STUDENT ANALYTICS FOR COURSE RECOMMENDATION

BY

RAMYA DHANI SRINATHA

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF MASSACHUSETTS LOWELL
SPRING 2013

Signature of Author: R D.S. Date: 04/30/2013

Signature of Thesis Advisor: Fred Martin

Name: Fred Martin

Signature of Thesis Co-Advisor: Karen M. Daniels

Name: Karen Daniels

UNIVERSITY OF MASSACHUSETTS LOWELL

ABSTRACT

Student Analytics for Course Recommendation

by Ramya Dhani Srinatha

Thesis Committee: Professor Fred Martin, Department of Computer Science

Professor Karen Daniels, Department of Computer Science

Academic advisors often develop anecdotal guidelines about how each student's past performance relates to their performance in later courses in a specified major. While these guidelines can be useful, a more formal statistical analysis of these relationships can help predict student's performance in later courses, which can help professors guide their students to focus on potential areas of success. In addition, such analyses can identify the courses which are key indicators of later performance in the major. This additional insight into relationships between the courses in the curriculum can help develop a recommender system for automatic student advising.

ACKNOWLEDGMENTS

This dissertation would not have been possible without the guidance and the help of several individuals who have contributed and extended their valuable assistance in the preparation and completion of this study.

I would like to express my sincere gratitude to my thesis advisor Prof. Fred Martin who offered his continuous advice and encouragement throughout the course of this thesis work.

I acknowledge my gratitude to Prof. Karen Daniels for her encouragement and support to make this thesis possible.

I would like to thank Charlotte Mandell, the Vice Provost for Undergraduate Education and the Provost's Office at UMass Lowell for supporting me in beginning this research work in Summer 2012.

I would also like to thank Julie Alig, Director of Institutional Research, for her assistance in providing the data used in this study.

Finally, I take this opportunity to thank my family for their continuous support and inspiration.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Recommender systems have been used to process vast amounts of information successfully in many production systems (Venetis *et al.* 2011). The most popular domains that use recommender systems include movies, books and other entertainment media (Salter *et al.* 2006). The same techniques can be used to recommend courses in an academic setting. There are many related articles in the area of academic performance analysis, particularly in the realm of better advising systems. Much of the work involves the construction of a systematic model which considers only the prescribed curriculum structure (Sharma *et al.* 2003). There has also been a great deal of research in the literature, though most of the literature involved predicting success based on secondary education data and entrance exam scores, rather than current coursework performance (D'Agostino *et al.* 2009; Donnelly *et al.* 2010).

In an academic setting, usually students are presented with a host of courses to choose from as they plan their subsequent semesters. Often, having many choices is a good thing, but it does make it hard for a student to wade through and read all of the information on each course. Universities usually employ academic counselors; people who are tasked with helping students make their choice. But in practice the counselors are often overloaded with too many students and not enough time, and in some scenarios, the counselor may not be specialized within a major to advise students (Young *et al.* 2011).

The motivation of this dissertation is to explore various machine learning techniques for data analysis and implement an efficient recommender system for academic advising that could enhance the students' success rate in a chosen major.

To build better recommender systems, implementation of an efficient prediction model is necessary. One such model was developed using Regression Analysis technique (developed in language R, used specifically for machine learning). The input to the model constituted of students' grades in their past courses and the output was their predicted grades in the courses they would pursue in future.

Preparation:

Prior to beginning this work, it required training on research ethics education known as CITI Course Program. Upon successful completion of this program, a request was submitted to Institutional Research Board to approve this research as it involved human subjects. Following the approval from IRB (see Appendix A), the development of the model was initiated and had the following phases:

Collection of research data:

Data sets for this research were requested twice from the university. The initial data set included all freshmen students who started their undergraduate study in the university as computer science majors even though some would have left the major subsequently. The data set was from past 6 years, totaling to 330 students. After analysis of the model, it was determined that there were insufficient numbers for analytical purposes. For example, there were only 40 students in Operating Systems (91.308), which is one of the upper division courses in the CS program. Therefore a second set was requested which would satisfy the following conditions:

For a given semester (over the range of the last six years):

For any student who is enrolled in a core CS course (e.g., 91.101, 102, 201, 203, 204, 301, 304, 305, 308, 404): Data on all courses and grades for specified student for that semester.

