

**Predictive analytics with online data for WSO2  
Machine Learner with the support of Ensemble  
method**

**Software Requirements Specification**

**Project ID:16-054**

Author:  
IT13007734  
Pathinayake I.M

Supervisor:  
Mr. Lakmal Rupasinghe

Bachelor of Science (Special Honors) in Information Technology

Sri Lanka Institute of Information Technology

Submitted on 01.04.2016

## DECLARATION

I hereby declare that the submitted project Software Requirements Specification document for Predictive analytics with online data for WSO2 Machine Learner with the support of Ensemble method is an original work done by Pathinayake I.M. This document is proprietary and an exclusive property of the SLIIT project group 16-054. List of references I referred for the preparation of this document are given as references at the end of the document.

Member : IT13037734 – Pathinayake I.M

Signature : .....

Supervisor

.....  
Mr. Lakmal Rupasinghe

## **List of Figures**

Figure 1 - System Overview .....	8
Figure 2- Home Page .....	11
Figure 3 - Load streaming data interface .....	11
Figure 4 - Building dataset interface.....	12
Figure 5 - Applying the algorithm interface .....	12
Figure 6 - Evaluating model and predicting interface .....	13
Figure 7 - Data visualization interface.....	13
Figure 8 - Use case diagram.....	17
Figure 9 - Class Diagram .....	22
Figure 10 - how to select the algorithm .....	25
Figure 11- Work Breakdown Structure.....	26
Figure 12 - Gantt Chart.....	26
Figure 13 - WSO2 machine learner data base ER diagram .....	26

## **List of Tables**

Table 1 - Comparison of existing Products with the proposed Machine learner.....	10
Table 2- Use case scenario for extracting streaming data.....	15
Table 3 - Use case scenario for train the model.....	16
Table 4 - Use case scenario for Evaluate the model and predict .....	16
Table 5 - User Characteristic .....	18

# **1. Introduction**

## **1.1. Purpose**

This document provides detailed descriptions of all the requirements related to the system ‘Predictive analysis with streaming data for WSO2 Machine Learner’, with respect to the scope covered by the project, users involved in this project, features of the system, interfaces of the system, functionality, the constraints under which it must operate. All parts of this document are intended primarily for giving a continuous, consistent and unambiguous details to the customers of the application, but will also be of interest to software engineers building or maintaining the software. The document is used by the developers to implement the functionalities and to ensure traceability of the software, by testers to know the interfaces and to test the software accordingly and by the users to verify that the requirements specified satisfy their needs.

## **1.2. Scope**

“Predictive analysis with streaming data for WSO2 machine learner using the ensemble method” is the system which is described by this document. This SRS includes the requirements for the initial release of the system. If the requirement changes in future it is possible to change the specification accordingly. It also covers the details of hardware and software requirements need to implement the system and gives detailed description about externally visible behavior of the system and covers the areas that contain limitations while completing the system.

The proposed system aims at giving the most accurate prediction by implementing an ensemble method for streaming data. Furthermore the proposed system focusing on giving the unambiguous decisions in the right time and provide the opportunity to the user to change the parameter of the algorithm on the fly and get the expected prediction.

### **1.2.1. Objectives**

- Design Incremental learning component and combine it with existing WSO2 Machine Learner.
- Design Predictive model with streaming data and combine it with existing WSO2 Machine Learner.
- Design the system which allows user to change the parameters of the algorithm on the fly.
- Design and deliver data visualization component.
- Implement Ensemble method and combine it with existing WSO2 Machine Learner.

### **1.2.2. Other Objectives**

- To make WSO2 Machine Learner more user friendly with data visualization and more understandable user interfaces.
- To develop the system with high-accuracy, efficiency, understandability, flexibility and satisfy other non-functional requirements.

### **1.2.3. Benefits**

- Since the proposed system allows user to change the parameters of the algorithm on the fly, user can get the predictions for any field by changing the parameters.
- Provide multiple visualizations to explore your data; scatter plots, histograms, Trellis charts, cluster diagrams and so on.
- It gives fast prediction result.
- Since the system implements an ensemble method, use can get unambiguous, more accurate prediction result.
- Since the system use user friendly interfaces, No need of expert knowledge in order to deal with the system.

### 1.3. Definitions, Acronyms, and Abbreviations

ML	Machine Learner
RAM	Random Access Memory
SRS	Software Requirements Specification
SE	Standard Edition
JDK	Java Development Kit
UI	User Interface
SVM	Simple Vector Machine
HTML5	Hyper Text Markup Language 5
IDE	Interactive Development Environment
FIFA	International Federation of Football Associations
SVN	Subversion
ActiveMQ	Active Messaging Queue
JMS	Java Message Service
API	Application Programming Interface
PC	Personal Computer
WEKA	Waikato Environment for Knowledge Analysis
HPCC	High Performance Computing Cluster

### 1.4. Overview

This SRS document intends to cover all the functional and non-functional requirements of the proposed system. Each of them has been discussed clearly in detail. All are described under three chapters.

The first chapter provides a full description of the project to the users who are interested in big-data prediction. The purpose of the SRS, particular audience, what will the system do, how the system going to perform their actions, general objectives, goals, benefits that can gain, definitions, acronyms, abbreviations and overview of the system.

And also contains user characteristics which can be useful to identify the intended customer and potential user. The second chapter concerns details of each of the system functions and actions in full for the software developers' assistance. These two sections are cross-referenced by topic; to increase understanding by both groups involved.

Major components and how they are subdivided into smaller components and the processes are taking place to develop the system are described in the chapter 2. A detailed overview of product perspective, user interfaces, system interfaces, hardware interfaces, communication interfaces, memory constraint, operations, site adaption requirements with the use of use case, given in the rest of the part of this chapter.

The third chapter includes all the supporting information such as references, indexes and appendices. Finally this will focus on the system deployment and usage procedures in the real world environment and how it will meet the requirements of the Stakeholders. This document can be used as a guide by the development team in the development phase.

The proposed ML is capable of identifying streaming data patterns in the recent history and updating the patterns with incoming data without catastrophic forgetting, in order to improve the quality and the accuracy of the prediction component. Furthermore to make the prediction component more efficient and accurate the proposed system going to implement an ensemble method which is going to combine multiple algorithms to give the most accurate prediction results. And also as the last part of this project the incremental learning component and ensemble method component will be combined with existing WSO2 Machine Learner.

#### **1.4.1. Goals**

- Design cost-effective, user-friendly Machine Learner.
- Turns data into predictions.
- Develop data-driven prediction model based on online data

- Enable user to change the parameters of the algorithm on the fly.

