Adaboost Visualizer

Technical Document

Arpita Chowdhury (0724)
9/29/2018

Submitted to:

Dr. Naushin Nower

Assistant Professor

Institute of Information Technology

University of Dhaka

Dr. Ahmedul Kabir

Assistant Professor

Institute of Information Technology

University of Dhaka

Submitted by:

Arpita Chowdhury (BSSE 0724)

4th Year, 8th Semester, 7th Batch

Session: 2014-2015

Institute of Information Technology

University of Dhaka

Submission Date: 29th September, 2018

Letter of Transmittal

29th September 2018

Coordinator

Software Project Lab 3

Institute of Information Technology

University of Dhaka

Subject : Submission of technical document on "Adaboost Visualizer"

Madam/Sir, I, the student on whom the project on Adaboost Visualizer was assigned, am submitting my report with due respect. I have tried my best for the report. However, it may lack perfection. So, may I therefore, hope that you would be kind enough to accept my report and oblige thereby.

Yours sincerely

Arpita Chowdhury

BSSE 0724

4th Year, 8th Semester, 7th Batch

Session: 2014-2015

Institute of Information Technology

University of Dhaka

Document of Authentication

Role	Name	Signature
Analyzer, Developer and Tester	Arpita Chowdhury BSSE 0724 Institute of Information Technology University of Dhaka	
Supervisor	Dr. Ahmedul Kabir Assistant Professor Institute of Information Technology, University of Dhaka	

Acknowledgement

I am highly indebted for getting such a tremendous opportunity to prepare the technical document on Adaboost Visualizer. I would like to thank whole-heartedly our teacher, Dr. Kazi Muheymin-Us-Sakib, Professor, Institute of Information Technology, University of Dhaka, for giving me guideline about how I can prepare this report. In completing this paper I have collected various important data and information from IIT.

Special thanks to:

Dr. Ahmedul Kabir

Assistant Professor

Institute of Information Technology,

University of Dhaka

Abstract

The aim of this project is to make a research or learning tool for researchers who want to visualize ensemble learning method- Adaboost (Adaptive Boosting). It will implement AdaBoost in multidimensional dataset, showing instance weight changes. For ease of visualization, it will use a dimensionality reduction technique. It is important for users to view step by step weight change for the classifier and the dataset. This is what motivated me for taking the initiative to make a research tool.

My final product will be a fully functional website where user will give training data with attributes and corresponding output class to the application. Then it will perform a dimensionality reduction technique (principal component analysis) on the n-dimension data. After that, it will perform Adaboost on the training data showing the user step by step weight changes of the dataset. It will also show corresponding graphical representation of decision tree and graph plotting data set. User can also fill up questionnaire and system will provide him/her with learning curve or rate from those who used this website. It will also provide user with guideline for proper understanding of website. I have plan to extend this website in future implementing visualization of more machine learning algorithm for better understanding of these algorithm.

Contents

Table of Figures

Chapter 1: Elicitation

1.1 Quality Function Deployment

Normal Requirements:
Normal requirements consist of objectives and goals that are stated during the meeting with the clients. Normal requirements of the project are:

- Working with user provided multidimensional datasets
- Using one-hot encoding for categorical attributes
- Using principal component analysis (PCA) in dimensionality reduction module
- Visualizing instances/samples in two dimension graph with different colors for different classes
- Performing Adaboost on the training data showing the user step by step weight changes of the dataset
- Showing corresponding graphical representation of decision tree
- Highlighting the sub dataset that is currently being used to train the classifier
- Having feature to proceed and undo the process to find the optimal weighted classifiers
- Comparing results from test data and showing prediction accuracy

Expected Requirements:
These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state those. Expected requirements of the project are:

- Providing a user friendly interface for the users
- Accurate analysis algorithm and techniques
- Efficient and fast data processing

Exciting Requirements:
These requirements are for features that go beyond the customer's expectation and prove to be very satisfying when present. Exciting requirements of the project are:

- Having different preprocessing techniques
- Showing attribute information - unique, distinct, maximum value, minimum value, class percentage
- Providing learning curve

1.2 Usage Scenario

Data Processing Module

- **Preprocessing Module**

Before processing with algorithm and visualization, data preprocessing is required. The program will work both with categorical and numeric attributes.

For scaling of dataset, minmax scaling and standard scaling option will be given.

Normalization makes training less sensitive to the scale of features, so we can better solve for coefficients. It will have feature for normalization on dataset.

Often features are not given as continuous values but categorical. To convert categorical features to such integer codes, it will use one hot encoding technique. For various reasons, many real world datasets contain missing values, often encoded as blanks, Nan or other placeholders. Missing values will be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.

It will calculate type, minimum and maximum value, unique value count and percentage, distinct value count and percentage of each attribute.

- **Dimensionality reduction module**

We will need dimension reduction for showing dataset to user in two dimension graph in case it is multi dimensional dataset. For dimensionality reduction, it will have options of different dimension reduction techniques:

❏ Multidimensional Scaling (MDS)

It is a form of non-linear dimensionality reduction. An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible. Each object is then assigned coordinates in each of the N dimensions[6].

❏ Principal Component Analysis (PCA)

The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set [2].

- ❑ T-distributed Stochastic Neighbor Embedding (t-SNE)

It models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability[5].

- **Boosting Module**

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. Adaboost is a type of ensemble learning which focuses on classification problems and aims to convert a set of weak classifiers into a strong one. It works by choosing a base algorithm (e.g. decision trees) and iteratively improving it by accounting for the incorrectly classified examples in the training set. At each step of iteration, it applies the base algorithm to the randomly selected subset of training set and increases the weights of the incorrectly classified examples. It iterates n times, each time applying base learner on the training set with updated weights. The final model is the weighted sum of the n learners[3][4].

