
THE UNIVERSITY
of ADELAIDE
EXAMPLES OF MILESTONE 1
adelaide.edu.au
seek LIGHT

Milestone template

MCI project First Milestone Report

Team number :
Project Title:

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan .	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?		

Milestone template

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

Supervisor assessment	Please, rate your team (1) effort, (2) project progress and (3) their self-reflection for milestone 1 Rating scale 1-10 as per standard marking scheme, ie 5 is a Pass and 7 is a credit. Add some comments to explain your rating
Effort: Progress: Reflection:	

Download Milestone Template

- Link to download the Milestone template. You can find the link in Week 6

https://myuni.adelaide.edu.au/courses/50165/files/5864423?module_item_id=1710419

- Video about how to fill the Milestone template up. Click on the video named Preparing for Milestone 1

https://myuni.adelaide.edu.au/courses/50165/external_tools/82

Example 1

MCI project First Milestone Report

Team number: 4

Project Title: Building a Machine Learning Classifier to Predict Edits to Stack Overflow posts

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan.	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
Fetch the posts information of Stack Overflow from Google BigQuery and feed them to the five specified machine learning algorithms, SMO, Naive Bayes, J48, Random Forest and KNN to predict whether the posts will be edited or not. Based on the existing attributes, extend or generate more attributes to make the prediction more accurate.	To know the structure of dataset and download it. And be familiar with Weka Gui.	Be able to access the data on Google BigQuery and to use Weka to classify simple datasets.	Same as planned
	Processing Date type and String type attributes to fit the requirement of the specific machine learning algorithms.	1. "Date" type attributes are converted to numeric type. 2. "String" type attributes are converted to nominal type.	Same as planned
	Employing the 5 machine learning algorithms(SMO, IBK, NaiveBayse, RandomForest and J48) on the dataset.	In each algorithm, we change a little bit of the attributes and parameters to see how those changes reflect on the prediction results.	Same as planned
	Extend more attributes from the	Employ the attribute filters and other	Same as planned

Example 1

	text content of the posts.	techniques to quantify the text content of the posts. (i.e. String Vectors, Readability Metrics)	
Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?		

Question1 :
Our team communicate through meetings, E-mail and the communication app. Regular meetings with the supervisor are held on Tuesday, we meet the supervisor at 3:30 pm and after that, we have a team meeting. We update what we have done so far, ask for suggestions to the problems we encountered in the meeting with the supervisor. We exchange some ideas and double check the tasks of the current week in the team meeting. According to the issues discussed in the meeting, we may adjust the plan or reassign the tasks.

At the first meeting, the supervisor suggested that we can divide the project into two parts, one is about the application of Weka and another one is about the dataset processing. After the discussion, we decided that Shuai take the Weka part, and Cheng-En will be responsible for the data part.

We employ five algorithms in this project. Shuai learned the basic principles with SMO, KNN and Naive Bayes and Cheng-En learned J48 and RandomForest. We integrate what have we learnt, so we can know how to fit the data to each algorithm.

As part of the deliverables of the first milestone, we have to automate the process of reading the data and classify the data by Weka. Cheng-En writes a script to get and process the data and Shuai writes a script to run the algorithms automatically. Finally, we present our training result in an Excel form.

Example 1

Question2: The outcome of each activity go along with the plan mostly. One minor difference is that we were not going to add the Readability parameters at the beginning. Yet, we found that the Readability parameters could be some very useful attributes, so we finally decided to add them to our features.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

We encountered a problem with this project. The most difficult problems in our project were the String data type. The first time we inputted data into Weka, almost all the classifiers could not deal with the String type data. Then, the supervisor suggested that we can use StringtoVector Filter in Weka. The StringtoVector splits the content of the specified string, counts the number of occurrences of each word in the string and turns them into features. However, even the String was converted into a numeric type, words like some HTML syntax tags was still inefficient to the machine learning algorithms. Thus, we had other two steps to manage the problem. First, we analysed the text content of the post body, separate it into three parts, the HTML tags, readable text and code snippets. Then, we employed the StringToVecotr filter again and applied the readability metrics on the readable text in order to get more informative attributes. Finally, we try different attribute combinations to train the machine learning algorithms and get the highest F-Measure we have got so far (0.69).

