

Behaviour analysis through Machine learning techniques

Raphael Sculati

Bachelor's Thesis
Business Information Technology
15.07.2015



Table of Contents

| | |
|---|----|
| Table of illustrations | 4 |
| Acknowledgements | 5 |
| Abstract | 6 |
| 1 Introduction | 7 |
| 2 Research question..... | 8 |
| 2.1 Related researches..... | 8 |
| 3 Methods | 9 |
| 3.1 Methodology..... | 9 |
| 3.2 Workflow processes..... | 10 |
| 4 Background..... | 11 |
| 5 Design..... | 11 |
| 5.1 Conceptual Architecture | 11 |
| 5.2 Generation of the simulated data | 12 |
| 5.3 Machine learning..... | 13 |
| 5.3.1 Choice of the algorithm..... | 14 |
| 5.3.2 Bayes classifier..... | 15 |
| 5.3.3 Utilisation of Bayesian Network | 16 |
| 5.4 Technologies..... | 17 |
| 5.4.1 Input data | 17 |
| 5.4.2 Machine learning software | 17 |
| 5.4.3 Output solution..... | 18 |
| 6 Implementation..... | 18 |
| 6.1 1st iteration | 18 |

| | | |
|-------|---|----|
| 6.1.1 | Generation of the data population | 18 |
| 6.1.2 | Pre-processing | 19 |
| 6.2 | 2nd iteration..... | 21 |
| 6.2.1 | Creation of the model..... | 21 |
| 6.2.2 | Training the data on the model..... | 22 |
| 6.2.3 | Evaluation of the model..... | 23 |
| 6.3 | 3rd iteration | 24 |
| 6.3.1 | Deployment of the solution..... | 24 |
| 6.3.2 | Machine learning implementation | 26 |
| 6.3.3 | Weka API – Java & .Net Interoperability | 27 |
| 6.3.4 | Console application test..... | 27 |
| 6.3.5 | WCF Service | 30 |
| 7 | Results | 31 |
| 7.1 | Current results..... | 31 |
| 7.2 | Future amelioration..... | 31 |
| 7.2.1 | Classifier model storage..... | 31 |
| 7.2.2 | Top ranking algorithm..... | 32 |
| 8 | Conclusion..... | 33 |
| 9 | References..... | 34 |
| 10 | Appendices | 37 |
| | Appendix : Nurses Survey..... | 37 |

Table of illustrations

| | |
|--|----|
| Figure 1 - Predictive applications | 9 |
| Figure 2 - CRISP/DM method: Intelligent data analysis processing | 10 |
| Figure 3 - Conceptual architecture | 12 |
| Figure 4 - Example of the created dataset | 19 |
| Figure 5 - MetaNode - pre-processing | 20 |
| Figure 6 - Pre-processing results | 21 |
| Figure 7 - Weka library | 21 |
| Figure 8 - BayesNet | 22 |
| Figure 9 - Partitioning | 22 |
| Figure 10 - Evaluation of the model..... | 23 |
| Figure 11 - Confusion matrix..... | 23 |
| Figure 12 - Full KNIME Workflow | 24 |
| Figure 13 - Detailed Architecture..... | 25 |
| Figure 14 - ARFF file example | 29 |
| Figure 15 - Top ranking algorithm..... | 32 |

Acknowledgements

First, I would like to thank my school's advisor of this thesis in Finland, Mr. Amir Dirin, for his advices and encouragements throughout this thesis project. I would also like to sincerely show my gratitude to my second advisor from Switzerland, Mr. David Wannier, Head of Business Information Technology department of HES-SO Valais/Wallis, who made this international exchange possible between our two respective universities.

To my two main colleagues of this project, Mr. Michael Dayen, for his effectiveness and work during this year and Mr. Nathaniel Ham for his help and great support.

I would show as well my appreciation to Mr. Juhani Välimäki, my international coordinator, and Mr. Jarmo Peltoniemi, Head of Business Information Technology department of Haaga-Helia, for their assistances and their encouragements.

Finally, I would like to thank, Pr.Dr Dominique Genoud and Mr. Yann Bochi, for their supports and for the possibility of being able to work with their team in the Institute of Information Systems, in Sierre.

Abstract

| | |
|--|---------------------------------|
| Authors Mr. Raphaël Sculati | Group mHealth project |
| Title Behaviour analysis through Machine learning techniques | Number of pages 38 |
| Supervisors Mr. Amir Dirin - School's advisor of the thesis in Finland Mr. David Wannier - School's advisor of the thesis in Switzerland | |
| <p>Behaviour analysis is the science of studying the comportment of a person to establish a specific profile about it. It has firstly been used in psychology and since a few years, it has been implemented in information technology programs to improve and suggest in different forms the content of an application for users. With the growth of artificial intelligence, it tends to become the new trend that gives the possibility for applications to be personalized and centred on the user's needs.</p> <p>Machine Learning is a subcategory of artificial intelligence and has the goal to develop solutions to implement automatic methods to make our computers capable of evolving by themselves. The activities and actions of users start to be analysed to determine rules that can be integrated to align software applications in parallel with the daily routine and comportment of a person.</p> <p>This thesis is part of a healthcare mobile application project (mHealth) that has for objectives to develop a management tools for the medical personal to administer the patients in the hospital. Moreover, this application would like to use the location of a user and his habits of utilisation of the software, to quickly provide information for the nurse and therefore, reduce human-machine interaction and save precious time for better purposes.</p> <p>These goals are starting to be feasible through the utilisation of correct technologies and technics. This thesis analyses the different data that can be provided and uses machine learning algorithm technics to study the behaviour of a user to predict his needs and suggest him content.</p> <p>Specifically, we simulate the comportment of a nurse to subsequently be construed by our machine learning solution. Thereafter, we provide the predicted content for the user via a Web Service.</p> <p>The solution that we have developed has a current accuracy of 75% and the model created with simulated data will progressively adjust itself with the real data in the healthcare environment.</p> <p>Key words Machine Learning, Behaviour analysis, WCF Service, Data mining, KNIME, Weka</p> | |

1 Introduction

The task of this thesis project is to create a solution based on machine learning and pattern recognition for analysing the behaviour and the habits of the different types of users of a mobile application to provide row data to suggest useful content for the user.

Currently we have the idea of the advantages that can offer this kind of algorithm and services for the medical personal (e.g., reduce human-machine interaction, time saving and intelligent information), now we want to have a proof of concept. Furthermore, the future possible generic solutions could be integrated and used to several kind of environments.

The research results will be the solution that implements our model of the machine learning algorithm that will be used to send specific intelligent row data. This document will describe in detail the workflow and the processes of the data analysis. Firstly, we will study the background of the project and explain the choices of the different techniques, software and technologies to reach the goal of the project. After the design phase, we will try to implement the machine learning algorithm. We will work with different implementations to be able to gradually improve our solution. Finally, we will discuss about the possible results, the different points that can be improved and the future of the project.

There are more than a few points that could be learned during the realization of this project thesis. First of all, the different analyses that have to be done to understand machine learning concept before starting any kind of implementation. Secondly, based on the nature of the project, it will be also interesting to be able to wisely choose different kind of technologies and solutions that could fulfil our needs. There is also a collaborative aspect that is very important. Indeed, we are a few people on the project, thereby we will learn to collaborate between diverse actors to satisfy the final requirements. Finally, it will also be a challenge to produce a concrete solution in the respect of the given deadlines.