### 1.4.2. System Overview

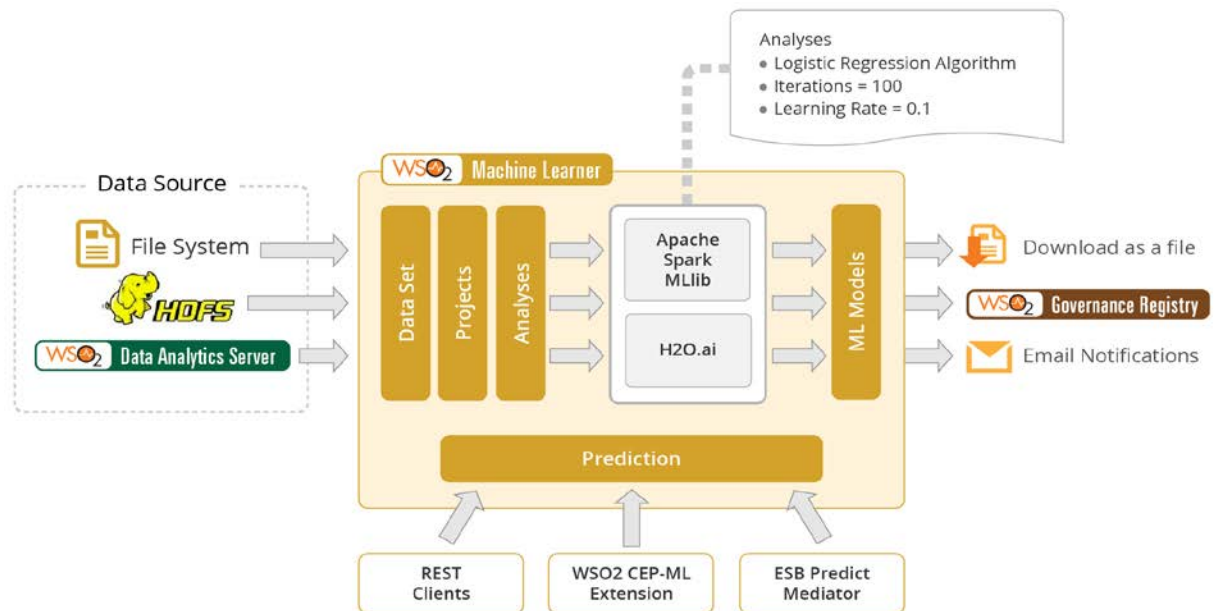


Figure 1 - System Overview

## 2. Overall Descriptions

In real-world scenarios, many types of data are acquired sequentially. Traditional way of waiting for the data to be collected and identifying models is a unique process. With the arrival of new data, these models are changing and acquired habits must be changed accordingly and decisions based on these models are also changing. Concept of machine learning algorithms with data streaming was born in these circumstances.

The proposed system focused on several objectives. With the completion of the system is supposed to accomplish these objectives. The idea of incremental learning with streaming data focuses on two objectives.

1. Identifying patterns in the recent history.
2. Updating the patterns with incoming data without catastrophic forgetting.



Apart from the above mentioned objectives the proposed system focus on the following objectives as well, in order to improve the efficiency and the accuracy of the existing WSO2 ML,

1. Implement an ensemble method(s), to combine multiple algorithms.
2. Create a UI to include the end to end flow of training the algorithm.
3. Integrate it to the WSO2 Machine Learner

These objectives can be approached in two methods [1].

- Incremental algorithms - there are machine learning algorithms which can be modified to support incremental learning. Eg: Mini-batch k-means, stochastic gradient descent based linear models, SVM, etc.
- Periodic re-training - machine learning models are trained with buffered data periodically.

Implementation of the proposed system will be done by using several technologies and tools such as Machine learning, Streaming data, Java, Scala, HTML5, JavaScript, Apache spark and Eclipse IDE.

There are several main components of the application,

- Incremental learning component
- Predictive model with streaming data
- User enabled parameter changed of the algorithm on the fly
- Data visualization component
- Implement Ensemble methods.

Final outcome is combining these components with the existing WSO2 Machine Learner and give the ability to the Machine Learner (ML) to give more efficient and accurate predictions.

## **2.1. Product perspective**

There are many existing applications to predict the outcome of an experiment from gathered data. Consider an example like this. Football is one of the most watched sports in the entire world. People come from all over the world to watch the FIFA finals. People are anxious to know which teams will get selected to the semifinals, finals and ultimately become the champions. Many software have been produced to

predict scenarios like this. The specialty of proposed WSO2 Machine Learner is that it takes into account the online streaming data and according to the expected outcome it can change the parameters of the algorithm on the fly and provide the most accurate result. In order to make the prediction most accurate the proposed system going to implement an ensemble method which is going to combine multiple algorithms. This is not just applicable for a sport but rather it can be used in many fields. Even though WSO2 has already developed a Machine Learner that doesn't fulfill those requirements. It doesn't use streaming data to get predictions and it doesn't allow user to change the parameters of the algorithm on the fly.

Features	WEKA	HPCC System	Existing WSO2 ML	Proposed WSO2 ML
User Friendly Interfaces	✓	✓	✓	✓
Outlier detection when generating graphs	X	X	X	✓
Model Generation	X	X	✓	✓
Can configure with spark	X	X	X	✓
Display prediction result rather than value	X	X	X	✓
Validate File Formats	X	X	X	✓
Improved Performance	X	X	X	✓
Successfully displaying cluster diagrams	X	X	X	✓
Use Streaming data	X	X	X	✓
Use ensemble method	X	X	X	✓

Table 1 - Comparison of existing Products with the proposed Machine learner

### 2.1.1. System interfaces

- Data Extraction Interface.
- Web – Desktop connectivity Interface.
- Command Prompt