The second set not only had freshmen students but also had transfer students and students who pursued at least a preliminary CS course irrespective of their major. There were a total of 645 students in this set.

Data De-identification:

The original data set was de-identified using a script written in R. The script chose the entire set of original student IDs, replaced them with random numbers from a specified range and finally returned the de-identified data file along with the mapping file. The mapping file included one-one mapping for original and replaced student IDs and this file was subsequently stored in a secure location.

Data division:

During machine learning one often needs to divide the data set into training and testing datasets. To accomplish this, initially the students' data was sorted according to the number of CS courses they have taken in increasing order. After re-organizing the data set, it was divided into training set constituting the first 80 percent of this sorted set and the other 20 percent was labeled as test set for certain number of iterations to implement cross-validation procedure.

These training and test sets ensured that all the records of a student is included either in training or test set and both training and test sets involved representative number of students at each stage of completion of CS degree program. Then the supervised prediction model was built using the training set and the test set was used to test the prediction model.

Data Preprocessing:

The analysis only included CS courses, Natural Science electives, Math courses and selected ethics courses (see Appendix B). The student data from the training set was classified according to the courses taken and their enrollment terms. Statistics on overall courses taken, enrollment term codes, and repetition of courses by students were extracted from the data set. Further, the data set was subdivided into different data frames, each corresponding to information on the set of students in each course which included their student ID enrollment term code, number of attempts and the grade obtained. The grades were represented in a numerical form which ranged from 0-4.

Chapter 2

METHODOLOGY

Predictive analytics encompasses a variety of techniques from statistical modeling, machine learning, and data mining that analyze current and historical facts to make predictions about future, or otherwise unknown events.

Determining good predictors is an essential part of development of prediction models. The initial assumption was to explore the impact of performance of one course on the other. This assumption was further analyzed by fitting models involving linear dependence. The predictors were deduced based on similarity between the performances of students in all course pairs. The similarity measure refers to a class of statistical relationships involving dependence.

The similarity is often measured in terms of correlation coefficients. There are many correlation coefficients and the most common of these is the Pearson Correlation which is sensitive to linear relationship between variables. Thus, the similarity measure chosen was Pearson Correlation as this is a good fit to linear models (Correlation and Dependence 2013).

The correlation value ranges from -1 to 1 where '0' indicates no correlation, '1' indicates strong correlation and '-1' indicates negative correlation. Thus the correlation between two courses was formulated using the Eq (1).

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

X, Y *Indicates performances in Course X and Course Y*

$cov(X,Y)$ *Indicates Covariance between two courses*

$\sigma_X \sigma_Y$ *Variance in two courses*

μ_X, μ_Y *Mean value in two courses*

Eq. 1 Pearson Correlation Coefficient

Therefore the Pearson correlation was used to detect how the performance of any course pair varies with each other over same set of students. Correlation matrix with common enrollment of at least 100 is as shown in Figure 1.

From_Course	To_Course	Correlation	No_of_Common_Students
16.265	42.22	0.2290321	103
16.265	91.304	0.5519394	107
16.265	91.301	0.5225482	114
16.265	91.305	0.5215177	119
16.265	92.386	0.3736424	124
16.265	91.102	0.5769994	126
16.265	92.322	0.497169	129
16.265	91.201	0.571813	137
16.265	91.203	0.6323061	138
16.265	92.321	0.5783268	139
16.265	91.204	0.4957861	143
42.101	42.102	0.4397904	161
91.101	16.265	0.5683726	106
91.101	91.203	0.6230991	114
91.101	91.201	0.5922722	120
91.101	92.321	0.6174955	127
91.101	92.132	0.6148526	141
91.101	92.131	0.6805934	147
91.101	42.102	0.3347729	183
91.101	91.102	0.7032127	209
91.101	42.101	0.4844432	227
91.102	91.204	0.3392686	115
91.102	91.203	0.6382345	141
91.102	91.201	0.5615547	145
91.102	42.101	0.4413346	148
91.102	42.102	0.4250738	166

