DESIGN AND IMPLEMENTATION OF REAL-TIME STUDENT PERFORMACE EVALUATION AND FEEDBACK SYSTEM

A Thesis Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Karthik Bibireddy

May 2017

DESIGN AND IMPLEMENTATION OF REAL-TIME STUDENT PERFORMACE EVALUATION AND FEEDBACK SYSTEM

_____

Karthik Bibireddy

APPROVED:

_____

Dr. Christoph F. Eick
Department of Computer Science

_____

Dr. Sara G. McNeil
Department of Curriculum and Instruction

_____

Dr. Weidong Shi
Department of Computer Science

_____

Dean, College of Natural Sciences and Mathematics

Acknowledgements

First and foremost, I would like to thank and convey my sincere gratitude to my graduate advisors, Dr. Christoph F. Eick, and Dr. Nouhad Rizk for giving me an opportunity to work in the data analysis and intelligent systems team. This opportunity eventually led to the work done in this thesis. Their advice, guidance, and support were instrumental throughout the course of this study. It is difficult to envisage this work without their help and insight.

I would also like to thank Dr. Weidong Shi and Dr. Sara G. McNeil for agreeing to be a part of my Thesis committee. My appreciation also goes to the Educational Data Mining team, led by Dr. Nouhad Rizk, for their continuous efforts in obtaining the real datasets, conducting a survey and collecting information.

I would like to especially acknowledge the significant role played by my friends at the University of Houston, for making sure that I ensconce in my personal as well as academic life at the University of Houston. Also, I would like to take this opportunity to thank my family for encouraging and motivating me in all my endeavors. Finally, I would like to express my deepest gratitude towards my parents and my sister for their unparalleled love and support. They continue to be the greatest source of inspiration for me.

DESIGN AND IMPLEMENTATION OF REAL-TIME STUDENT PERFORMACE EVALUATION AND FEEDBACK SYSTEM

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Karthik Bibireddy

May 2017

Abstract

Undergraduate education is challenged by high dropout rates and by delayed student graduation due to dropping courses or having to repeat courses due to low academic performance. In this context, an early prediction of student-performance may help students to understand where they stand amongst their peers and to change the attitude with about the course they are taking. Moreover, it is important to identify students in time who need special attention and providing appropriate interventions, such as mentoring and conducting review sessions. The goal of this thesis is the design and implementation of real-time student-performance evaluation and feedback system (RSPEF) to improve graduation rates. RSPEF is an interactive, web-based system consisting of a Predictive Analysis System (PAS) that uses machine learning techniques to interpolate past student-performance into future, and the development of an Emergency Warning System (EWS) that identifies poor-performing students in courses. Moreover, a unified representation of student-background and student-performance data is provided in form of a relational database schema that is suitable to be used to assess student's performance across multiple courses, which is critical for the generalizability of RSPEF system. The system design includes core machine learning & data analysis engine, a relational database that is reusable across courses and an interactive web-based interface to continuously collect data and create dashboards for users.

Table of Contents

List of Figures

List of Tables

1. Introduction

Despite a strong and intensive effort over decades by colleges and universities to improve their graduation rates by trying to help students graduate in time, the graduation rates in the US remain flat. According to a report by the National Center for Education Statistics [1], about 28% of bachelor's degree students entered a STEM field (i.e., chose a STEM major) at some point within six years of entering postsecondary education in 2003-2004; however, 48% left the field either by changing majors or leaving college without completing a degree. Other studies indicated that many of the students who left the STEM fields were actually high-performing students who might have made valuable additions to the STEM workforce had they stayed [2,3]. To produce more graduates in STEM fields, some recent U.S policies have focused on reducing student's attrition from STEM fields in college, arguing that increasing STEM graduation even by a small percentage can be a cost efficient way to contribute substantially to the supply of STEM workers. [1,4,5] The University of Houston has reported its graduation rates for a full-time first-time in college undergraduates between 2008 and 2011. In 2008, 3486 freshmen enrolled at UH, a cumulative graduation rate in four years was only 18%, in five years was 38% and in six years was 48.1%. In 2009, 3100 freshmen enrolled at UH, a cumulative graduation rate in four years was 19.7%, in five years was 41.6% and in six years was 51%. In 2010, 3453 freshmen enrolled at UH, a cumulative graduation rate in four years was 22.7% and in five years was 42.5%. In 2011, 3556 freshmen enrolled at UH, a cumulative graduation rate in

four years was only 25.2%. In this research, we tackle these problems by educational data-mining techniques.

Educational data-mining is a relatively new discipline that employs data-driven methods to analyze educational datasets such as student, professor or instructor, course, and school data, to better understand students and the environment they are learning in. Educational data is usually collected using computer-assisted or web-based learning systems or the administration of the school or the university will provide the data. The data is often observed to be complex in nature and highly interrelated to one another. It is critical to identify poorly performing students in an early phase of the semester so that instructors can take appropriate interventions with such students like mentoring or reviewing topics with them. Faculty members will need functional tools that will allow them to identify such students. Additionally, students need a system that helps them to understand where they stand among their peers so that they can assess themselves and plan accordingly without waiting for the instructor to intervene.

The goal of this thesis is to alleviate the problems mentioned in the last paragraphs; its focus is the design and implementation of real-time student performance evaluation and feedback system (RSPEF). The RSPEF system has three components: a data pre-processing layer, a machine learning and data analysis engine and an application layer. The data pre-processing layer consists of the relational database and the dataset selector. The data required to build models is collected through surveys which are answered by students

and instructors who upload student course scores through an interface. The survey and performance data are stored in a relational database.

The role of data selectors is to extract relevant training datasets from the relational database, to clean and standardize them and passes these datasets to the machine learning and data analysis engine which learns models from training data and makes predictions using learned models. The machine learning and data analysis system also called MLDA engine has three sub-systems: a Predictive Analysis System (PAS), a Report Generation System (RGS) and an Emergency Warning System (EWS). The student-performance dataset is passed as input to PAS, which builds models by learning from the training datasets and uses these models to predict student grades, starting very early in the semester. The EWS learns models from student survey datasets and classifies students who currently take course into either "passing" or "failing" the course. The RGS uses both student-performance and survey datasets to compute statistical summaries, such as mean, median and standard deviation of course elements such as attendance, quizzes, assignments, and exams. The summaries, predicted grades and the list of identified poor-performing students are passed on to the application layer. The application layer is web-based and will visualize visualizes the information and predictions from these models, which are displayed to the students and instructors. Students can register and log into their account to view their predicted grades for various courses they are enrolled in, assess themselves based on other information available on their dashboard. Instructors can view the list of students identified as poor-performing for various courses in a very early stage so that the instructors can give special attention and help them to graduate.

The remainder of this Master Thesis is organized as follows. In chapter 2, we will discuss the background and the related work done in the field of educational data-mining and machine learning using educational datasets, In chapter 3, we will discuss the architecture of our proposed system and the functionality of its components. In chapter 4, we will briefly explain the design of the relational database and introduce the relational schema of the designed database. In chapter 5, we will discuss various models built by machine learning algorithms and evaluate and compare their performance. In chapter 6, we will mention the work done during this research and our accomplishments.

2. Background & Related Work

This chapter we will discuss the background and related work concerning educational data-mining, prediction and classification models, and the entity-relationship model used in this research.

2.1 Background of Machine Learning Techniques

Machine Learning is a form of artificial intelligence that provides intelligence to computers so they can learn without being programmed. Machine learning is essentially classified into three types: supervised learning, unsupervised learning, and reinforcement learning. The main goal of supervised learning is to learn a model from labeled or training data which will allow us to make predictions about future data. Supervised learning can be further classified into two tasks: a classification task, where the predicted outcome is a categorical value, and a regression task, where the outcome is a continuous value. In reinforcement learning, the goal is to develop a system or agent that improves its performance based on interactions with the environment there are rewards and penalties in this approach, but it is closely related to supervised learning. In unsupervised learning, we are dealing with unlabeled data we extract meaningful information without any prior knowledge about the data. Clustering is a technique of unsupervised learning [23]. In this section, we will discuss various classification and regression algorithms which are relevant to our thesis.

2.1.1 Naïve Bayes

Naïve Bayes classifier belongs to the family of probabilistic classifiers, built upon Bayes theorem with strong independence assumptions between the features or attributes [22]. In the equation 2.1.1.1, the X denotes evidence and C is a hypothesis. P (C|X), given examples or evidence X and assuming C, we will calculate the probability of occurrence of C given evidence X using the equation 2.1.1.1. This is called Posterior probability.

$$P(C \mid X) = \frac{P(X \mid C) \times P(C)}{P(X)} \qquad (2.1.1.1)$$

P(C) is prior, the probability of occurrence of our hypothesis C. P (X|C) is called likelihood, the probability of the evidence given our hypothesis, the ratio of examples where our hypothesis C is true. P(X) is the prior probability of evidence X. Evidence X is in the form shown in the equation 2.1.1.2.

$$X = \{X_1, X_2, \ldots. X_n\} \qquad (2.1.1.2)$$

X is a collection of features or attributes, n represents the number of attributes. In our case, X is a collection of course elements or the survey elements/questions. Given an instance of evidence X, the task of Naïve Bayes is to predict the hypothesis C having the highest posterior probability.

$$P(C_i \mid X) > P(C_j \mid X) \qquad (2.1.1.3)$$

The class C_i with maximum $P(C_i|X)$ is called the maximum posterior hypothesis. An unseen example is classified into a class with maximum posterior. Usually, our datasets

have many attributes X_i to X_n, it is computationally expensive to acquire all the necessary probabilities to compute $P(X|C_i)$ (likelihood). To simplify the computing process of $P(X|C_i)$, an assumption of class conditional independence is made in equation 2.1.1.4, which assumes that the probabilities of different attributes have specific values are conditionally independent of one another, and we obtain:

$$P(X|C_i) = \prod_{k=1}^{n} P(X|C_i)$$

$$= P(X_1|C_i) \times P(X_2|C_i) \times \ldots \times P(X_k|C_i) \qquad (2.1.1.4)$$

With the conditional independence assumption, we only need to estimate the conditional probability of each X_i, given C and the prior probabilities of X_i and C instead of knowing the class conditional probabilities for every combination of X. To classify an unseen record of a student, the Naïve Bayes classifier computes the posterior probability for each class C using the formula given in equation 2.1.1.5

$$P(C \mid X) = \frac{P(C)\prod_{i=1}^{d} P(X_i \mid C)}{P(X)} \qquad (2.1.1.5)$$

Since P(X) does not depend on the class membership, it is fixed for each class. Therefore, it is sufficient to choose the class that maximizes the numerator term. From the equations 2.1.1.5 and dropping the denominator, we can deduce equation 2.1.1.6; where $\propto$ represents proportional.

$$P(C \mid X) \propto P(C)\prod_{i=1}^{d} P(X_i \mid C) \qquad (2.1.1.6)$$

2.1.2 Logistic Regression Classification

Multinomial logistic regression classification uses one versus all policy, dividing the problem into K (number of classes) simple logistic regression problems [24]. In each simple logistic regression problem, one of the K classes is represented as 1 and the rest of the classes is represented as 0 of the dichotomy. In this approach, K simple logistic regression models are built one for each class. Given an unseen record as input, each model will return a probability of that the record belongs to the class at 1's boundary in that simple logistic regression model. The student is classified into the class who's probability returned is highest.

If there are K classes for N instances with M attributes, the parameter matrix B to be calculated is an M x (K-1) matrix. The probability for a class j with the exception of the last class (Since ridge estimator is added as an L2 regularization for all parameters except the last parameter we divide the equation into two parts 2.1.2.1 and 2.1.2.2) is given in the equation 2.1.2.1 and the probability of last class is given in equation 2.1.2.2.

$$P_j(X_i) = \frac{e^{\left(X_i B_j\right)}}{\sum_{j=1}^{(K-1)} e^{\left(X_i B_j\right)} + 1} \qquad (2.1.2.1)$$

$$1 - \left(\sum_{j=1}^{(K-1)} P_j(X_i)\right) = \frac{1}{\sum_{j=1}^{(K-1)} e^{\left(X_i B_j\right)} + 1} \qquad (2.1.2.2)$$

2.1.3 Random Forest

Random forests use an ensemble of simple decision trees as classification models [25]. An ensemble approach constructs a set of base classifiers from training data and performs classification by taking a vote from the predictions made by each base classifier.

Random Forests use an ensemble learning approach called boosting to obtain a set of base classifiers by generating different training sets from the training data using weighted sampling with replacement approach. Boosting assigns a weight to each training example and which adaptively changes at the end of each boosting round. AdaBoost [26] is the most popular boosting algorithm. The AdaBoost algorithm initially assigns equal weights to the training examples. A training set is generated by drawing examples from the training set. Next, a classifier is induced from the training set and used to classify all the examples in the original data. The weights of the training examples are updated at the end of each boosting round. Examples that are classified incorrectly will have their weights increased while those that are classified correctly will have their weights decreased. This forces the classifier to focus on examples that are difficult to classify in subsequent iterations which are important to obtain a diverse set of base classifiers.

2.1.4 Multi-Layer Neural Network

A Neural Network is an artificial neural network which is a system build based on a biological neural network such as a brain [27]. Like the neuron, a Neural Network consists of nodes which are connected to each other. Each node will have an inhibition and excitation threshold. Inhibition and excitation thresholds are the minimum and maximum

input it expects to receive. Each node has an activation function, these inputs from one or more nodes are fed as input to this function and the output of this function is the output of the node. The node will transmit only if the activation function's output is above the excitation threshold. A multilayer Neural Network is an artificial neural network with one or more hidden layer. An MLNN with a single hidden layer is illustrated in figure 2.1.

