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

Software Requirements Specification

Project ID:16-054

Author:
IT13028756
K. Indujayani

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 : IT13028756 – K.Indujayani

Signature : ………………………………….

Table Of Content

List Of Figures

List Of Tables

1 Introduction

1.1 Purpose

This document basically focuses on providing entire requirement set about creating Incremental Learning Component using Streaming Linear Regression Algorithm [1] for WSO2 [2] Machine Learner [3]. The purpose of this SRS document is to provide a detail overview of project software product, its parameters and goals. This document is targeting the client, designers, developers and other interested parties as its audience. A description of the problem to be solved will be provided relating what the system would do in order to overcome the problem. How the system performs its tasks will not be uncovered. This document will serve the purpose of providing the potential users to determine if the software specified meets their needs or how the software must be modified to meet their needs.

1.2 Scope

This SRS includes the requirements for the initial release of the proposed WSO2 Machine Learner. 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 WSO2 Machine Learner aims at getting Streaming data using APIs[4] and creating a Incremental Learning Component using Streaming Linear Regression Algorithm[5]. Incremental Learning Component can train the machine to identify patterns in the recent history and updating the patterns with incoming data without catastrophic forgetting. MLlib[6] currently supports streaming linear regression using ordinary least squares. MLlib implements a simple distributed version of stochastic gradient descent (SGD)[7]. Stochastic gradient descent based linear models support machine learning [8].

We are living in an era where the technology is at its climax. Most problems nowadays can be solved with the knowledge from Computer Science and Informatics If we are capable of predicting the future incidents just using a machine that will be really useful in many streams. Proposed WSO2 Machine Learner make it possible to

reach unambiguous decisions in the right time by providing an accurate prediction using gathered streaming data.

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

Term	Definition
ML	Machine Learner
OS	Operating System
API	Application Programming Interface
SQL	Structured Query Language
SRS	Software Requirements Specification
IDE	Integrated Development Environment
XML	Extensible Markup Language
JDK	Java Development Kit
UI	User Interface
JVM	Java Message Service
WEKA	Waikato Environment for Knowledge Analysis
HPCC	High Performance Computing Cluster
CEP	Complex Event Processor
SGD	Stochastic Gradient Descent
DAS	Data Analytics Server
ESB	Enterprise Service Bus

Table 1: Definitions, Acronyms, and Abbreviations

1.4 Overview

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

The first chapter explains the purpose of preparing this document. Scope describes clearly what the project will and will not do. In the overview it explains how the SRS is organized and describes what the rest of this document contains briefly.

Chapter 2 describes the overall description in a non-technical way which is understandable by the user. It includes product perspective, product functions, user characteristics, constraints, assumptions and dependencies and apportioning of requirements. Main target of the product perspective is to find whether the existing system is available in regard for the developing application. Product functions are also described as a summary of all major functions of the application. In user characteristics, it describes the kind of people who the application is intended to be used. In constraints sub section it describes all condition that may limit developer's options. Assumptions and dependencies describe that any assumptions being made while the apportioning of requirements describes the order in which the requirements are to be implemented when developing the application.

Under 3rd chapter it describes the developer's point of view of proposed WSO2 Machine Learner. It uses technical words or phrases understandable by the software engineers, developers, and testers. External interface requirements, performance requirements, design constraints, application attributes and other requirements are also explained in advance

Chapter 4 describes the information that is useful for the readers. This gives all the supporting information. It also contains additional diagrams and instructions of the code. Those are not directly relevant to the project but can use as additional information for readers.

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.

2 Overall Descriptions

WSO2 Machine Learner has a versatile set of characteristics. It can extract features from a dataset made available from a file system, Hadoop Distributed File System (HDFS) or WSO2 Data Analytics Server (DAS). This data is passed on to the Machine Learner Core, which allows you to explore your datasets, pre-process your data and apply various machine learning algorithms to make sense out of it all. Using Apache spark[9], it then analyzes and builds models with the chosen algorithms. Recommendation engine capability allows you to provide product recommendations for users.

As mentioned in the abstract there are two main objectives that we hope to achieve at the end of this research.

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

These objectives can be approached in two methods.[10]

- 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

The goal of WSO2 Machine Learner is to make machine learning accessible to WSO2 Data Analytics Platform. The expected outcome of this process is a deployable model in many of WSO2 products such as WSO2 Enterprise Service Bus (ESB), WSO2 Complex Event Processor (CEP). As a latter part of this project we hope to implement Ensemble methods support for WSO2 Machine Learner. The WSO2 Machine Learner can be utilized to build machine learning models for various tasks such as fraud detection, anomaly detection, classification etc. It will also come in handy for developers and data scientists to implement machine learning algorithms quickly. WSO2 Machine Learner is built up on top of the WSO2 Carbon platform, which is based on the Open Service Gateway Initiative (OSGI) framework enabling better modularity for your service oriented architecture (SOA) and all its operations are exposed through a RESTful API. WSO2 Machine Learner is released under Apache

Software License Version 2.0, one of the most business-friendly licenses available today [3].

The prominent features of WSO2 Machine Learner are as follows[12]:

- Data Exploration

- Model generation

- Model comparison

- Prediction

Data exploration focuses on the ability to convert datasets into a format which is suitable for machine learning algorithms (data preprocessing) and ability to do explorative data analysis. Model generation supports various machine learning algorithms in classification, numerical prediction and clustering spaces. Also it generates models and persist them or download them via WSO2 machine learning REST API. At the same time it has the ability to examine the characteristics of a model. Model comparison comprises of figures that help you to compare models across a machine learning project.In Prediction the generated models can be used to perform predictions using a REST API client, WSO2 Enterprise Service Bus (ESB) mediator or WSO2 Complex Event Processor (CEP).

