Lab 1 assignment solution

The objective of lab 1 assignment is to practice data pre-processing that includes: integration, conversion and identification of the useful features for data mining. A software tool will be developed to automate the data pre-processing.

The details of the MATLAB code are presented below to perform the data preprocessing.

```
%% Read the Excel file
%% the function xlsread will read the Excel file and will return 3
     variables:
   %"Numbers variable" contains all the numeric data in the Excel file.
     It contains only the entries, without the title row located in row
     1 in the Excel file)
   %"Text" contains only the text values. It has the dimensions of the
     data found in the Excel file.
   %"AllData" saves all the numeric and text data in cell format

[Numbers,Text,allData]=xlsread('appmon_1.xls');
```

```
%%Initialization
% These parameters are needed for future use in the program, thus they
need to be initialized first.

%a%Find the column of the attribute, the following function will return
the column number that contains the selected attribute

Column_number_mouseclicks= find(strcmp(allData(1,:),'mouseclicks'));

Column_number_keystrokes= find(strcmp(allData(1,:),'keystrokes'));

Column_number_mSec_from_start= find(strcmp(allData(1,:),'mSec from
start'));

Column_number_focus_app_name=
find(strcmp(allData(1,:),'focus_app_name'));

Column_number_focus_app_title=
find(strcmp(allData(1,:),'focus_app_title'));
```

```
Column_number_opened_windows=
find(strcmp(allData(1,:),'opened_windows'));

Column_number_mousemoves= find(strcmp(allData(1,:),'mousemoves'));
Column_number_start_time= find(strcmp(allData(1,:),'Start time'));
%b% Find the total number of rows in the Excel file
Total_number_of_rows= length(allData);




%c% Find the Start time in the excel sheet.
To be able to use the started time "08:07:50 AM", I added it to the
input excel file in the row 1, column 12. The software should read this
time and add the value of the mSec from start. The value will be then
saved in a new column under a new attribute named "actual time".

The following function will return a double number representing the
start time in MATLAB

Start_time = allData{2,Column_number_start_time};




%d% Add columns to accommodate new extracted features.
% Add Actual time feature, window switch, number of opened windows and
mousemoves2 categorized.

allData{1,Column_number_start_time+1}='Actual time';
allData{1,Column_number_start_time+2}='Window Switch';
allData{1,Column_number_start_time+3}='Number of opened Windows';
allData{1,Column_number_start_time+4}= 'Discretization of mousemoves';




%e % Save their column number

Column_number_Actual_time= find(strcmp(allData(1,:),'Actual time'));

Column_number_Window_Switch= find(strcmp(allData(1,:),'Window
Switch'));

Column_number_number_opened_windows= find(strcmp(allData(1,:),'Number
of opened Windows'));
```

```
Column_number_mousemoves2= find(strcmp(allData(1,:), 'Discretization of
mousemoves'));
```