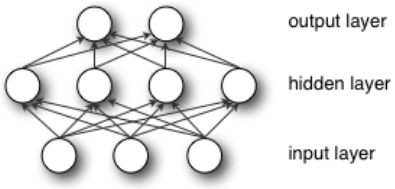

Figure 2.1: Multi-Layer Neural Network with single hidden layer

The activation flows through the network, through hidden layers, until it reaches the output nodes. The output nodes then reflect the input in a meaningful way to the outside world. The difference between predicted value and actual value (error) will be propagated backward by apportioning them to each node's weights according to the amount of this error the node is responsible for (e.g., gradient descent algorithm).

The transfer function translates the input signals to output signals. Four types of transfer functions are commonly used but, we will discuss only unit step and sigmoid. Unit step, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value (see figure 2.2). The sigmoid function consists of 2 functions, logistic and tangential. The values of logistic function range from 0 and 1 and -1 to +1 for tangential (see figure 2.3)

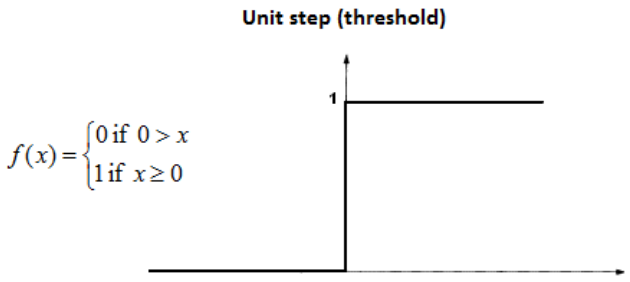

Figure 2.2: Unit-step transfer function for MLNN

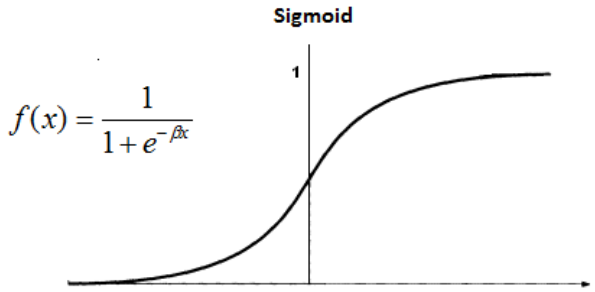

Figure 2.3: Sigmoid transfer function for MLNN

2.1.5 Support Vector Regression

An SMOreg or Support Vector Regression is a popular machine learning tool for regression problems [28, 29]. Suppose a training dataset where we have N observations with observed labels. The goal of SMOreg is to find a function f(x) that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as

possible. We will describe the working of SMOreg using a linear function, taking the form shown in equation 2.1.6.1.

$$f(x) = x'\beta + b \qquad (2.1.6.1)$$

Now our cost function J is given by 2.1.6.2, our goal is to minimize the cost function which is by minimizing the norm value of (β'β). This is now a convex optimization problem to minimize. Our goal here is to minimize 2.1.6.2 subject to the condition stated in 2.1.6.3

$$J(\beta) = \frac{1}{2}\beta'\beta \qquad (2.1.6.2)$$

$$\forall n : |y_n - (x'_n\beta + b)| \leq \varepsilon \qquad (2.1.6.3)$$

No such function f(x) might exist to satisfy this constraint for all points. To deal with this problem we introduce slack variables ξ_n and ξ_n^* for each point. This approach is similar to a soft margin concept in SVM classification, which will let the SMOreg consider points up till the slack variables, yet still, satisfy the required conditions. We introduce slack variables to our objective function in the equation 2.1.6.4 and the conditions this function is constrained by is mentioned in 2.1.6.5.

$$J(\beta) = \frac{1}{2}\beta'\beta + C\sum_{n=1}^{N}(\xi_n + \xi_n^*) \qquad (2.1.6.4)$$

$$\forall n : y_n - (x'_n\beta + b) \leq \varepsilon + \xi_n \qquad (2.1.6.5)$$

$$\forall n : (x'_n\beta + b) - y_n \leq \varepsilon + \xi_n^*$$

$$\forall n : \xi_n \geq 0$$

$$\forall n : \xi_n^* \geq 0$$

2.2 Background of the Entity-Relationship Model

The entity-relationship (ER) model [7] is one of the popular database-design models. It is popular for its simple graphical representations and safe constructs. ER models are currently being used widely in databases, information systems, and software engineering. An entity can be a real-world object, which is animate or inanimate and can exist independently. All entities have attributes which are properties of the entity. There can be different types of attributes, simple attribute, composite attribute, derived attribute and multi-value attribute. A key is an attribute which will uniquely identify an instance in that entity. See figure 2.4. A relationship is an association among entities. For example, an employee works_at a department. Works_at is the relationship here.

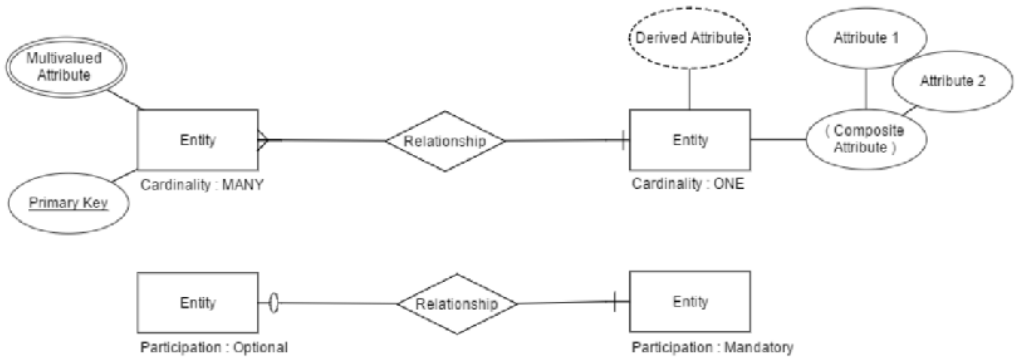

Figure 2.4: Components in an Entity-Relationship Model

A cardinality defines the number of instances in one entity type, which can be associated with the number of instances of another entity type via relationship. Four kinds of relationships can be distinguished:

- One to one: one instance in A can be associated with at most one instance in B (A and B are entities in a relationship R)
- One to many: one instance in A can be associated with more than one instance in B
- Many to one: many instances in A can be associated with at most one instance in B
- Many to many: many instances in A can be associated with many instances in B

There are two types of participation constraints:

- Total/mandatory Participation – Each instance of an entity id involved in the relationship. Total participation is represented by a small line cutting the line between relationship and entity perpendicularly close to the entity rectangle.
- Partial/optional Participation – Not all instances are involved in the relationship. A small oval near the entity rectangle represents the partial participation of that entity.

2.3 Related Work

Educational data-mining (EDM) research has been growing at a fast pace over the last decade. The main aim of EDM is to develop models capable of better understanding how students learn. In this section, we will briefly discuss approaches are closely related to our study. Han and Kamber [6] describe an approach that uses association rule mining to identify the factors influencing the success of students, and decision tree models are used to predict student academic performance. Priya et al. [8] applied a decision tree

classification technique towards the end of the semester, leading to the discovery of valuable knowledge that can be used to improve students' performances. Many studies (Ardila, 2001; Boxus, 1993; Busato et al., 1999, 2000; Chidolue, 1996; Furnham et al., 1999; Gallagher, 1996; Garton et al., 2002; King, 2000; Minnaert and Janssen, 1999; Parmentier, 1994.) were undertaken to try to explain the academic student-performance or to predict their success or their failure. Galit [9] presented a case study that used students data to analyze their learning behavior to predict the results and to warn students at risk before their final exams.

Goyal and Vohra [10] showed that if data-mining techniques such as clustering, decision tree induction, and association analysis were applied to higher education processes, it helped to predict student retention rate. Surjeet Kumar et al. [11] used decision-tree classifiers to predict student drop-out rates. Pathan et al. [12] developed a decision-tree model to improve students' programming skills in the C language where they collected data from 70 students of Structured Programming Language (SLP) course and generated two datasets.

Minaei-Bidgoli [13] used a combination of multiple classifiers to predict student grades based on features extracted from logged data in an education web-based system. Furthermore, they tried to learn an appropriate weighting of the features using a genetic algorithm approach, which further improved prediction accuracy. Pittman [14] performed a study to explore the effectiveness of data-mining methods in identifying success or failure of students in a course and found that SVM-based approaches and random forests methods accomplished the highest accuracies. Romero et al. [15] compared different, data-mining

methods for classifying students based on their Moodle (e-learning system) usage data and the final marks obtained in their respective programs and they observed that decision trees were the most suitable approach for this task. Nguyen et al. [16] compared the accuracy of the decision tree and bayesian models for predicting the academic performance of undergraduate and postgraduate students and observed that decision tree models accomplished better accuracy than a bayesian classifier.

Al-Radaideh et al. [17] proposed a classification model to enhance the quality of the higher educational system by evaluating students by predicting their final grades in courses. They used three different classification methods ID3, C4.5, and the Naïve Bayes. The results indicated that the decision tree model had better prediction accuracy than the other models. Muslihan et al. [18] compared artificial neural network and the combination of clustering and decision tree classification techniques for predicting and classifying student's academic performance and found that combination of clustering and decision tree techniques achieved high accuracies. Ramaswami and Bhaskaran [19] constructed a predictive model called CHAID with 7-class response variables by using predictive variables obtained through feature selection, to evaluate the academic achievement of students at higher secondary schools in India. Tripti et al. [20] used J48 decision trees and random forests to predict the performance of Masters of Computer Applications (MCA) students in their work and observed that random tree has better accuracy, and it consume less time than the J48 decision tree algorithm. B, Minaei-Bifgoli et al. [21] developed a web-based interface to track learning patterns in students and used genetic algorithms to predict student's final grade.

Our proposed approach is different from what has been proposed in the following aspects: Work has been done to predict failing students or predicting final grades but in our approach, we are building a system to predict grades, detect poor-performing students and presenting these insights to students and instructors so that they can act well informed. Moreover, our approach uses student personal data and class performance data to build models. A relational database schema was designed to store and use the student information. Additionally, the relational database is scalable across various courses and universities. Moreover, unlike others, our approach predict grades and detects poor-performing students based on the data which is collected in the first two to three weeks of the start of the semester to present them to students and professors so that they have enough time to take necessary actions.

3. Architecture of the Proposed System

This thesis centers on the design and implementation of the RSPEF system whose primary goal is to improve the graduation rate at the University of Houston, by providing feedback to students and instructors. In this chapter, we will present and discuss the architecture of RSPEF.

The RSPEF system architecture has three layers: i) a Data Pre-processing Layer ii) a Machine learning & Data Analysis (MLDA) Engine and iii) an Application Layer. Figure 3.1 illustrates the RSPEF system architecture.

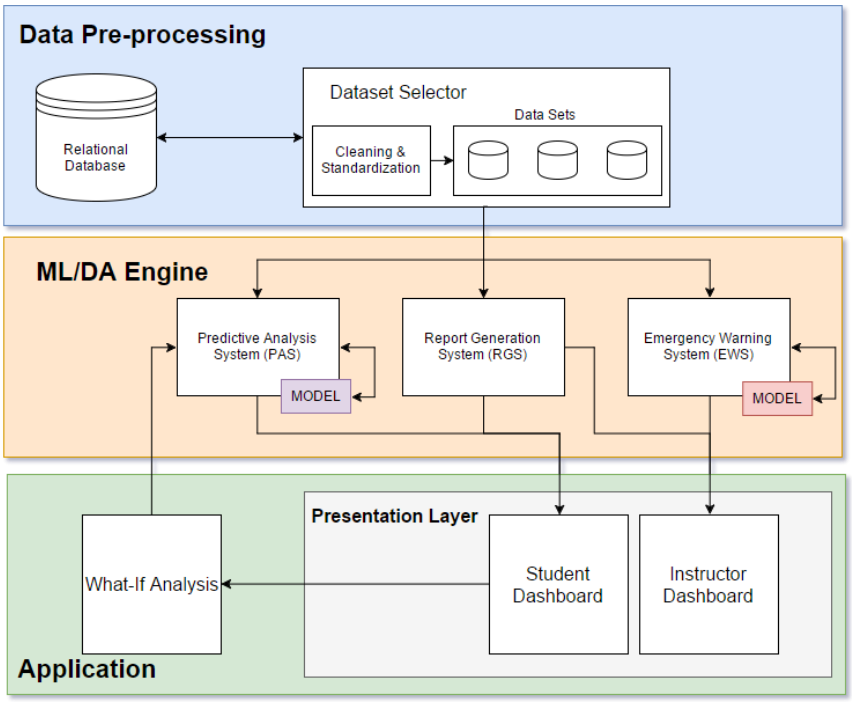

Figure 3.1: Architecture of RSPEF system

3.1 Data Pre-Processing Layer

The Data Pre-processing Layer has two components a relational database and a dataset selector. Performance and personal data of students are stored in the relational database. A dataset selector is a programmed API that creates datasets from the relational database, cleans them and standardizes them; these datasets will be later passed as input to the machine learning and data analysis (MLDA) engine.

When new information is inserted or updated in the relational database such as student submitting a survey will insert student responses in the database or instructor updating the scores will update existing information in the database. When such an insert or update event takes place, will invoke the dataset selector. The dataset selector is programmed to obtain metadata from these tables which will be used to identify training datasets. When a student submits a survey an insert event happens which will evoke dataset selector which in turn collects metadata by identifying the student by "student ID" and using this ID to retrieve a list of courses that this student has taken. By using the information in the metadata, the dataset selector will generate SQL commands to obtain all survey training datasets for each of the courses. Similarly, when an instructor updates student's scores, an update event happens which will evoke the dataset selector to collect metadata by identifying the course in which performance data was updated and retrieves all the relevant training datasets which have the same structure of the current course. The dataset selector extracts these datasets and passes them as input to MLDA engine. Current performance dataset and its training dataset are passed as input to Predictive Analysis System (PAS) and Report Generation System (RGS). Current survey data and its corresponding training

datasets are passed into Emergency Warning System (EWS) and Report Generation System (RGS).