Example 2

MCI project First Milestone Report

Team number: 14
Project Title: Debugging, user testing and additional features for a tutor booking system

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan.	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
The first task for our project is a full testing of the existing system to make sure that everything is functioning and documenting any issues in the issues section of the project GitHub repository.	1. Learning PHP and doing PHP exercises.	1. Learning theory and doing exercises from W3School.	1. Same as planned.
	2. Understanding and researching this system.	2. Checking every feature in the existing tutor booking system.	2. Same as planned
	3. Testing.	3. Writing testing report.	3. Same as planned
	4. Analysing the codes	4. Analysing 70% of codes and asking questions for codes.	4. Same as planned
Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?		

1. Our project progressed smoothly and the plans were completed on time. Firstly, we tested every feature in the tutor booking system. Secondly, we recorded the process of our test in detail and highlighted each bug in red colour. Thirdly, we defined severity and relation of each bug and explain severity and relation in our testing report. Finally, in order to make an appropriate foundation for our milestone 2, we did PHP exercises every week from W3School, submitted our practice codes into 'Draft-codes' branch in our GitHub, downloaded codes and asked any issues of codes for supervisor.

2. No. There are several reasons why we can finish milestone 1 smoothly. First and foremost, each team member plays his role and fulfils his responsibilities seriously. Additionally, our team implements the project in strict accordance with the plan of our feasibility analysis. Last but not least, for each task, we got useful advice from our supervisor.

Example 2

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?
Yes. We have encountered problems during our project. Every week, we would summary all issues and then write them in our client meeting agenda. In every client meeting, we would ask our supervisor for any impediments. After client meeting, we would write client meeting minute to summary the feedback from the supervisor and address our issues according to supervisor's feedback as soon as possible.	

Example 3

MCI project First Milestone Report

Team number: 03
Project Title: A Julia package for Parsing and Visualizing Tobii Pro Glasses 2 Eye-Tracking Data

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan.	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outrow the actual outputs compared to what was projected (or type "same as planned")
• GUI creation • Parsing of JSON files • Fill the data of the Jason file into the data frame • Generate a CSV from the data frame • Create a test file	Create a GUI to convert raw eye tracking data	Select the local file from the pop-up window of the "open" button, and convert the data	Convert files by entering the path to the target file
	Parse the JSON file to enter the data into the data frame	Convert a JSON file to a dictionary type file	same as planned
	Input the parsed dictionary file into the data frame	Create a data frame containing eye tracking data	same as planned
	Generate a CSV from the data frame	Generate a CSV file locally to display the sorted eye tracking data	same as planned
	Create a test file	Use the selected data to detect if the correct data is stored in the CSV	same as planned

Example 3

Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?

1. Project progressing

Overall, the progress of the project corresponds to our expectations. In the first to third weeks, we learned the basic syntax and knowledge of Julia. From the fourth week, we started to enter the stage of writing code. In the process of writing the code, although we encountered a lot of problems and difficulties, fortunately, we finally solved most of the problems and correspond with our planned progress. Therefore, we are satisfied with the progress of our current project, and we hope that we can continue to work hard to complete the tasks on time and with high quality.

- We have completed the framework and some functions of the GUI, which will help users to convert and read data through an interface.
- We convert the JSON file into a dictionary file that can be easily read and create a data frame and lay the foundation for the next step.
- We fill all the eye tracking data into a data frame to make it readable.
- We convert the data frame to CSV and save it locally so that the data can be read and edited easily.
- Finally, we used a test file to test whether our output is correct.