## 2.1.2. User interfaces

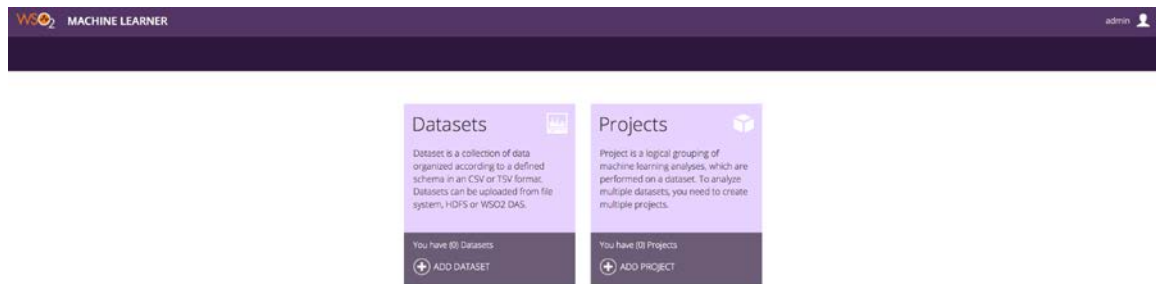


Figure 2- Home Page

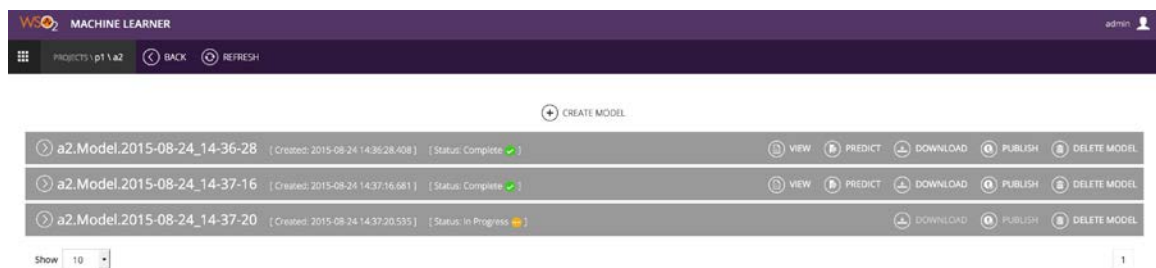


Figure 3 - Load streaming data interface

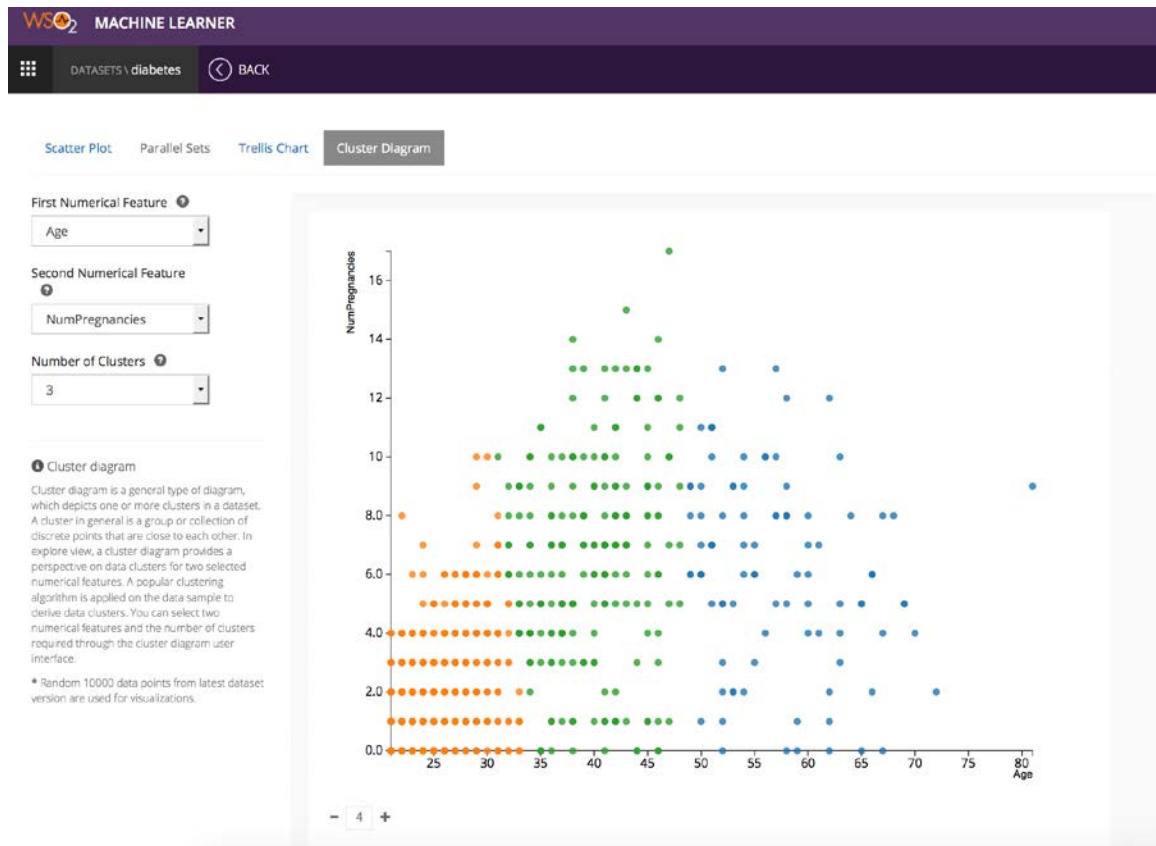


Figure 4 - Building dataset interface

### Algorithm

Algorithm name \*  
COLLABORATIVE FILTERING (Implicit Feedback Data)