It will apply boosting algorithm on dataset and save information of each iteration. This information contains-

- ❑ Selected instances
- ❑ Updated weight of each selected instance
- ❑ Updated weight of each classifier
- ❑ Probability of each instance selection
- ❑ Iteration Number
- ❑ Training Data accuracy
- ❑ Test Data accuracy
- ❑ Threshold
- ❑ All classifier Training Data Accuracy till this iteration
- ❑ All classifier Test Data Accuracy till this iteration

Presentation Module

- **Data Preprocessing Module**

User will upload dataset which will in ARFF format and CSV format. In case of CSV format, user has to specify if the dataset contains attribute name. If not, user will have to specify attribute name, output column. After proper upload of dataset, there will be options given to user for data preprocessing.

User will be shown attribute type, minimum and maximum value, unique value count and percentage, distinct value count and percentage, missing value percentage of each attribute. User can choose from different type scaling options to scale data. She/he can normalize data, impute absent data, and choose which attribute she/he wants to proceed with for boosting algorithm.

For visualization of instances and dataset, user can choose from different types of dimensionality reduction techniques mentioned above.

- **Boosting Module**

When user proceeds for Adaboost algorithm, it will show corresponding decision tree, highlight corresponding selected random instances of the dataset in two dimensional graphs. It will show user before and after selection probability, weight of those selected instances. It will also have an excel sheet for showing row number of instances, for each iteration their updated weight. User can click next for viewing next iteration and previous for viewing previous iteration. She/he can stop at any iteration.

- **Training-Testing Module**

At each iteration, user will be shown training accuracy of current classifier, testing accuracy of current classifier, Training accuracy of summation of all classifiers till this iteration, testing accuracy of summation of all classifiers till this iteration.

- **Learning Rate Module**

User will be given an option for user manual of how the website works. There will also be a questionnaire assessing user experience, understanding of algorithm. This will be used for generating a learning curve which will show user how much better people who used this website have understood algorithm with respect to time.

Chapter 2: Scenario Based Modeling

2.1 Definition of Use Case

A use case captures a contact that describes the system behavior under various conditions as the system responds to a request from one of its

Stakeholders:

In essence, a use case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A use case diagram simply describes a story using corresponding actors, who perform important role in the story and makes the story understandable for the users. The first step in writing a use case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators.

Primary Actor:

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor:

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

2.2 Use Case Diagrams

Level 1 : Adaboost Visualizer

Primary Actor : User

Goal in context: The diagram refers to the details of the project **Adaboost Visualizer.**

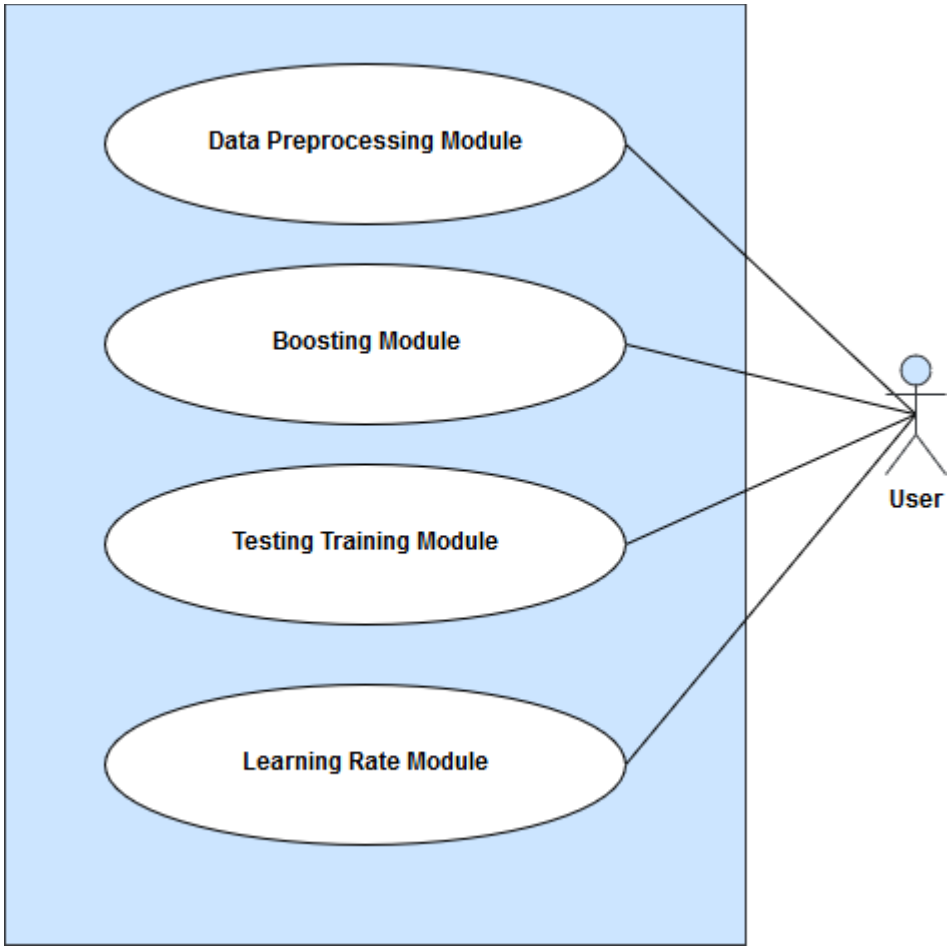

Figure 1 Level 1 Use Case

Actions and Replies

A1: User requests services of data preprocessing module.
R1: System provides data preprocessing module services to the user.
A2: User requests services of boosting module.
R2: System provides boosting module services to the user.
A3: User requests services of testing training module.
R3: System provides testing training module services to the user.
A4: User requests services of learning rate module.
R4: System provides learning rate module services to the user.

Level 1.1: Data Processing Module

Primary actor: User.

Goal in context: The diagram refers to the details of the **Data Processing Module** of level 1.