3.2 Machine Learning & Data Analysis Engine

Machine Learning & Data Analysis (MLDA) Engine has three sub-systems: i) a Predictive Analysis System (PAS) ii) a Report Generation System (RGS) and iii) an Emergency Warning System (EWS).

The purpose of the Predictive Analysis System, also called PAS, is to predict student grades. PAS receives training and current performance datasets from the dataset selector. A model is learned from the training dataset. This model uses current performance datasets and predicts grades for students who are currently taking a course.

The purpose of the Report Generation System helps students assess themselves amongst their peers by generating summaries specific to courses from the student-performance and survey datasets so they can be displayed to students and instructors using the application layer. RGS receives student survey and performance datasets from the dataset selector and generates statistical summaries from them. For performance datasets, RGS will compute mean, median and standard deviation for each column in the dataset. For survey datasets, RGS computes the individual conditional probabilities of student passing the course given his age, gender, race, major or minor. For example, given that a student is male the probability of the student passing the course is computed. These outputs are passed into application layer as input.

The purpose of the Emergency Warning System (EWS) is to detect poor-performing students, very early in the semester by building models to classify students into “Pass” and “Fail” classes. EWS receives training datasets and the dataset which needs classification from the dataset selector. EWS learns models from the training datasets which will be used to determine if the student will fail the course for each course the student is enrolled into. A list of students ID’s and their corresponding course ID’s who are detected to fail that course is created which will be displayed on instructors dashboard by the application layer.

3.3 Application Layer

In the Application Layer, the presentation layer is responsible for visualizing the outputs from MLDA Engine for displaying them on dashboards and the what-if analysis component supports students to enter their future scores and see how it affects the predicted grade. The student dashboard displays student’s grade predicted by the PAS, summaries of each course elements such as mean, median score, and standard deviation, and individual probabilities of a student passing the course given his gender, race, major and age group computed by RGS. Figure 3.2 illustrates an example display of the student dashboard. The instructor dashboard displays a list of poor-performing students detected by EWS. The instructor can click on a student from the list and view information about the student such as student name, description, grade predicted by PAS and summaries generated by RGS. Figure 3.3 illustrates an example display of the instructor dashboard.

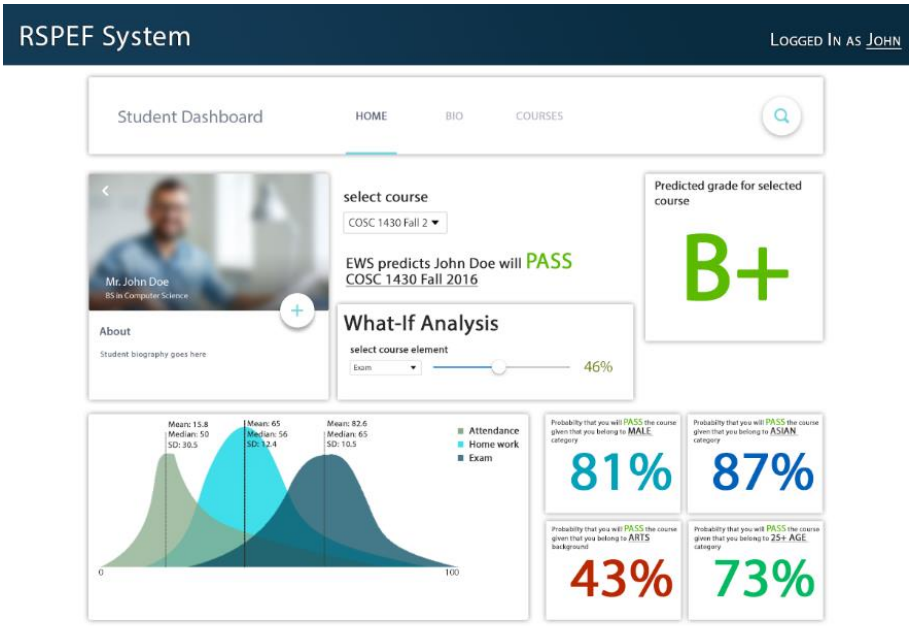

Figure 3.2: Example display of the student dashboard

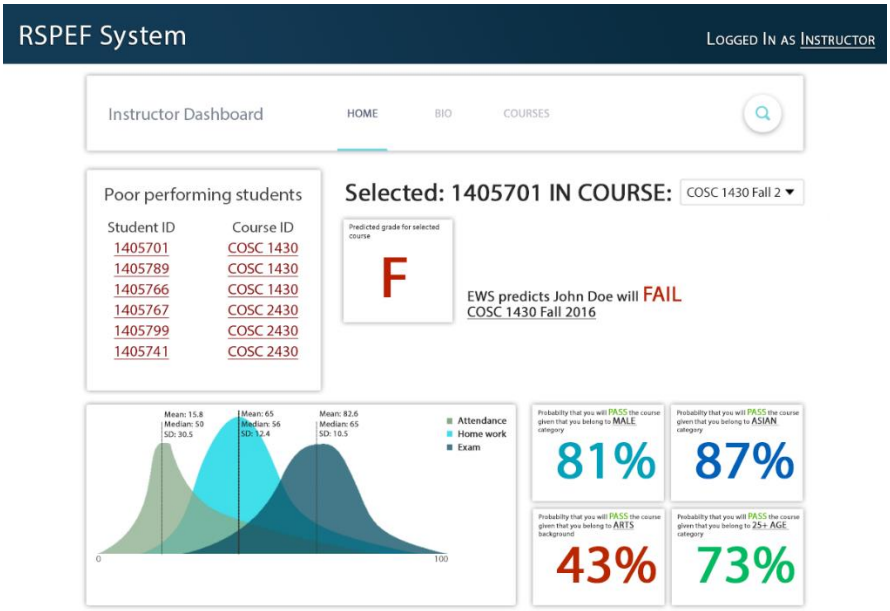

Figure 3.3: Example display of the instructor dashboard

Student dashboard supports what-if analysis: students can use sliders to change the scores they obtained or can input future scores; this information will then be used to predict their final grade based on the information they provided. The role of the application layer is to create specific dashboards for students and instructors. These dashboards visualize and display information such as predictions and summaries from PAS, EWS, and RGS which can be easily comprehended by students as well as instructors.

4. Design of Relational Database to Store Student Data

We collect a wide range of student information from surveys to student scores in courses they take. The RSPEF system continually collects and updates data from students when submitting surveys and instructors when uploading student scores through an interface. The relational database stores all information necessary to build models, it is provided with interfaces to extract relevant information from students and allow instructors to upload and save course and student-performance information, a dataset selector interface on top of the relational database will extract relevant datasets that are required to build models. Moreover, a schema is designed that provides a unified representation of student and course data that can be used across courses. The data we collect is naturally relational and requires extensive capabilities to perform complex join operations between related tables. Therefore, there is a need to store them in independent tables which share relationships. The relational database best fits our needs as it is a well-understood approach, maps well to data with multiple relationships between multiple entities and support extensive join and subset capabilities. In this chapter, we will discuss data acquisition, pre-processing, archiving and design of the relational database.

4.1 Data Acquisition

We collect a wide variety of datasets to perform statistical analysis and build models. The primary sources of data are students and instructors. The data is collected via a web application through which students are asked to fill out surveys and instructors use an

interface to upload data along with metadata (course name, semester etc.) into the database. Instructors upload old labeled data from previous semesters and new data from the current semester is continuously updated.

Students are asked to register themselves on our web application. During the process of registration, they are asked to answer 35-questions-long questionnaire and responses are recorded in our database and we will refer to this data as student survey data. Instructors upload and update student scores continuously during the semester as and when they are available. We will refer this data as student-performance data. Section 4.1.2 and 4.1.3 discuss student survey data and student-performance data respectively.

4.1.1 Student Survey Data

We have conducted surveys to uncover answers to specific and important questions related to students. Rather than using academic performance to determine student success, we consider social, economic, institutional, and individual-specific factors to determine success. Students fill out this survey on a web application. We will use this data to build a machine learning models to classify students into the pass and fail categories. Students classified as fail are detected as poor-performing / failing students. All questions in the survey are classified into five categories.

1. Demographic & Personal Questions
2. Financial Questions
3. Course-Specific Questions
4. Student-Specific Questions

5. Student-Institution Involvement Questions

Table 4.1 through table 4.5 will list each of the questions, their appropriate response range and which classification they come under. For example, the question “What is your gender?” can accept either of male or female as a response. Other responses are strictly not allowed so, the response range for this question contains only male and female. Questions like “Which high school did you graduate?” will have a textbox to allow students to type the name of the high school they have graduated from. The response range for this question is however very large as there are over 36,000 high schools in the USA and is dealt by replacing the text box with a drop-down list of high schools.

Table 4.1 lists out thirteen questions which collect demographics (age, gender, ethnicity) and some questions are personal information (major, job, health). All questions are encoded with a unique code, the first letter of this code represents the category they belong to, following digit is the identifier for the question. D1, asks students to enter their ID which will be used to associate student scores with this survey. D2 & D3 are age and gender. D4, student’s major, for example, computer science. D5 & D6 are racial ethnicities and high school students graduated from. D7, D8 & D11 evaluate the quantity and quality of time a student spends every day. D7 asks if his/her day-time is occupied by a job, if the answer to this question is a no it means the student has more time in hand which he could utilize on something more productive. D8, if the job is major related, if yes, then the time he spends on his job is productive in building his career. Similarly D11, a number of hours the student spends on a job will determine the remaining hours in hand he could use to prepare for

homework or an exam for the current course. D9, D10, D12 & D13 are useful to determine student's average health.

Category Name	Code	Question	Response Range
Demographic &	D1	What is Your ID?	{text}
Personal	D2	What is your age?	{0 to 100}
Questions	D3	What is your gender?	{male, female, other}
(D)	D4	What is your major?	{text}
	D5	What is your racial ethnicity?	{Hispanic, Asian, native, African American, white}
	D6	Which high school did you graduate?	{text}
	D7	Do you work on/off campus?	{yes, no}
	D8	Is the job, major related?	{yes, no}
	D9	Do you have any health issues?	{yes, no}
	D10	What are your health issues?	{multiple texts}
	D11	How many hours a week you work?	{0 to 40}
	D12	How many hours of sleep, you get every day?	{0 to 20}
	D13	How many meals have you a day?	{0 to 5}

Table 4.1: Demographic & personal questions and their respective response ranges

Table 4.2 lists out five questions which focus on financial standing. F1, asks students if their parents approve the major they have taken in the university, the student's response will determine how much, morally and financially his/her parents support their education. F2 and F3 are follow up questions to F1. These questions have an important effect on academic success. F4 & F5, Financial aid reduced a student's burden towards paying tuition or a loan. Students work overtime or night shifts to earn their tuition. A financial aid can help such students to cease working overtime or night shifts and spend more time on their education. The amount of scholarship or aid received is proportional to the burden lifted upon students.

Category Name	Code	Question	Response Range
Financial	F1	Do your parents approve of your major you chose in the university?	{yes, somewhat, no, don't know}
Questions	F2	Please estimate your household's annual income in USD.	{Positive Integers}
(F)	F3	What percent does your parent support financially towards your study?	{0 to 100}
	F4	In the previous semester, did you receive any financial aid, grants, or scholarships?	{yes, no, don't know}
	F5	How much did you receive as aid, grant, or scholarship?	{Positive Integers}

Table 4.2: Financial questions and their respective response ranges

Table 4.3 lists out four course-specific questions. As a student cannot evaluate a course before it begins we conduct this part of the survey towards the end of the course. This category of the survey is summarized and presented to the course preparers and instructors, which in turn will help them to improve the course. Other than using this survey as a feedback to instructors, we will use it to identify the students who really love the course and enjoy it, and the ones who do not like the course and disappointed with it. Assuming students satisfaction towards the course will have its influence on student achievement, we include these questions in our survey.

Category Name	code	Question	Response Range
Course-Specific	C1	On a scale of 0 to 5, how difficult do you find this course?	{0 to 5}
Questions	C2	On a scale of 0 to 5, how heavy do you find the workload of this course?	{0 to 5}
(C)	C3	On a scale of 0 to 5, how will you rate the helpfulness of your instructor?	{0 to 5}
	C4	On a scale of 0 to 5, how will you rate the helpfulness of the teaching assistant?	{0 to 5}

Table 4.3: Course specific questions and their respective response ranges

Table 4.4 lists out six student-specific questions. S1 & S2 will record student's proficiency and experience. While S3 records if the student is getting any extra help for the student

group. S4 & S5 assess students based on the time they take to complete tasks or preparing for exams. S6 is a feedback question on learning methods students generally prefer.