User variable \*  
USER\_ID

Product variable \*  
PRODUCT\_ID

Observation list  
2,3,6

### Parameters

Set Hyper-Parameters for Recommendation\COLLABORATIVE FILTERING IMPLICIT

Rank \*  
8

Iterations \*  
20

Lambda \*  
0.01

Blocks \*  
3

Alpha \*  
40

Weights \*  
3,4,7,6,5

Figure 5 - Applying the algorithm interface

### Predict

Prediction Source ⓘ  
Feature values ▼

sepal\_length \*  
1.2

sepal\_width \*  
3.2

petal\_length \*  
4.3

petal\_width \*  
1.0  
1.0

Predict

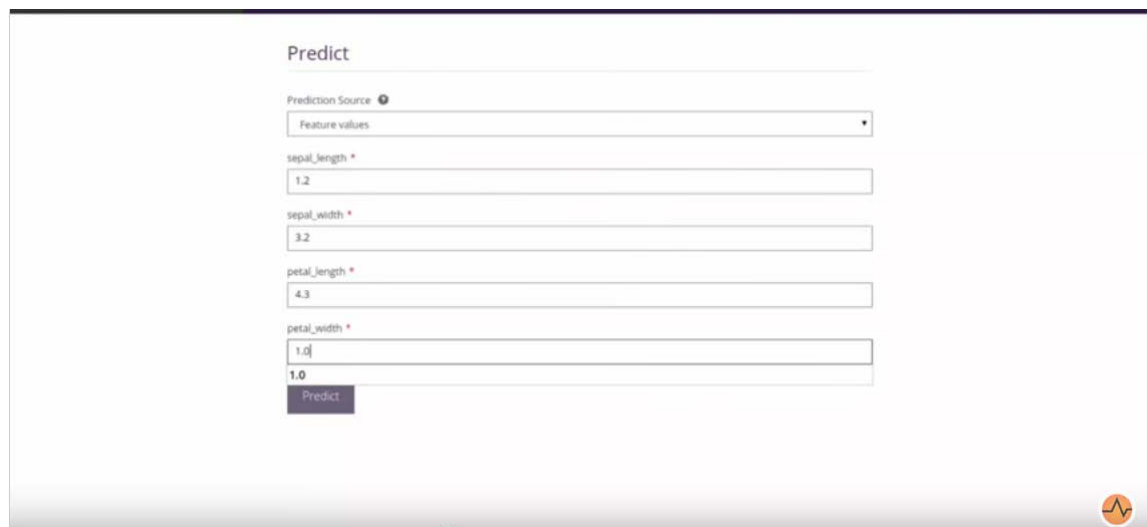


Figure 6 - Evaluating model and predicting interface

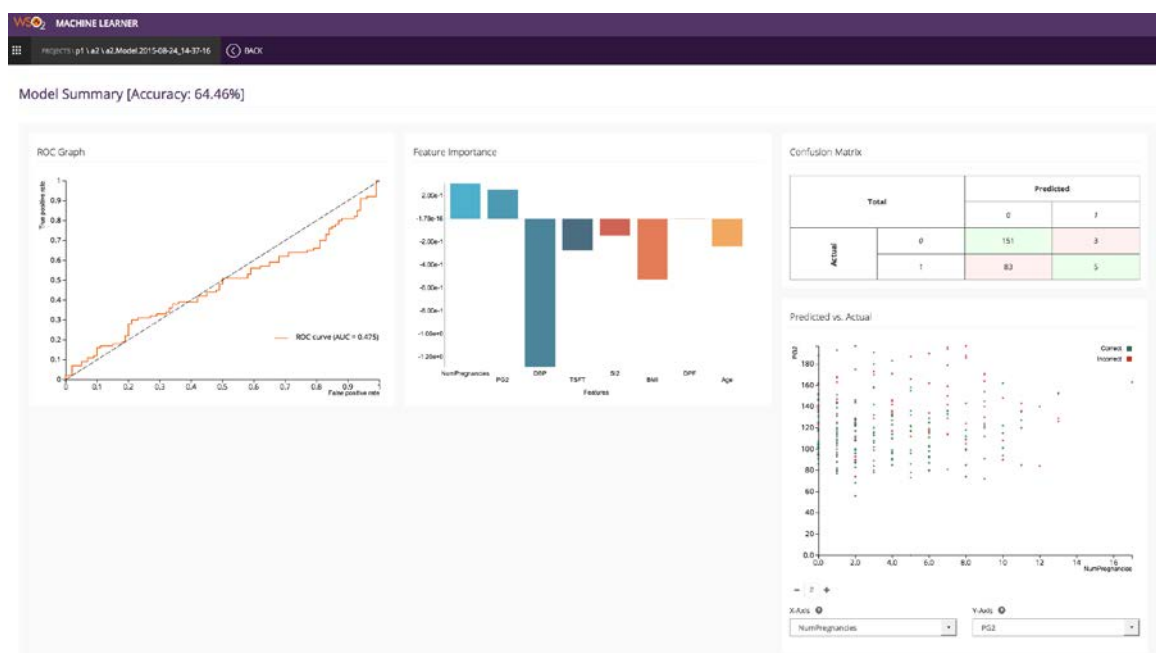


Figure 7 - Data visualization interface

### **2.1.3. Hardware interfaces**

No special hardware interfaces are used for the system.

### **2.1.4. Software interfaces**

- Oracle Java SE Development Kit (JDK) 1.6.24 or later / 1.7.\*
- Apache ActiveMQ JMS Provider 5.5.0 or later
- Apache Ant 1.7.0 or later
- SVN Client
- Apache Maven 3.0 or later
- JavaScript enabled Web Browser
- MySQL

### **2.1.5. Communication interfaces**

- Internet
- Database connection interface

### **2.1.6. Memory constraints**

- Memory - 2 GB minimum, 512 MB heap size.
- Disk - 1 GB minimum, excluding space allocated for log files and databases.

### **2.1.7. Operations**

- Extract streaming data from sources using APIs [3].
- Cleansing of data with necessary attributes and get it transformed.
- Load the streaming data set into testing or training model in Machine Learner (ML).
- Launch the Machine Learner prediction model and get prediction.
- Interpret with the dashboard.
- Viewing reports.

### 2.1.8. Site adaptation requirements

- Oracle Java SE Development Kit (JDK) must be installed.
- To enable the product's JMS transport and try out JMS samples, the ActiveMQ client libraries must be installed in the product's class path before enable the JMS transport [4].
- To compile and run the Apache Ant must be installed.
- SVN Client to check out the code to build the product from the source distribution.
- Apache Maven must be installed in order to build the product from the source distribution.
- Should be available JavaScript enabled Web Browser.

### 2.2. Product functions

Use case Name	Extract Steaming Data
Pre –Condition	Internet connection is active. User must be registered with the system.
Actor	Registered user
Main Success Scenarios	<ol style="list-style-type: none"><li>1. Click “Add dataset”.</li><li>2. Give a desired name and a version.</li><li>3. Select file system as “Streaming data”.</li><li>4. Click “Choose file”.</li><li>5. Click “Create Dataset”.</li><li>6. Click “Explore” to explore the dataset</li></ol>
Extension	5a. Invalid source file 5b. Select a valid source file and click “Create Dataset”.

*Table 2- Use case scenario for extracting streaming data*

Use case Name	Train the model
Pre –Condition	Internet connection is active. User must be registered with the system. User must have created dataset.
Actor	Registered user
Main Success Scenarios	1. Click “Add project”. 2. Give a desired name. 3. Select previously created dataset. 4. Click “Create Project”. 5. Click “Create Analysis”. 6. Select k- means clustering algorithm to be used. 7. Select Hyper- parameters according to k- means clustering algorithm. 8. Click “Build” icon to train the model
Extension	3a. Empty Dataset 3b. Go to previous level and create a dataset.

*Table 3 - Use case scenario for train the model*

Use case Name	Evaluate the model and predict
Pre –Condition	Internet connection is active. User must be registered with the system. User must have created dataset. User must have trained the model.
Actor	Registered user.
Main Success Scenarios	1. Click “Projects”. 2. Click “Models”. 3. Click “Predict” icon. 4. Select “Featured value” as prediction source. 5. Click on “Predict” to get the prediction.