2. The differences between projected and actual outputs/outcomes

- Regarding the GUI, our outcome and the planned outcome are not consistent. In the plan, we expect to pop up a window by clicking the "open" button, allowing the user to select the JSON file that they wish to be converted. However, since the “CImGui” package in Julia does not currently support pop-up operations, we have to choose to compile the file by directly entering the path for the target file. Although the GUI interface is inconsistent with expectations, the outcome is same.
- The column names in the CSV are not same as the projected. We changed the column names to make them readable. In the plan, we hope that the final names are "gd_x", "pd_l", "ac_x", etc. Finally, our column names are "GazeDirectionLX", "PupilDiaL", "AccelerometerX" and so on. This will help the user understand the specific meaning of each column.
- In the original plan, we expect to put two rows at each point in time, which represent the data for the left and right eyes. However, in our outcome, we let all the data at the same time stamp have only one row, which is essential for milestone 2 and later data analysis.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

1. The first problem we encountered was the inability to pop up a window in the GUI. The reason for this problem is that we did not have a deep understanding of GUI in Julia when setting goals, which led us to find this problem in the process of programming. Therefore, our solution is to compile the file by directly entering the path for the target file. We looked at a lot of information and made a lot of attempts to achieve this outcome finally. Although this brings a little inconvenience in use, the final function is the same.
2. The second problem we encountered was that the parse JSON file was only parsed one row because the function of the “JSON” package conflicted with the data type we read, because our raw data is not a standard JSON file. Therefore, our solution is to replace the “LazyJSON” package to compile the file. In the end, we solved this problem.
3. The third problem we encountered was the inability to combine multiple data frames into one frame. All data frames are continuously generated in a for loop, but since the number and name of column names in each row are different, our idea is to enumerate all possible column names and set empty values. Assigning a value of 0 ensures that the column names of all rows are the same. But unfortunately, we spent 2-3 days trying, but all failed because we could not assign values to empty column names. Finally, after reviewing the information online and continually working, we changed the “LazyJSON” package back to the JSON package. In addition, we selected the vcat() in the join(), push!(), append!() and vcat() functions because it can ignore the order of the columns, and finally we solved the problem.
4. The fourth problem we encountered was the sorting problem of column names. It will automatically sort all column names by alphabet when using the Dict() method, but we can't modify the order of the columns in the data frame. We consulted our supervisors on this issue, and through his guidance, we found a solution to this problem. We created a new empty data frame at the beginning and ordered the columns in order, and then we merged the generated data frames into the completed frame. In the end, this problem was solved.

Example 4

MCI project First Milestone Report

Team number : 2
Project Title: A Julia package for Electrodermal Activity Analysis

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan .	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
In milestone 1, we plan to build a basic GUI. The GUI allows users to select data files that they want to import to the program. Also, the GUI has a function to show the plotted data in the line chart. And we plan to build a function to synchronize different frequency data.	Installing CImGui package by the instruction given by Dr. Zygmunt Szpak. Using CImGui package to build a GUI window with a checkbox and a slider. We read the source code of demo.jl from Github and we want to figure out which functions needed in our GUI.	Create a basic GUI	Same as planned.
	Using CImGui package to build a text box. This box can allow users to select data files. We read the GimGui manual to figure out which API we want to use.	Allow users to select files that they want to import	Same as planned. And did some modifications according to the client's requirements.
	We use DataFrames package to import data files. And then we use Gadfly to plot data. We read Gadfly manual to study how to use this package.	Plot data and show in the GUI	Same as planned.
	We build a function to synchronize the different frequency data.	Build a function to synchronize different frequency data	Because the GUI package we used (CImGui) does not provide two dimensions plot function. We moved this part to milestone 2 under supervisor's advice.
	Not mentioned in draft plan	Not mentioned in draft plan	We added a new function which allow users to slide the line chart.

Example 4

Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?