This project will cover the machine learning and pattern recognition aspects of the mHealth application. It is entirely present in the backend part of the application that will be described in the architecture section. Although, the management of the basic database and server won't be part of this thesis. Furthermore, the user interface part will be produced by other actors of the project.

2 Research question

The main research question of this thesis is to determine if it is possible to predict the future function that a user will need based on his previous location and past behaviour at a certain hour of a day.

2.1 Related researches

Behaviour analysis comes at first from psychology and was used to explain the comportment of a subject to establish rules and models about a person. With the growth of data and interactions of users with computer programs, couple of researches have been done concerning behaviour analysis in information technology.

At the beginning, the idea was to establish a profile of a user and suggest him information or content (mostly to sell products) as we could do it in marketing when we want to determine the profile of our future customers. Those first solutions that we can call “basic recommender systems”, didn’t use any machine learning technics (Note that machine learning will be explain in chapter 13) but were only based on some conditional rules. It was principally implemented into e-commerce website.

Few years ago, we saw the emergence of new recommender systems that build a list of recommendation for the user based on his past behaviour using machine learning algorithm. It is still mostly use for research content and social tags and has replaced the old engines for e-commerce websites.

(Cherry, s.d.) (Tim Jones, 2013)

Nowadays, the actual generation of applications are even more user-centred than before and tend to adapt their content to be unique and personalized. With the arrivals of smartphones, new engines have been developed and we have seen the apparition of new advance smart applications that use the same ideas and take into account a lot of different information to predict different elements. The next figure resumes the most advanced app to anticipate a person’s needs:

| NAME | Cue | Google Now | Osito | Tempo AI | Dark Sky |
|-------------|---|--|---|--|---|
| RAISED | \$4.7 million | N/a | \$1.1 million | Incubated at SRI International | \$39,376 |
| FOUNDED BY | Y Combinator graduates Daniel Gross and Robby Walker | An internal Google team | Bill Ferrell, a former Google developer | Raj Singh, Corey Hulen, and Thierry Donneau-Golencer | Jack Turner and Adam Grossman |
| PREDICTIONS | Summarizes a person's day based on information scavenged from calendar, e-mail, and documents | Directions, traffic, and weather based on a person's location and calendar | Handles transactions like checking in for a flight or calling a cab after you land at the airport | Directions to appointments. Also sends messages if you're running late | Provides minute-by-minute weather forecasts for user's exact location |

Figure 1 - Predictive applications

(Simonite, 2013)

3 Methods

3.1 Methodology

We will mainly work with different iterations along the project and try to increase the efficiency of the results progressively. The first part of the project will be focused on the establishment of a proof of concept that can demonstrate the feasibility of our solution. After verification of the achievability of this concept, we will try to deploy our solution to make it accessible to others components of our architecture's project.

Regarding to the project management methodology, we will follow a few Kanban¹ concepts, e.g. Scrum table, dashboard, and try to focus on programming. Furthermore, by having weekly meetings with the other parts of the team, we will improve the collaboration between the actors and try to stay updated about the advancement of the project.

¹ Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team. Like scrum, Kanban is a process designed to help teams work together more effectively. (What is Kanban? Kanban Software Tools, s.d.) (Morris)

3.2 Workflow processes

It is very crucial to follow a correct process from the starting point until the creation and the deployment of the solution. We will use during this thesis project, the CRISP/DM² (Cross Industry Standard Process for Data Mining) describes on the following figure.

Problem understanding: We need to consider the problem and how the solution should look like.

Data understanding: Take in consideration the amount, the availability, the quality of the data. If it is relevant to our current problem.

Data pre-processing: Clean to data to concentrate on what is really important, find the best way for modelling and increase the quality of the data.

Modelling: Finding the model that suits to our current problematic, what is the best method?

Evaluation: Answering the question, if our model fulfil our requirements.

Deployment: Understanding how the model should be deployed and establish that the model that we have is valid for real application.

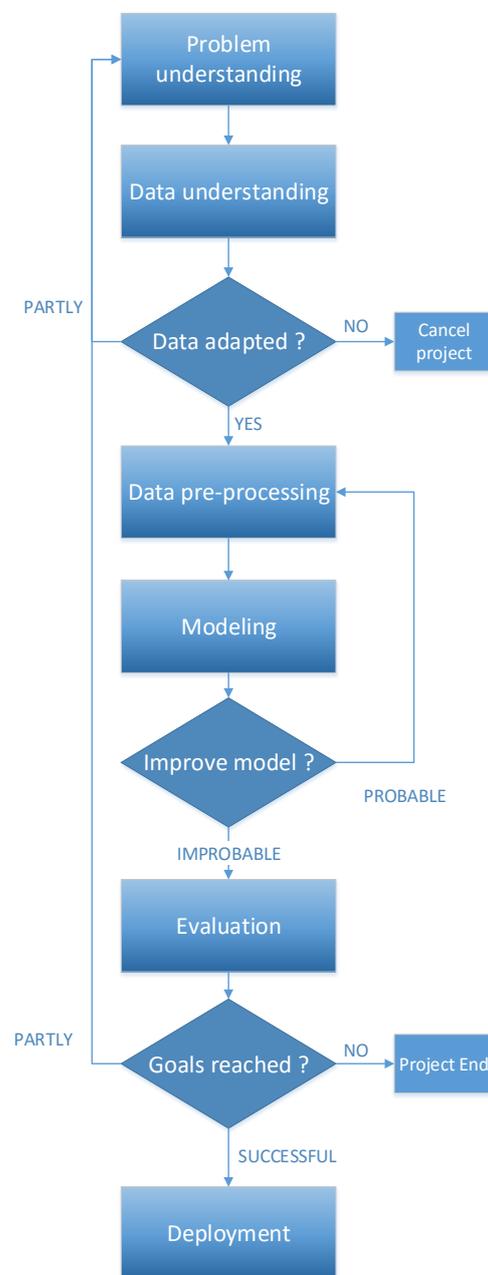


Figure 2 - CRISP/DM method: Intelligent data analysis processing

² Cross Industry Standard Process for Data Mining is a data mining process model that describes commonly used approaches that data mining experts use to tackle problems. (Chapman, 2000)

4 Background

During our last year in Haaga-Helia University of Applied Science, in the Business Information Technology program, we have been part of a healthcare mobile application project. The idea of this application was to develop a solution to provide different management tools and information about the patients in the hospital for the medical personal.

For the moment, the project is in a prototype stage where we have a basic mobile display that shows the main information about the patients directly from a database through an OData service. The information of the application is translated into two main languages: Finnish and English.

This project has for objectives to improve the work's efficiency of the collaborators in a healthcare environment and limit the amount of their administrative charges. We also want to integrate some context awareness principles with the location of the user of the application that will work with our machine learning algorithm.

5 Design

This part will describe and explain the conceptual architecture of the project from the machine learning part point of view, the workflow of the processes for intelligent data analysing, the choice of the machine learning algorithm and the generation of the simulated data. We will also justify the choice of the different technologies.

5.1 Conceptual Architecture

The following figure shows us the conceptual architecture of the mHealth project. The red components are part of this project thesis and concerns mainly the machine learning side.