Proposed WSO2 Machine Learner aims at getting Streaming data using APIs[4] and creating a Incremental Learning Component using Streaming Linear Regression Algorithm[5]. Incremental Learning Component can train the machine to identify patterns in the recent history and updating the patterns with incoming data without catastrophic forgetting. MLlib[6] currently supports streaming linear regression using ordinary least squares. MLlib implements a simple distributed version of stochastic gradient descent (SGD)[7]. Stochastic gradient descent based linear models support machine learning [8].
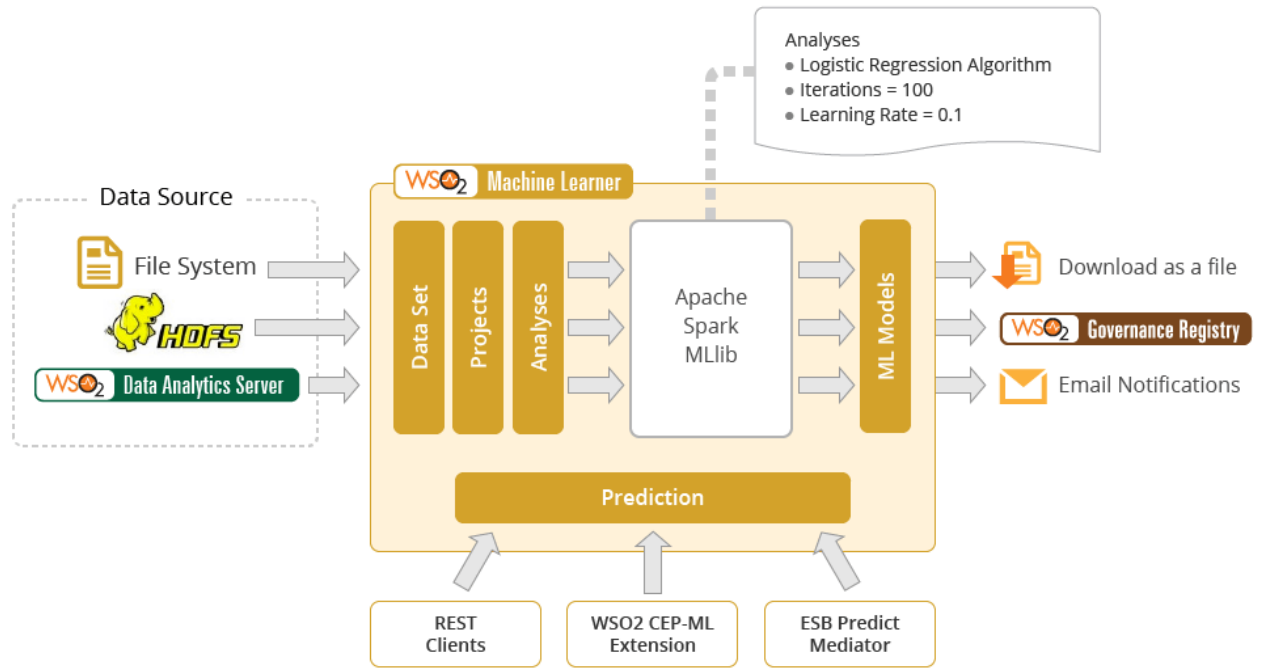
Figure 1: System Overview

2.1 Product perspective

Features	WEKA	HPCC System	Existing WSO2 ML	Proposed WSO2 ML
User Friendly Interfaces	✔	✔	✔	✔
Outlier detection when generating graphs	✗	✗	✗	✔
Model Generation	✗	✗	✔	✔
Can configure with spark	✗	✗	✗	✔
Display prediction result rather than value	✗	✗	✗	✔
Validate File Formats	✗	✗	✗	✔

Features	WEKA	HPCC System	Existing WSO2 ML	Proposed WSO2 ML
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 2: Product perspective
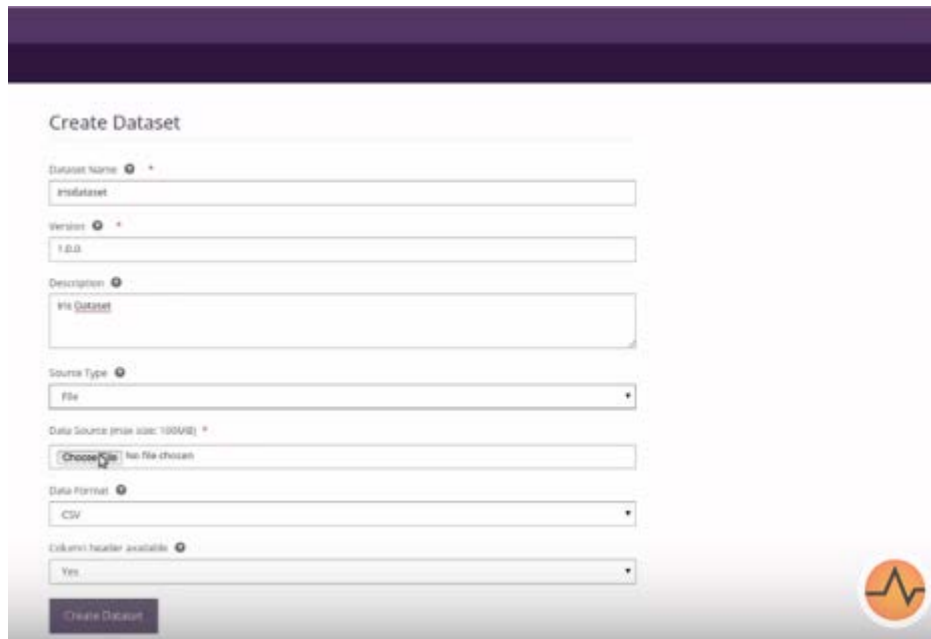
2.1.1 System interfaces

- Data Extraction Interface.
- Web – Desktop connectivity Interface.
- JVM

2.1.2 User interfaces
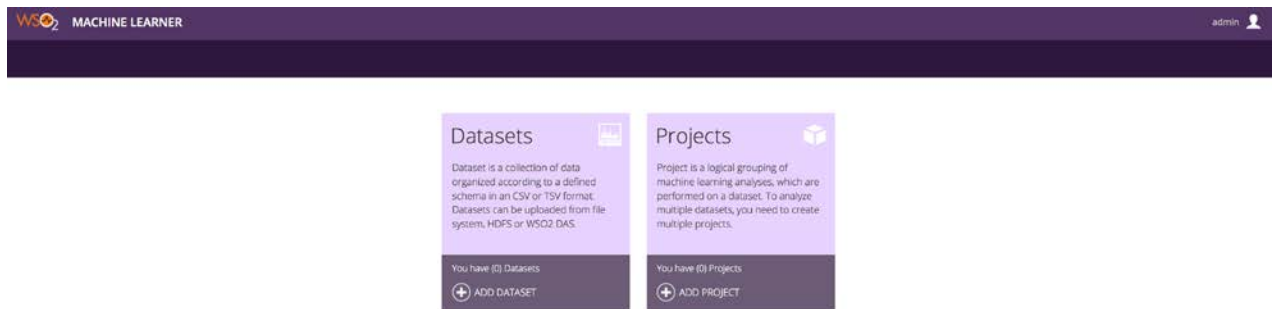
Figure 2 :WSO2 Machine Learner - Dashboard
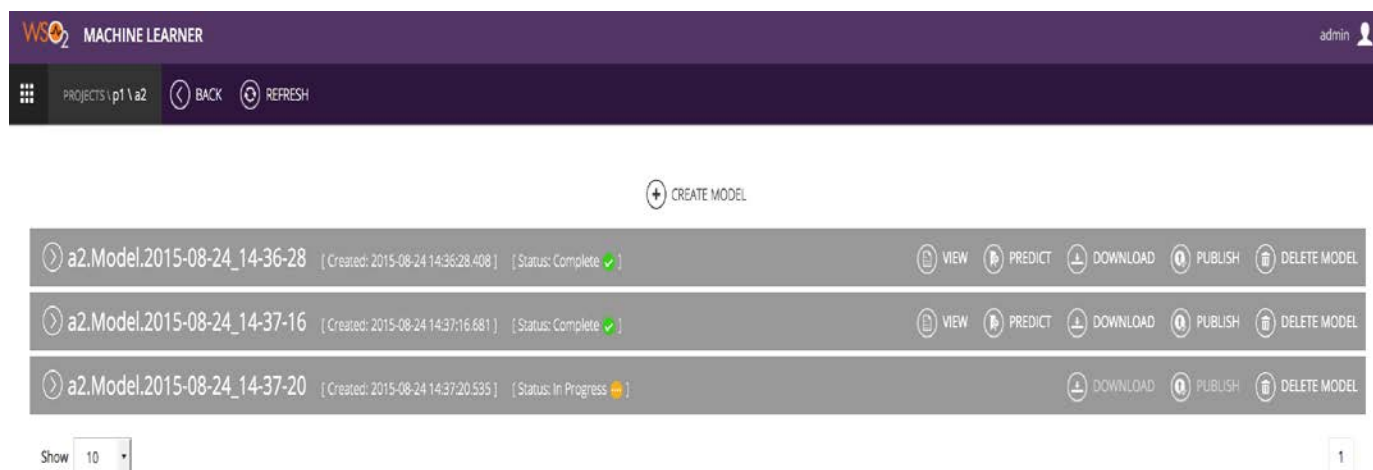
Figure 3: Data Extraction Interface