In general, we progressed as our plan. We finished most of the tasks in milestone 1 although many difficulties are in the progress. We have not executed the synchronize function and decided to put it to our milestone 2. CImGui does not provide multi-dimensional plotting, we changed to Makie according to the client's suggestion, and then found that Makie and our CImGui original GUI conflict, cannot be displayed at the same time. We are going to do this synchronization in milestone 2, and then follow the client's suggestion, and use CImGui's plotlines package which is modified by ourselves to implement this function. We added a new function in our GUI, and it can allow users to slide the line chart. We read the document of CImGui package and found an API to execute this slider. And then we moved the synchronize function to our milestone 2 plan. Therefore, we did not finish all tasks in our draft plan. Apart from the synchronize function, we did all tasks the same as planned. Also, we did some modifications to our plan according to the client's requirements. And we decided to change our GUI under client's suggestions. We would build a more complicated file dialog to select files.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

The first difficulty is that we are not familiar with the Julia language. We need to spend a lot of time to read the Julia document to find the solution. To manage this problem, we worked face to face for many times, and helped each other while programming. It is effective and efficient because we faced some same problems and one of us might have a solution, and we did not need to waste time in repeated questions.
The second problem is that we need to use CImGui packages that we have never used. It is a big challenge for us because some of them do not have enough support documents. When we were implementing the basic GUI, Dr. Szpak provided a blank GUI to help us. And when we were implementing functions in the blank GUI, we read the source code of the CImGui package to find APIs we needed. CImGui did not have the built-in file browser function, and we had to make effort to implement one.
The third difficulty is that plotting csv data. To manage this problem, we break it to three steps. The first step was to transfer csv file to dataframe. In this step, we used dataFrames package. And the second step was to transfer the dataframes to an array. The final step was to plot the array into the GUI. In this step, we changed the package we used. We used Makie package to plot interactive figures. And then after tests, we found that Makie and CImGui have some conflicts and the plotted figures cannot display in our GUI. Therefore, the only way we can do is that we make some changes in CImGui package's built-in plotline function.
The fourth problem is time management. To manage this problem, we need to follow the plan step by step. We made a detailed schedule which contains daily tasks. And we all followed the task list and finished the required tasks on time.
Dr. Szpak gave many suggestions to our project which helps us to handle difficulties.

Example 5

MCI project First Milestone Report

Team number : team23
Project Title: A web-based environment for running code and tasks in containers

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan .	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
In the first milestone plan, we should implement two main functions. Firstly, When a user clicks the start button in the web interface, the server side run the code in containers and then the web interface shows the feedback of the code. The second function is to click stop button and then the container is deleted.	1. Building the interface	Design two buttons and a text area in one webpage	Complete one more button which can choose the file to upload, the other parts are same as planned. The code are included in the html1.html and html1.css
	2. Create a server	Using express framework to create a server that can receive request and reply response	Same as planned
	3. Connecting the container	Using the docker remote API implements the connection with the docker server and create the container. Using AJAX implements that client can call server function Users can build and delete the container through the interface button	The user can upload the dockerfile to build a container and delete the container.
	4. Unit testing	Beginning to learn how to do the unit testing of html and javascript and design some tests for milestone1.	Finish two parts (create container and output results functions).
	5. Integration testing	Beginning to learn how to do the integration testing of html and javascript and design some tests for milestone1.	Doing the AJAX tests using the mockjax and doing the javascript tests using the mocha.

Example 5

Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?

We have completed the milestone 1. We planned to connect the container using the docker remote API and using AJAX implements client call to server in milestone1. The projected outputs are to carry out two functions. The first function is to click start button to build container and get feedback in the interface. The second function is to click the stop button to delete the container. After we finished projected outputs, we added one more function in actual outputs, uploading a dockerfile.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?

We have met two major problems.

1. We didn't know how we started the project in the first few weeks, because we had no experience in building web.
2. We have some problems when we connected front-end and back-end.

For solving these problems:

1. We enrolled one web course, which is web and database computing, to learn about how to design web. This course gives us a guidance on how to start our project. Meanwhile, we also try to get relative knowledge from some public courses like Udacity.
2. In the connection problem, we wrote a document to separately regulate the input and output in the front-end and the back-end. At the same time, we coded together when we were in debug part, which can make us communicate errors more smoothly and directly.

Questions?