% Start Data Preprocessing

```
% For every entry or row, the program will be able to check perform
data clean up, data transformation and feature extraction. Data
preprocessing will be performed for every row by implementing the
requested 1-6 steps in the Lab0 assignment. These functions will be
repeated for each row, starting row 2 where the entries values begin
till the total number of entries.


for i=2: total_number_of_rows % the entries start at row 2. Row 1 is
the title of each column.



%1% Data Clean up ==> If mouseclicks[i] > 200 it is an outlier and
     should be set to 0. Since "Numbers" array contains only entries,
     the index of the entry will be (i-1) in the Numbers array
    if(allData{i,Column_number_mouseclicks}>200)
        allData{i,Column_number_mouseclicks}= 0;
    end



 %2% Data Clean up ==> If keystrokes[i] > 100 it is an outlier and
     should be set to 0
    if(allData{i,Column_number_keystrokes}>100)
        allData{i,Column_number_keystrokes}= 0;
    end



 %3% Feature extraction ==> Extract the actual time at row [i] knowing
     that the experiment started at 08:07:50 AM.

     %The start time was included in the Excel file in row 2 column 12.
     %The MATLAB can read the Start time from Excel by
     (Start_time =allData{2,Column_number_start_time}) as a number of
     type double that represents the fraction of the day and can be
     converted later to a desired format 'HH:MM:SS AM or PM'

     % For instance:
          % The number 0 represents the time '12:00:00 AM'
          % The number 0.3388 represents the time '08:07:50 AM'
            according to the following computations:
```

```
The number of seconds equivalent to '08:07:50 AM'= 8*3600+
7*60+50= 29270 seconds

The number of seconds in 24 hours = 24*3600 = 86400 seconds.

Thus, the ratio or the fraction of the day representing the time
'08:07:50 AM' is equal to 29270/86400 = 0.3388 (the number that
was read by MATLAB from EXCEL)

Accordingly, the number of "mSec from start[i]" can be represented
as a fraction of the day and added to the above number 0.3388 that
represents the start time in MATLAB. The number will be then
converted into the time format and saved in a new column under
"Actual time" feature.


% Find "mSec from start[i]"
duration_from_start= allData{i,Column_number_mSec_from_start}



% Find ratio "mSec from start[i]" fraction of the day
duration_from_start_fraction= duration_from_start/ 1000/(24*3600)



% Find the actual time by adding the start time to the fraction
above. The fraction can be found by diving by
(1000msec*24hours*3600 seconds)
Actual_time= Start_time + duration_from_start_fraction



% Save the number actual time in the database under feature Actual
time in format 'HH:MM:SS AM or PM'

allData{i,Column_number_Actual_time}=datestr(Actual_time,14);



%4% Feature extraction ==> Extract a Window Switch feature that is set
to 1 whenever focus_app_name[i]!= focus_app_name[i-1] and
focus_app_title[i]!= focus_app_title[i-1], and that is set to 0
otherwise.

%Assume the first entry located at row [2] has a window switch=1;
```

```
    if (i==2)
       allData{i,Column_number_Window_Switch}=1;

    else %for other rows greater than 2 check the above conditions

         if( ~strcmp(allData(i, Column_number_focus_app_name),
         allData(i-1, Column_number_focus_app_name)) &&
         ~strcmp(allData(i, Column_number_focus_app_title), allData(i-
         1, Column_number_focus_app_title)))

           allData{i,Column_number_Window_Switch}=1; %conditions true

         else

           allData{i,Column_number_Window_Switch}=0; %conditions false
         end

     end
```

```
%5% Feature extraction ==> Extract the number of opened windows, by
    parsing opened_windows[i]
    % The opened_window feature is a set if opened windows delimited
    by   "|". Thus, the number of opened windows will be equal to the
    number of      "|" in opened_windows[i], the value will be stored
    under 'Number  of opened Windows' feature.

    % The below function will find the delimiter "|", find the number
    of times it was repeated in opened_windows[i], and store the
    number under 'Number of opened Windows' feature.


   allData{i,Column_number_number_opened_windows}=
length(findstr(allData{i,Column_number_opened_windows},'|'));
```

```
%6% Feature extraction ==> Extract a (discretization) categorization
    of mousemoves[i] according to the following categories:
   %a. No Move, if value is equal to 0
   %b. Slow, if value is greater than 0 and less than 36
   %c. Moderate, if value is greater or equal to 36 and less than 55
   %d. Fast, if value is greater or equal to 55
   %The extracted feature will be saved under a new column
    "mousemoves2"

   % a. No moves, value equal 0
```

```
    if(allData{i,Column_number_mousemoves}==0)
        allData{i,Column_number_mousemoves2}='No Move';
    end

    %b. Slow, if value is greater than 0 and less than 36
    if(allData{i,Column_number_mousemoves}>0 &&
allData{i,Column_number_mousemoves}<36)
        allData{i,Column_number_mousemoves2}='Slow';
    end

    %c. Moderate, if value is greater or equal to 36 and less than 55
    if(allData{i,Column_number_mousemoves}>=36 &&
allData{i,Column_number_mousemoves}<55)
        allData{i,Column_number_mousemoves2}='Moderate';
    end

    %d. Fast, if value is greater or equal to 55
    if(allData{i,Column_number_mousemoves}>=55)
        allData{i,Column_number_mousemoves2}='Fast';
    end



end  %% End the for loop (for i=1:total_number_of_rows)
```

%7% Delete unused features

```
% Pick the features are needed and were used when performing data
preprocessing.
%The features are: “mouseclicks”, “keystrokes”, “Actual time”, “Window
Switch”, “Number of opened windows” and “Discretization of mousemoves”.

Needed_Columns=[Column_number_mouseclicks, Column_number_keystrokes,
Column_number_Actual_time, Column_number_Window_Switch,
Column_number_number_opened_windows, Column_number_mousemoves2];
```

% Write the Data Preprocessed in a New Excel file

```
 xlswrite('appmon_1_out.xls',allData(:,Needed_Columns))
```