Figure 4: Created Models Interface

13
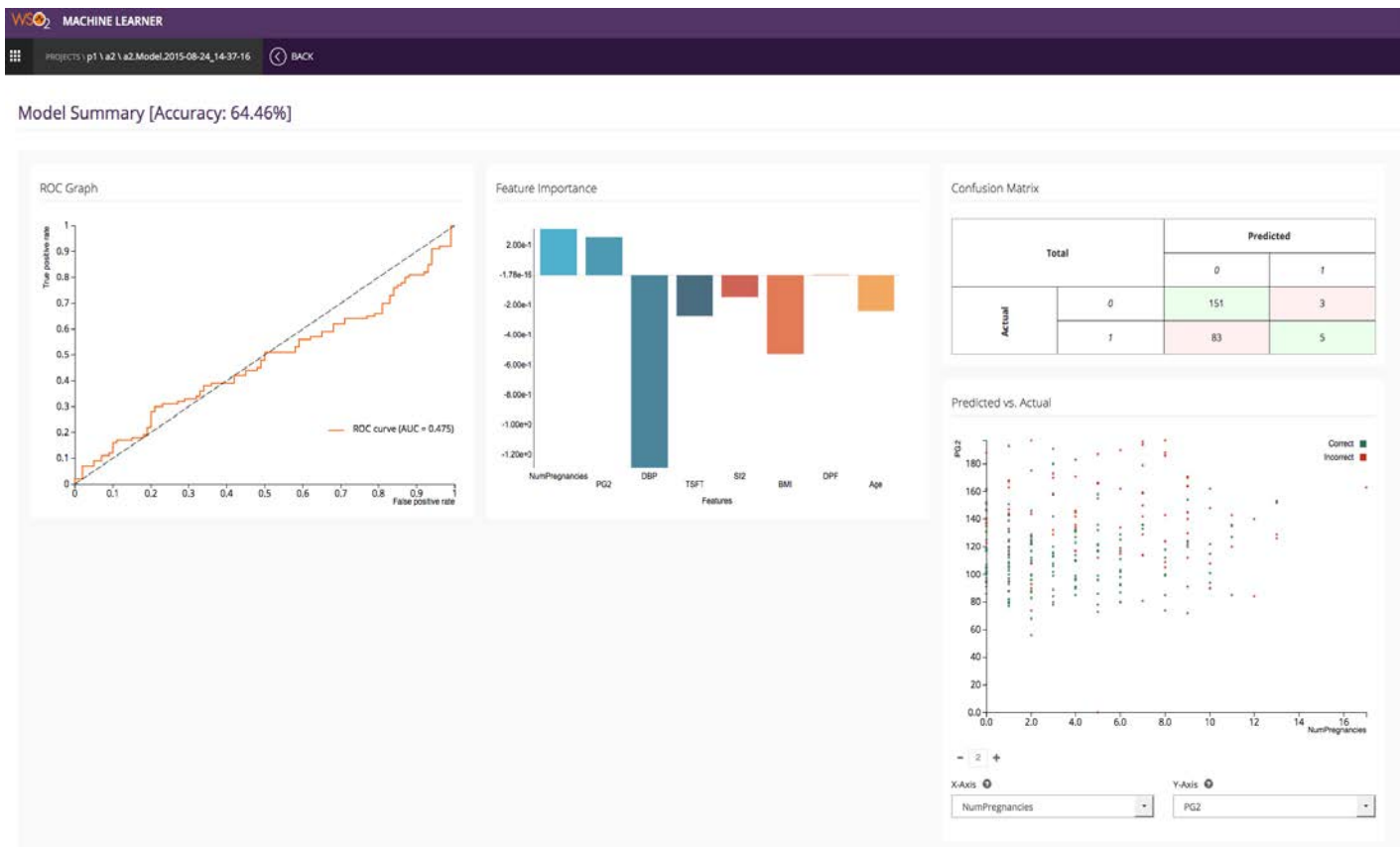
Figure 5: *Prediction Result Interface*
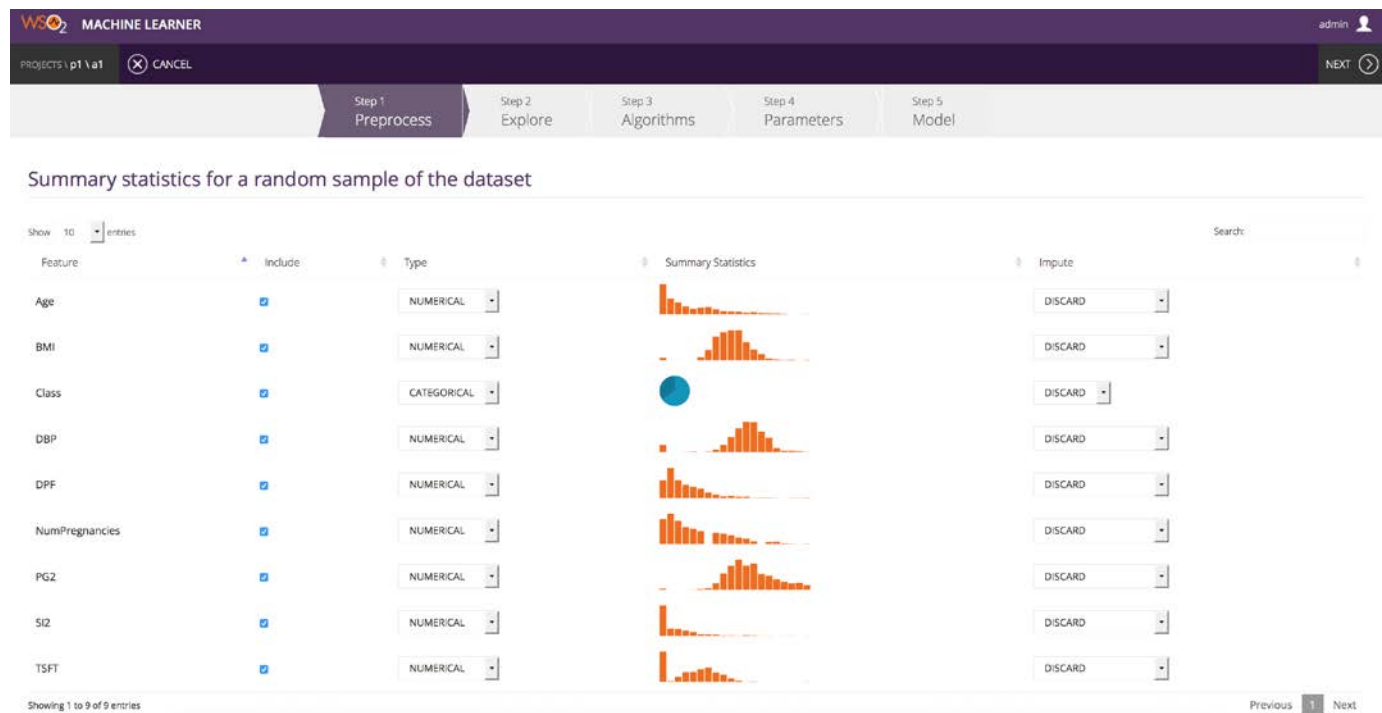
Figure 6: Summery Report Interface