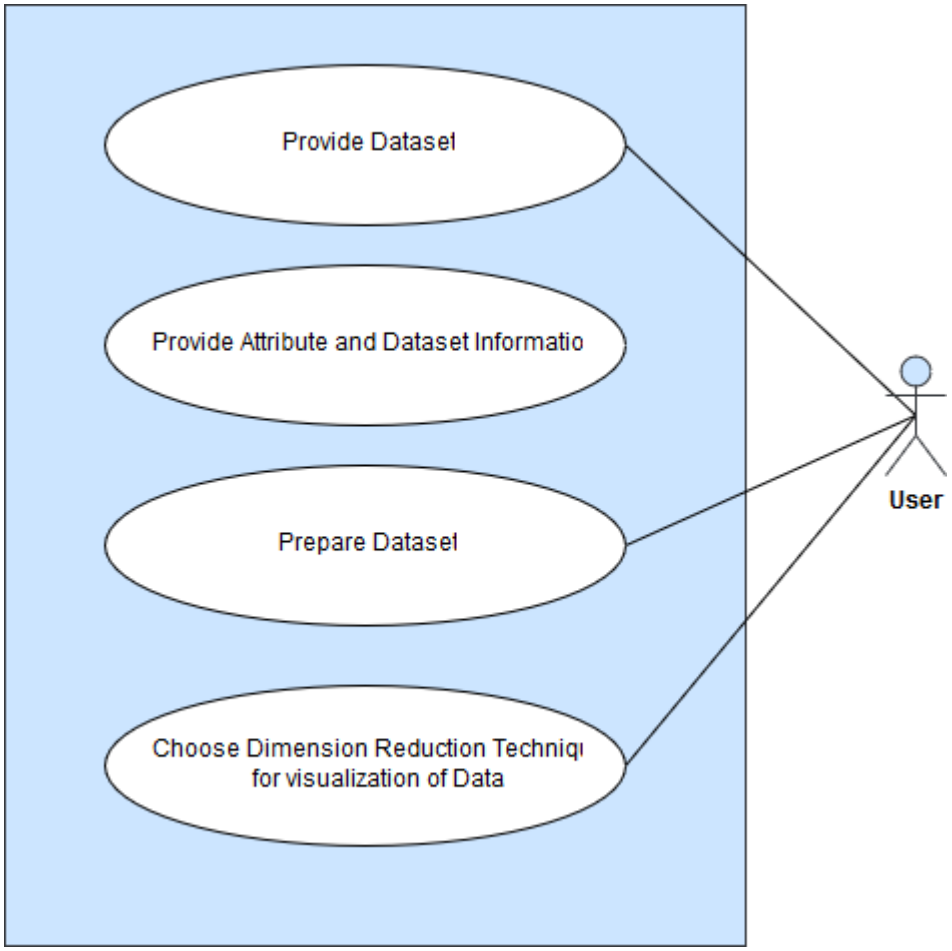

Figure 2 Level 1.1 Use Case

Actions and Replies

A1: User provides dataset to analyze.

R1: System conducts analysis on the dataset and shows the information of the dataset.

A2: User chooses options to prepare dataset.

R2: System provides services to prepare dataset.

A3: User chooses dimension reduction technique for visualization of Dataset.

R3: System visualizes dataset using user selected technique.

Level 1.2: Boosting Module

Primary actor: User.

Goal in context: The diagram refers to the details of the **Boosting Module** of level 1.

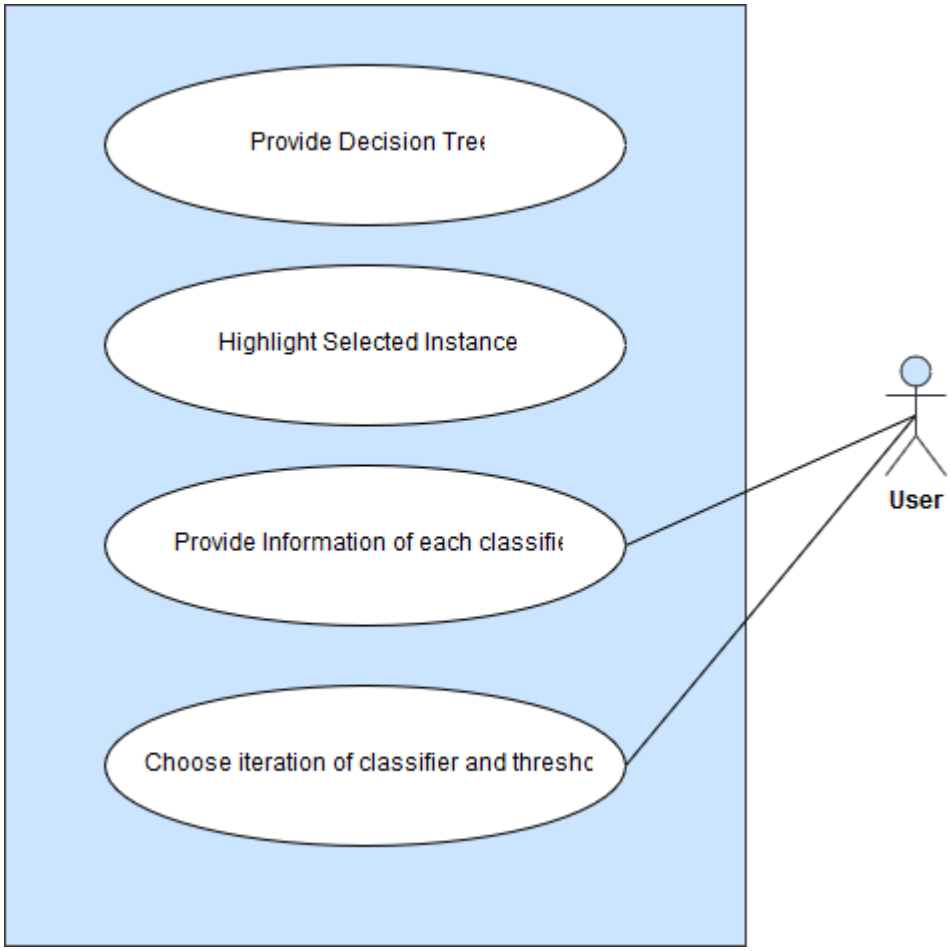

Figure 3 Level 1.2 Use Case

Actions and Replies

A1: User wants information of each classifier.