*Table 4 - Use case scenario for Evaluate the model and predict*



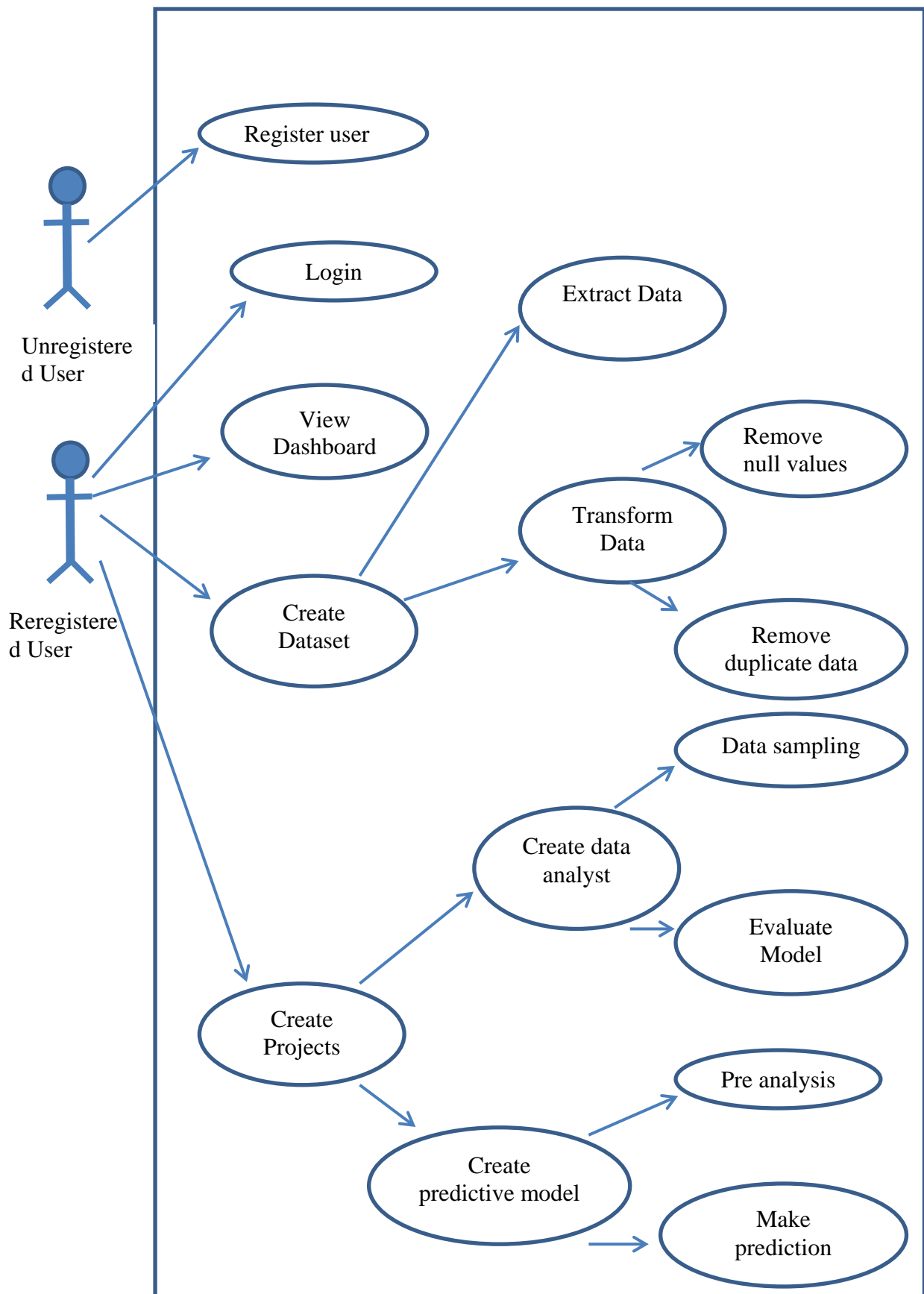


Figure 8 - Use case diagram

### 2.3. User characteristics

User	Privilege	Activities
Unregistered user		Register with the system
Registered user	Full Access to feeding streaming data to System and Visualization.	1. Extract Streaming data. 2. Cleansing of data with necessary attributes and get it transformed. 3. Load the data set to Machine Learner 4. Interpret with the dashboard.

*Table 5 - User Characteristic*

### 2.4. Constraints

- A Major constraint that will be facing is the limitation of available time. The project group is expected to complete the project within 8 months.
- All the tools and technologies used for the development should be open source.

### 2.5. Assumptions and dependencies

- All the users have basic knowledge using computer and internet.
- Apache Server is up and running 24x7.
- There's sufficient memory and processing power in all user PC's.

## **2.6. Apportioning of requirements**

The requirements described in sections 1 and 2 of this document are referred to as primary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the application will be built from functional specification since it is more detailed.

## **3. Specific requirements**

### **3.1. External interface requirements**

#### **3.1.1. User interfaces**

- **Streaming Data Extraction Interface:**

This is the first stage of the data acquisition process. The relevant streaming data should be loaded into the database which has collected using APIs. The source file should be in from a permitted file type. Then name the dataset and name the version of the dataset and finally, click create dataset to start the data extraction process.

- **View Data Interface:**

In this interface user can see the actual data in a more attractive way. Collected streaming data will be represented in Scatter plot, Trellis Chart and Cluster diagram. User can either transform the displayed data or analyses them from there.

- **Data Transformation Interface:**

This is where the data transformation is done. First the user must select the name of the field which should be transformed. Then the form of transformation needs to be specified. Transformation can be in different forms like correcting data that is incorrect, out-of-date, redundant, incomplete, or formatted incorrectly. Finally user has to select the date range which the data should be.

- **Evaluating model and predicting Interface:**

This interface is used to apply the algorithm to the dataset which has transformed earlier. User can select the k-means clustering algorithm to apply the evaluation to the previously created analysis. When create the analysis machine will get trained

according that that dataset. User can change hyper parameters according to k- means clustering algorithm and generate samples from a whole population data. Finally using the predict interface user can select the project where they were working on (which has created the dataset), select the model and then predict the value.

- **Data Visualization Interface**

Finally the predictive value will be represented in Scatter plot, Trellis Chart and Cluster diagram. User can make decisions by exploring these diagrams.

### **3.1.2. Hardware interfaces**

No special hardware interfaces are used for the system.

### **3.1.3. Software interfaces**

- Oracle Java SE Development Kit (JDK) 1.6.24 or later / 1.7.\*  
To launch the product as each product is a Java application
- Apache ActiveMQ JMS Provider 5.5.0 or later  
To enable the product's JMS transport and try out JMS samples. The ActiveMQ client libraries must be installed in the product's classpath before you can enable the JMS transport.
- Apache Ant 1.7.0 or later  
To compile and run the product samples
- SVN Client  
To check out the code to build the product from the source distribution. If you are installing by downloading and extracting the binary distribution instead of building from the source code, you do not need to install SVN.
- Apache Maven 3.0 or later  
To build the product from the source distribution (both JDK and Apache Maven are required). If you are installing by downloading and extracting the

binary distribution instead of building from the source code, you do not need to install Maven.

- JavaScript enabled Web Browser

To access each product's Management Console. The Web Browser must be JavaScript enabled to take full advantage of the Management console.

- MySQL

My-SQL is used as the database management system. As the project is basically based on data mining extraction and transformation, My-SQL would be used regularly for major operations of the system.

#### **3.1.4. Communication interfaces**

- **Internet:**

Since the system use online data (Streaming data) to make predictions to get the data system require internet. Because the system use APIs to get streaming data. So in order to access the APIs the system requires an internet connection.

- **Database connection interface:**

Database connection interface is used to exchange data between the application and the database. It acts as the adapter which converts the database queries into application data and vise-versa.