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) version 1.6.24 or later or 1.7.*

- Apache ActiveMQ JMS Provider 5.5.0 or later

- Apache Ant 1.7.0 or later

- SVN Client

- Apache Maven 3.0.*

- Web browser – Java script enabled

2.1.5 Communication interfaces

- Internet

- Database connection Interface

2.1.6 Memory constraints

- RAM of 2GB or higher

2.1.7 Operations

- Extract Streaming data from given source.

- Cleansing of data with necessary attributes and get it transformed.

- Change Parameters of the Algorithm on the fly.

- Train the machine according to the data source.

- Produce a prediction result.

2.1.8 Site adaptation requirements

- Server must have MySQL server installed on it.
- Server machine must be running Apache server in order to deploy the web application.
- Oracle Java SE Development Kit (JDK) must be installed.
- The user machine should have Java Virtual Machine installed.
- Web Browser should be java script enabled.

2.2 Product functions

Use case Name	Extract Streaming Data
Pre –Condition	Database connection is active.
Actor	Administrator / Logged User
Main Success Scenarios	1. Select the source file. 2. Select the target database. 3. Provide a table name. 4. Click "Extract".
Extension	4a. Invalid source file 4b. Select a valid source file and click "Extract".

Table 3: Use Case Scenario for Extract Streaming Data

Use case Name	Cleanse Data
Pre –Condition	Database connection is active.
Actor	Administrator / Logged User
Main Success Scenarios	1. User select the data 2. Select transformation type 3.Select Date range 4.Click "Transform"
Extension	4a. User press "Cancel" 4b. Cancel the transformation process.

Table 4: Use Case Scenario for Cleanse Data

16

Use case Name	Train Machine
Pre –Condition	Database connection is active.
Actor	Administrator / Logged User
Main Success Scenarios	1. Select data 2. Select incremental learning component using Streaming Linear Regression Algorithm 3. Change Parameters 4. Click "OK"
Extension	4a. User press "Cancel" 4b. Cancel the transformation process.

Table 5:Use Case Scenario for Train Machine

Use case Name	Produce prediction result
Pre –Condition	Database connection is active.
Actor	Administrator / Logged User
Main Success Scenarios	1. Select data 2. Select incremental learning component using Streaming Linear Regression Algorithm 3. Change Parameters 4. Click "OK"
Extension	4a. User press "Cancel" 4b. Cancel the transformation process.

Table 6:Use Case Scenario for produce prediction result
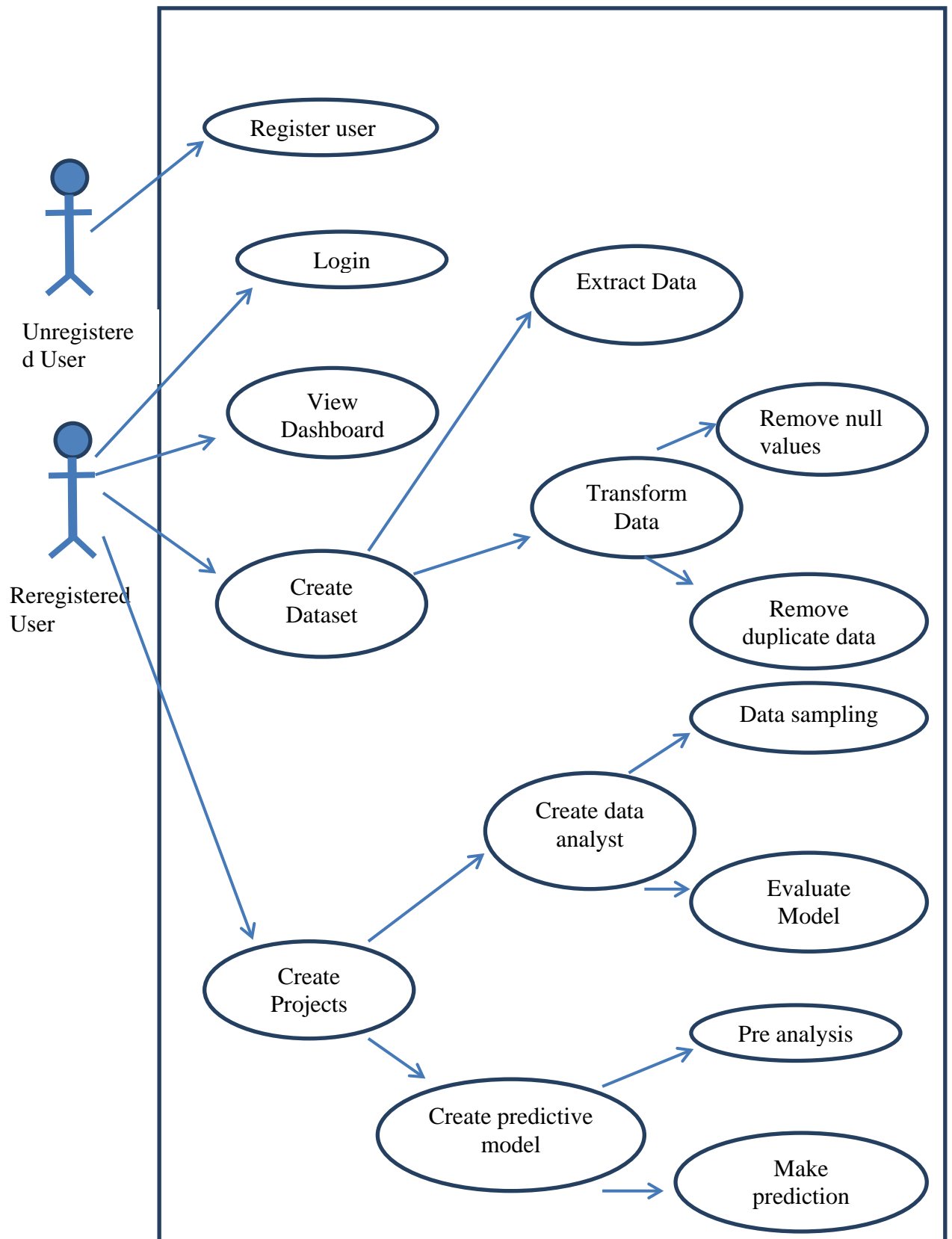
Figure 7: Use Case Diagram