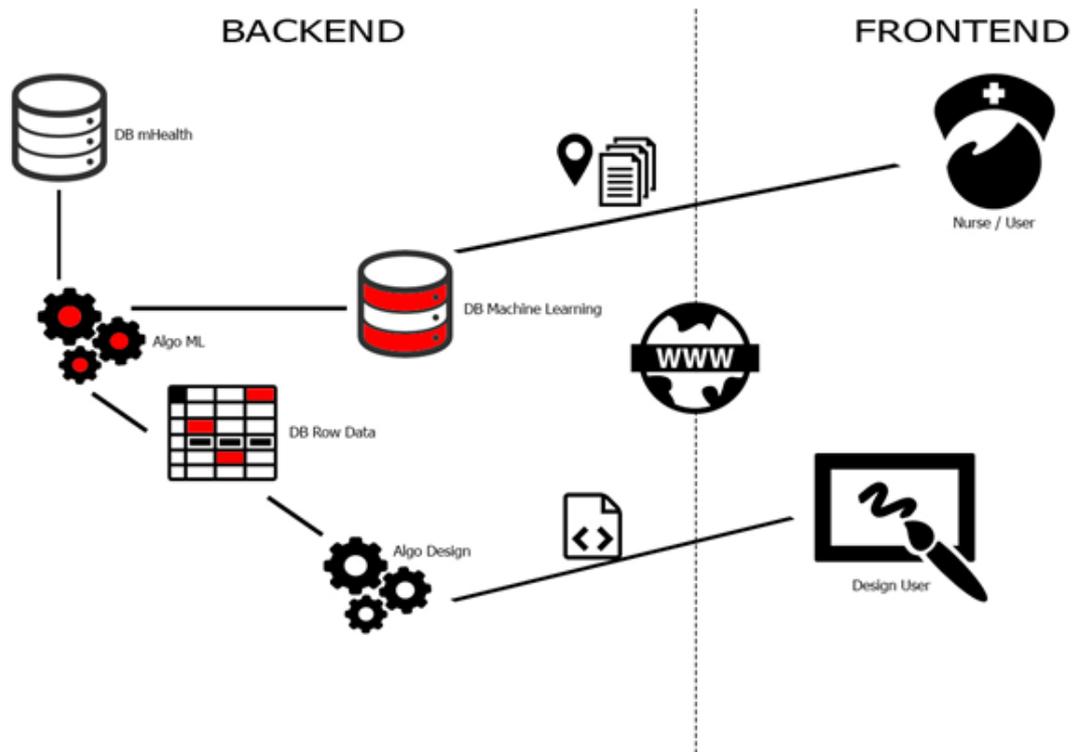


Figure 3 - Conceptual architecture

The general idea is that in the frontend of our application, every time a user is using the program, the different information, e.g., location, date, hours etc. are transferring into the backend and collecting into the Machine Learning Database. (Note that, the Machine Learning DB, is separated from the mHealth DB in that case, for a better understanding of the whole process. In reality, the Machine Learning DB is a table of the mHealth DB.) This DB contains the history of the behaviour of our users that will be the future input of the solution for our algorithms and machine learning processes to send, based on a model, the specific row data for further applications. A detailed architecture with the different technologies will be explained in the chapter 6.3.1, Deployment of the solution.

5.2 Generation of the simulated data

To be able to work and make predictions on the behaviour of our users, it is crucial to have an efficient dataset. Unfortunately, we are currently in a prototype phase and the user test experience has not started, this is the reason why we will have to generate simulated data. The generation of data is an important point that has to be taken with precautions. Indeed, we could create data that works with our example and match with our expected results, but

that doesn't correspond to the reality of our environment. We decided to ask different nurses to answer an online survey to help us generate a correct dataset. This survey can be found in the appendices of this document.

The following table describes the different variables that we will try to generate for our dataset:

| Variables | Description | Type | Example |
|--------------|--|----------|----------------|
| LastFunction | Last function/path that the user used or touched in the application. (This is what we will try to predict for suggesting content.) | Text | "measurements" |
| User | Type of user | Text | "Nurse" |
| Location | The location of the user at a current time. | Text | "Room 404" |
| Date | Date of the current day | DateTime | "2015-02-15" |
| Time | Time at the current moment | DateTime | "14:02:00" |

Table 1 – Dataset variables

During our implementation, after creating this dataset and generating our data population based on the survey, we will go through different phases of pre-processing of this data to establish what is important to keep for our classifier. It will probably result of new classes.

5.3 Machine learning

Machine learning as a definition is part of artificial intelligence to develop, analyse and implement automatic methods to be able to evolve to accomplish complex tasks. The idea can be summarized with a famous quote from Arthur Samuel³ :

³ Arthur Samuel (1901-1990) was a pioneer of artificial intelligence research. From 1949 through the late 1960s, he did the best work in making computers learn from their experience.

“Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.”

However, this definition is still evolving due to the fast expansion of the field.

There are two main types of Machine Learning processes that needs to be explained and taken in account when developing a solution:

Supervised machine learning: the application is trained with a training datasets that can be consider as examples.

Unsupervised machine learning: the application is given an amount of data and the program has to find patterns and relations between the different attributes.

Under both of this types, we can find subcategories containing different categories of machine learning algorithm. We won't go in details about all the different algorithms that exist but we will explain and justify the choice of a specific one in the next chapter.

(McCrea, 2014)

5.3.1 Choice of the algorithm

Choosing a machine learning algorithm is a complicated task and the choice can easily changed depending on the problem and the structure of the data that we are working with. Computational complexity is also an important point that has to be considered when we want to compare the resources that will be needed for a given algorithm. Furthermore, the number of dimensions of the dataset has to be taken in account.

In our case, most of the data will be categorical⁴. We won't have a huge amount of data and the model has to be able to constantly evolve and improve itself.

⁴ A categorical variable (sometimes called a nominal variable) is one that has two or more categories, but there is no intrinsic ordering to the categories. For example, gender is a categorical variable having two categories (male and female) and there is no intrinsic ordering to the categories. Hair color is also a categorical variable having a number of categories (blonde, brown, brunette, red, etc.) and again, there is no agreed way to order these from highest to lowest. A purely categorical variable is one that simply allows you to assign categories but you cannot clearly order the variables. If the variable has a clear ordering, then that variable would be an ordinal variable, as described below. (Statistics, s.d.)

Finally, we want to construct our model based on a past behaviour of our user. Owing this last fact, we will use a **supervised machine learning**.

Supervised machine learning can be from two subcategories: regression systems that are built to determine real numbers or specific values and classification system that can predict nominal value, true-false statements.

We want to define the correct function for the user and due to the fact that our dataset contains nominal values, therefore, we will use a **classification algorithm**.

The next table compares the three main group of classification algorithms:

| Algorithms | Advantages | Disadvantages |
|-----------------------|---|---|
| Bayes | <ul style="list-style-type: none"> - Fast and converge quickly - Less training data - Memory Usage Low | <ul style="list-style-type: none"> - Don't take in account interactions between the features |
| Decision Trees | <ul style="list-style-type: none"> - Easy to interpret - Handle interaction - Memory Usage Low | <ul style="list-style-type: none"> - Easily over fit - Speed : medium |
| SVMs | <ul style="list-style-type: none"> - High accuracy - Good for text classification | <ul style="list-style-type: none"> - Memory-intensive - Hard to interpret |

Table 2 - Classification algorithms

Based on the previous table, we made the choice for this project to work with a Bayes classifier that will exploit the Bayes' rule. It is a simple machine learning algorithm that could satisfy our needs. However, if the results in our first iterations weren't concluding, we will try with a different algorithm by following a correct development methodology.

(MathWorks.Inc, 2015) (Chen, s.d.)

5.3.2 Bayes classifier

The concept of Bayes' rule can be summarize with this following formula:

$$\text{posterior probability} = \frac{\text{conditional probability} * \text{prior probability}}{\text{evidence}}$$

The **posterior probability** answers the following question: “How probable is our hypothesis given the observed evidence?”