R1: System provides information of each classifier.

A2: User chooses iteration of classifier and threshold.

R2: System provides results accordingly.

Level 1.3: Testing Training Module

Primary actor: User.

Goal in context: The diagram refers to the details of the **Testing Training Module** of level 1.

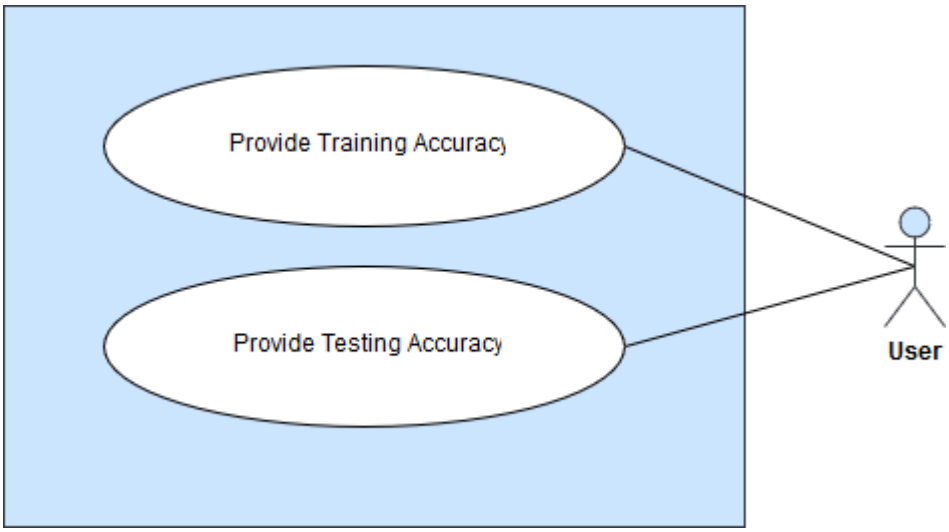

Figure 4 Level 1.3 Use Case

Actions and Replies

A1: User wants to know training accuracy.

R1: System provides information of training accuracy.

A2: User wants to know testing accuracy.

R2: System provides information of testing accuracy.

Level 1.4: Learning Rate Module

Primary actor: User.

Goal in context: The diagram refers to the details of the **Learning Rate Module** of level 1.

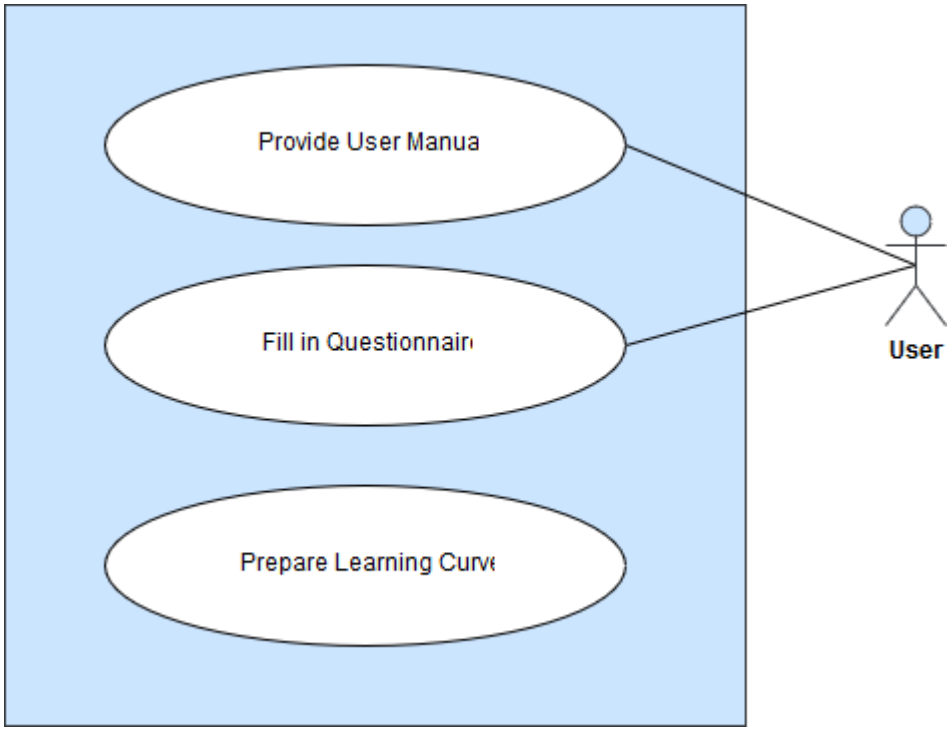

Figure 5 Level 1.4 Use Case

Actions and Replies

A1: User wants to know user manual.

R1: System provides information of user manual.

A2: User wants to fill in questionnaire.

R2: System provides questionnaire.

Chapter 3: Data Based Modeling

3.1 Entity Relationship Diagram

Figure 6 shows Entity Relationship diagram of this project. For simplicity only primary key is shown in ER diagram.