Category Name	code	Question	Response Range
Student-Specific	S1	On a scale of 0 to 5, how will you rate your proficiency in the field you are majoring in?	{0 to 5}
Questions	S2	On a scale of 0 to 5, how will you rate your experience in the field you are majoring in?	{0 to 5}
(S)	S3	Do you have a study group for this course?	{yes, no}
	S4	How many hours do you plan to spend each week on studying or preparing homework?	{0 to 20}
	S5	How many hours do you plan to prepare for an upcoming exam?	{0 to 20}
	S6	What learning method do you prefer?	{visual, auditory, kinesthetic}

Table 4.4: Student specific questions and their respective response ranges

Table 4.5 lists out seven questions, identifying student's involvement in institutional support and other activities. The University of Houston supports academic achievement in students by establishing tutoring and support centers with advisors and counselors to advise them. Also, these centers provide tutoring to students for several courses. I1 & I2 are questions related to these support centers. I3 records student's average attendance. As attendance to any course is important to get maximum knowledge out of that course, it is considered that a student will low average attendance will grasp less information or knowledge of the course. Which is why this feature is considered important. I4 asks students, the number of years they believe it is going to take them to graduate with a degree. I4 reveals the confidence of the student towards his academics and is an important factor. I5, I6 & I7 inquiries students if they are involved in a student organization and if it was a major related one. These questions are reasonably good factors in determining student success as they add to the student's holistic experience in the major they have chosen.

Category Name	code	Question	Response Range
Student-Institution Involvement Questions (I)	I1	How often do you meet a counselor/advisor?	{weekly, monthly, quarterly, never}
	I2	How often do you go to tutoring centers like CASA or Learning Support Service?	{weekly, monthly, quarterly, never}
	I3	How often do you skip classes?	{weekly, monthly, quarterly, never}
	I4	How many years will it take you to complete your degree requirements?	{0 to 20}
	I5	Are you a member of a student organization at the university?	{yes, no}
	I6	Are you a member of a student organization at the university related to your major?	{yes, no}
	I7	What is the name of the major related organization?	{text}

Table 4.5: Student-Institution involvement questions and their respective response ranges

4.1.2 Student-Performance Data

A student-performance dataset will be used to predict student's final grade. This dataset comprises scores obtained by students. To evaluate a student's coursework, instructors conduct evaluation procedures, both subjective and objective like exams, quizzes, assignments, or projects. These are referred as course elements. Each of these elements is given scores based on how students have performed. Usually, the instructors have a certain weight for each of the course elements which are later used to calculate weighted scores. In our tables, these weights are written next to the course element in brackets. The instructor then records scores in an EXCEL or a CSV file. There are two versions of this dataset: labeled and unlabeled or new data. Labeled data is older data with the final grades and are used to train and evaluate ML models. These models are used to predict labels, in this case, grades, for new data. These datasets are uploaded to the database by instructors

using a web application interface. A typical student-performance dataset is illustrated in Table 4.6

	Course_Element 1(10)	Course_Element2 (20)	Course_Element 3(70)	Weighted Average	GRADE I
0001	50	100	50	60	B-
0002	100	100	100	100	A
0003	25	25	25	25	F

Table 4.6: Student performance dataset with labels, and weight of each course element indicated in brackets as percentages (they add up to 100%)

4.2 Data Pre-Processing

Data pre-processing is a necessary task to maintain quality data/datasets prior to the use of machine learning algorithms. Real world data, specifically of those collected from surveys could be dirty and could drive the process of building ML models useless. This is mainly due to incomplete data, noise, and inconsistent data.

When a student intentionally or unintentionally leaves out questions in the survey, Incomplete datasets are created. In such cases, data values are imputed but in most cases where imputing doesn't work, we remove that instance entirely.

Noise is dealt with in the data collection stage. Rather than using text boxes to capture responses, we use drop-down lists, sliding bars and radio buttons to eliminate noise. For example, a question in the survey asks students for high school they graduated from which can have infinite possible inputs. To avoid incorrect or non-standard inputs, we use a web feature, which upon starting to write the name of the high school in the input box, the browser send a request to a database server to look for all possible matches for the partial

word (involves Natural Language Processing & Hidden Markov Model) for which the plugins are already available (Flexselect, Autocomplete.js), and returns a drop down menu with few suggested high schools. So, now the student has to select from the drop down menu.

Inconsistent data is nothing but sparseness in data, they are capable of learning wrong or biased models, we normalize student scores to keep the data consistent within a small range. In student-performance, data student scores can be sparse, for example, one student has a final exam score of 100, first best score. The second best score is 55 and the rest of the scores are distributed below 55. In this case, we could learn a biased model, to avoid this situation we normalize the scores using the Z-score procedure. Equation 4.2.1 describes the formula to calculate the z-score for a particular value in a data column.

$$X_{new} = \frac{X-y}{\sigma} \tag{4.2.1}$$

Once we have pre-processed the datasets we pass them to next stage of building ML models for PAS and EWS.

4.3 Database System Architecture

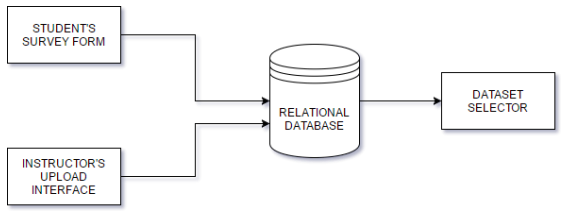

Figure 4.1: Database system architecture

The architecture of the database is required to understand how the data in populated in the database and the role of the database in providing datasets to learn models. The student's survey form is a web-based questionnaire where students enter responses to the questions mentioned in section 4.1.1. Instructor's upload interface is also a web-based interface, lets instructors upload CSV (comma separated values) or EXCEL (usually tab separated values) files to the database. Dataset selector identifies datasets which need predictions and determines appropriate training sets. These datasets are preprocessed and further divided into smaller datasets. If a course is identified for prediction, it will extract datasets of that course from previous semesters to represent training data. Now from both the current and training datasets student-performance dataset and student survey datasets are extracted. These datasets are then fed as input to PAS, EWS, and RGS.

4.4 Design of Relational Database

The data we collect in our research is naturally relational and requires extensive capabilities to perform complex join operations between related tables. In the following sections, we will explain the entities and relationships in our ER model, entity-relationship model, and relational schema model.

4.4.1 Entity

Our model has seven entities,

- Student
- Professor
- Course

- Section
- Course Element
- Survey
- Performance Summary

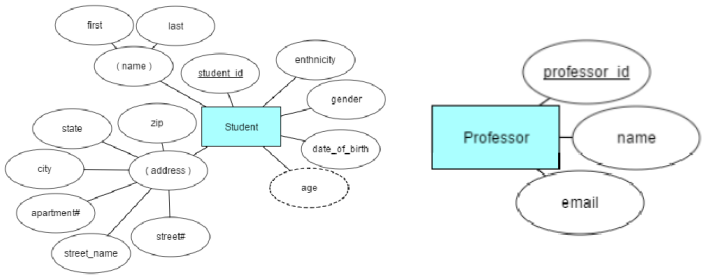

Figure 4.2: Student and professor entities and their associated properties

Student entity has an identifier student_id; the name is a composite property and comprises two individual properties: first and last name; the address is a composite property comprising: street number, name, apartment number, city, state, and zip code; ethnicity, gender, and date of birth. Age is a derived property (from the date of birth). Professor entity type has an identifier professor_id, name, and email address. See figure 4.2.

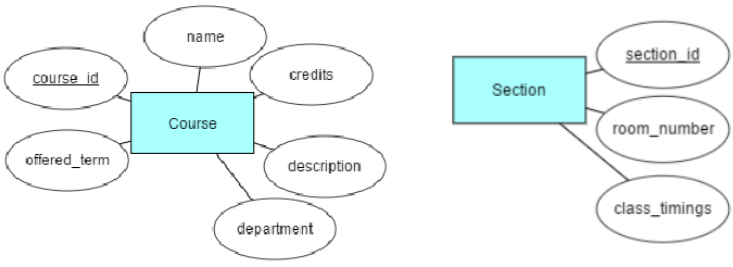

Figure 4.3: Course and section entities and their associated properties

Course entity has an identifier course_id, the name of the course, optional description, term in which it is being offered, department and the number of credits awarded. Section entity has an identifier section_id, room number, and timings. See figure 4.3.

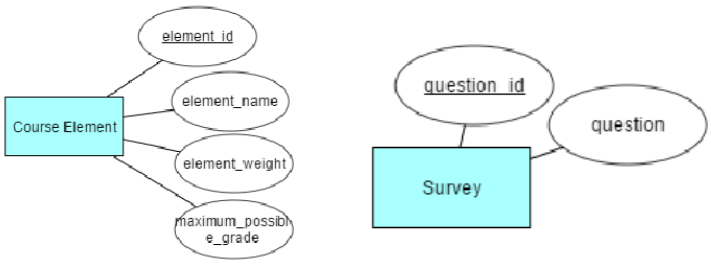

Figure 4.4: Course element and survey entities and their associated properties

Course element has an identifier element_id, the name of the element, weight, and maximum possible score. Survey entity has an identifier survey_id and question. This table has only 35 instances as there are 35 survey questions. The table remains same throughout unless the administrator decides to add or remove questions. See figure 4.4.

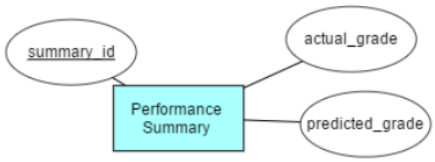

Figure 4.5: Performance summary entity and its associated properties

Performance summary entity is where we intend to store student results and ML predictions. It has a summary_id, actual grade obtained and grade predicted by PAS. See figure 4.5.

4.4.2 Relationship

A relationship defines how one entity depends on another. We have identified eight relationships among seven entities. A relationship represents a table in RDBMS, mapping instances of one entity with another.

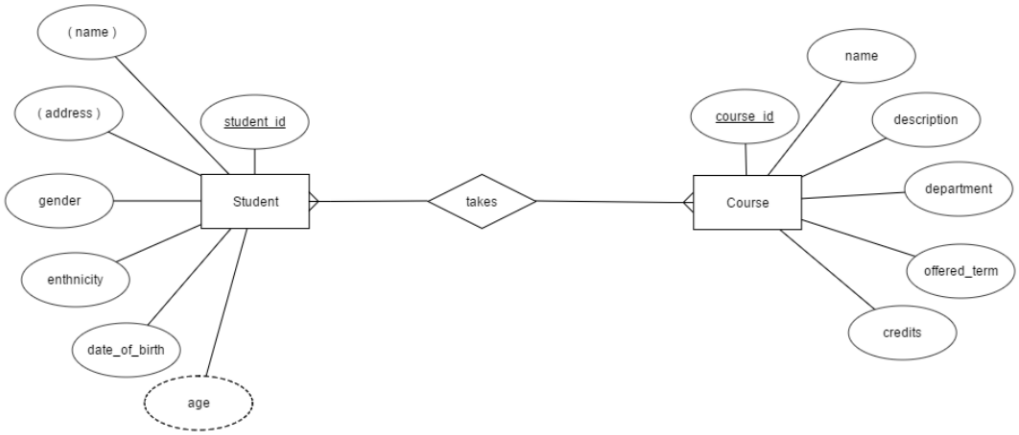

Figure 4.6: Illustration of “student takes course” relationship

The “student takes course” relationship is many-to-many cardinality, one or more students can enroll in one or more courses. The table “takes” has student_id and course_id as foreign keys which represent a list of student and course mappings. See figure 4.6.

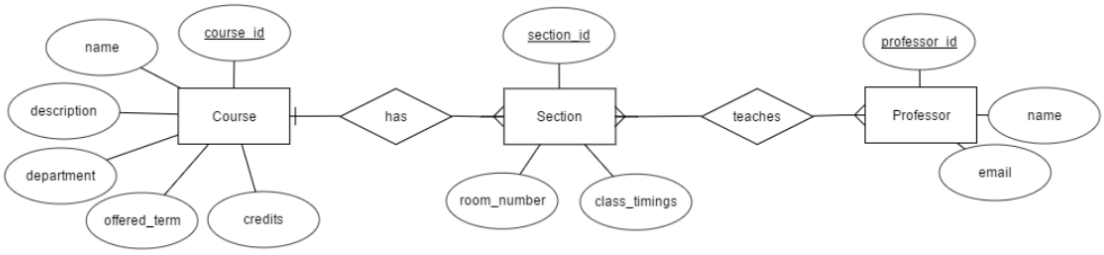

Figure 4.7: Illustration of “course has section” and “professor teaches section” relationship

The “course has section” is a special kind of relationship. Unlike the prior relationship, there is no explicit table for ‘has’ instead section_id is placed into course as a foreign key. It follows one-to-many cardinality, one course can have one or more sections. The “professor teaches section” is many-to-many cardinality, one or more professors can teach one or more sections. The table “teaches” has professor_id and section_id as foreign keys which represents a list of professor and section mappings. See figure 4.7.

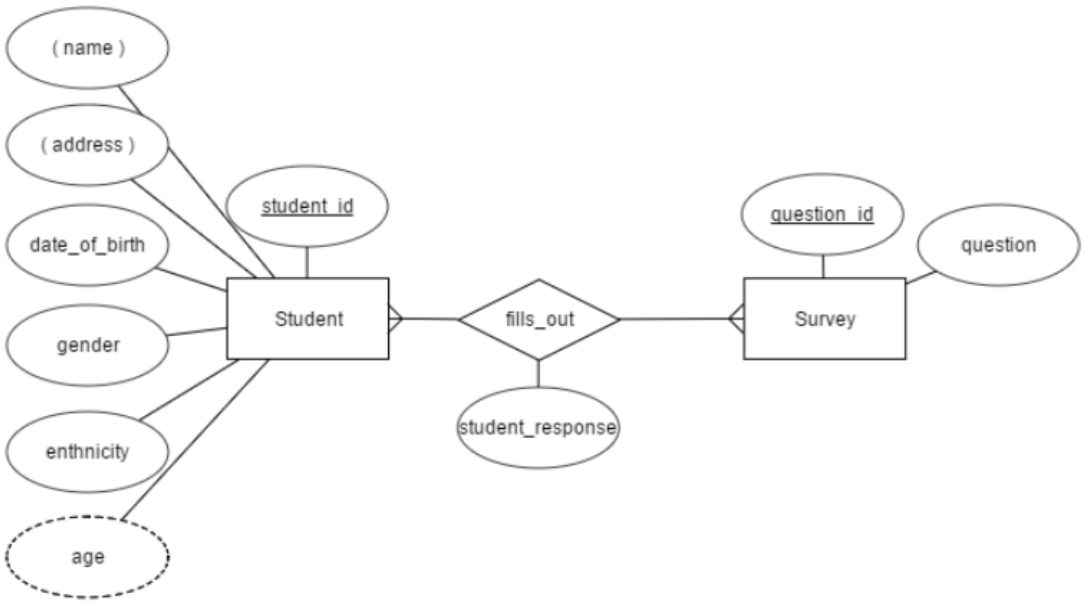

Figure 4.8: Illustration of “student fills_out survey” relationship

The “student fills_out survey” relationship is many-to-many cardinality, every student must answer every question in the survey. Responses are recorded in “fills_out” table under ‘student_response’ column. See figure 4.8.