Fig 1. Correlation matrix

The correlation matrix lists the course pairs, corresponding number of common students between them, and their correlation value. For example, the number of students between Computing I (91.101) and Calculus I (92.131) are 147 students and the correlation value is 0.6805. This value of correlation indicates positive correlation and can be inspected using scatterplot in Figure 2(a). This suggests that students, who do better in Computing I, mostly tend to do better in Calculus I as most of the students in Calculus I with score above 3.0 have a score of minimum 2.7 to score above 3.8.

Whereas the pair Calculus I (92.131) and Calculus II (92.132) has a correlation value of 0.57 and scatterplot in Figure 2(b) indicates positive correlation as the students score 3.0 or more in 92.131 score 3.0 and above in 92.132. But the pair, Probability & Statistics (92.386) and Logic Design (16.265) does have a correlation of 0.42 but the performances in the both courses do not reveal much information as it is evident from the scatterplot shown in Figure 2(c) that the performance in logic design is not predictable only using Probability and Statistics.

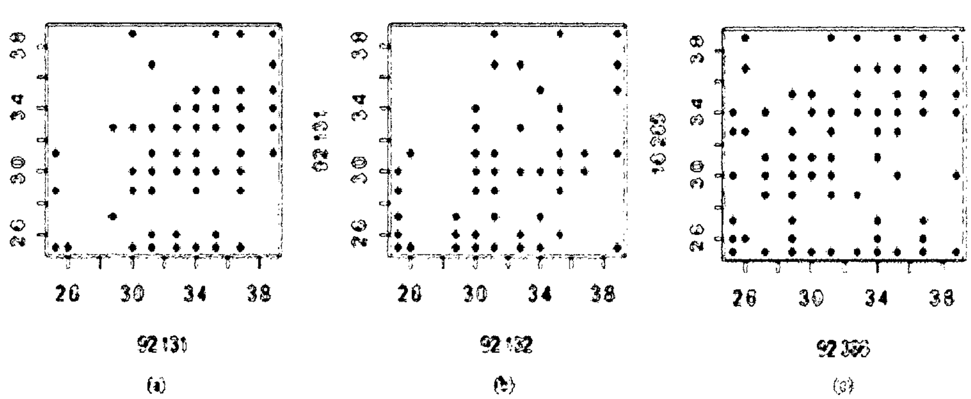

Fig 2. Scatterplots of various course pairs. Here 91.101 – Computing I, 92.321 – Discrete Structures I, 92.131 & 92.132 – Calculus I & II, 92.386 – Probability and Statistics I.

Once the correlation data on distinct course pair combinations were derived, the next step was to evaluate their statistical significance. One of the methods chosen was to implement the T-test (T-test 2013). The T-test is the most commonly used statistical data analysis procedure for hypothesis testing. This test finds a numerical metric based on the number of samples used in correlation and the variance in the sample mean. Thus the numerical metric was formulated using the Eq (2).

$$t = \frac{(x' - \mu)}{s/sqrt\ (n)}$$

Eq. 2 T-test equation

Where x' indicates the sample mean, μ is the population mean, s is the standard deviation of the sample, n is the sample size and degrees of freedom are equal to n-1.

Corresponding cumulative probability is found for the hypothesis such that, the probability that "the T-score is less than expected mean of the population" by chance (12). If this probability is less than 0.05, then our hypothesis fails and the correlation and T-score are significant. Thus the correlation of each course pair with significance was chosen based on the T-score and cumulative probability. This was obtained by "corr package" in R. Example of corr-test is as shown in the Figure 3 for random data x_test and y_test.

```
data:  x_test and y_test
t = 8.822, df = 229, p-value = 4.441e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4006063 0.5940972
sample estimates:
     cor
0.503641
```

Fig 3. Example for corr-test in R

Prediction model depends on the order of courses taken as they tend to be the predictors of the future courses. So, the analysis on the order of courses taken by the students was carried out.