### 3.2. Classes/Objects

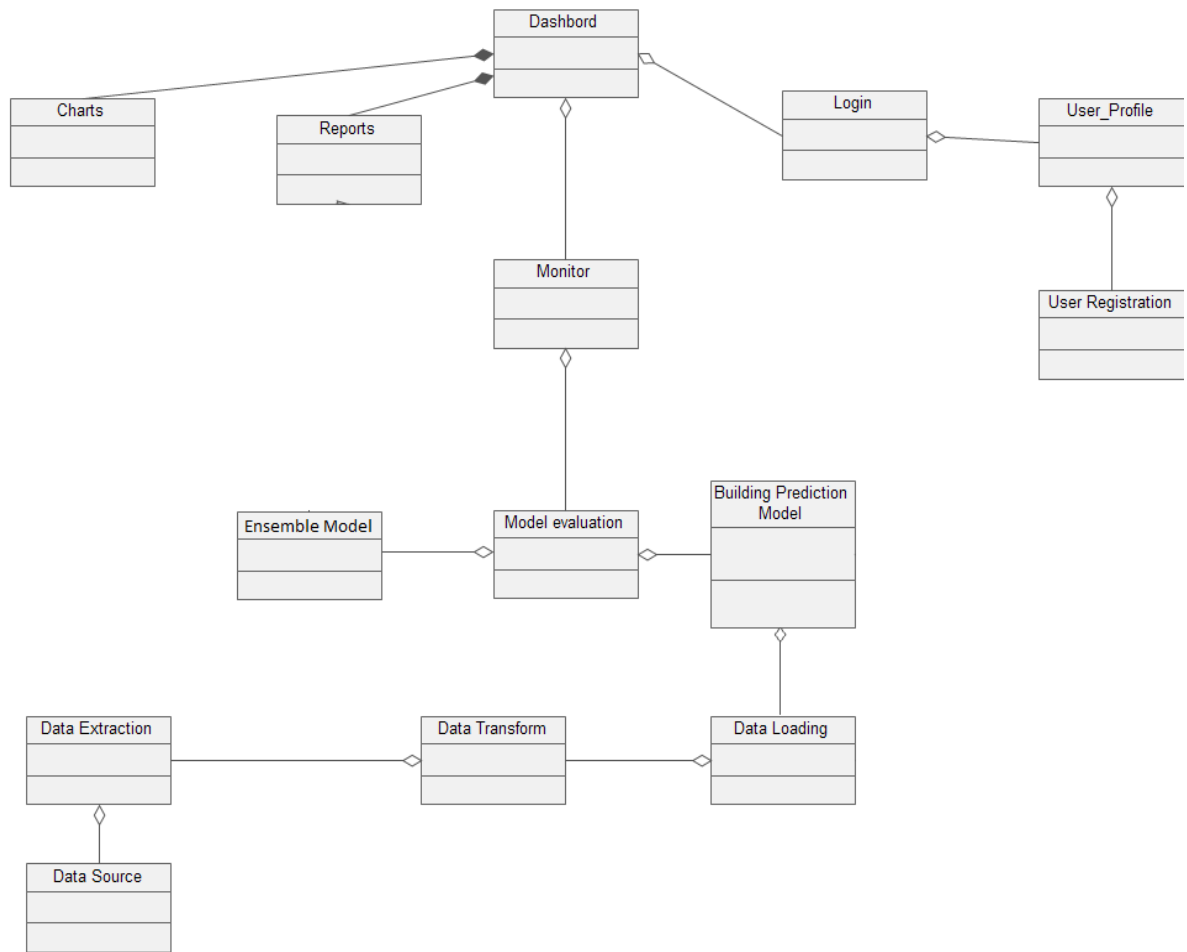


Figure 9 - Class Diagram

### 3.3. Performance requirements

The development team expected that the proposed system will perform all the functions as stated under functional requirements. Some identified performance requirements are listed below:

- The system should be able to accommodate a minimum of 50,000 records in the database.
- Predictive list should be generated within 1 hour.
- Based on the derived predictive lists, reports must be generated within 10 seconds.

### **3.4. Design constraints**

- A Major constraint that will be facing is the limitation of available time. The project group is expected to complete the project within 8 months.
- The predictive tool will be only focusing on several fields (at least) for this 10 month implementation phase.
- Predictive model building, evaluation, testing and all other operations will be carried out using some old datasets.

### **3.5. Software system attributes**

#### **3.5.1. Reliability**

Reliability is the ability of the management system with the minimum number of defects. The system has to go through a proof of this is the application must prove by fixing each error. Each component will be tested and finally integrated system will also be tested and ensure that the desired result is achieved. The output of the system must be tested to ensure that the result is significant. Since the proposed system strengthens and supports decision making .Hence system reliability is expected.

#### **3.5.2. Availability**

WSO2 Machine Learner (ML) can be accessed at any time even intended users ranging from superior to other decision makers -level management. All and information needed for the requested user can see. There is no restriction of time for users to access the system WSO2 Machine Learner (ML) according to their access privileges.

#### **3.5.3. Security**

Safety is a fundamental non-functional requirement of any system. This should keep all critical data safely. So good policies need to be developed and followed. The data are classified according to different requirements of safety and handling. All critical data is secured during storage and transmission through appropriate access controls.

#### **3.5.4. Maintainability**

High maintainability is one of the key virtues of stable and highly productive products. Even in product implementation of the proposed machine Learner (ML), we are more focused on creating very easy to maintain system. Standards of coding practices will be followed throughout the implementation of the system and minimize errors as much as possible. Each time new requirements arrive, the system can be modified to accommodate these requirements by maintaining system stability.

#### **3.6. Other requirements**

- Data acquisition:

The data for the predictive model building, evaluation and testing should be acquired from valid sources, otherwise it can affect the accuracy of the model.

- Use of source open source technologies:

The product should be developed only using open source technologies.