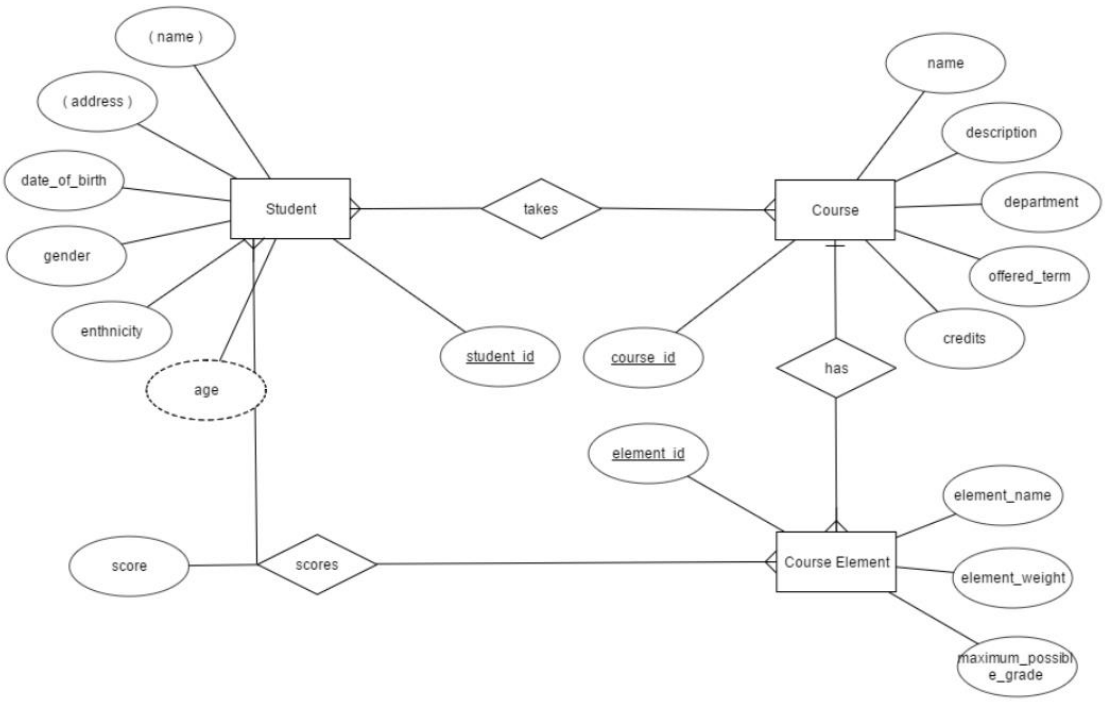

Figure 4.9: Illustration of "course has course element" and "course element scores students" relationships

The "course has course element" and "course element scores students" are one-to-many cardinalities, every course has more than one course elements while every student enrolled in these courses are scored for all the associated course elements. For example, a student's score in the final exam for the enrolled course 0001 is recorded under 'score' column in table scores. See figure 4.9.

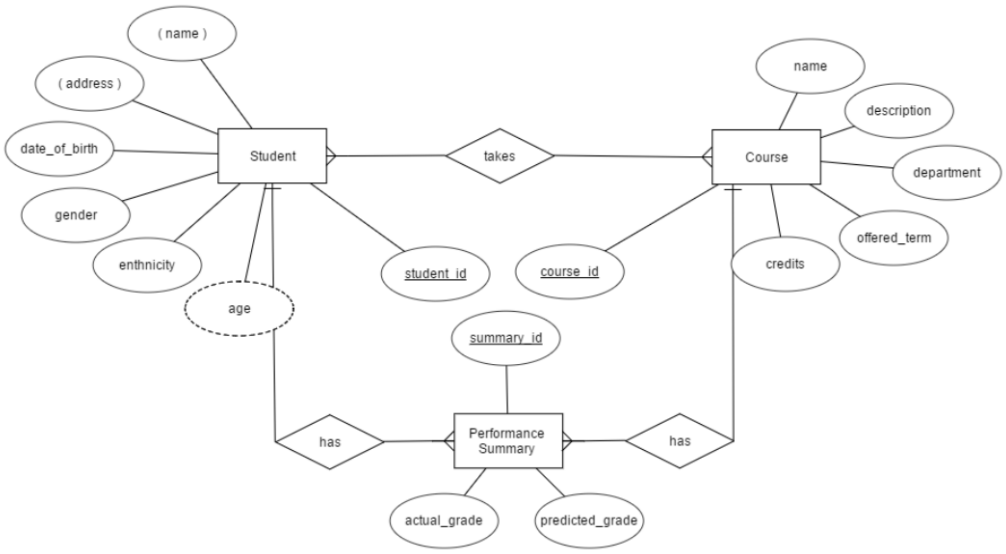

Figure 4.10: Illustration of "course has performance summary" and "student has performance summary" relationships

The "course has performance summary" and "student has performance summary" are one-to-many cardinalities. These relationships allow capturing student's summary for a particular course in performance summary table.

4.4.3 Entity-Relationship Model

An entity-relationship diagram also popularly called ER diagram is a schematic representation of a database, generally used by database practitioners to design databases. For a given problem there can be many possible ER Representations, we try to deduce the set of all possible ER diagrams to one good diagram or model which can satisfy our goal [22]. We have used an online tool called ERDPlus to design our model. Figure 4.11 shows the ER model used.

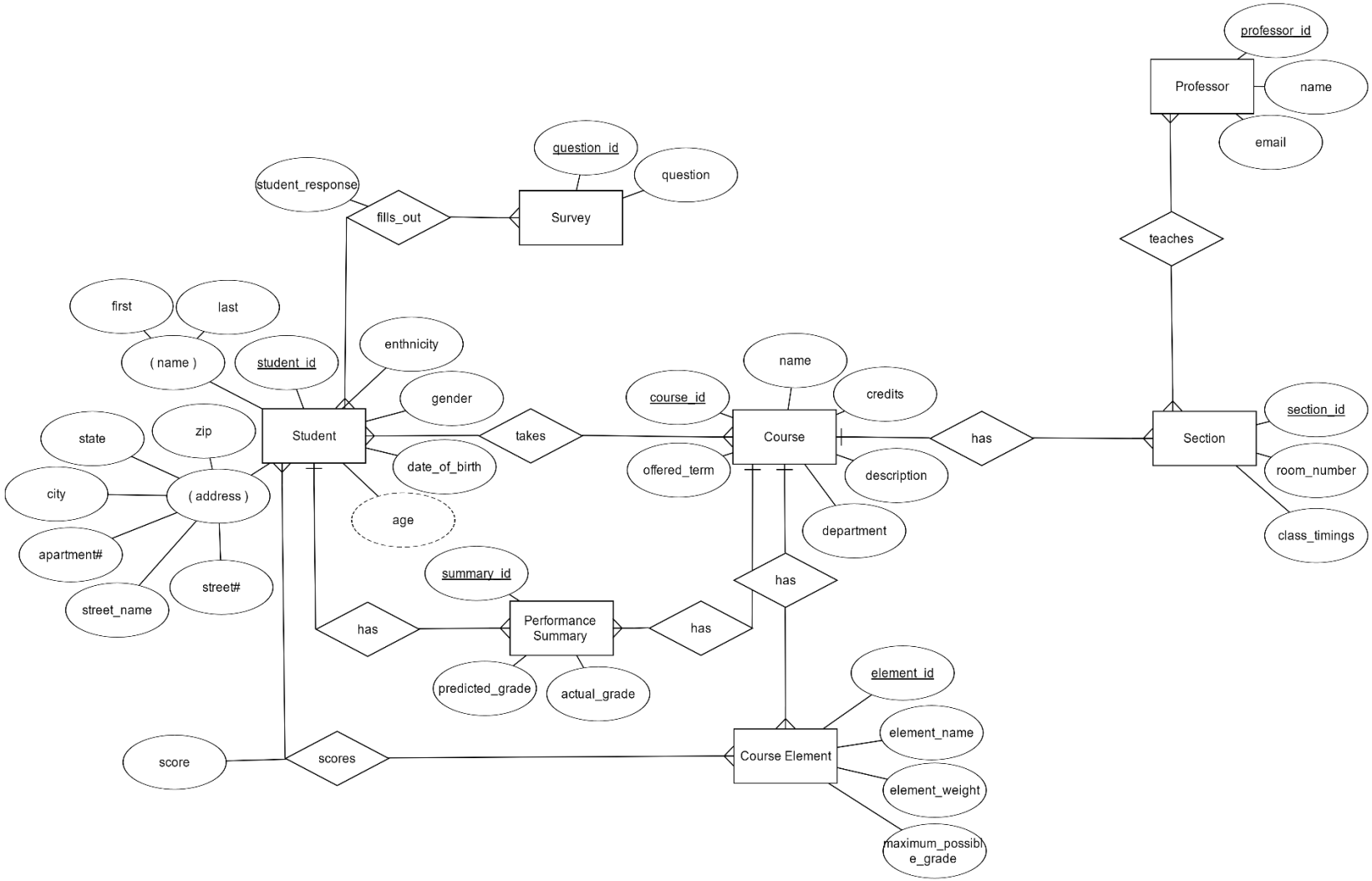

Figure 4.11: Entity-Relationship diagram for storing student data

This model recites the structure of the database. This model captures student's data from more than one courses which have one or more sections taught by one or more professors. Student, course, section and the professor can have specific information associated with them. A survey entity is a collection of 35 mandatory questions and each student must be associated with all of them. And the responses go into the relationship 'fills_out'. A course will have more than one course element associated with it. And every student is scored for the course elements which belong to the course they have taken. Performance summary shares a similar relationship with student and course. Therefore, information like final grades and predicted grades which are specific to student and course are recorded here.

4.4.4 Relational Schema

The below figure 4.12 depicts the relational schema for storing student data. The tool, ERDPlus has a feature to convert ER diagrams into relationship schema diagram. All entities and relationships convert into tables. The relationship 'has' will not convert into a table. Rather these relations are treated specially, a foreign key in placed in one of the entities which will reference a primary key in another entity.

In " student takes course" and "professor teaches section" relations, 'takes' and 'teaches' tables have student-course and professor-section mappings respectively. In "course has section", which has many-to-one cardinality, section_id is put in course table as a foreign key.

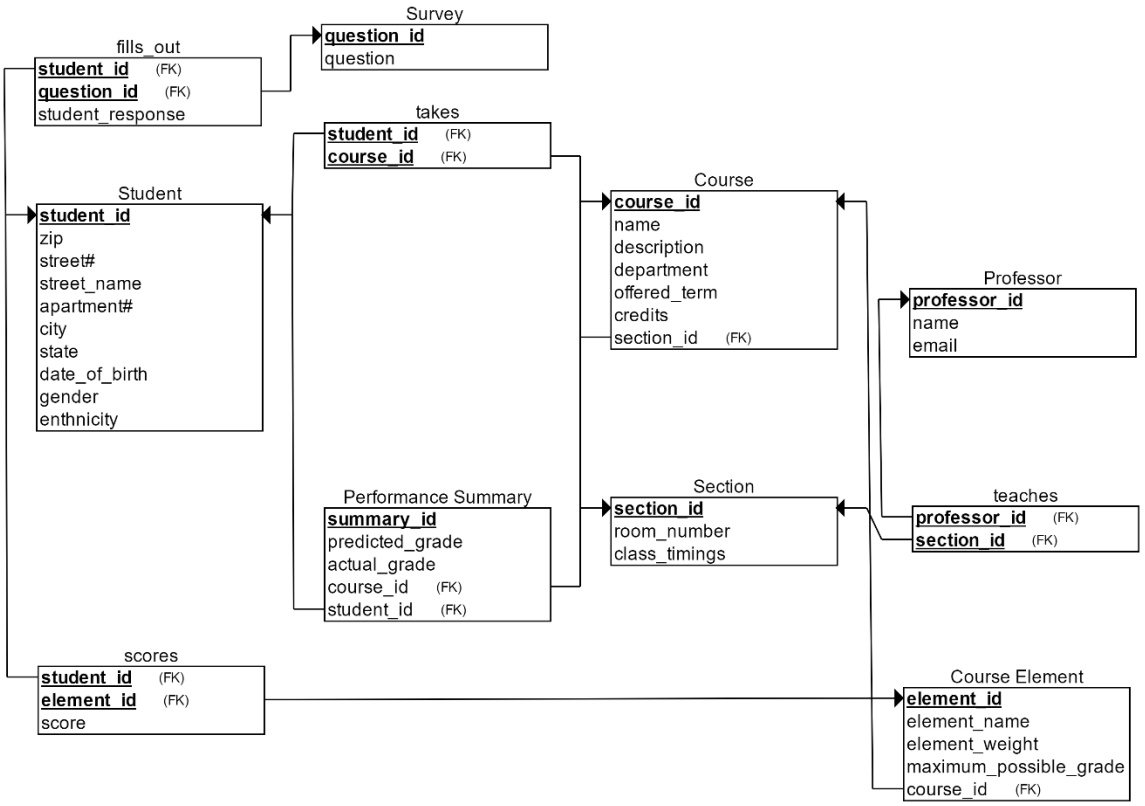

Figure 4.12: Relational schema for storing student data

In “student has performance summary” and “course has performance summary” relationships, which are one-to-many cardinalities, student_id, and course_id are placed in ‘performance summary’ table as a foreign key. To identify a particular student's grade in a particular course, we join three tables (student, course and performance summary) and filter by specific student_id and course_id. Similarly, “course has course element” puts course_id as a foreign key in ‘course element’ table.