For every course pair, three possible orders were formed. They were: Course 1 taken before Course 2, Course 2 taken before Course 1, and Course1 and Course2 taken simultaneously. Each course pair was given a score based on the number of students who took them in the above mentioned orders. Their corresponding correlation value and significance were also derived. Thus, all the analysis data on each course pair was consolidated into a matrix for each order as shown below in Figure 4.

For example, consider the course pair 91.101 (Computing I) and Calculus I (92.131). It is evident that the course pair has 87 students taking the courses simultaneously (Figure 4). But the number of students taking Computing I before Calculus I are 37 (Figure 6). It is difficult to make a decision on the most probable order as the order between courses is more predictive. This tie was resolved using their status on significance. In this example, the "courses taken simultaneously" is the most probable order for the course pair.

Course 1	Course 2	C1=C2 N	C1=C2 R	C1=C2 tcalc	C1=C2 Status	C1 PASS	C2 PASS
92.131	91.101	87	0.59291886303007	6.78839376317358	Significant	79	85
16.265	91.101	1		0	NA	0	1
16.265	92.131	1		0	NA	0	0
92.132	91.101	21	0.655138302919751	3.77931171517668	Significant	16	19
92.132	92.131	0	0	0	0	0	0
92.132	16.265	7	0.791673536220459	2.89754019585095	Significant	3	4
92.321	91.101	19	0.867975399254824	7.20640837752077	Significant	17	16
92.321	92.131	2	1		NA	2	1
92.321	16.265	41	0.66744404787043	5.59743643541214	Significant	37	36
92.321	92.132	38	0.647837679572978	5.10256794261094	Significant	26	21
92.386	91.101	5	0.961986456084279	6.10115983483941	Significant	4	4
92.386	92.131	0	0	0	0	0	0
92.386	16.265	24	0.254458955707229	1.2341418339431	Not Significant	21	19
92.386	92.132	2	-1	0	Not Significant	1	1

Fig 4. Course pair analysis data for simultaneous order

The first two columns of the matrix indicate course pair, followed by number of students, correlation value for simultaneous order, T-score, Significance status and last two columns indicate the no of students who successfully completed the course pair

Consider the example 92.132 (Calculus II) and 16.265 (Logic Design). This course pair has 7 students taking them simultaneously (Figure 4), 52 of them taking them in Calculus II before Logic Design order (Figure 5) and 14 students taking them in Logic Design before Calculus II order (Figure 6). The interesting feature about the course pair is that all the correlation values are statistically significant.

Deciding the most probable order was difficult. Therefore, to solve this, a normative order was followed. If 75% or more students took a course pair in a particular order, we termed it as normative order and only this order was used for prediction as this is based on the correlation metric. Further, if the course pairs were suffering a tie even at this level, the pass percentage of students for that order was used as a metric to resolve it. Thus the most probable order was found for each course pair for efficient prediction.

Course 1	Course 2	C1>C2 N	C1>C2 R	C1>C2 tcalc	C1>C2 Status	C1 PASS	C2 PASS
92.131	91.101	1		0	NA	0	0
16.265	91.101	0	0	0	0	0	0
16.265	92.131	1		0	NA	0	0
92.132	91.101	1		0	NA	0	0
92.132	92.131	0	0	0	0	0	0
92.132	16.265	52	0.421215554987017	3.28398376531742	Significant	42	43
92.321	91.101	2	-1		NA	1	1
92.321	92.131	1		0	NA	0	1
92.321	16.265	63	0.36944724791951	3.10516002629247	Significant	52	47
92.321	92.132	14	0.163013179312867	0.572350033872904	Not Significant	8	7
92.386	91.101	1		0	NA	0	1
92.386	92.131	0	0	0	0	0	0
92.386	16.265	22	0.58498396426247	3.22562654104123	Significant	18	15
92.386	92.132	0	0	0	0	0	0