2.3 User characteristics

User	Privilege	Activities
Registered User/ Administrator	Full Access to feeding Streaming data to System and visualization.	1. Extract Streaming data from a given source. 2. Cleansing of data with necessary attributes and get it transformed. 3. Load the data set to Machine Leaner 4. Interpret with the dashboard.
Unregistered User		1. Register as a user

2.4 Constraints

- All the tools and technologies used for the development should be open source.
- Machine Leaner supports only English language.

2.5 Assumptions and dependencies

- All the users have basic knowledge using computer and internet.

- There is an active internet connection.

- 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

- **Data Extraction Interface:**

This is the first stage of the data acquisition process. The relevant data should be taken using Streaming data APIs. Then the targeted database must be selected where the data is to be extracted. Finally a name for the extracted data table should be provided. After all the details are provided, click extract to start the data extraction process.

- **Data Transformation Interface:**

This is where the data transformation is done. First the user must select data to 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.

- **Train Machine and Predict Result Interface:**

Model is created using gathered streaming data, with the use of Streaming Linear Regression algorithm machine is trained to identify data patterns and accurate prediction value is generated. Along with prediction scatter plot, histogram, trellis chart, parallel set and cluster diagram is created to give a better visualization of data for the user.

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) version 1.6.24 or later or 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.***

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.

- **Web browser – Java script enabled**

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

3.1.4 Communication interfaces

- **Internet:**

In order to access the web application user needs 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
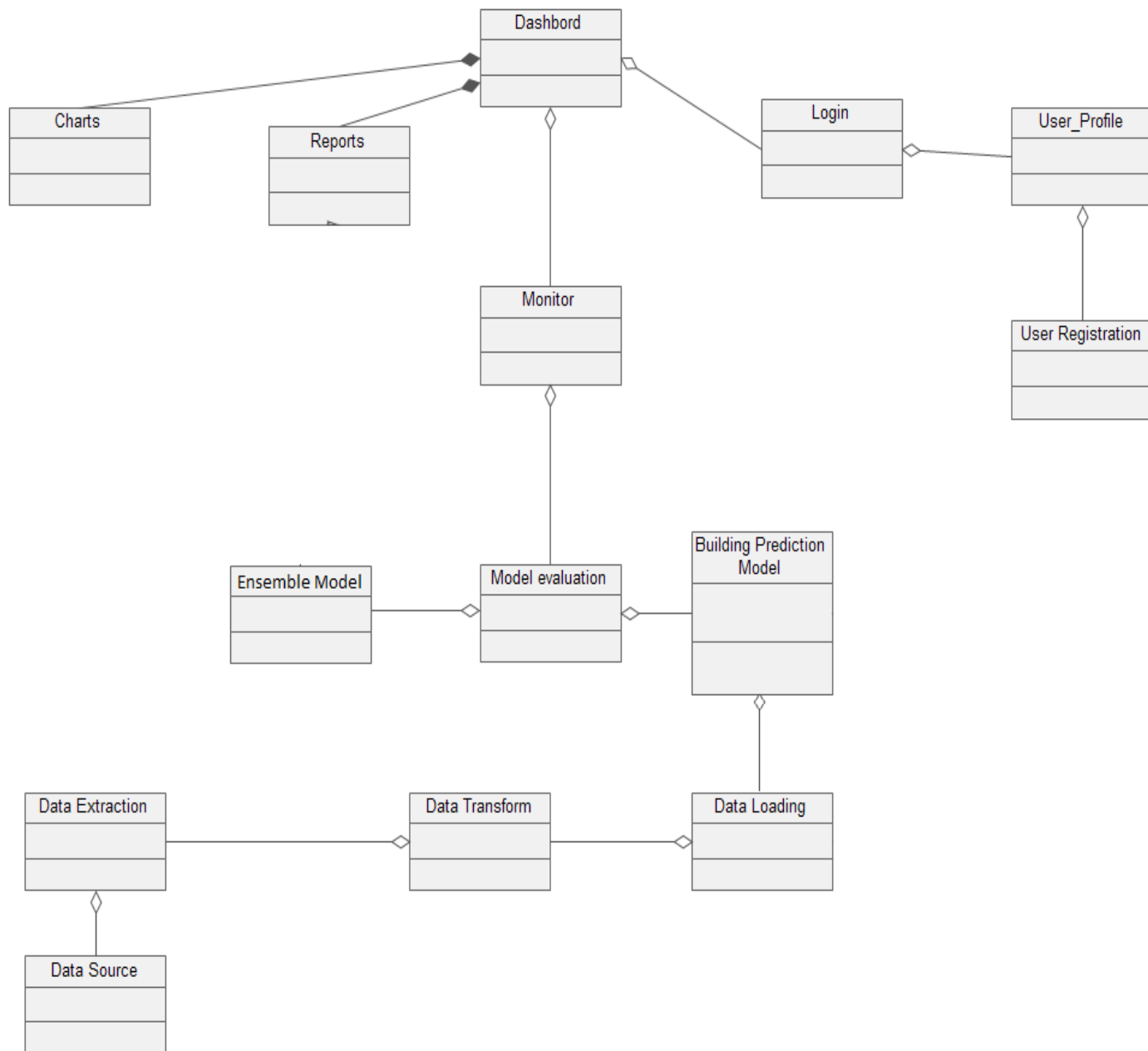
Figure 8: Class Diagram

3.3 Performance requirements

It is expected that the proposed system will perform all the requirements stated under the functional requirements section. Some performance requirements identified is 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

- All the tools and technologies used for the development should be open source.
- Machine Leaner supports only English language.
- 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 system run with minimum number of failures. The system has to go through a testing i.e. application must test by fixing each and every possible bug. Each and every component will be tested and finally integrated system also be tested and to make sure the desired output is obtained. System output also has to be tested to make sure output is meaningful. Since, Machine Leaner is a tool to enhance and support decision maker. Hence reliability of system is much expected.