## 4. Supporting information

### 4.1. Appendices

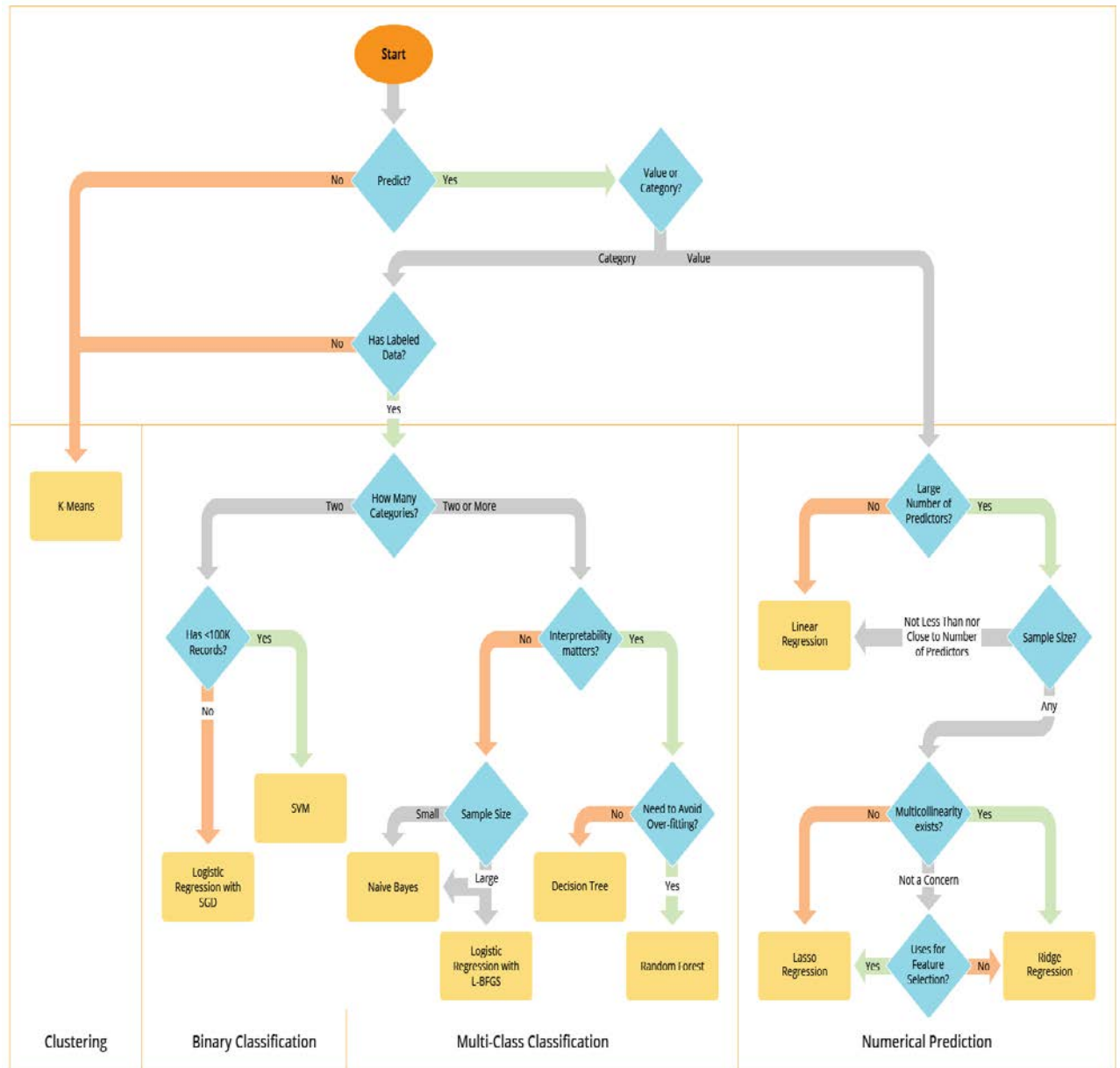


Figure 10 - how to select the algorithm

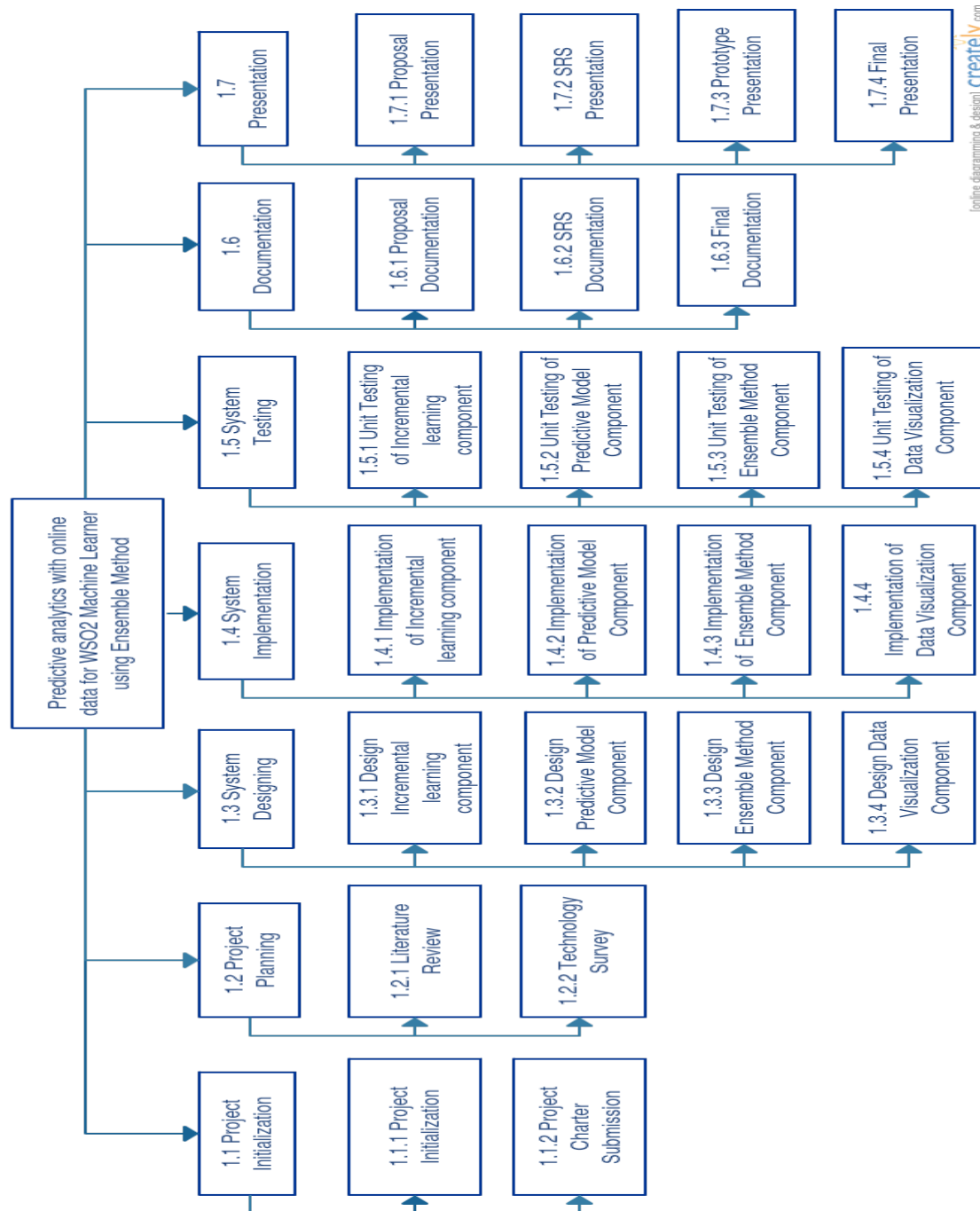


Figure 11- Work Breakdown Structure

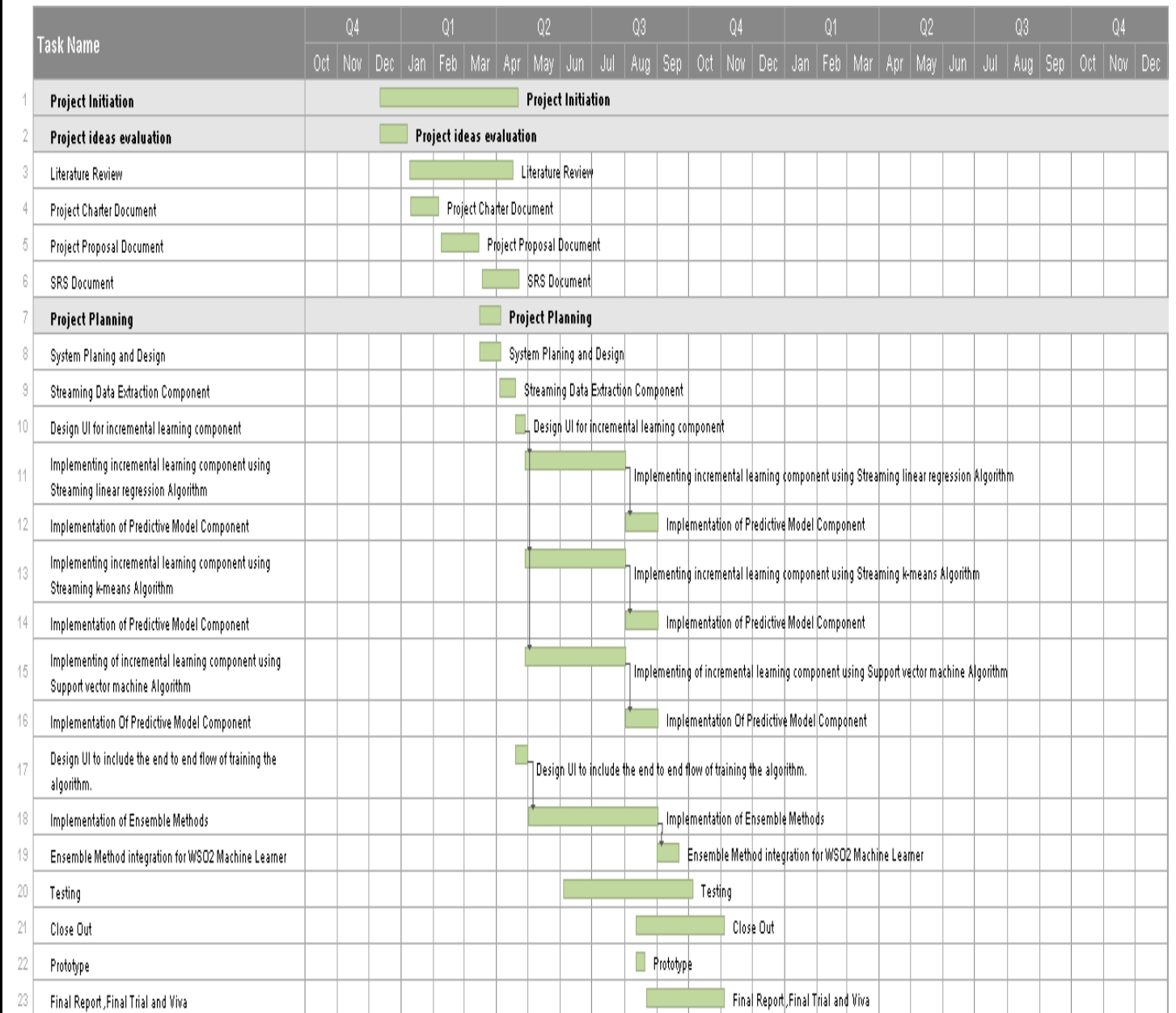


Figure 12 - Gantt Chart

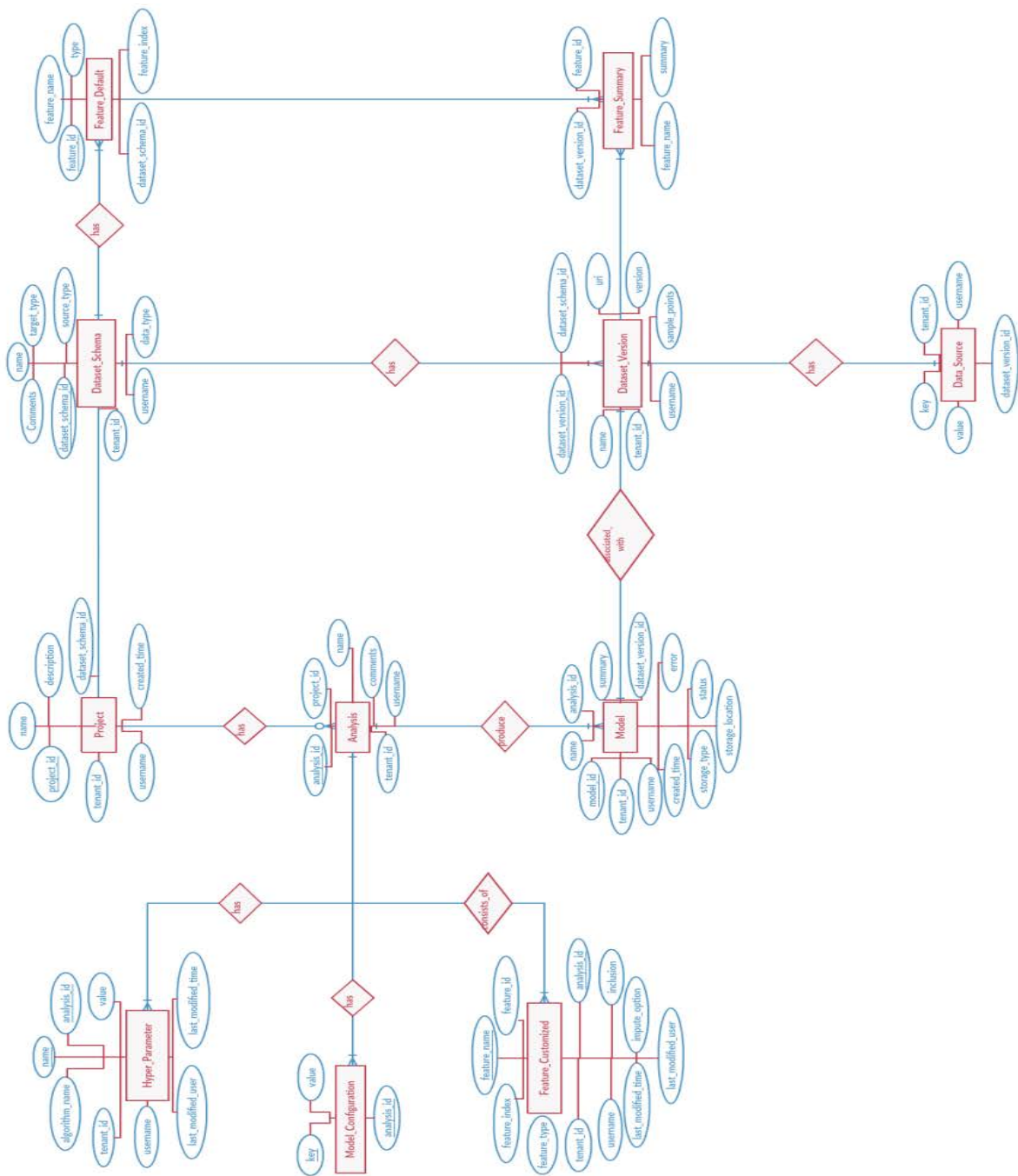


Figure 13 - WSO2 machine learner data base ER diagram

## 4.2. References

[1] [Online] Available:

[https://docs.wso2.com/display/GSoC/Project+Proposals+for+2016#ProjectProposalsfor2016-Proposal6:\[ML\]PredictiveanalyticswithonlinedataforWSO2MachineLearner](https://docs.wso2.com/display/GSoC/Project+Proposals+for+2016#ProjectProposalsfor2016-Proposal6:[ML]PredictiveanalyticswithonlinedataforWSO2MachineLearner)

[Accessed: March 29<sup>th</sup>, 2016]

[2] [Online] Available:

<https://docs.wso2.com/display/APPM100/Installation+Prerequisites> [Accessed:

March 29<sup>th</sup>, 2016]

[3] [Online] Available: <http://dev.datasift.com/docs/api/streaming-api> [Accessed:

March 28<sup>th</sup>, 2016]

[4] [Online] Available: <http://activemq.apache.org/> [Accessed: March 29<sup>th</sup>, 2016]

[5] [Online] Available: <http://whatis.techtarget.com/definition/machine-learning>

[Accessed: March 28<sup>th</sup>, 2016]

[6] [Online]. Available: <https://docs.wso2.com/display/ML110/Features> [Accessed:

March 29<sup>th</sup>, 2016]

[7] [Online]. Available:

<http://www.laits.utexas.edu/~anorman/BUS.FOR/course.mat/Alex/> [Accessed: March 27<sup>th</sup>, 2016]

[8] [Online]. Available:

<http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm> [Accessed: March 28<sup>th</sup>, 2016]

[9] [Online]. Available: <https://spark.apache.org/docs/1.4.1/mllib-linear-methods.html#streaming-linear-regression>

[Accessed: March 30<sup>th</sup>, 2016]

[10] [Online]. Available: <https://spark.apache.org/docs/1.4.1/mllib-clustering.html#streaming-k-means>

[Accessed: March 31<sup>st</sup>, 2016]

[11] [Online]. Available: <https://spark.apache.org/docs/1.4.1/mllib-guide.html>

[Accessed: March 28<sup>th</sup>, 2016]