Fig 5. Course pair analysis data for Coursel before Course2 order

The first two columns of the matrix indicate course pair, followed by number of students, correlation value for "Coursel before Course2" order, T-score, Significance status and last two columns indicate the no of students who successfully completed the course pair

Course_1	Course_2	C2>C1_N	C2>C1_R	C2>C1_tcalc	C2>C1_Status	C1 PASS	C2 PASS
92.131	91.101	37	0.163579169467201	0.980960752721857	Not Significant	18	33
16.265	91.101	88	0.535321511623086	5.87743682683345	Significant	65	83
16.265	92.131	48	0.674656667893669	6.19908920925342	Significant	35	41
92.132	91.101	105	0.56371421391781	6.92649258970642	Significant	65	102
92.132	92.131	84	0.571330910829044	6.30376096450697	Significant	54	77
92.132	16.265	14	0.787950379487982	4.43295182228283	Significant	9	8
92.321	91.101	103	0.534676504262045	6.35866395342538	Significant	77	99
92.321	92.131	66	0.370618112011207	3.19228217321968	Significant	48	56
92.321	16.265	16	0.479246103382724	2.04308355043002	Significant	14	6
92.321	92.132	49	0.283343026154147	2.02551010305129	Significant	41	43
92.386	91.101	73	0.141149043604071	1.20137067576514	Not Significant	63	69
92.386	92.131	39	0.084507169664382	0.5158824230425	Not Significant	33	34
92.386	16.265	56	0.440967807296003	3.61042407762648	Significant	43	42
92.386	92.132	59	0.443621253575	3.73712713940447	Significant	48	44

Fig 6. Course pair analysis data for Course2 before Course1 order

The first two columns of the matrix indicate course pair, followed by number of students, correlation value for "Course1 before Course2" order, T-score, Significance status and last two columns indicate the no of students who successfully completed the course pair

This normative order was only used for prediction model as this was based on the correlation metric. A separate order analysis was carried out to rate each course pair order and deduce the most probable order for recommender system. This order diagram used for recommender system is as shown in Figure 7.

