all the machines.

Acknowledgement

Since it has come to an end to our course with this project report submission, we would like to take a chance in thanking out instruction, Dr. Ravi Mukkamala for his guidance and the knowledge that he has shared with us in training and helping us with any difficulty we have faced. His quick response for the questions we had through any mode of communication that we have approached him. For this, we would like to express a deep sense of gratitude to our instructor for providing us a huge amount of precious time and effort for us.

References

- https://ieeexplore.ieee.org/xpl/topAccessedArticles.jsp?punumber=69
- https://ieeexplore.ieee.org/document/7944072/
- https://ieeexplore.ieee.org/document/8078809/
- https://acadpubl.eu/jsi/2017-117-20-22/articles/20/68.pdf
- https://www.cs.indiana.edu/~predrag/classes/2010springi211/week6_m.pdf
- https://www.researchgate.net/publication/288825433_STEP_BY_STEP_DATA_PREPROCESSING_FORDATA_MINING_A_CASE_STUDY