The **conditional probability** answers the following question: “How probable is our evidence given that our hypothesis is true?”

The **prior probability** answers the following question: “How probable was our hypothesis before observing the evidence?”

The **evidence** answers the following question: “How probable is the new evidence under all possible hypotheses?”

Our chosen algorithm will try to predict the most probable class, by computing the posterior distribution over the parameters given the observed data as

$$\text{pred}(\mathbf{x}) = \arg \max_{y \in \text{dom}(Y)} \frac{P(\mathbf{x} | y)P(y)}{P(\mathbf{x})}.$$

Here \mathbf{X} is an instantiation of the descriptive attributes and \mathbf{Y} is the class attribute.

We won't describe here in details how the algorithm works with all the different dimensions and probabilities, please refer to Michael R. Berthold reference for further information.

(Raschka, 2014), (Geller, 2012), (Grosse, s.d.), (Michael R. Berthold, 2010)

5.3.3 Utilisation of Bayesian Network

The exact Bayes classifier that we are going to use will be a Bayesian Network. Basic or naïve Bayes classifiers make the assumptions that all the classes of our dataset are independent of each other.

At the opposite, the Bayesian Network takes in account the dependences between those classes. In our case, the link between our classes are very crucial to determine the correct probability.

5.4 Technologies

This section will describe the main technologies that we will use for this thesis project. It is separated in three short chapters, the input that contains the data, the machine learning and the output of our suggested row data.

5.4.1 Input data

In the first iteration, we will start by generating our data in a simple Excel or CSV file to save time and concentrate our efforts to establish a proof of concept. During the third iteration phase, when we will want to deploy our solution, we will generate a database in Microsoft SQL Server and try to retrieve data from it.

5.4.2 Machine learning software

At the beginning, we started to work for the machine learning part with KNIME Analytics⁵ that is an open source data mining platform. It helped us for the first part of our implementation to determine to best way to predict our class. After the establishment of a correct model with an efficient score of prediction, we started to deploy our solution.

To run the application correctly in production, we want to have a Web Service that implements our solution and that can communicate the prediction to the user interface. The technologies used for this part are also detailed in the chapter 6.3, Deployment of the solution.

⁵ KNIME Analytics with its visual workbench combines data access, data transformation, initial investigation, powerful predictive analytics and visualization tool. The KNIME Analytics incorporates hundreds of processing nodes for data I/O, pre-processing and cleansing, modelling, analysis and data mining as well as various interactive views, such as scatter plots, parallel coordinates and others. It integrates all of the analysis modules of the well-known Weka data mining environment and additional plugins allow R-scripts to be run, offering access to a vast library of statistical routines. KNIME is based on the Eclipse platform and, through its modular API, easily extensible. (KNIME, s.d.)

5.4.3 Output solution

Concerning the output of our solution, we will try to generate a string of the predicted function for the user to retrieve the related information from the Microsoft SQL Server database. In the case where we couldn't share this data, we will provide the model of our algorithm.

6 Implementation

The two first iterations will be performed outside the mHealth project and totally in local. It mainly concerns the establishment of our proof of concept. At the opposite, the third iteration concerns the online deployment of our solution and the integration with the other parts of the project.

6.1 1st iteration

The first iteration will focus on the generation of the data population and the pre-processing phases.

6.1.1 Generation of the data population

As indicated in chapter 5.2, we don't have real data concerning the historic of the behaviour of our user. We already defined the structure of our dataset, now we need to generate our data. To create a realistic Bayesian Network model that can simulate real life environment, we established some "generic day" and rules for a nurse based on our previous survey.

There are three main shifts with the different teams in the hospital every eight hours. The schedule of the shifts will approximately be:

- 7am-3pm
- 3pm-11pm
- 11pm-7am

Three times a day, the patients have a nutrition phase at around:

- 7am
- 12am
- 7pm

To simulate the different location, we will have five different room, e.g. room1, room2 etc. and one location that is the office. We also need to take in consideration the break time of the nurse. We want to create a dataset for 15 consecutive days for a nurse.

The next figure shows a sample of the created dataset:

| Row ID | S LastFunction | S User | S Location | S Date | S Time |
|--------|----------------|--------|------------|--------|---------|
| 1.0 | measurements | nurse | room1 | 1/1/15 | 6:30:00 |
| 2.0 | activities | nurse | room1 | 1/1/15 | 6:33:00 |
| 3.0 | measurements | nurse | room2 | 1/1/15 | 6:36:00 |
| 4.0 | activities | nurse | room2 | 1/1/15 | 6:39:00 |
| 5.0 | measurements | nurse | room3 | 1/1/15 | 6:42:00 |
| 6.0 | activities | nurse | room2 | 1/1/15 | 6:45:00 |
| 7.0 | measurements | nurse | room4 | 1/1/15 | 6:48:00 |
| 8.0 | activities | nurse | room4 | 1/1/15 | 6:51:00 |
| 9.0 | measurements | nurse | room5 | 1/1/15 | 6:54:00 |
| 10.0 | activities | nurse | room5 | 1/1/15 | 6:57:00 |
| 11.0 | nutrition | nurse | room1 | 1/1/15 | 7:00:00 |
| 12.0 | nutrition | nurse | room2 | 1/1/15 | 7:10:00 |
| 13.0 | nutrition | nurse | room3 | 1/1/15 | 7:20:00 |

Figure 4 - Example of the created dataset

6.1.2 Pre-processing

The pre-processing phase is a crucial phase before the creation of our classifier model. It consists of cleaning the different data, removing the noises in the observation and correcting the missing data. Without a correct pre-processing, the results of our solution could be worthless and could represent a mistaken reality of our environment.

In KNIME, we created a MetaNode 1:1 to manage all our different pre-processing nodes. The MetaNode can be understand as a class like in programming, to encapsulate the different methods of the pre-processing phase.

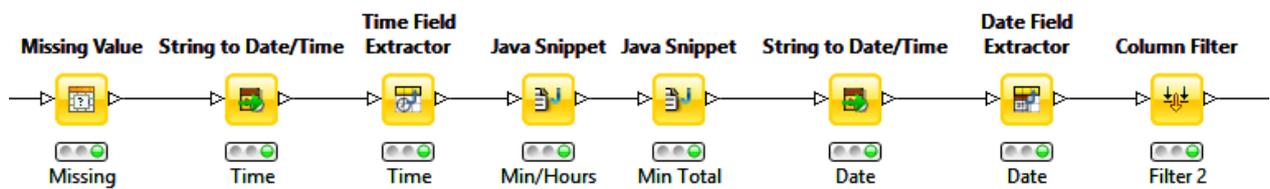


Figure 5 - MetaNode - pre-processing

We start to delete all the missing value and convert the format of our column “Time”. With the Time Field Extractor, we extract the minutes and the hours of the time to have two new columns.

With the two Java Snippets, we round the time to the nearest five minutes and create a new column that contains the amount of minutes of the corresponding time. The use of a rounded method is useful due to the fact that during a day, we have 1440 minutes. It means 1440 variables to analyse subsequently for our classifier. It limits the amount of calculations by five. Moreover, the exact time of the behaviour of the user doesn’t need to be as accurate because it can have some variations depending of the day and the situation.

| | |
|--|---|
| <p style="text-align: center;">Java Snippet</p>  <p style="text-align: center;">Min/Hours</p> | <p style="text-align: center;">Java Snippet</p>  <p style="text-align: center;">Min Total</p> |
| <pre> out_Minute = (c_Minute + 4) / 5 * 5; out_Hour = c_Hour; if(out_Minute == 60){ out_Minute = 0; } if(c_Minute > 55){ out_Hour = c_Hour + 1; } out_TimeRounded = out_Hour + ":" + out_Minute ; </pre> | <pre> int minutesHour = c_HourModified * 60; out_MinutesTotal = minutesHour + c_MinuteRounded5; </pre> |

(Note that this Java code is KNIME oriented)

Afterward, we extract the days of the week with the column “Date”. This column will be important when we will have a sufficient amount of data that can compare the differences and the similitudes depending of the day of the week.