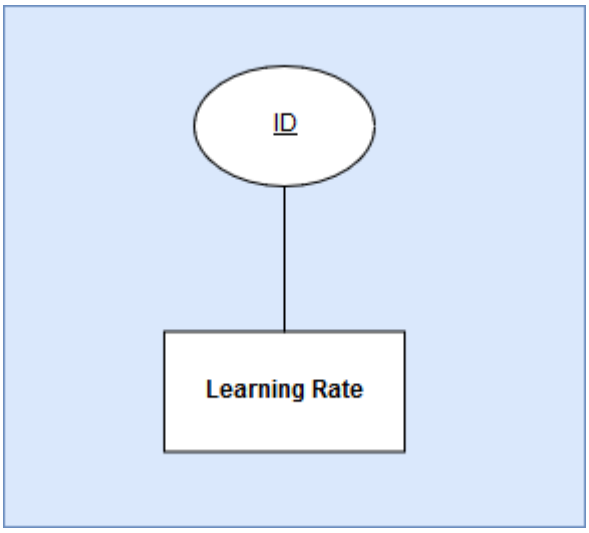

Figure 6 ER Diagram

3.2 Schema Table

Below is the table from ER diagram for table Learning Rate.

Attribute	Type	Size
ID	Varchar2	30
Time	Varchar2	30
Date	Date	
Correct answer percentage	float	

Chapter 4: Class Based Modeling

4.1 Analysis Classes

Data Preprocessing Module			
Serial No	Class	Attributes	Methods
1	Upload-Management	Data format, containsAttributeName, outputColumnNumber, AttributesColumnName, temporaryDataSavePath	CheckDataFormat()+ saveTempData()+ checkAttrbuteNames()+ SetOutputColumn()+ ChangeDatafile()+ setAttributeColumnName()+ getDataSet()
2	Preprocess-Management	AttributeName, AttributeType, AttributeMinimumValue, AttributeMaximumValue, AttributeUniqueValueCount, AttributeDsitinctValueCount, AttributeMissingValueCount, ScalingOptions,	normalizeData()+ calculateAttributeInformation() + scaleData()+ ImputeData()+ oneHotEncoding()+ chooseAttributeToWork()+ saveTempData()+getData()
3	Dimension-Reducer	ReductionOptions, ChosenReductionAlgorithm, reducedGraph	chooseReductionAlgorithm()+ MDS()+ PCA()+tSne()+ getGraph()+makeWeightedReduced Graph()

Boosting Module			
Serial No	Class	Attributes	Methods
1	Booster	IterationNumber, SelectedInstances, weightOfCurrentClassifier, probablityofEachInstance, decisionTreeGraph, ReductionGraph, previousWeightOfSamples, currentWeightofSamples, threshold	ShowDecisionTree()+ HighLightSelectedSamples() + getSelectedSamples()+ Adaboost()+ viewSelectedSamplesChange()+ Update()
2	BoostingManagement	Threshold, TotalIteration,	checkIfThresholdMatched()+ getRequiredIterationInfo()+ getTrainingInfo()+ viewDataChangesforAll()+stop() + getDataset()+ getReducedGraph()

Training Testing Module			
Serial No	Class	Attributes	Methods
1	TrainingManagement	iterationNumber, CurrentTrainingAccuracy, TotalTrainingAccuaracy	getDataset()+ getTrainingDataofCurrentIteration() + calculateCurrentTrainingAccuracy() + calculateTotalTrainingAccuaracy()

2	TestingManagement	iterationNumber, CurrentTestingAccuracy, TotalTestingAccuaracy	getDataset()+ getTestingDataofCurrentIteration()+ calculateCurrentTestingAccuracy()+ calculateTotalTestingAccuaracy()

Learning Rate Module			
Serial No	Class	Attributes	Methods
1	LearningRater	UserManual, questionnaireAnswers, LearningRateGraph,	ShowUserManual()+ getQuestionAnswer()+ calculatePercentage()+ saveinDatabase()+ showGraph()

4.2 Class Cards

UploadManagement	
Responsibility	**Collaborators**
Getting Dataset	
Updating Dataset	
Saving Dataset	

PreprocessManagement	
Responsibility	**Collaborators**
Saving Dataset	UploadManagement
Imputing Dataset	UploadManagement
Normalizing Dataset	UploadManagement
Calculating Attribute Information	UploadManagement

DimensionReducer	
Responsibility	**Collaborators**
Reducing Dataset	PreprocessManagement

Booster	
Responsibility	**Collaborators**
Boosting Dataset	BoostingManagement
Making Decision Tree	BoostingManagement

BoostingManagement	
Responsibility	**Collaborators**
Presenting	Booster
Managing Boosting	PreprocessManagement DimensionReducer Booster

TrainingManagement	
Responsibility	**Collaborators**
Presententing	
Calculating Training Accuracy	PreprocessManagement BoostingManagement

TestingManagement	
Responsibility	**Collaborators**
Presenting	
Calculating Testing Accuracy	PreprocessManagement BoostingManagement

LearningRater	
Responsibility	**Collaborators**
Showing User Manual	
Calculating Percentage	UploadManagement
Making Learning Curve	UploadManagement