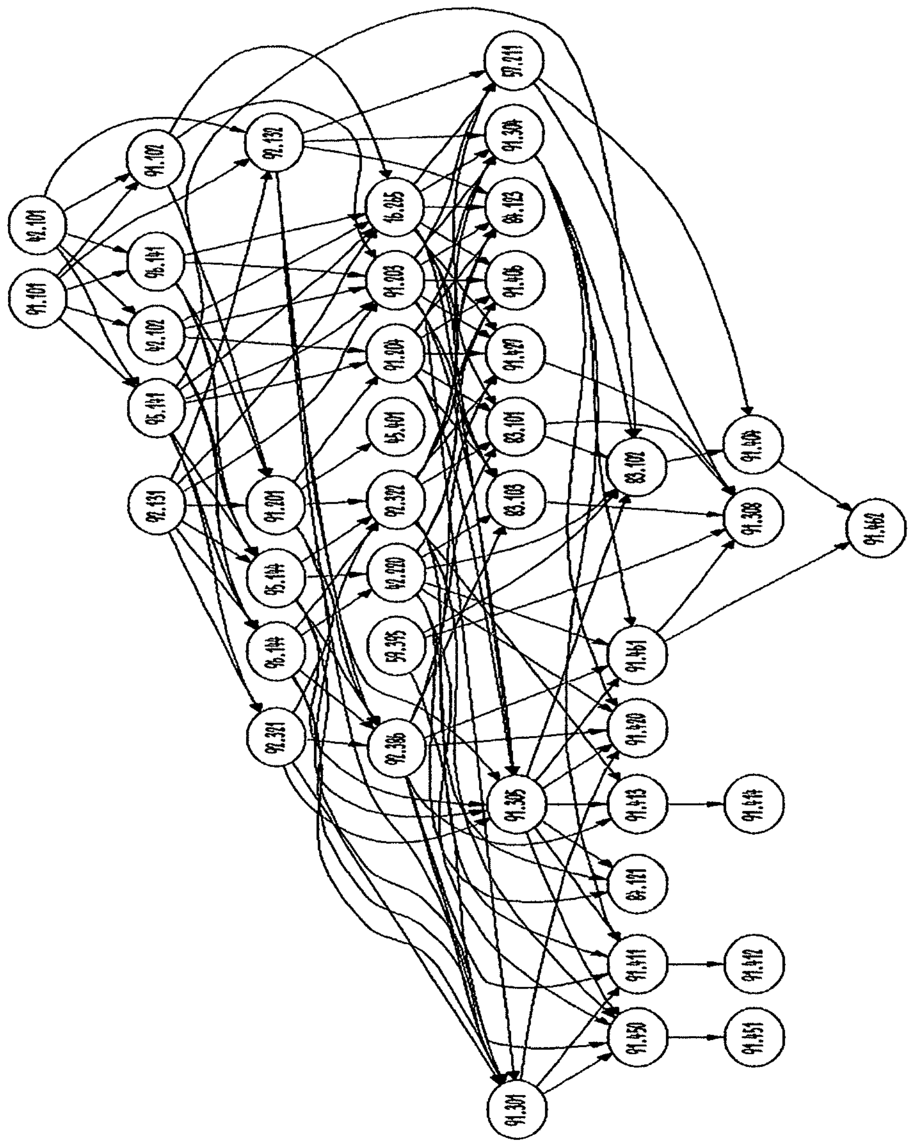

Fig 7: Order Diagram for recommender system

The order diagram could be compared to the published curriculum model of CS degree program for consistency (see Appendix C). As this order adhered with the pre-requisites as prescribed by the university, this was further used in recommendation model which is explained in Chapter 4. This order included both the required CS courses and also some of the prescribed elective courses which were termed as "Soft Pre-requisites". These included General ethics courses, Natural Science electives and Math courses.

The next step was to process the normative order of courses which included removing of duplicates and applying transitivity relations and order selection based on correlative significance. Thus the final list of courses along with their predictors is as shown in Figure 8(a), 8(b), 8(c), 8(d), 8(e) and 8(f).

PREDICTORS	PREDICTED COURSE
College Writing I, Calculus I	Computing I (91.101)
Computing I, College Writing I	Calculus I (92.131)
Calculus I, Computing I, Physics I Lab, Physics I, College Writing II Computing IV, Computer Org, Computing III, Computing II, Discrete Structures I Calculus II	Logic Design (16.265)
Calculus I, Computing I, Physics I Lab, Physics I, College Writing II College Writing I. Computing II	Calculus II (92.132)
Calculus II, Computing I, Physics II, Physics I Lab, Physics I College Writing II, Computer Org, Computing III, Computing II	Discrete Structures I (92.321)

Fig 8(a). List of courses and their predictors

PREDICTORS	PREDICTED COURSE
Discrete Structures I, Calculus II, Logic Design, Computing I, College Writing II Computing IV, Computer Org, Computing III, Computing II, Discrete Structures II	Probability & Stats (92.386)
Discrete Structures I, Calculus II, Logic Design, Calculus I, Computing I Physics II Lab, Physics II, Physics I Lab, Physics I, College Writing II Computing IV, Computer Org, Computing III, Computing II	Discrete Structures II (92.322)
Discrete Structures II, Prob & Stats, Discrete Structures I, Logic Design, Calculus I Computing I, Physics II, Life Science I, SW Eng II, Org Prog Lang Operating Systems, Computing IV, Computer Arch, Computer Org, Computing III Foundations, Computing II	Algorithms (91.404)
Calculus I, Computing I, Physics I Lab, College Writing II, College Writing I	Computing II (91.102)
Computing II, Discrete Structures II, Prob & Stats, Discrete Structures I, Logic Design, Calculus I, Computing I, Life Science I, Org Prog Lang, Computing IV, Computer Arch, Computer Org, Computing III	Foundations (91.304)