Finally, we filter our dataset to have the following result:

| Row ID | S LastFu... | S User | S Location | HourMo... | Minute... | Minutes... | S Day of ... |
|--------|--------------|--------|------------|-----------|-----------|------------|--------------|
| 1.0 | measurements | nurse | room1 | 6 | 30 | 390 | Thursday |
| 2.0 | activities | nurse | room1 | 6 | 35 | 395 | Thursday |
| 3.0 | measurements | nurse | room2 | 6 | 40 | 400 | Thursday |
| 4.0 | activities | nurse | room2 | 6 | 40 | 400 | Thursday |
| 5.0 | measurements | nurse | room3 | 6 | 45 | 405 | Thursday |
| 6.0 | activities | nurse | room2 | 6 | 45 | 405 | Thursday |
| 7.0 | measurements | nurse | room4 | 6 | 50 | 410 | Thursday |
| 8.0 | activities | nurse | room4 | 6 | 55 | 415 | Thursday |
| 9.0 | measurements | nurse | room5 | 6 | 55 | 415 | Thursday |
| 10.0 | activities | nurse | room5 | 7 | 0 | 420 | Thursday |
| 11.0 | nutrition | nurse | room1 | 7 | 0 | 420 | Thursday |
| 12.0 | nutrition | nurse | room2 | 7 | 10 | 430 | Thursday |
| 13.0 | nutrition | nurse | room3 | 7 | 20 | 440 | Thursday |

Figure 6 - Pre-processing results

The pre-processing phase is finished, now we can start to create our classifier model.

6.2 2nd iteration

The second iteration contains the creation of the model, the training phase and the evaluation of our results.

6.2.1 Creation of the model

As explained in the analysis phase, we will use a Bayes Network Classifier to predict the function of the user. KNIME provides different extensions that contains a lot of different classification algorithm. We will use the Weka (3.7) KNIME library developed by the Machine Learning Group at the University of Waikato. (Note that this extension need to be downloaded via KNIME

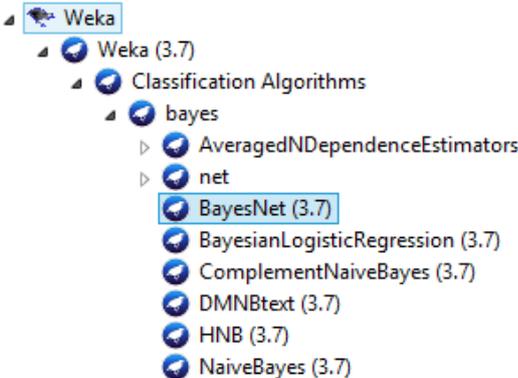


Figure 7 - Weka library

extension wizard). After the installation of the library, the Weka node are available into the node repository.

The BayesNet node classifier will be modify as the following figure:

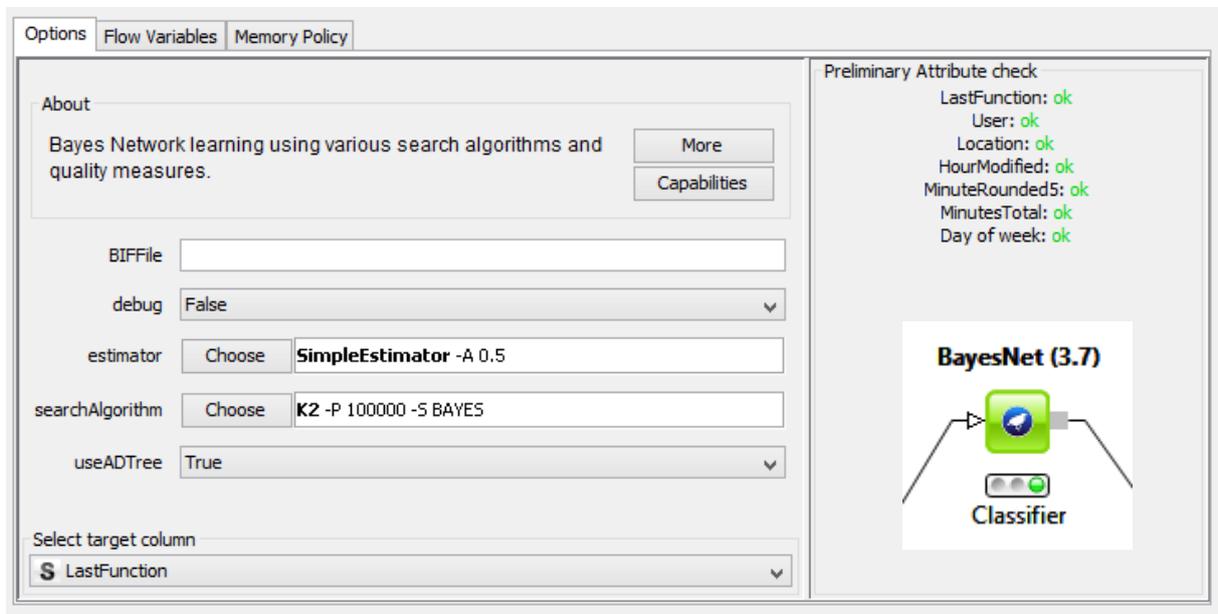


Figure 8 - BayesNet

Our target column is the column “LastFunction” and the preliminary attributes are the followings: User, Location, HourModified, MinuteRounded5, MinutesTotal and Day of week.

6.2.2 Training the data on the model

To train our model, we will partition our data into two datasets, one for the training, and the other for testing our classifier model. KNIME provides a nodes call “Partitioning” to execute this function. We will configure it to have a partition with 80% of the amount of the data for the training and 20% for the prediction test. It is important to split the data by hierarchical order to respect the time of the action of the user during the day.

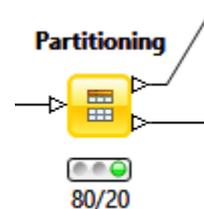


Figure 9 - Partitioning

6.2.3 Evaluation of the model

To evaluate the score of our model, we will use the Weka node: “Weka Predictor” to compare our predictions with the LastFunction column. Finally, the node “Score” establishes the score of our prediction and provides different statistics information.

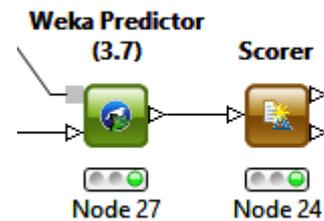


Figure 10 - Evaluation of the model

The next figure show the result of our model with a Confusion Matrix from the Scorer:

| LastFunction \ Prediction (...) | measurem... | activities | nutrition | physical ex... | prescription | doctor visit |
|---------------------------------|-------------|------------|-----------|----------------|--------------|--------------|
| measurements | 60 | 3 | 0 | 0 | 3 | 12 |
| activities | 0 | 84 | 0 | 0 | 0 | 3 |
| nutrition | 0 | 0 | 45 | 0 | 0 | 0 |
| physical examination | 15 | 0 | 0 | 75 | 0 | 0 |
| prescription | 27 | 0 | 0 | 0 | 21 | 12 |
| doctor visit | 0 | 0 | 0 | 0 | 0 | 69 |

| | |
|----------------------------------|----------------------|
| Correct classified: 354 | Wrong classified: 75 |
| Accuracy: 82.517 % | Error: 17.483 % |
| Cohen's kappa (κ) 0.788 | |

Figure 11 - Confusion matrix

In green, we can see the correct predictions for the different column. Per example for the function “Measurements”, it predicted it correctly 60 times.