Both “course element scores student” and “student fills_out survey” are many-to-many cardinalities. Table ‘scores’ has both associated foreign keys and the column ‘score’. Table ‘fills_out’ has both associated foreign keys and the column ‘student_response’.

By following these standard procedures to convert an ER diagram, we generate a reasonable relational schema diagram from which one can easily write DDL (Data Definition Language) in SQL to create tables for any relational database in the market. We are using SQL server 2016 from Microsoft.

4.5 Data Archiving

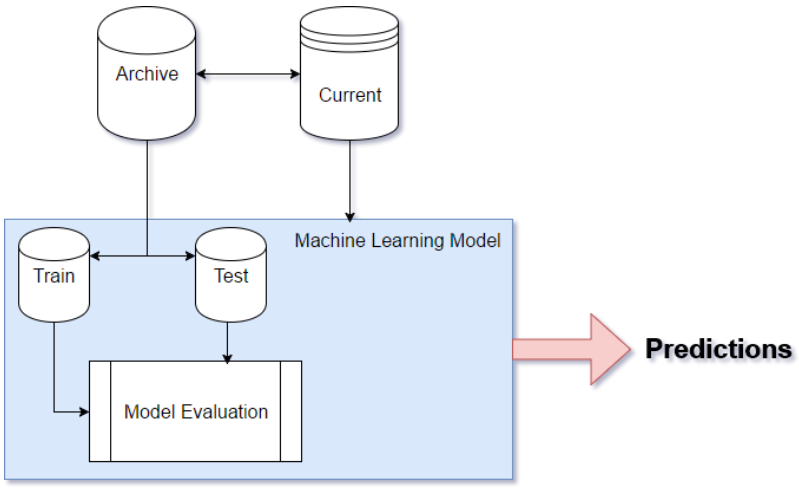

Figure 4.13: Learning models from archived datasets

Archived data is the major source for training and test data. We only store current semester data in our database. Once the data is a more than 12 months old we move it into an archive database wich is a replica of our current database. When building ML models data from the archive is requested. The data is proportioned into training and test sets. A model is built and evaluated using the archive data itself. During deployment, current data is passed into the model to make predictions for the current or new data. See figure 4.13

5. Learning Models that Predict Student Performance

Various models are learned by Predictive Analysis System (PAS) and the Emergency Warning System (EWS) to predict student grades and to identify poor-performing students in a course. In this chapter, we will briefly discuss the different approaches that were used to obtain these models. Moreover, we will discuss the results obtained by using various datasets to train these models and will compare the performance of different types of models using a set of evaluation measures. Sections 5.1 and 5.2 will discuss the datasets used in the experiments and introduce naming conventions; sections 5.3 and 5.4 will discuss the machine learning tools used to build and evaluate models and sections 5.5 and 5.6 will give an overview of experiments and compare the obtained results.

5.1 Datasets

Over the period of two semesters (FALL 2015 & SPRING 2016) we have collected survey and performance data from the course: COSC1430 (Introduction to Programming). The data this course in the fall semester of 2015 has 104 instances and will be used to perform experiments. We will refer to this dataset as COSC1430 (FALL 2015). Models are trained and tested by cross-validation technique.

5.1.1 Student Survey Dataset

Survey responses from COSC1430 (FALL 2015) reside in the relational database and have the following structure:

Student_id, response1, response2 … response 35, grade

(See section 4.1.2 for questions and their possible responses). Target class here is the grade obtained by the student at the end of that semester.

5.1.2 Student Performance Dataset

This dataset comprises of student-performance data from the course, COSC1430 (FALL 2015). This course has the following course elements: attendance, homework, class project, exam, and quiz. A final grade is one of 12 grades which is recorded in the column called "grade I". Columns, grade II (see table 5.6 for the mapping between grade I and grade II) and grade point can be derived from grade I. See table 5.1 and 5.2.

Identifier	Course_Element1	Course_Element2	Course_Element3	GRADE I	GRADE II	GRADE POINT
0001	50	100	70	B+	B	3.33
0002	100	100	80	A-	A	3.66
0003	25	25	25	F	F	0

Table 5.1: Structure of COSC1430 (FALL 2015) dataset in train mode

Identifier	Course_Element1	Course_Element2	Course_Element3
0001	50	100	70
0002	100	100	80
0003	25	25	25

Table 5.2: Structure of COSC1430 (FALL 2015) dataset in test mode

5.1.2.1 Datasets to Predict a Grade I

This dataset is a subset of table 5.1, all columns except grade II and grade point columns from this dataset.Typically in the University of Houston, a student is awarded a grade from a pool of 12 grades: A, A-, B+, B, B-, C+, C, C-, D+, D, D- and F. Special grades such as I (Incomplete), W (withdraw) and NO-grade are removed from the dataset. We will call this dataset "dataset-o1" (o1 stands for output 1) indicating that our target class is column "grade I". See table 5.3.

Identifier	Course_Element1	Course_Element2	Course_Element3	GRADE I
0001	50	100	70	B+
0002	100	100	80	A-
0003	25	25	25	F

Table 5.3: Structure of dataset-o1

5.1.2.2 Datasets to Predict a Grade II

For this dataset, we have reduced the number of class variables from twelve to five, just using five grades: A, B, C, D, and F. From COSC1430 (FALL 2015), a subset of data is extracted by leaving out "grade I" and "grade point" columns. We will call this dataset "dataset-o2" (o2 stands for output 2) indicating that our target class is column "grade II". Table 5.4 illustrates the structure of dataset-o2.

Identifier	Course_Element1	Course_Element2	Course_Element3	GRADE II
0001	50	100	70	B
0002	100	100	80	A
0003	25	25	25	F

Table 5.4: Structure of dataset-o2

5.1.2.3 Datasets to Predict Grade Point

Because all grades represent numeric values, we convert can convert the grades in column "grade I" to their respective grade points. The column "grade point" in COSC1430 (FALL 2015) dataset is derived from "grade I", see table 5.5. We will call this dataset "dataset-o3" (o3 stands for output 3) indicating that our target class is column "grade point". Table 5.6 illustrates the structure of dataset-o3.

grade I	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
grade Point	4.00	3.66	3.33	3.00	2.66	2.33	2.00	1.66	1.33	1.00	0.66	0.00
grade II	A		B			C			D			F

Table 5.5: Mapping between grades I, grade II and their respective numeric grade points

Identifier	Course_Element1	Course_Element2	Course_Element3	GRADE POINT
0001	50	100	70	3.33
0002	100	100	80	3.66
0003	25	25	25	0.00

Table 5.6: Structure of dataset-o3

5.2 Dataset Representations

In the previous sections, we discussed the structure of dataset COSC1430 (FALL 2015) and extracted three datasets: dataset-o1, dataset-o2, and dataset-o3. In our experimental evaluation we will use different subsets of dataset-o1, dataset-o2, and dataset-o3, therefore, we have notations for the datasets so that we can easily describe their subsets using notations. See table 5.7. These notations start by mentioning all attributes by starting with a character ‘a’ followed by a sequence of respective column indices, then character ‘o’ followed by a sequence of target class indices.

Index	Meaning	Comment
1	Attendance	Attribute/Feature
2	Homework	Attribute/Feature
3	Class project	Attribute/Feature
4	Exam	Attribute/Feature
5	Quiz	Attribute/Feature
1	Grade I	Target Class
2	Grade II	Target Class
3	Grade Point	Target Class

Table 5.7: Notations used in our experiments I

When making predictions we frequently do not have data for all the course elements. For example, early in the semester, we might have only data only for attendance and homework. But, we need to generate prediction models for these partial datasets.

We use the following nomenclature to refer to such datasets: The course elements are numbered sequentially from left to right. To represent columns which are attributes of dataset-o1 (see table 5.3), we write the letter ‘a’ before “o1” and list all the columns by

their numbers after ‘a’. For Example, dataset a12o1 will have attendance, homework, and grade I attribute. If there are two same course elements but are added to the dataset at a later time they are merged into one in the pre-processing step. For example, if the dataset-o1 has two columns of the quiz (say, quiz 1 and quiz 2) but are added to the dataset on different dates, they are merged into one course element, quiz and the scores are averaged. Table 5.8 and 5.9 illustrates some examples of our nomenclature.

Identifier	attendance	homework	class project	GRADE I
0001	50	100	70	B+
0002	100	100	80	A-
0003	25	25	25	F

Table 5.8: Structure of dataset-a123o1

Identifier	attendance	exam	GRADE II
0001	50	100	B
0002	100	100	A
0003	25	25	F

Table 5.9: Structure of dataset-a14o2

5.3 Machine Learning Methods

Mathematically backed intelligence exhibited by machines or computers is called machine learning. Machine learning is applied in the process of predictive modeling to predict outcomes. If the outcome is categorical it is called classification and if the outcome is numerical it is called regression.

In this research, we will be using Naïve Bayes Classifier, Logistic Regression Classifier, Random Forests, Multi-Layered Neural Network/ Neural Networks, Linear Regression, and Support Vector Regression. A Data-mining and machine learning tool called Weka developed at the University of Waikato in Australia, will be used in this research. A Java code is written using Weka API to learn models. A notation which defines the dataset is passed as an input to the Java code. Then the code loads the dataset with respect to the notation entered, into the system memory. By analyzing the dataset the program will decide if it must choose classification tool or regression tool. If we are predicting a grade or classifying students into a pass or fail categories it is a classification problem and if we are predicting a grade point it is a regression problem. If program identifies them dataset to be a classification problem we build models using Naïve Bayes Classifier, Logistic Regression Classifier, Random Forests, and Multi-Layered Neural Network classifier otherwise we use Linear Regression and SMOreg to build models. Then we select the number of cross-validation folds that we like to use, which is 10 in our case.

The Naïve Bayes Classifier has two parameters which can be toggled based on the algorithm and user requirement which are "-K" which provokes the model builder to use kernel density estimator rather than normal distribution for numeric attributes, "-D" which allows the use of supervised discretization to process numeric attributes. Both parameters are turned on in our approach. Logistic Regression Classifier has two parameters which are "-R" and "-M" The first is turned on which use the ridge in the log-likelihood and the later represents a maximum number of iterations before convergence which is set to 10000. The Random Forest uses default parameters which are: "P", size of each bag which is set to

100, “I”, number of iterations which is set to 100, “S”, seed for random number generator which is set to 1 and “B”, to break ties randomly when several attributes look equally good. The Multi-Layer Neural Network uses 0.3 for learning rate, momentum rate for back propagation algorithm is set to default 0.2 and the seed value for the random number generator is set to 0.

Linear Regression in Weka uses Akaike criterion for model selection. The selection method is set to greedy and the ridge parameter is set to 1.0e-8. Weka uses SMOreg which is a Support Vector Regression for regression problems. Parameters are learned using various algorithms. The algorithm is selected by setting the RegOptimizer implemented in RegSMOImporved. The model parameters are set to default. Which are: “C”, complexity constant is set to 1, “T”, tolerance parameter for checking the stopping criterion is set to 0.001, “P”, and the epsilon for round-off error is set to 1.0e-12 and “L” the epsilon parameter in the epsilon –insensitive loss function is set to 1.0e-3. We build the models using these parameters, output evaluation metrics for different algorithms used and can compare them in the following sections.

5.4 Evaluation Methods.

Every model needs to be evaluated to understand how well it is performing in comparison to other models and how credible it is to be utilized in the real world. Error on the training data is not a good indicator of performance on future data. Because the model learns from the training data, estimates made to this data by the model are usually optimistic. In this

section, we will discuss various metrics of evaluation used to measure a model's performance.

5.4.1 Confusion Matrix

It is a table often used to describe the performance of a machine learning based classification model on the set of test data of which the true values are known. There are four basic terms in related to this matrix:

True Positive (TP): These are cases which are predicted to be true and are actually true.

True Negative (TN): These are cases which are predicted false and they are actually false.

False Positive (FP): These are predicted true but are actually false. (Type I error)

False Negative (FN): These cases are predicted false but are actually true. (Type II error)

	CONFUSION MATRIX	***Predicted*** TRUE	***Predicted*** FALSE
Actual	TRUE	TP	FN
Actual	FALSE	FP	TN

Table 5.10: Confusion Matrix

5.4.2 Accuracy

For classification, accuracy is straight forward. It is defined as the percentage of instances which are correctly classified (for all class variables). The formula for accuracy is given in 5.4.2.1.

$$Accuracy = \frac{\sum_{i=1}^{K} Correctly\ Classified\ Instances_i}{Total\ Number\ of\ Instances} \qquad (5.4.2.1)$$

When calculating accuracy for a regression model we can convert the grade points to respective grades (from the pool of 12 grades).Then use the formula to compute accuracy of the regression model.