4.3 CRC Diagram

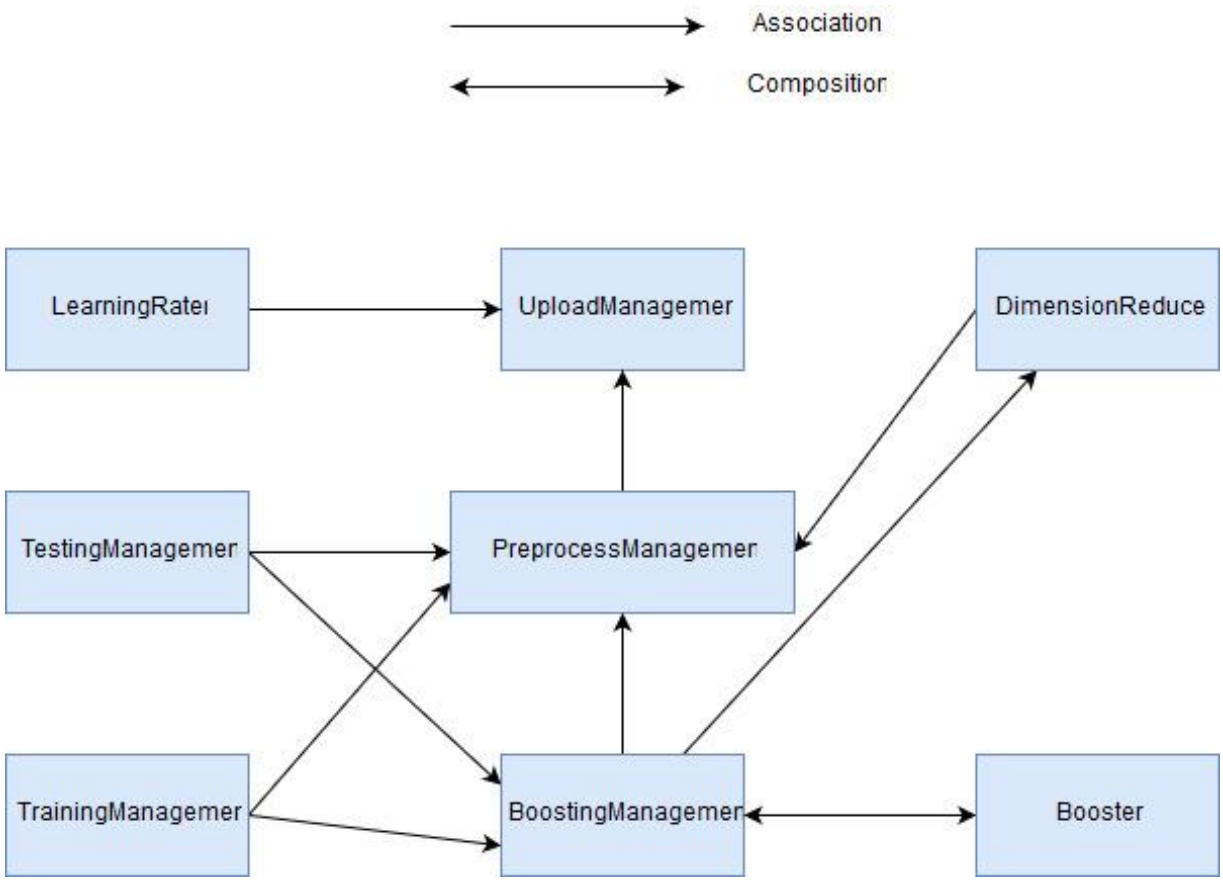

Figure 7 CRC Diagram

Chapter 5: Architectural Design

Software architectural design represents the structure of the data and program components that are required to build a computer – based system. There are 4 steps in architectural design:

- Represent the system in context
- Define archetypes
- Refine the architecture into components
- Describe instantiations of the system

5.1 Representing the system in context

Following is the architectural context diagram of this project.

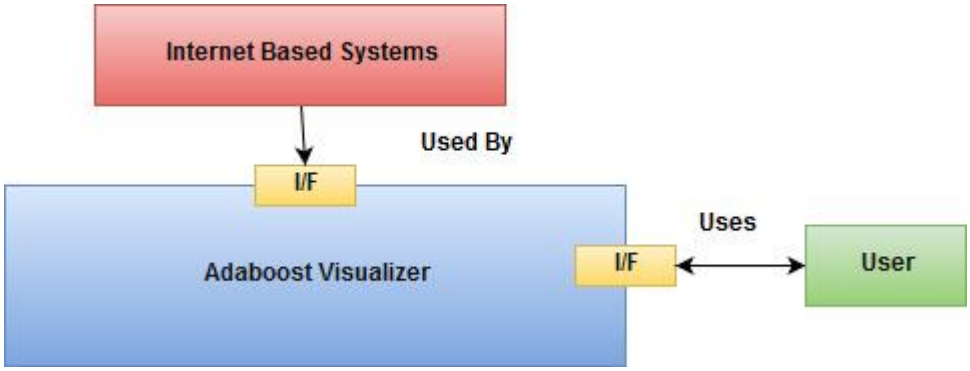

Figure 8 Architectural Context Diagram

This application is based on the classic 3-tier architecture. The software is divided into a presentation layer, logic layer and a persistence layer.

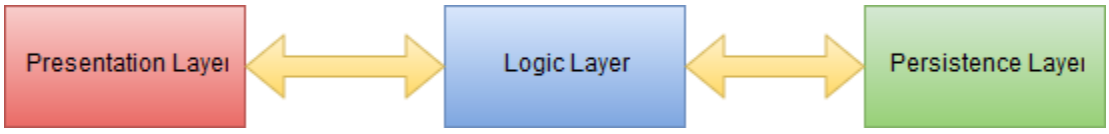

Figure 9 3-tier architecture

The presentation layer is where all the user interactions take place. The presentation layer communicates with the logic layer. This logic layer is a REST API that provides URL endpoints for the presentation layer to communicate. Through the logic layer user inputs are processed and information are returned to the client. The logic layer communicates with the persistence layer to store and retrieve data from the system.