In red as an example, we can see that our model predict 15 times wrong the physical examination and believed that it was “Measurements”.

In blue, we can have an idea of the accuracy of the model, here: 82 %. However, this percentage has to be taken with precautions because of the nature of our dataset. (Unreal generated data to test the solution). Nonetheless, the accuracy of the model will improve by itself when this data will be replace by real data after a certain amount of time.

In orange, you can see the kappa coefficient (k): 0.788. It defines a number that indicates if our results and predictions are made randomly or represents something reliable. The close to 1.0 the most realistic. The next tables indicate the level of agreement of the kappa coefficient:

| Value of Kappa | Level of Agreement |
|----------------|--------------------|
| 0-0.20 | NONE |
| 0.21-0.39 | MINIMAL |
| 0.40-0.59 | WEAK |
| 0.60-0.79 | MODERATE |
| 0.80-0.90 | STRONG |
| +0.90 | ALMOST PERFECT |

Table 3 - Kappa coefficient

Finally, the next figure represents the whole picture of our KNIME workflow:

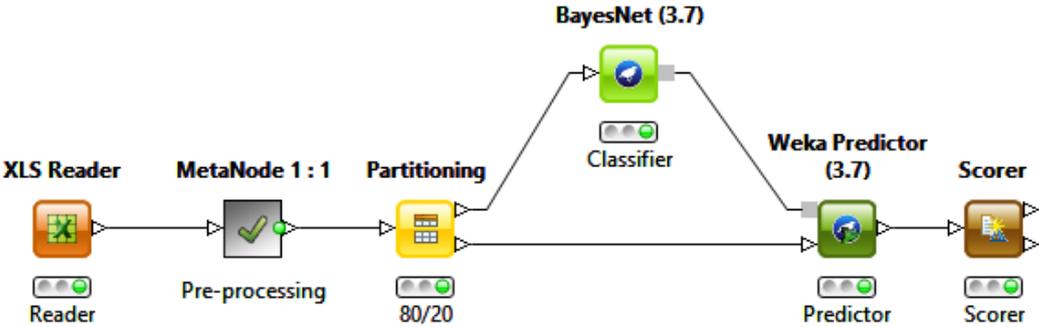


Figure 12 - Full KNIME Workflow

6.3 3rd iteration

6.3.1 Deployment of the solution

We had two different possibilities to deploy our KNIME solution:

- Use KNIME Server to discuss with its native Web Service to launch our workflow by remote access.

- Recreate the KNIME workflow with coding, implement this future library in the backend of our server and create our own Web Service to share the information.

Unfortunately, it was impossible to purchase or use a version of KNIME server so we decided to take the second option. To be able to access our predictions in the backend via a web service, we rethought some components of the architecture of the project. The next figure shows the updated and detailed architecture of the whole project:

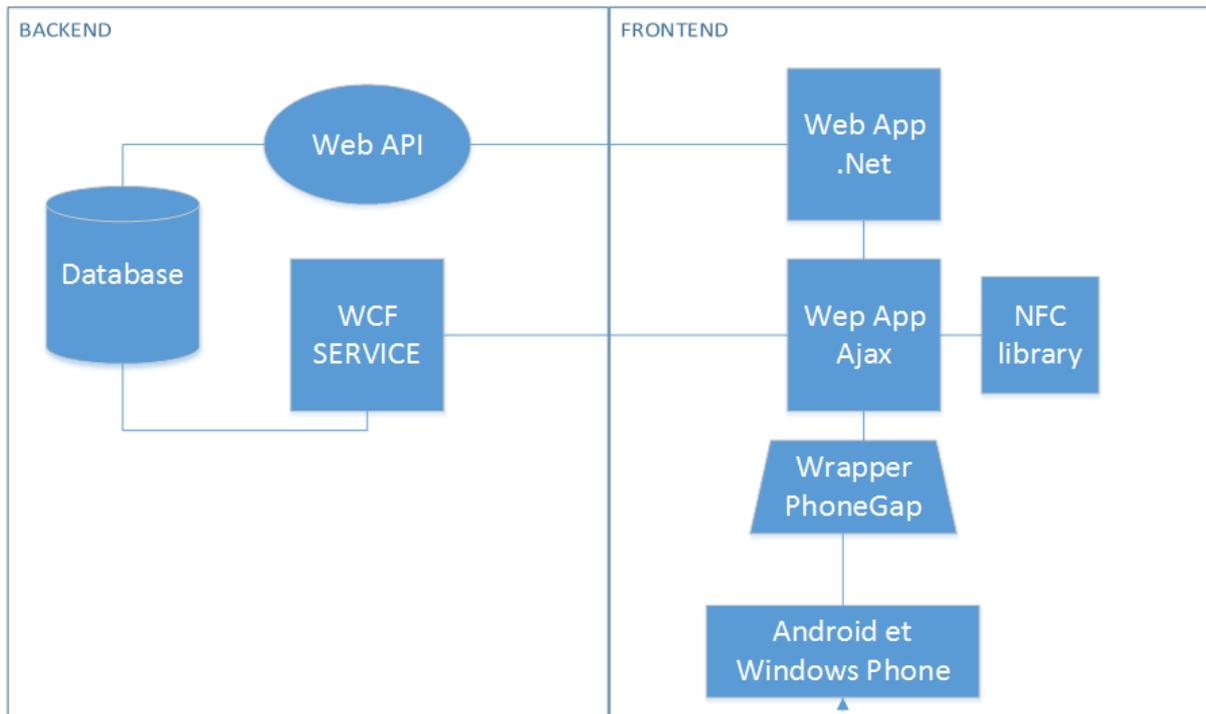


Figure 13 - Detailed Architecture

In the backend side, we have the database that contains all the information for the User Interface of the mobile application. Those data are available from the Front End via a Web API with an OData service. It also contains the historical behaviour of the users of the application that will be useful in our project to train the future implementation of the classifier model.

We took the decision to work with a WCF ⁶ Service because the first implementation of our user interface was based on a .Net Web Application. Therefore, it is also easy to share this information with other components with this solution.

In the front end, we have the implementation of the user interface of the application in .Net and another Web Application that interacts with the .Net Web App and the PhoneGap ⁷ Wrapper, due to fact that PhoneGap cannot be use with a .Net environment. This Web App is linked with the NFC library that triggers the changes of location of our user during the day.

Every time the user is using the application, the front end keeps in memory the history of his behaviour. It is only when the user changes his location that the front end will communicate with our WCF Service. It will send a list of the historical behaviours to be stored in the database, and the information concerning the new change of the location to get in return the predicted useful function for the user at this emplacement and suggest him related content. We think that is better to only send the historical data when a change of the location is triggered to limit the amount of connections with the database.