3.5.2 Availability

Machine Leaner can be access at any time even from intended users that range from top –level management to other decision makers. All and necessary information for user requested can be viewed. There is no time restriction to users to get access to the Machine Leaner system according to their access privilege.

3.5.3 Security

There are different types of users logging into the system such as Administrators, Registered Users. Each and every user have their own access levels within the system and they can only perform tasks according to those access levels. Access to the various subsystems will be protected by a user log in screen that requires a user name and password. Password should be encrypted and store in the database. Integrity of the data must be protected at all cost.

3.5.4 Maintainability

High maintainability is one of the key virtues of stable and highly productive products. Even in product implementation of Machine Leaner, we are more focused on creating highly maintainable system. The standards of the coding practices will be followed throughout system implementation and it will minimize the bugs as much as possible. Whenever there are new requirements, the system can be modified to accommodate these requirements by maintaining the stability of the system.

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.
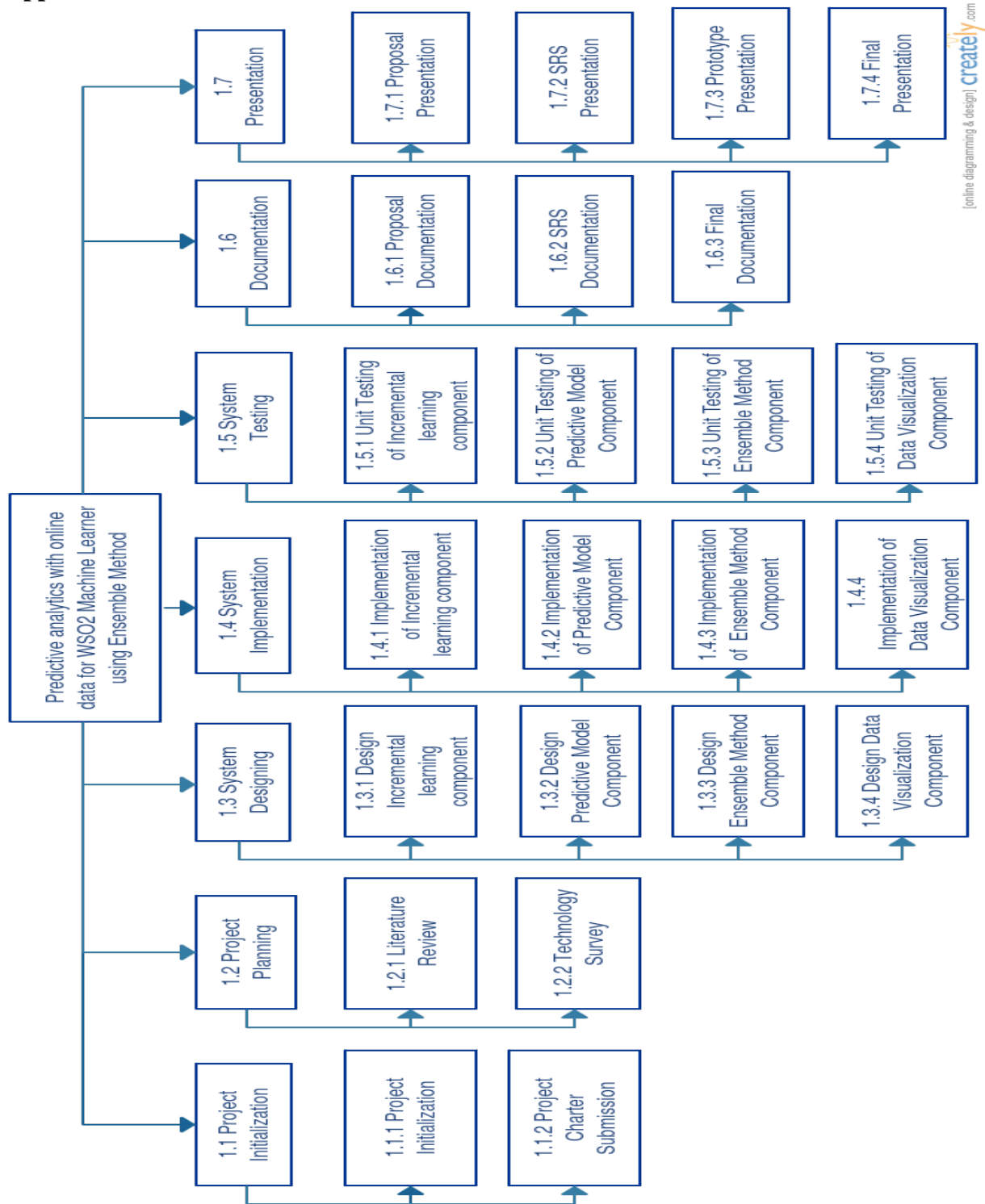
4 Supporting information
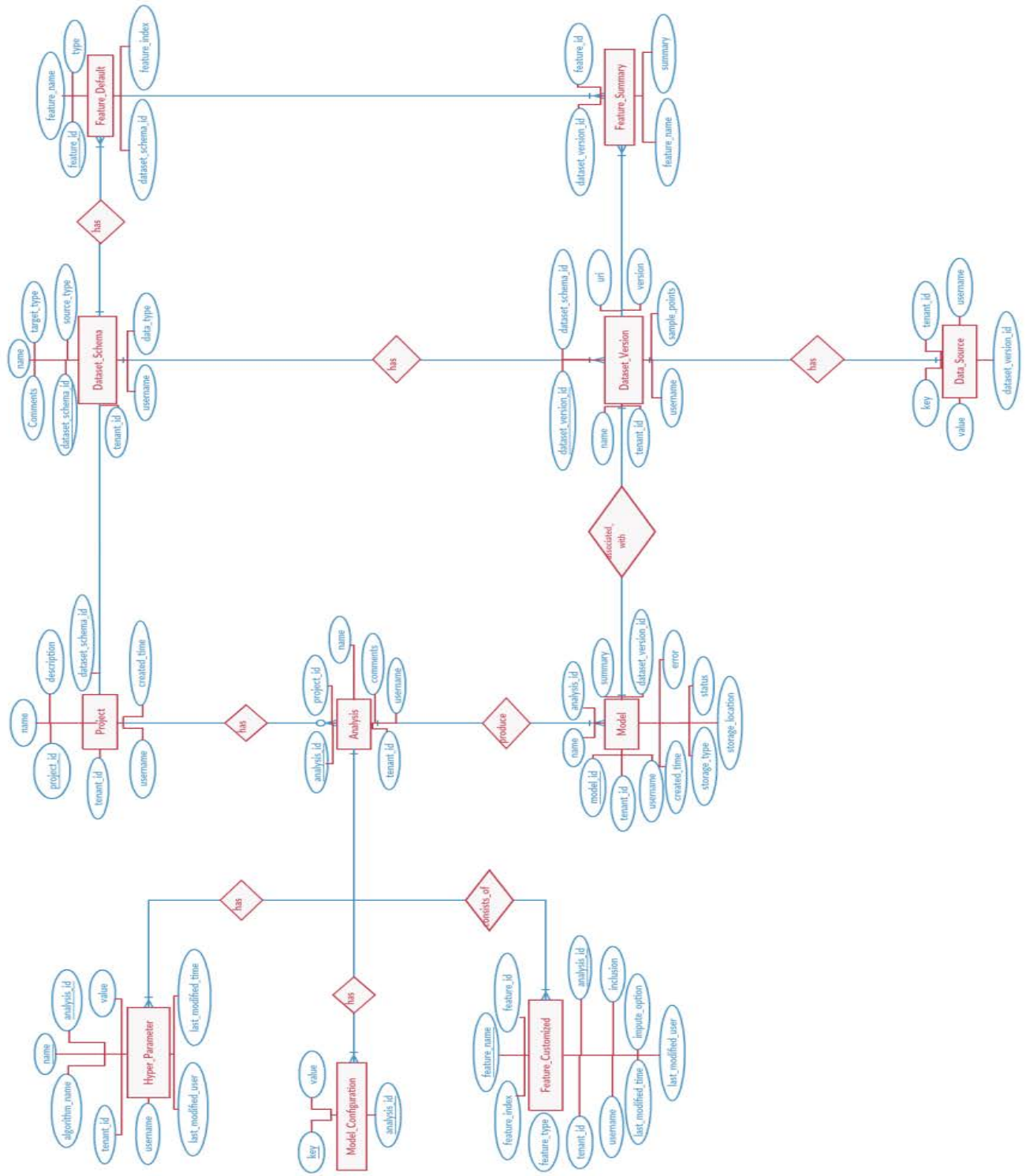
4.1 Appendices