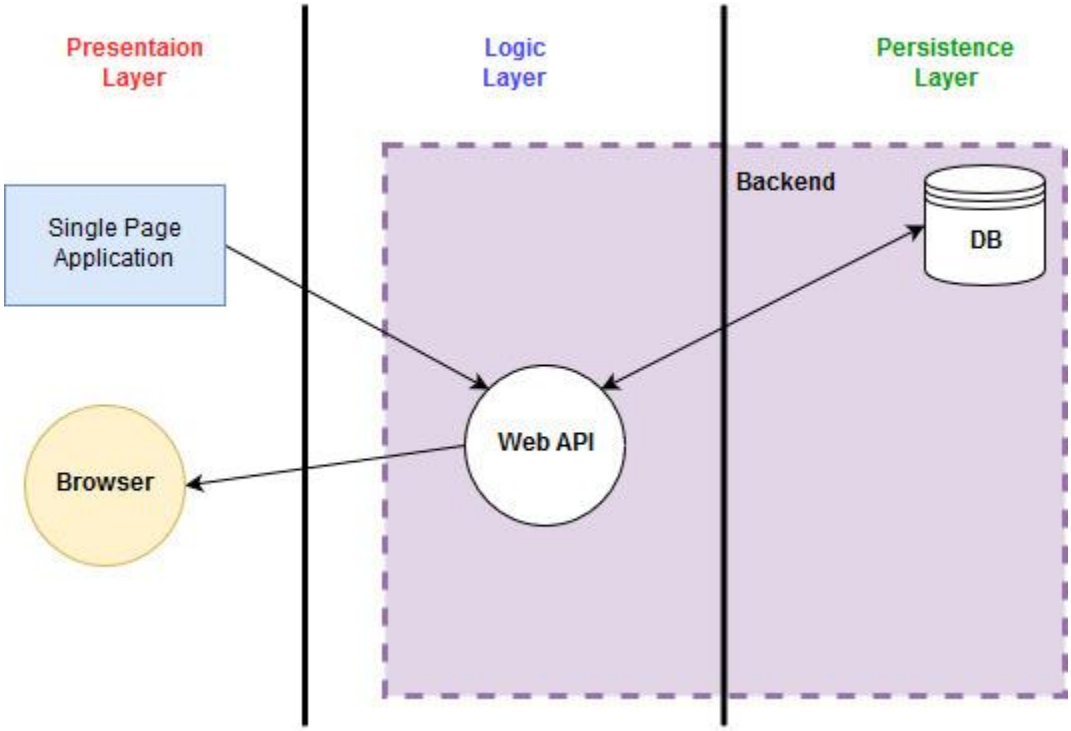

Figure 10 Architectural Design

When the client requests the server for the application a single page application will be loaded. Later on all requests to the server will be AJAX calls. The server will reply accordingly. The server stores and retrieves data from the database server.

5.2 Defining Archetypes

An archetype is a class or pattern that represents a core abstraction that is critical to the design of architecture for the target system. They represent stable elements of the architecture. Archetypes can be derived by examining the analysis classes defined as part of the requirements model. The archetypes of Adaboost Visualizer are:

- UploadManagement
- PreprocessManagement
- DimensionReducer
- Booster
- BoostingManagement
- TrainingManagement
- TestingManagement
- LearningRater