6.3.2 Machine learning implementation

We needed to choose a correct solution to implement the Bayesian Network algorithm in our .NET environment. We looked for possibilities and found two potential solutions:

- **Infer.NET**, a Microsoft Research framework for running Bayesian inference
- **Weka API**, a collection of machine learning algorithms for data mining tasks

To make easier the implementation of our classifier algorithm, we choose to work with the API from Weka owing to the fact that we used the Weka nodes that are based on the same solution for our Bayesian Network in KNIME. Firstly, we thought that using the same algorithm will logically result to an equivalent solution and accuracy that in KNIME

⁶ Windows Communication Foundation (WCF) is a framework for building service-oriented applications. (Microsoft.com, 2015)

⁷ PhoneGap is a free and open source framework that creates mobile apps using standardized web APIs for the multi-platforms.

Analysis. Secondly, the Infer.NET though the fact that it is .NET oriented, doesn't contain a lot of support and possibilities as Weka.

6.3.3 Weka API – Java & .Net Interoperability

For the moment, the Weka API has only been created for Java applications, however, the backend of our server is working in a .NET environment. To be able to achieve a .NET and Java interoperability, we can use **IKVM**⁸ that is an implementation of Java for .NET.

Firstly, we need to download the corresponding Weka application. After downloading it, we need to extract the weka.jar that contains the Java classes for Weka. Then we need to download and use IKVM console with the next command:

```
> ikvmc -target:library weka.jar
```

It will create a .dll library of the weka.jar to be used in our .NET environment as a reference.

Furthermore, we will also need to add a reference to IKVM.OpenJDK.Core (available in the IKVM folder) that contains some general Java classes that we will need for our future code. Indeed, some methods don't work with the **System.IO** class, therefore, we need to use the class **Java.io**.

(Weka, 2015)

6.3.4 Console application test

To test our implementation of the Weka API, we will firstly create a basic Console Application. After having a functional solution, we will create the WCF service that implements our previous tested code.

To start with a correct structure, we created a new library with two classes, Behaviour.cs and BehaviourController.cs.

⁸ IKVM.NET is an implementation of Java for Mono and the Microsoft .NET Framework. It includes the following components: a Java Virtual Machine implemented in .NET, a .NET implementation of the Java class libraries, tools that enable Java and .NET interoperability

Behaviour.cs defines the class structure of the behaviour of a normal user as the following:

```
public class Behaviour
{
    public String LastFunction { get; set; }
    public String User { get; set; }
    public String Location { get; set; }
    public int HourModified { get; set; }
    public int MinuteRounded10 { get; set; }
    public int MinutesTotal { get; set; }
    public String Dayofweek { get; set; }
}
```

Note that we modified the previous attribute MinuteRounded5 to MinuteRounded10 (To the nearest 10 minutes). It gave us better results and improved the efficiency of the algorithm.

The BehaviourController.cs manages the different methods to predict the function for the user and save the historical behaviour of our user. The following explanations concern this class.

Firstly, we want to retrieve the data. At this point, we won't use a .csv file as we used to do in KNIME to simulate the database but we will directly connect to the real backend of the server.

Weka API provides different methods to generate the instances of our data from a .csv or from a database. Unfortunately, the methods to read from database don't work so well even with the IKVM implementation. Consequently, we will need to make some adjustments.

We will use a [SqlConnection](#) with a basic SQL query to retrieve the data into a [SqlDataReader](#).

To bypass the interoperability problem, we will write in a temporary .csv file the data contained in the [SqlDataReader](#).

Weka API uses **ARFF**⁹ files to manage the creation of the instances¹⁰ of the data. The next figure shows an example of a structure of an ARFF file:

```
@relation train

@attribute lastFunction {measurements,activities,nut:
@attribute userType {nurse}
@attribute location {room1,room2,room3,room4,room5,o:
@attribute hourModified numeric
@attribute minuteRounded10 numeric
@attribute totalMinutes numeric
@attribute dayOfTheWeek {Thursday,Friday,Saturday,Su

@data
activities,nurse,room1,6,30,390,Thursday
```

Figure 14 - ARFF file example

From our .csv we can use an [ArffSaver](#) to create the corresponding ARFF file. During the creation of the instances, it is important to set the class index of our instances. Here we will define the index at 0 to indicate that the prediction will be for the attribute "LastFunction". In our code:

```
data.setClassIndex(0)
```

After creating the instance for the training of our model, we need to create the Bayes Network classifier.

```
Classifier cls = new BayesNet();
cls.buildClassifier(train);
```

We can save the model of the classifier into a file if needed:

```
weka.core.SerializationHelper.write("BayesNet.model", cls);
```

⁹ An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

¹⁰ Instances – an object that contains all of a dataset.

After that, we will create the instances for using our model corresponding with the data that we want to predict. Note that is very important that the ARFF file has the same attributes of the training file to be able to calculate the correct probability.

We will retrieve the probability of the attribute with the following line:

```
double predi = cls.classifyInstance(test.instance(0));
```

Finally we will return the name of the predicted class:

```
return test.classAttribute().value((int)predi);
```

6.3.5 WCF Service

The WCF Service “BehaviourService” is used to communicate in OData with the frontend for two main points:

- **Storage** of historical behaviour of the user
- **Prediction** of the needed function for the user

We implemented the “BehaviourLibrary” as a reference to this WCF Service to be able to use our different classes that we tested in the Console Application test with the Weka library.

Concerning the prediction, we have a “GetPrediction()” string method that takes in parameters a **Behaviour** object to send back a display of the correct prediction of the function to the user interface.

Regarding the storage, we have a void method “SaveHistoricalData()” that has a list of **Behaviour** as parameters. This method read all the list and insert the different element of objects into the database.

It is important that we have two independent methods, therefore, if in one of them an error occurred, it won’t affect the result of the other one.

We firstly started by implementing those methods and test it in local with the WCF test client ¹¹ tool. Thereafter, we deployed and published our method into the web server.

To host our service, we firstly compiled it into a DLL. Then we added this dll into the website folder. Afterwards, we created a .svc file in this website to make the service reachable by the WCF Client.

7 Results

7.1 Current results

We currently have a Windows Communication Foundation Service that implements our controller class to be able to predict with the Weka API a suggested function for a user.

This WCF Service, also implement a method to store the historical data to keep updated the past behaviour of the users concerning the mobile application. Therefore, the efficiency of the Bayesian network classifier model used for the prediction will progressively improve every day.

At this time, we have an accuracy of about 75 %. Nonetheless, this research was performed with generated data and to get a realistic result, we need to compare it with real historical data.

7.2 Future amelioration

7.2.1 Classifier model storage

At the moment, the classifier model evolves every time a user performs an action when the WCF Service get the historical data. We could imagine to store the classifier model in a file when the level of accuracy is satisfying to limit the amount of calculation every time. For that, we will need to wait the implementation of real data in the database.

¹¹ Windows Communication Foundation (WCF) Test Client (WcfTestClient.exe) is a GUI tool that enables users to input test parameters, submit that input to the service, and view the response that the service sends back. It provides a seamless service testing experience when combined with WCF Service Host.

7.2.2 Top ranking algorithm

We could also suggest contents for the user that will be based on the most used functions.