5.4.3 Precision

Precision is the fraction of retrieved instances that are relevant. It is the ratio of correctly predicted positive observation to the total predicted positive observations.

$$Precision = \frac{TP}{(TP+FP)} \qquad (5.4.3.1)$$

5.4.4 Recall

The Recall is the fraction of relevant instances that are retrieved. It is the ratio of correctly predicted positive observation to all observations in positive (TRUE) class.

$$Recall = \frac{TP}{(TP+FN)} \qquad (5.4.4.1)$$

5.4.5 F-1 Score

The f-1 score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to

understand as accuracy, but F-1score is usually more useful than accuracy. In the other perspective, it can be said that F-1 score is the harmonic mean between precision and recall.

$$F-1\ score = \frac{2(Precision\ X\ Recall)}{Precision+Recall)} \qquad (5.4.5.1)$$

5.4.6 Mean Absolute Error (MAE)

Mean absolute error (MAE) is a quantity used to measure how predictions are to the eventual outcomes. Mean absolute error formula is given by 5.4.6.1.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \qquad (5.4.6.1)$$

5.4.7 Mean Absolute Error 2 (MAE 2)

MAE 2 is a customized metric to measure closeness between nominal variables. Usually, MAE cannot be computed for data with nominal or categorical labels. But our work around lets us compute MAE for such datasets. Since the grades can be sorted from highest to lowest (A to F), we assign each of the grades with a number from 1 to 12 or 1 to 5 depending on what our outcome is. After conversion we calculate MAE. using the formula in 5.4.6.1.

5.4.8 Root Mean Squared Error (RMSE)

Root mean squared error (RMSE) is a slight modification to MAE, instead of or taking the absolute value of the error it takes the square root of the square of the error. Which will

effectively amplify large error and damp smaller error. The formula to compute RMSE is given by 5.4.8.1. ‘e’ stands for an error.

$$RMSE = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2} = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(e_i)^2} \qquad (5.4.8.1)$$

5.4.9 Root Mean Squared Error 2 (RMSE 2)

RMSE 2 is a customized metric to measure closeness between nominal variables. Usually, RMSE cannot be computed for data with nominal or categorical labels. But our work around lets us compute RMSE for such datasets. Since the grades can be sorted from highest to lowest (A to F), we assign each of the grades with a number from 1 to 12 or 1 to 5 depending on what our outcome is. After conversion we calculate RMSE. using the formula in 5.4.8.1.

5.5 Overview of Experiments.

In our experiments, we will use COSC1430 (FALL 2015) course data. The datasets that were described in chapter 5.1 were used as training data to build prediction and classification models; next, the obtained models were evaluated and compared using metrics such as accuracy, MAE, MAE 2, RMSE, RMSE 2 for performance dataset based models, whereas accuracy, precision, recall were used for survey dataset based models.

To perform the above-stated task we have written software in Java which operates on the top the Weka API (Application Interface). This code is customized to compute MAE 2 and RSME 2 which is not a part of Weka. We used 10-fold cross-validation to evaluate different models with respect to different evaluation metrics.

5.5.1 Predicting Student Grade I

In this experiment we extract 5 subset datasets of dataset o1: dataset-a1o1, dataset-a12o1, dataset-a123o1, dataset-a1234o1 and dataset-a12345o1. We are classifying students in this experiment into 12 classes: A, A-, B+, B, B-, C+, C, C-, D+, D, D- and F. In the experiment we used Naïve Bayes, Classification via Logistic Regression, Random Forests and Multi-Layered Neural Networks as models. We will run four of our selected algorithms on five of our extracted datasets and the results can be found in tables 5.13, 5.14, 5.15, 5.16 and 5.17.

Notation	schema
Dataset-o1	Id, attendance, homework, class project, exam, quiz, grade I
Dataset-a1o1	Id, attendance, grade I
Dataset-a12o1	Id, attendance, homework, grade I
Dataset-a123o1	Id, attendance, homework, class project, grade I
Dataset-a1234o1	Id, attendance, homework, class project, exam, grade I
Dataset-a12345o1	Id, attendance, homework, class project, exam, quiz, grade I

Table 5.11: Review of various dataset nomenclature used and the schema they represent

5.5.2 Predicting Student Grade II

In this experiment, from COSC1430 (FALL 2015) we extract student-performance dataset from which we extract dataset-o2. This dataset-o2 has five attributes: attendance, homework, class project, exam, and quiz. Therefore, we extract 5 other datasets: dataset-a1o2 (see chapter 5.1.3), dataset-a12o2, dataset-a123o2, dataset-a1234o2 and dataset-a12345o2. We are classifying students in these datasets into 5 classes (A, B, C, D, and F).

It is a multi-class classification problem. Therefore, from our bag of algorithms, we use Naïve Bayes, Classification via Logistic Regression, Random Forests, and Multi-Layered Neural Network. We will run four of our selected algorithms on five of our extracted datasets and compare results.

5.5.3 Predicting Student Grade Point

In experiment 3, from COSC1430 (FALL 2015) we extract student-performance dataset from which we extract dataset-o3. This dataset-o3 has five attributes: attendance, homework, class project, exam, and quiz. Therefore, we extract 5 other datasets: dataset-a1o3 (see chapter 5.1.3), dataset-a12o3, dataset-a123o3, dataset-a1234o3 and dataset-a12345o3. This is a regression problem as we are predicting a grade point which ranges from 0 to 4 and can take decimal values. Therefore, from our bag of algorithms, we use Linear Regression and Support Vector Regression popularly called SMOReg. We will run both of our selected algorithms on five of our extracted datasets and compare results.

5.5.4 Detecting Poor-Performing Students

In experiment 4, we use student survey dataset and then we convert the target class column which records the final letter grade into either “pass” or “fail” using the following logic: All grades from ‘A’ through ‘D-‘ are replaced with “pass” and F are replaced with “fail”; that is, we are dealing with a binary classification problem. In the experiment, we used Naïve Bayes, Classification via Logistic Regression, Random Forests, and Multi-Layered Neural Network to obtain models.

5.6 Experimental Results & Model-Performance Evaluation

In this section, we will present and discuss results from our experimentation. The goal of this experimentation is to determine best suitable and credible models for each of our problems so, they can be applied in real-world scenarios. See table 5.12 gives notations used in summaries of experimental results.

Notation	***Meaning***	***Comment***
a1	*Attendance*	*Attribute/Feature*
a2	*Homework*	*Attribute/Feature*
a3	*Class project*	*Attribute/Feature*
a4	*Exam*	*Attribute/Feature*
a5	*Quiz*	*Attribute/Feature*
o1	*Grade I*	*Target Class*
o2	*Grade II*	*Target Class*
o3	*Grade Point*	*Target Class*
MAE	*Mean Absolute Error*	*Denotes the number of grades, the prediction deviates from actual*
MAE2	*Modified Mean Absolute Error*	*Denotes a numeric value, the prediction deviates from actual*
RMSE	*Root Mean Squared Error*	*Denotes the number of grades, the prediction deviates from actual*
RMSE2	*Modified Root Mean Squared Error*	*Denotes a numeric value, the prediction deviates from actual*

Table 5.12: Notations used in our experiments II

5.6.1 Discussion of Results: Predicting Student Grade I

We have computed Accuracy, MAE, MAE 2, RMSE, RMSE 2 for the four classification algorithms over the five datasets that use 1, 2, 3, 4, and 5 attributes, respectively. Also, we have a color coding applied which visualizes the performance of each algorithm for a dataset: In each column, the cell colored in green is the best result and the one in red is the worst result.

Naïve Bayes, for the single attribute dataset, has an accuracy of 16.2% which is least among all four classifiers, whereas logistic regression model is the "best" method with an accuracy of 22.1%. The reason for such low accuracy is that the dataset has only 104 instances to learn models that need to distinguish between 12 classes. Additionally, if there is only one attribute to learn from then we can not expect good results. But, as we go from left to right in the table 5.13, accuracy increases for all algorithms. When attribute "a2" (see 5.12) is added there is a leap in accuracy for Naïve Bayes and logistic regression classification. This kind of leap is again seen when attribute "a4" is added. It can be inferred that attribute "a2" and "a4" could positively affect the accuracy of predicting student grades. Looking at the green strip, it can be observed that logistic regression has consistently outperformed other three algorithms. Random forests have lower accuracy in most cases. For multi-layer Neural Networks, accuracy steadily increases with the number of attributes involved. The highest accuracy achieved in 57.14% by logistic regression classification in five attribute category. Table 5.13 shows the accuracy in predicting grade I.

ACCURACY					
o1	**dataset-a1o1**	**dataset-a12o1**	**dataset-a123o1**	**dataset-a1234o1**	**dataset-a12345o1**
Naïve Bayes	16.20	31.42	32.85	43.57	46.42
Logistic Regression Classification	22.14	31.42	32.14	55.71	57.14
Random Forests	21.43	24.28	20.71	47.14	46.42
Multi-Layer Neural Network	21.43	27.85	28.57	52.14	52.14

Table 5.13: Accuracy for predicting grade I for dataset-o1

MAE, MAE 2, RMSE and RMSE 2 tell how close the model's predictions are to the actual values. MAE and MAE 2 have similar insights to accuracy. For one attribute datasets, the predictions are 3 grades off, for two and three attribute datasets the predictions are 2 grades

off and for more than 3 attributes the predictions are one grade off. In general, MAE gives us a better understanding of how far off our predictions are. In a worst-case scenario a student who actually receives a 'B' grade, the model might predict an 'A' or a 'C+', In the best case scenario, the model will predict a 'B+' or 'B-' or even 'B'.

MAE					
o1	**dataset-a1o1**	**dataset-a12o1**	**dataset-a123o1**	**dataset-a1234o1**	**dataset-a12345o1**
Naïve Bayes	3.87	2.52	2.61	1.79	1.64
Logistic Regression Classification	3.36	2.54	2.44	1.32	1.23
Random Forests	3.59	2.67	3.20	1.56	1.48
Multi-Layer Neural Network	3.45	2.61	2.72	1.38	1.41

Table 5.14: MAE for predicting grade I for dataset-o1

MAE 2					
o1	**dataset-a1o1**	**dataset-a12o1**	**dataset-a123o1**	**dataset-a1234o1**	**dataset-a12345o1**
Naïve Bayes	1.06	0.57	0.54	0.31	0.28
Logistic Regression Classification	0.90	0.58	0.56	0.21	0.20
Random Forests	1.01	0.65	0.70	0.28	0.28
Multi-Layer Neural Network	1.03	0.63	0.66	0.24	0.23

Table 5.15: MAE 2 for predicting grade I for dataset-o1

MAE 2, is nothing but MAE computed after converting all the grades into grade points. Grade points range from 0 to 4. The number represented by MAE 2 is the closeness to the actual grade point achieved by the student. Naïve Bayes worst is 1.06 with only one attribute and the best is 0.20 with Logistic Regression Classification involving all

attributes. See table 5.14 for MAE comparison and 5.15 for MAE 2 comparison. The worse RMSE is 1.37 for (Naïve Bayes, one attribute) and the best is 0.34 (Logistic Regression Classification, all attributes). Table 5.16 and 5.17 shows RMSE and RMSE 2 respectively for predicting grade I.

RMSE					
o1	**dataset-a1o1**	**dataset-a12o1**	**dataset-a123o1**	**dataset-a1234o1**	**dataset-a12345o1**
Naïve Bayes	4.90	3.75	4.05	3.17	3.15
Logistic Regression Classification	4.52	3.88	3.74	2.70	2.59
Random Forests	4.71	3.96	4.52	2.90	2.85
Multi-Layer Neural Network	4.56	3.86	4.02	2.75	2.80

Table 5.16: RMSE for predicting grade I for dataset-o1

RMSE 2					
o1	**dataset-a1o1**	**dataset-a12o1**	**dataset-a123o1**	**dataset-a1234o1**	**dataset-a12345o1**
Naïve Bayes	1.34	0.78	0.75	0.46	0.43
Logistic Regression Classification	1.18	0.80	0.76	0.34	0.34
Random Forests	1.33	0.86	0.91	0.42	0.42
Multi-Layer Neural Network	1.37	0.85	0.92	0.38	0.37

Table 5.17: RMSE 2 for predicting grade I for dataset-o1

5.6.2 Discussion of Results: Predicting Student Grade II

Hoping to improve performance and allow algorithms to learn more reliable models we have reduced the number of classes to be predicted from 12 to 5. Tables 5.18, 5.19, 5.20, 5.21 and 5.22 shows accuracy, MAE, MAE 2, RMSE and RMSE 2 for predicting grade II. Accuracy improved by a factor of 2 for the two best-performing algorithms which are Naïve Bayes and logistic regression classification. Logistic regression classification learns a better model than other classifiers with 78.57% accuracy when using all five attributes and 38.57% when only using a single attribute. Naïve Bayes underperformed in comparison when attribute "a4" and "a5" is added. Random forests consistently underperformed throughout. Multilayer Neural Network had its best results when attribute "a2" and "a4" was added.

ACCURACY					
o2	**dataset-a1o2**	**dataset-a12o2**	**dataset-a123o2**	**dataset-a1234o2**	**dataset-a12345o2**
Naïve Bayes	38.57	47.14	47.14	70.71	75.00
Logistic Regression Classification	38.57	47.86	49.29	77.14	78.57
Random Forests	35.00	43.57	37.86	71.43	76.42
Multi-Layer Neural Network	36.42	49.28	44.29	76.43	75.71

Table 5.18: Accuracy for predicting grade II for dataset-o2

MAE and MAE 2 shows us that logistic regression classification performs the best job in predicting the actual results. The best MAE is observed for logistic regression classification when five attributes are considered which is 0.45. When only one attribute is considered to build model there is a tie between Naïve Bayes and multi-layer Neural Network, both having MAE of 1.33. In that case, if we look at our MAE 2 to break the tie, Naïve Bayes

has 0.9 and Neural Network has 0.96 for MAE 2. Table 5.19 and 5.20 shows MAE and MAE 2 for predicting grade II.