5.3 Refine the architecture into components

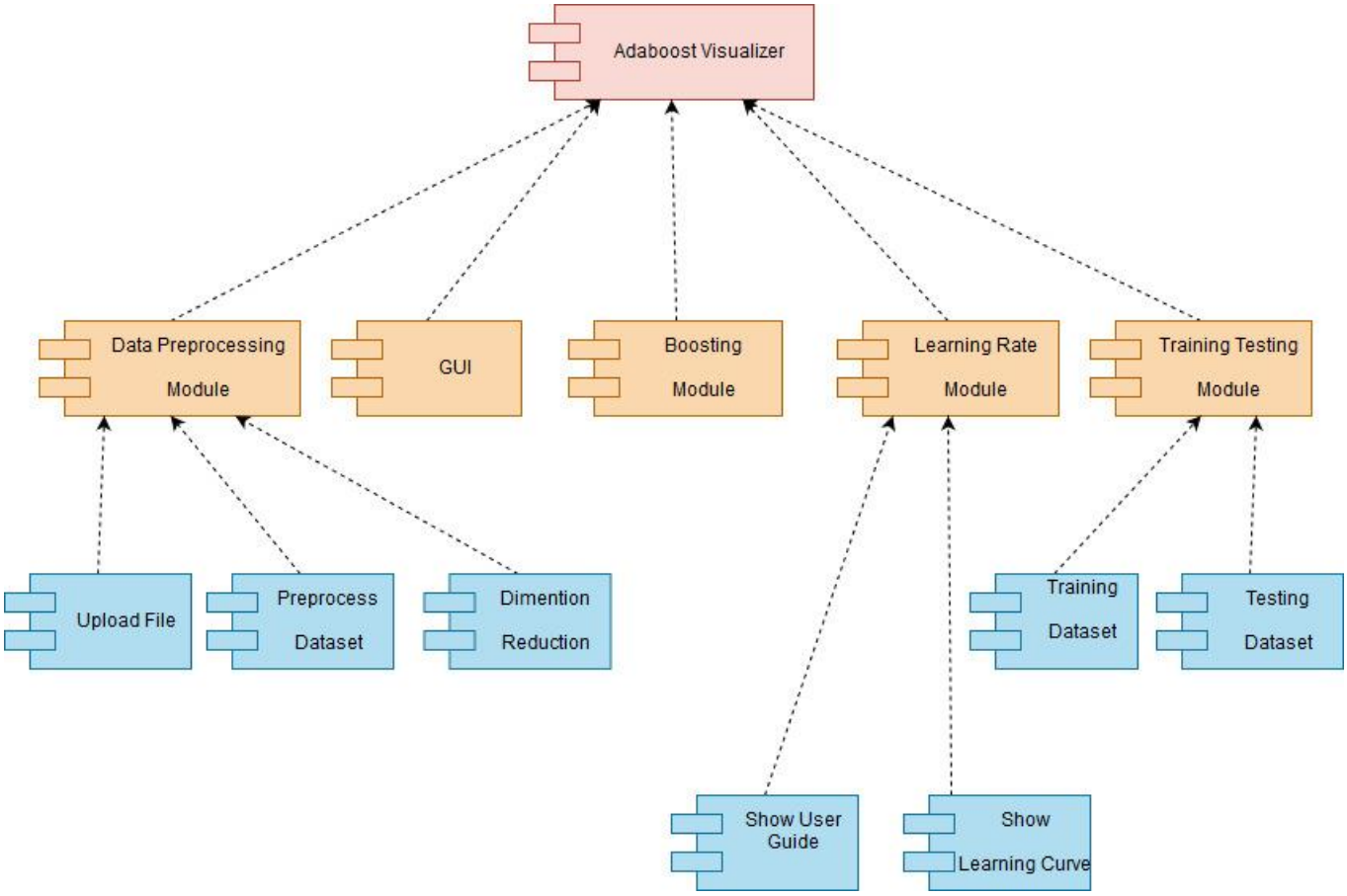

Figure 11 Refine the architecture into components

5.4 Describe instantiations of the system

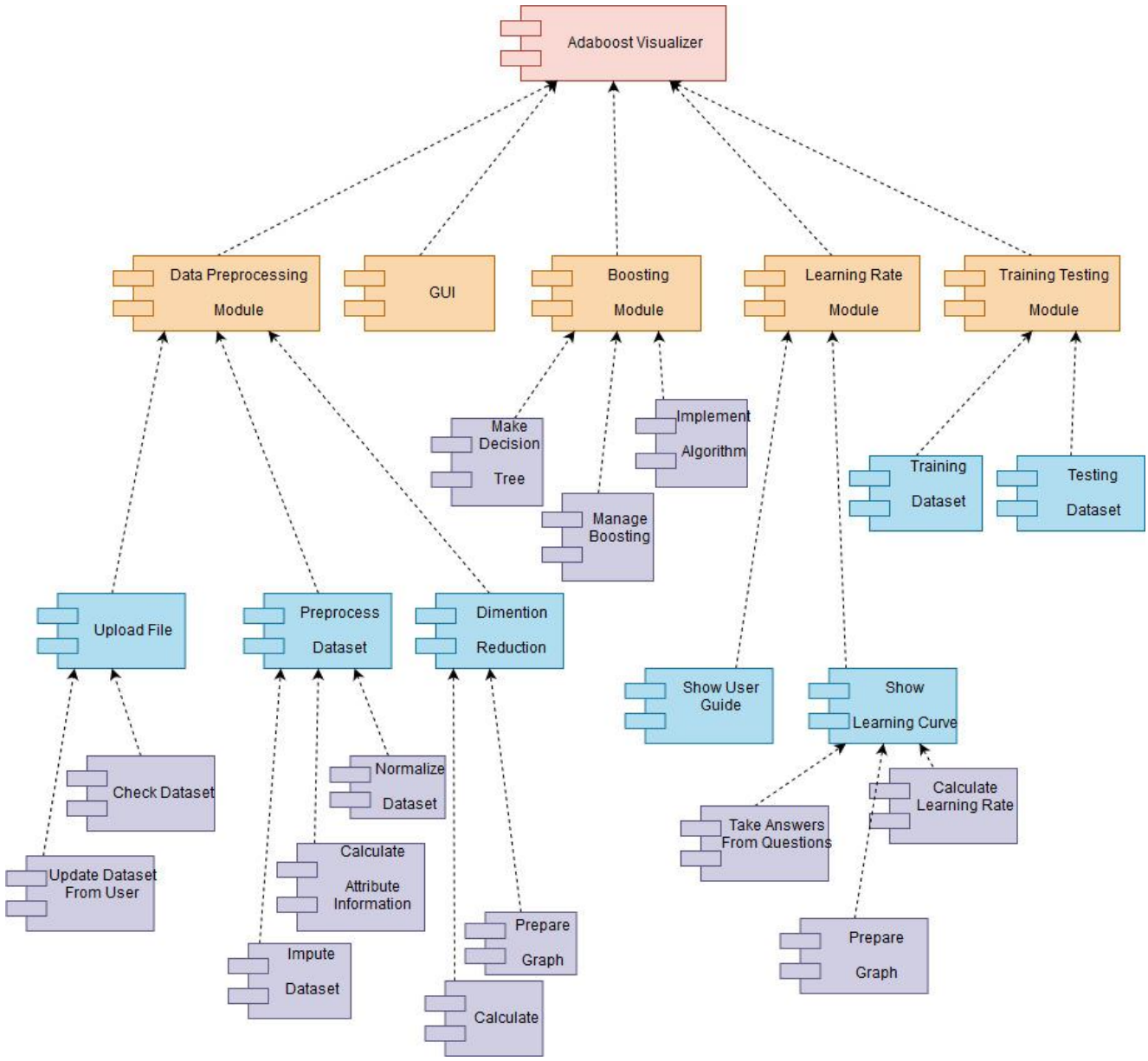

Figure 12 Describe instantiations of the system

Chapter 6: Implementation

The whole implementation has three parts - the database for storing data, the web API for data communication and the UI for presentation.

For database, I used SQLite. It was chosen for its simplicity and ease of use. It is easy to use, manage as well as scalable.

For serving the client we made a web API that receives client requests and serves accordingly. The web API is written in Python. I also uses the Flask framework for making the web API. Flask is a web micro framework for Python.

My front end is built using Angular 4. Angular is a framework for making single page applications. It also allows the testing of front end components.

Chapter 7: Conclusion

I am pleased to submit the final technical document of Adaboost Visualizer. From this report the readers will get a clear and easy view of the system. This technical document can be used effectively to maintain software development cycle. Hopefully, this document can also help our junior BSSE batch students. I tried my best to remove all dependencies and make an effective and fully designed document.

References

1) Pressman, Roger S. Software Engineering : A Practitioner's Approach (7th Edition)

2) Wold, Svante & Esbensen, Kim & Geladi, Paul. (1987). Principal Component Analysis. Chemometrics and Intelligent Laboratory Systems. 2. 37-52.

3) Dietterichl, Thomas G. (2002). Ensemble learning. In M. Arbib (ed.), The Handbook of Brain Theory and Neural Networks. MIT Press. pp. 405--408.

4) Freund, Yoav & E Schapire, Robert. (1999). A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence. 14. 771-780.

5) L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

6) van der Maaten, Laurens & Postma, Eric & Herik, H. (2007). Dimensionality Reduction: A Comparative Review. Journal of Machine Learning Research - JMLR. 10.