The next figure defines the possible uses of the top ranking algorithm:

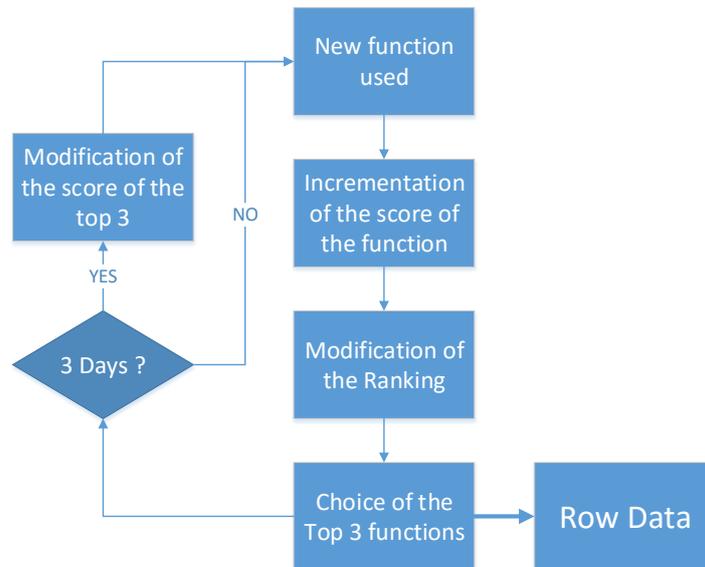


Figure 15 - Top ranking algorithm

Every time a function is used by the user, it will increment the score of this specific function and modify its rank. Afterward, we will choose the top three functions that have the best score and create row data based on their value. Every three days, the score of the top three functions could be modify to have to same value as the function with the fourth rank to be maintain the system efficient.

8 Conclusion

The first objective of this project was to determine the best way and technologies to establish a proof on concept that shows that is possible, by analysing the past behaviour of a user at a specific location and times, to suggest him personalized content.

We have been part of a healthcare project to test the application of this proof on concept in production. The goal was to develop a Web Service engine to provide the suggested information to frontend applications.

The machine learning algorithm to study the comportment of a user of a mobile application was based on a classification technics that used a Bayesian Network model.

We have worked with different iterations and tests phases along the project to implement at the end a functional and efficient solution that had satisfied our research requirements with a reasonable accuracy.

Through this entire thesis project, we have developed our analytics and researches skills but also improved our project management competencies by working with different collaborators. It was as well certainly worthy to have the opportunity to expose our technical abilities that have been learned and obtained during our formation.

The results of this thesis are profitable for general research to have an idea about the process involved in creating a solution using machine learning technics to suggest content. The utilisation of similar components and framework can be integrated and used to several kind of environments and programs. Nowadays, we are just at the beginning of the integration of that type of technologies in our daily life, but in the next years we will probably see increasingly emerge the creation of news applications.

9 References

- Chapman, P. (2000). *CRISP-DM 1.0*. Retrieved from the-modeling-agency.com:
<https://the-modeling-agency.com/crisp-dm.pdf>
- Chen, E. (n.d.). *What are the advantages of different classification algorithms?* Retrieved from Quora.com: <http://www.quora.com/What-are-the-advantages-of-different-classification-algorithms>
- Cherry, K. (n.d.). *What Is Behavior Analysis?* Retrieved from Psychology.about.com/:
<http://psychology.about.com/od/behavioralpsychology/f/behanalysis.htm>
- Francesco Ricci, L. R. (n.d.). *Introduction to Recommender Systems*. Faculty of Computer Science, Free University of Bozen-Bolzano. Retrieved from
<http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf>
- Geller, E. H. (2012, October 22). *Bayes' Rule and Bomb Threats*. Retrieved from Psychology In Action: <http://www.psychologyinaction.org/2012/10/22/bayes-rule-and-bomb-threats/>
- Grosse, R. (n.d.). *Bayesian machine learning*. Retrieved from Metacademy.com:
http://www.metacademy.org/roadmaps/rgrosse/bayesian_machine_learning
- Grossman, L. (2010, May 2010). *How Computers Know What We Want — Before We Do*. Retrieved from time.com:
<http://content.time.com/time/magazine/article/0,9171,1992403,00.html>
- KNIME. (n.d.). *KNIME Analytics Platform*. Retrieved from Knime.org:
<https://www.knime.org/knime>
- MathWorks.Inc. (2015). *Supervised Learning Workflow and Algorithms*. Retrieved from
<http://se.mathworks.com/>: <http://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html>
- McCarthy, J. (n.d.). *Arthur Samuel: Pioneer in Machine Learning*. Retrieved from infolab.stanford.edu: <http://infolab.stanford.edu/pub/voy/museum/samuel.html>

- McCrea, N. (2014). *An Introduction to Machine Learning*. Retrieved from toptal.com:
<http://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer>
- Michael R. Berthold, C. B. (2010). *Guide to Intelligent Data Analysis*. Konstanz,.
- Microsoft.com. (2015). *What Is Windows Communication Foundation*. Retrieved from
 msdn.microsoft.com: <https://msdn.microsoft.com/en-us/library/ms731082%28v=vs.110%29.aspx>
- Morris, B. (n.d.). *Yes We Kanban! Introducing an Agile Methodology*. Retrieved from verilab:
http://www.verilab.com/files/yes_we_kanban_morris.pdf
- Nedelcu, A. (2012, February 09). *How To Build a Naive Bayes Classifier*. Retrieved from
 bionicspirit.com : <https://www.bionicspirit.com/blog/2012/02/09/howto-build-naive-bayes-classifier.html>
- Raschka, S. (2014, October 4). *Naive Bayes and Text Classification I*. Retrieved from
<http://sebastianraschka.com/>: <http://arxiv.org/pdf/1410.5329.pdf>
- Simonite, T. (2013, March 13). *With Personal Data, Predictive Apps Stay a Step Ahead*. Retrieved
 from [technologyreview.com](http://www.technologyreview.com):
<http://www.technologyreview.com/news/514366/with-personal-data-predictive-apps-stay-a-step-ahead/>
- Statistics, U. . (n.d.). *What is the difference between categorical, ordinal and interval variables?*
 Retrieved from [ats.ucla.edu](http://www.ats.ucla.edu):
http://www.ats.ucla.edu/stat/mult_pkg/whatstat/nominal_ordinal_interval.htm
- StatSoft. (n.d.). *Naive Bayes Classifier*. Retrieved from Statsoft.com:
<http://www.statsoft.com/textbook/naive-bayes-classifier>
- Tim Jones. (2013, December 12). *Recommender systems*. Retrieved from
<http://www.ibm.com/>: <http://www.ibm.com/developerworks/library/os-recommender1/>

Walker, M. (2001). *Introduction to Genetic Programming*.

Weka. (2015). *IKVM with Weka tutorial*. Retrieved from <http://weka.wikispaces.com/>:
<http://weka.wikispaces.com/IKVM+with+Weka+tutorial>

What is Kanban? Kanban Software Tools. (n.d.). Retrieved from versionone:
<http://www.versionone.com/what-is-kanban/>

10 Appendices

Appendix : Nurses Survey

Hello, this quick survey has for objectives to establish some averages about the daily routine of nurses for science purpose.

All this information will stay confidential and won't be share with a third part.

Thank you for your participation

***Mandatory**

How many shifts do you have per days? *

- 2
- 3
- 4
- 5
- 5+

How many patients does a doctor have? *

- 1-10
- 10-20
- 20-30
- 30+

How many patients does a nurse have? *

- 3-5
- 5-8
- 8-12
- 12+

How many time do you see your patient per day? *

- 1
- 2
- 3
- 4
- 5
- 5+

How many hours do you spend on administrative work per day?

- 1
- 2
- 3
- 4
- 4+

How many breaks do you have per day? *

- 1
- 2
- 3
- 3+

How long is each break in minutes? *

- 5-10
- 10-15
- 15-20
- 20+

How many nurses can a patient have? *

- 1
- 2
- 3
- 3+