Fig 8(b). List of courses and their predictors

PREDICTORS	PREDICTED COURSE
Computing II, Discrete Structures I, Calculus II, Computing I, Physics II Lab, Physics II, Physics I Lab, Physics I, College Writing II, College Writing I, Computer Org	Computing III (91.201)
Computing III, Computing II, Discrete Structures I, Calculus II, Calculus I, Computing I, Physics II Lab, Physics II, Physics I Lab, Physics I, College Writing II, College Writing I	Computer Org (91.203)
Computer Org, Computing III, Computing II, Discrete Structures II, Discrete Structures I, Calculus II, Logic Design, Calculus I, Computing I, Physics I Lab, Physics I, Life Science I, College Writing II, Org Prog Lang, Computing IV	Computer Architecture (91.305)
Computer Org, Computing III, Computing II, Discrete Structures II, Prob & Stats, Discrete Structures I, Calculus II, Logic Design, Calculus I, Physics II, Physics I Lab, Physics I, College Writing II	Computing IV (91.204)

Fig 8(c). List of courses and their predictors

PREDICTORS	PREDICTED COURSE
Computing IV, Computer Arch, Computer Org, Foundations, Computing II, Algorithms, Discrete Structures II, Logic Design, Physics II, Life Science II, Life Science I, College Writing I, Sustainable Development, Org Prog Lang	Operating Systems (91.308)
Foundations	GUI Prog I (91.461)
GUI Prog I, Discrete Structures II	GUI Prog II (91.462)
Computing IV, Computer Arch, Computer Org, Computing III, Computing II, Discrete Structures II, Prob & Stats, Discrete Structures I, Calculus II, Logic Design, Calculus I, Computing I, Physics II Lab, Life Science II, College Writing II, Computers in Society, Sustainable Development	Org Prog Lang (91.301)
Computing IV, Discrete Structures II, Discrete Structures I, Logic Design	Data Communication I (91.413)
Data Comm I	Data Communication II (91.414)
College Writing II, College Writing I, Computing II, Calculus II, Calculus I, Computing I, Physics I Lab	Physics I (95.141)
Physics I, College Writing II, College Writing I, Computing III, Computing II, Discrete Structures I, Calculus I, Computing I, Physics II Lab	Physics II (95.144)

Fig 8(d). List of courses and their predictors

PREDICTORS	PREDICTED COURSE
Computing IV, Computing II	Compiler Construction (91.406)
Computing IV, Computer Arch, Computing III, Algorithms, Discrete Structures II	SW Eng II (91.412)
Computing IV, Computing III, Computing II, Discrete Structures II, Discrete Structures I, Logic Design, Computing I, College Writing II	Sustainable Development (57.211)
Calculus I, Computing I	College Writing I (42.101)
College Writing I, Computing II, Calculus II, Calculus I, Computing I	College Writing II (42.102)
Org Prog Lang, Computing IV, Computer Arch, Computer Org, Computing III, Foundations, Computing II, Discrete Structures II, Prob & Stats, Discrete Structures I, Calculus II, Logic Design, Calculus I, Computing I, Life Science I Lab	Life Science I (83.101)
Life Science I, Org Prog Lang, Computing IV, Computer Arch, Computer Org, Computing II, Discrete Structures II, Prob & Stats, Discrete Structures I, Calculus II, Logic Design, Calculus I, Computing I, Physics I Lab, Physics I	Life Science I Laboratory (83.103)