Figure 9:Work Breakdown Structure

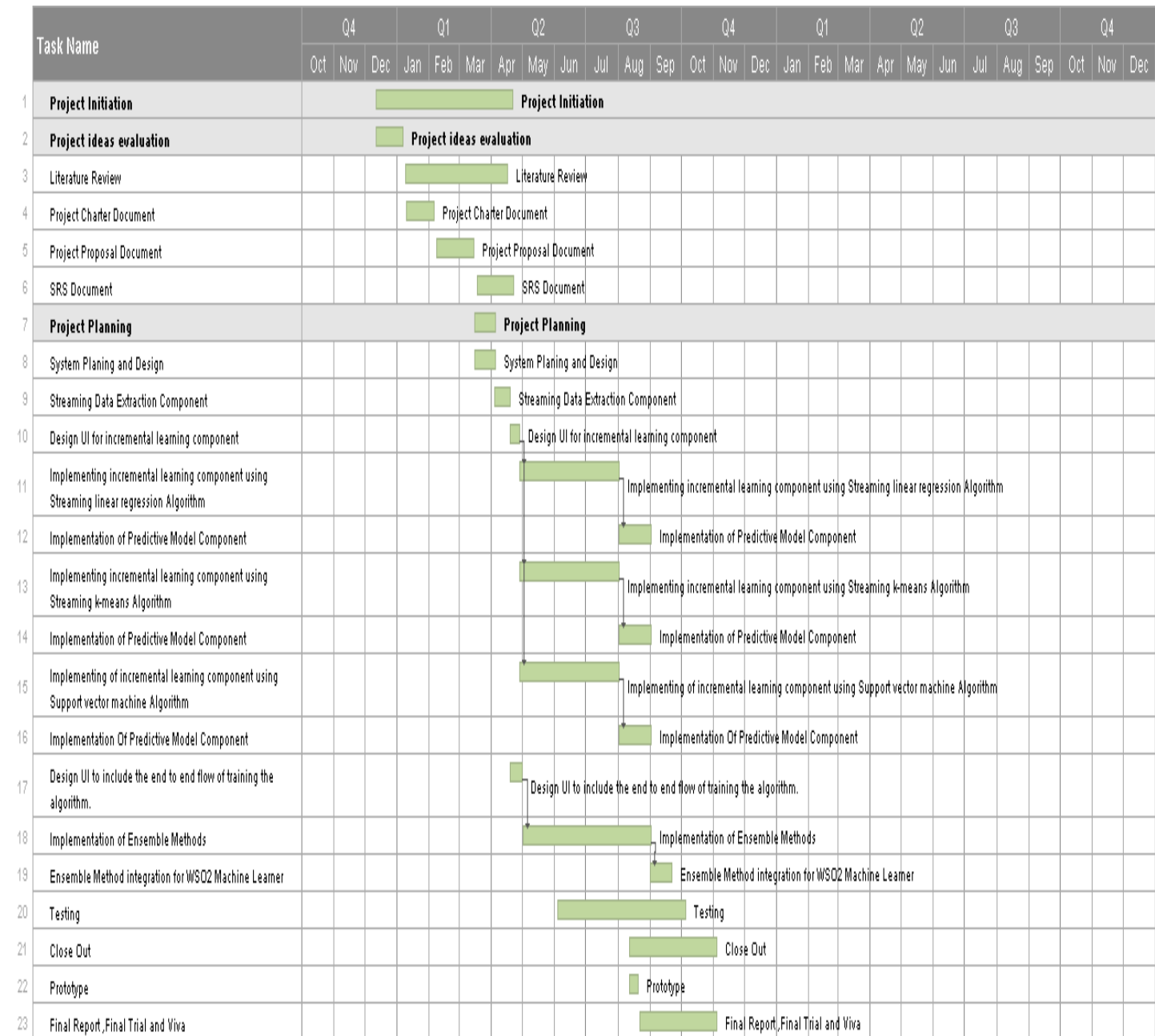*Figure 10:ER Diagram for WSO2 Machine Learner*