MAE					
o2	**dataset-a1o2**	**dataset-a12o2**	**dataset-a123o2**	**dataset-a1234o2**	**dataset-a12345o2**
Naïve Bayes	1.33	1.09	1.12	0.68	0.53
Logistic Regression Classification	1.34	1.04	1.06	0.48	0.45
Random Forests	1.92	1.14	1.36	0.58	0.53
Multi-Layer Neural Network	1.33	1.02	1.17	0.49	0.56

Table 5.19: MAE for predicting grade II for dataset-o2

MAE 2					
o2	**dataset-a1o2**	**dataset-a12o2**	**dataset-a123o2**	**dataset-a1234o2**	**dataset-a12345o2**
Naïve Bayes	0.90	0.67	0.66	0.29	0.25
Logistic Regression Classification	0.89	0.65	0.21	0.24	0.21
Random Forests	0.96	0.71	0.80	0.29	0.24
Multi-Layer Neural Network	0.96	0.64	0.71	0.24	0.25

Table 5.20: MAE 2 for predicting grade II for dataset-o2

For a dataset with one attribute, Neural Network works best according to RMSE and logistic regression classification works best according to RMSE 2. To determine the best algorithm we need to consider the difference between RMSE and RMSE 2. The difference between logistic regression and Neural Network errors in RMSE is 0.04 and for RMSE 2 is -0.08 which means if we choose the wrong algorithm our grades might be off by 0.04 which very insignificant change, in the case of RMSE 2 even though -0.08 which is also

insignificant but, when the grade points are converted back to grades they may be off by 1 grade which is significant. See table 5.21 for RMSE comparison and 5.22 for RMSE 2 comparison.

RMSE					
o2	**dataset-a1o2**	**dataset-a12o2**	**dataset-a123o2**	**dataset-a1234o2**	**dataset-a12345o2**
Naïve Bayes	1.89	1.67	1.73	1.40	1.15
Logistic Regression Classification	1.91	1.62	1.67	1.10	1.04
Random Forests	1.92	1.74	1.96	1.17	1.19
Multi-Layer Neural Network	1.87	1.62	1.78	1.11	1.21

Table 5.21: RMSE for predicting grade II for dataset-o2

RMSE2					
o2	**dataset-a1o2**	**dataset-a12o2**	**dataset-a123o2**	**dataset-a1234o2**	**dataset-a12345o2**
Naïve Bayes	1.26	0.98	0.96	0.53	0.49
Logistic Regression Classification	1.23	0.93	0.91	0.49	0.46
Random Forests	1.33	1.00	1.10	0.53	0.50
Multi-Layer Neural Network	1.31	0.92	1.02	0.47	0.53

Table 5.22: RMSE2 for predicting grade II for dataset-o2

In summary, reducing the number of classes from 12 to 5 has shown a significant increase in accuracy, MAE, MAE 2, RMSE and RMSE 2. Logistic regression classification significantly outperformed the other approaches in almost all experiments. Multi-layer Neural Networks performed second best, whereas Naïve Bayes performs moderately but better than random forests. It is also observed that the exam attribute (a4), is essential to predict student's grade.

5.6.3 Discussion of Results: Predicting Student Grade Point

In this experiment, we predict student grade point using linear regression and SMO regression on 5 datasets. The metrics used to compare the two algorithms were: accuracy, MAE2, and RMSE2.

ACCURACY					
o3	**dataset-a1o3**	**dataset-a12o3**	**dataset-a123o3**	**dataset-a1234o3**	**dataset-a12345o3**
Linear Regression	35.46	41.13	41.13	69.50	69.50
SMO Regression	34.75	41.13	42.55	73.75	73.04

Table 5.23: Accuracy for predicting grade point for dataset-o3

MAE2					
o3	**dataset-a1o3**	**dataset-a12o3**	**dataset-a123o3**	**dataset-a1234o3**	**dataset-a12345o3**
Linear Regression	0.89	0.61	0.62	0.30	0.30
SMO Regression	0.95	0.63	0.64	0.29	0.28

Table 5.24: MAE2 for predicting grade point for dataset-o3

RMSE2					
o3	**dataset-a1o3**	**dataset-a12o3**	**dataset-a123o3**	**dataset-a1234o3**	**dataset-a12345o3**
Linear Regression	0.96	0.65	0.65	0.30	0.31
SMO Regression	0.93	0.66	0.67	0.29	0.29

Table 5.25: RMSE2 for predicting grade point for dataset-o3

In terms of accuracy, SMO regression is consistently performing better than linear regression. In terms of MAE 2, SMO regression was better than linear regression until the attribute “a4” was added, but even after that, there is not much difference between two algorithms. While RMSE 2 also has the similar result. Overall we consider SMO regression to be a better approach in this experiment.

5.6.4 Discussion of Results 4: Detecting Poor-performing Student

This section will present performance metrics for experiment 4 in which we used the following machine learning algorithms: naïve Bayes, logistic regression classification, random forests and multi-layer Neural Network on student survey dataset that was obtained for the teaching of COSC1430 in Fall 2015. The metrics we used to compare the four approach are: accuracy, precision, recall and F-1 score; Table 5.26 summarizes the results of this experiment.

Survey Data	***Accuracy***	***Precision***	***Recall***	***F-1 Score***
Naïve Bayes	59.22	0.57	0.59	0.58
Logistic Regression	55.34	0.57	0.55	0.56
Random Forests	66.02	0.62	0.66	0.58
Multi-Layered Perceptron	52.42	0.53	0.52	0.53

Table 5.26: Metric comparison for detecting poor-performing student using student survey dataset of COSC1430 (FALL 2015)

	RANDOM FOREST	***Predicted*** TRUE	***Predicted*** FALSE
Actual	TRUE	64	4
Actual	FALSE	31	4

	NAÏVE BAYES	***Predicted*** TRUE	***Predicted*** FALSE
Actual	TRUE	51	17
Actual	FALSE	25	10

	LOGISTIC REGRESSION	Predicted TRUE	Predicted FALSE
Actual	TRUE	43	25
Actual	FALSE	21	14

	MULTI-LAYER NEURAL NETWORK	Predicted TRUE	Predicted FALSE
Actual	TRUE	43	25
Actual	FALSE	24	11

Table 5.27: Comparison of confusion matrices across various algorithms for COSC1430 (FALL 2015) survey dataset

In this experiment, Random forests outperform the other algorithms significantly in terms of accuracy, precision, and recall although the F-1 score is similar to that of Naïve Bayes and Logistic Regression. But our goal is not to predict accurately but to precisely detect poor-performing students. Confusion matrices for logistic regression show that it was abe to detect 14 students of 35 students who actually failed. Although the accuracy, in this case, is low but on a positive note, this algorithm will be abe to detect those 14 students accurately. Random forest algorithm will be able to learn strong models to classify students and the logistic regression model is better in identifying poor-performing students. We believe better models can be built when we use larger datasets to make predictions, also a good feature selection algorithm will be able to determine worthy features in the dataset and only use them to make predictions. Therefore logistic regression can be used as an early warning system in our real-time student-performance evaluation and feedback system.

6. Conclusion and Future Work

The overall goal of this research is to design and implement a system which provides instructors and student with valuable feedback with the goal to reduce the dropout rate. The Real-time student performance evaluation and feedback system is a tool which collects both student-performance data and personal data, builds models which can predict students grades in future from their current performance in the first or second week of the course and identify poor-performing students very early in the course. This tool will help instructors to identify poor-performing students very early in the course which will give them enough time to take necessary action such as mentoring the student or conducting review sessions. Moreover, students can use this tool to view their predicted grades and act without waiting for the instructor to intervene. This research has demonstrated that, by using classification models that are learned from past data, we can predict student grades and identify poor-performing students efficiently and very early in the course. Moreover, RSPEF provides a framework for an academic environment that motivates students and helps them to assess themselves better.

In this research, four classification models Naïve Bayes, Logistic Regression, Random Forests, and Multi-Layered Neural Networks were used, compared, and evaluated for student grade prediction and identification of poor-performing students. A benchmark of real-world datasets was collected from a computer science course at the University of Houston to assess and compare the classification models. The performance of these machine learning algorithms was evaluated based on different performance measures like

accuracy, precision, recall, MAE, RSME, MAE 2 and RMSE 2. The results show that Logistic Regression Classifier performed better when compared to other algorithms in predicting student grade and Random Forest performed better for identifying poor-performing students.

For future work, we aim to carry out more experiments using larger datasets consisting of thousand records of student-performance and survey data. Also, the approach to identify failing students accomplished quite a low accuracy; a larger survey dataset and using a feature selection algorithm will improve the accuracy. Also, approaches will be investigated to track student-performance and provide computer generated feedbacks to students using the web-application. Additionally, the achieved accuracy and MAE of the implemented classification models in RSPEF can be improved. We will explore different approaches like genetic algorithms, Bayesian belief networks to obtain higher accuracy. Finally, we are planning to build and deploy this system during the first week of fall 2017 and evaluate RSPEF system usability.

References

[1] ACT Institutional Data File, 2005.

[2] X. Chen, (2013). STEM Attrition: College Students' Paths Into and Out of STEM Fields (NCES 2014-001). National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education. Washington, DC, 2013.

[3] R. Levitz. 2016 National Freshman Motivation to Complete College Report. Cedar Rapids, Iowa. Retrieved from http://www.ruffaloNL.com.

[4] E. Seymour and N. M. Hewitt. Talking about leaving: Why undergraduates leave the sciences. Boulder, CO: Westview Press. 1997.

[5] B. L. Lowell, H. Salzman, H. Bernstein, and E. Henderson. Steady as she goes? Three generations of students through the science and engineering pipeline. Paper presented at the Annual Meeting of the Association for Public Policy Analysis and Management, Washington, DC., November 7, 2009.

[6] Han and Kamber. Data Mining: concepts and techniques. Elsevier, (2011).

[7] Thalheim, B. (2013). Entity-Relationship Modeling: Foundations of Database Technology. Springer Berlin Heidelberg.

[8] Shanmuga Priya and Senthil Kumar. Improving the Student's Performance Using Educational Data Mining. International Journal of Advanced Networking and Applications 4.4 (2013): 1806.

[9] Ben-Zadok and Galit. Examining online learning processes based on log files analysis: A Case Study. 5th International Conference on Multimedia and ICT in Education (2007).

[10] Goyal, Vohra R. Applications of Data Mining in Higher Education, International Journal of Computer Science Issues, Vol.9, Issue 2, No1 (2012).

[11] Surjeet Kumar, Brijesh Bharadwaj, and Saurabh Pal. Data mining applications: A comparative study for predicting student's performance (2012).

[12] Ashraful Alam Pathan, Mehedi Hasan, Ferdous Ahmed, and Dewan Farid. A Mining Model for Developing Students' Programming Skills. 8th International Conference on IEEE (2014): 1-5.

[13] Minaei, Bidgoli B. Predicting student performance: an application of data mining methods with an educational web-based system. In 33rd ASEE/IEEE Frontiers in Education Conference (2003): 1-6.

[14] Pittman K. Comparison of data mining techniques used to predict student retention. Ph.D. Thesis, Nova Southeastern University (2008).

[15] Romero C. Data mining algorithms to classify students. In 1st International Educational Data Mining Conference (2008): 8-17.

[16] Nguyen, Paul, and Peter H. A Comparative Analysis of Techniques for Predicting Academic Performance. In Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference (2007): 7-12.

[17] Al-Radaideh, Al-Shawakfa, and AI-Najjar M. Mining Student Data using Decision Trees. In Proceedings of the International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan (2006).

[18] Muslihah, Yuhanim, Norshahriah, Mohd Rizal, Fatimah, and Hoo Y. S. Predicting NDUM Student's Academic Performance Using Data Mining Techniques. In Proceedings of the Second International Conference on Computer and Electrical Engineering, IEEE Computer Society (2009).

[19] Ramaswami, Bhaskaran. CHAID Based Performance Prediction Model in Educational Data Mining. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1 (2010): 212-217.

[20] Mishra, Kumar, and Gupta S. Mining Students Data for Prediction Performance. In Advanced Computing & Communication Technologies (ACCT), Fourth International Conference on IEEE (2014): 255-262.

[21] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer and W. F. Punch, "Predicting student performance: an application of data mining methods with an educational Web-based system," 33rd Annual Frontiers in Education, 2003. FIE 2003., Westminster, CO, 2003, pp. T2A-13.

[22] Narasimha Murty, M. and Susheela Devi, V. (2011). Pattern Recognition: An Algorithmic Approach. ISBN 0857294946.

[23] Raschka, S. (2015). Python Machine Learning. Packt Publishing.

[24] le Cessie, S. & van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics, 41, 191-201.

[25] Breiman Leo. Random forests. Machine learning 45.1 (2001): 5-32.

[26] Chan, Jonathan Cheung-Wai, and Desiré Paelinckx. Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotype mapping using airborne hyperspectral imagery. Remote Sensing of Environment 112.6 (2008): 2999-3011.

[27] Sayad, S. (2011). Real-time data mining. Canada: Self-Help Publishers.

[28] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, K.R.K. Murthy: Improvements to the SMO Algorithm for SVM Regression. In: IEEE Transactions on Neural Networks, 1999.

[29] A.J. Smola, B. Schoelkopf (1998). A tutorial on support vector regression.