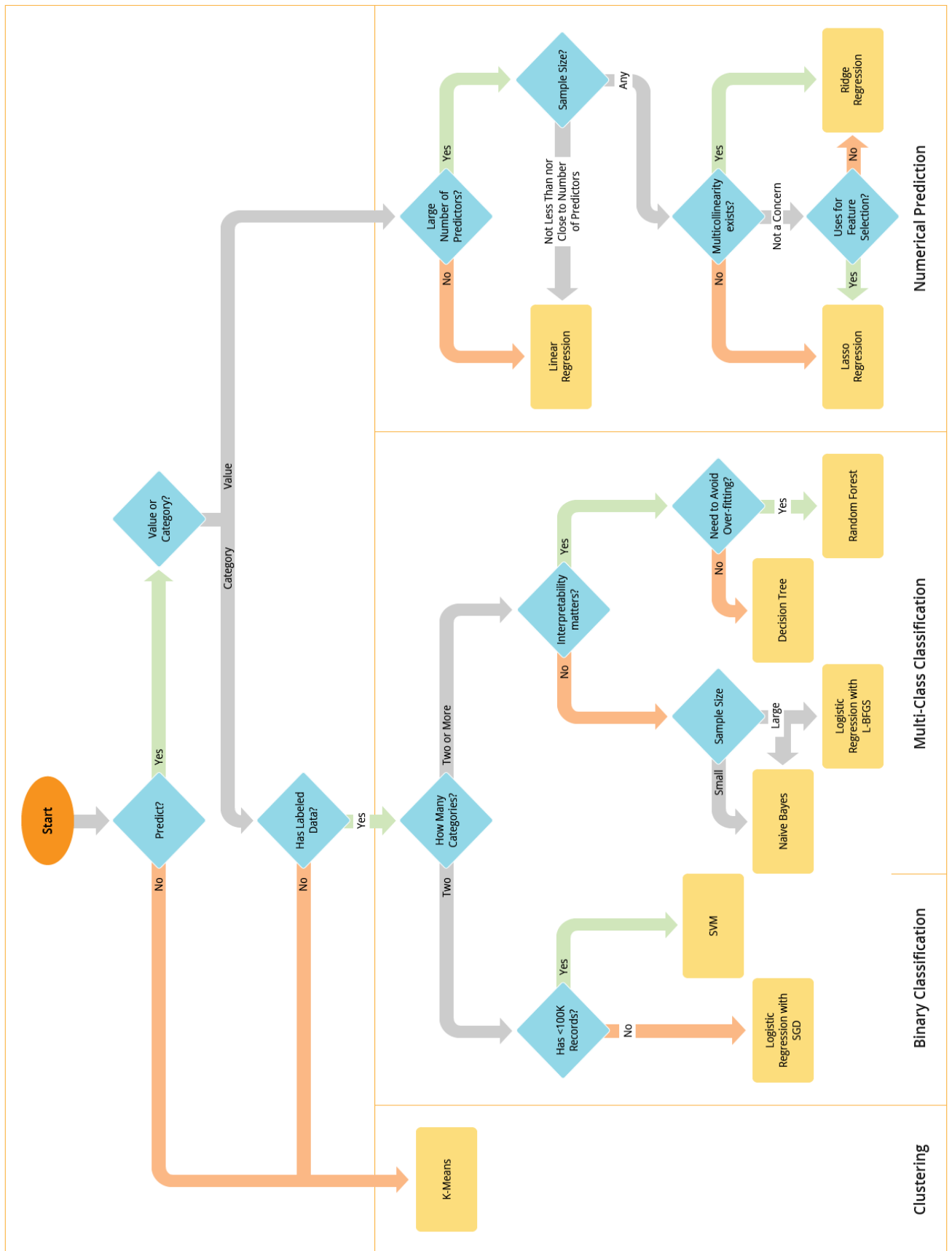| Task Name	Q4			Q1			Q2			Q3			Q4			Q1			Q2			Q3			Q4		
	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1 **Project Initiation**									Project Initiation																		
2 **Project ideas evaluation**						Project ideas evaluation																					
3 Literature Review								Literature Review																			
4 Project Charter Document					Project Charter Document																						
5 Project Proposal Document						Project Proposal Document																				
6 SRS Document							SRS Document																				
7 **Project Planning**							Project Planning																				
8 System Planing and Design							System Planing and Design																				
9 Streaming Data Extraction Component							Streaming Data Extraction Component																				
10 Design UI for incremental learning component							Design UI for incremental learning component																				
11 Implementing incremental learning component using Streaming linear regression Algorithm									Implementing incremental learning component using Streaming linear regression Algorithm																		
12 Implementation of Predictive Model Component										Implementation of Predictive Model Component																
13 Implementing incremental learning component using Streaming k-means Algorithm									Implementing incremental learning component using Streaming k-means Algorithm																		
14 Implementation of Predictive Model Component										Implementation of Predictive Model Component																
15 Implementing of incremental learning component using Support vector machine Algorithm									Implementing of incremental learning component using Support vector machine Algorithm																		
16 Implementation Of Predictive Model Component										Implementation Of Predictive Model Component																
17 Design UI to include the end to end flow of training the algorithm.							Design UI to include the end to end flow of training the algorithm.																				
18 Implementation of Ensemble Methods									Implementation of Ensemble Methods																		
19 Ensemble Method integration for WSO2 Machine Learner										Ensemble Method integration for WSO2 Machine Learner																
20 Testing								Testing																			
21 Close Out										Close Out																
22 Prototype									Prototype																		
23 Final Report ,Final Trial and Viva										Final Report ,Final Trial and Viva																

Figure 11: Gantt Chart

28

Figure 12:How to choose the Algorithm

REFERENCES

[1] [Online] Available: https://spark.apache.org/docs/1.4.1/mllib-linear-methods.html#streaming-linear-regression [Accessed: March 8th, 2016]

[2] [Online] Available: http://wso2.com [Accessed: February 29th, 2016]

[3] [Online]. Available: http://wso2.com/products/machine-learner/ [Accessed: February 29th, 2016]

[4] [Online]. Available: http://dev.datasift.com/docs/api/streaming-api [Accessed: March 1st, 2016]

[5] [Online]. Available: https://spark.apache.org/docs/1.4.1/streaming-programming-guide.html [Accessed: March 1st, 2016]

[6] [Online]. Available: https://spark.apache.org/docs/1.4.1/mllib-guide.html [Accessed: March 1st, 2016]

[7] [Online]. Available: http://scikit-learn.org/stable/modules/sgd.html [Accessed: March 3rd, 2016]

[8] [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning [Accessed: March 8th , 2016]

[9] [Online]. Available: http://spark.apache.org/docs/latest/ [Accessed: March 8th, 2016]

[10] [Online]. Available: https://docs.wso2.com/display/GSoC/Project+Proposals+for+2016#ProjectProposalsfor2016-Proposal6:[ML]PredictiveanalyticswithonlinedataforWSO2MachineLearner [Accessed: March 8th, 2016]

[11] [Online]. Available http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm [Accessed: February 29th, 2016]

[12] [Online]. Available: https://docs.wso2.com/display/ML110/Features [Accessed: March 8th, 2016]