Fig 8(e). List of courses and their predictors

PREDICTORS	PREDICTED COURSE
Life Science I, Org Prog Lang, Operating Systems, Computing IV, Computer Arch, Computing III, Foundations, Prob & Stats, Logic Design	Life Science II (83.102)
College Writing I, Computing III, Computing II, Calculus II, Logic Design, Computing I, Physics I Lab, Physics I, Chemistry I Lab	Chemistry I (84.121)
Chemistry I, College Writing I, Computer Org, Calculus I, Computing I	Chemistry I Laboratory (84.123)
Physics I, College Writing II, Computing II, Calculus II, Calculus I, Computing I	Physics II (96.141)
Physics II, College Writing II, College Writing I, Computing III, Computing II, Computing I	Physics II Lab (96.144)

Fig 8(f). List of courses and their predictors

Apart from the above mentioned list, the other courses do not have prediction model as they have no correlated courses. Their predictions are always replaced by the average score in that particular course as they have no predictor courses. The list is shown in Figure 9(a) and 9(b).

Course	Average score (No Predictors found)
Computer & Network Security I (91.561)	3.425
Graphics I (91.427)	2.877272727
Robotics I (91.450)	3.888888889
Graphics II (91.428)	3.3
Robotics II (91.451)	3.314285714
Database I (91.309)	2.833333333
Database II (91.310)	3.283333333
SW Eng I (91.411)	3.485185185
Artificial Intelligence (91.420)	2.486363636
Machine Learning (91.421)	3.175
Computers in Society (59.395)	2.765
Engineering and Ethics (45.334)	2.516666667
Bioethics and Genetic Research (45.401)	2.228571429
Oral & Written Communication for CS (42.220)	3.28
Principles of Biology I (81.111)	1.82
Experimental Biology I (81.117)	2.381818182

Fig 9(a). List of courses for which no predictors were found

Course	Average score (No Predictors found)
Principles of Biology II (81.112)	1.1625
Experimental Biology II (81.118)	1.575
Life Science II Lab (83.104)	2.683333333
Chemistry II (84.122)	1.625
Chemistry II Laboratory (84.124)	3.68
Earth and Environmental Systems I (87.201)	2.7
Earth And Environmental Systems Lab (87.203)	2.777777778
Earth And Environmental Systems II (87.202)	0.666666667
Hydrogeology (89.314)	1.675

Fig 9(b). List of courses for which no predictors were found

Thus these predictors were used to fit a prediction model to predict the future course performances.

Building Predictive models:

Linear regression models were used as approximations of the functional relationship between a predicted course value and a predictor course variable (Linear Regression 2013). The simple linear regression model with one predictor variable β was formulated using the Eq (3).

$$\boldsymbol{y_i = x_i \beta + \varepsilon_i}$$

y_i *Indicates observed responses of predicted course with index i*

x_i *Indicates observations for predictor course with index i*

β *Indicates an unknown parameter*

ε_i *Indicates the error between actual values and predictions with index i*

Eq 3. Single predictor - Linear regression model

This model was used to build prediction models for single predictor scenarios. In most of the cases of predicted courses, there were multiple predictors for each of them. Therefore, multi- variable regression model for these courses were formulated using Eq (4).

$$\boldsymbol{y_i = X\beta' + \varepsilon_i'}$$

y_i *Indicates observed responses of a predicted course with index i*

X *Indicates vector of observations for predictor courses*

β' *Indicates vector of unknown parameters*

ε_i'*Indicates the error between actual values and predictions with index i*

Eq 4. Multi-predictor - Linear regression model

Apart from linear regression models, an alternative approach - "Quadratic regression model" was formulated to explore non-linear predictive modeling technique for fitting the data. The formulation for the same is as shown in Eq (5).

$$y_i = X\alpha + X^2\beta'' + \varepsilon_i''$$

y_i *Indicates observed responses of a predicted course with index i*

X *Indicates vector of observations for predictor courses*

α, β'' *Indicates vectors of unknown parameters*

ε_i'' *Indicates the error between actual values and predictions with index i*

Eq 5. Multi-predictor - Quadratic regression model

The demonstration of these models with a specific example can be found in the next chapter.

Multicollinearity:

Multicollinearity is a statistical phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, *i.e.*, one can be linearly predicted from the others with a non-trivial degree of accuracy. In this situation the coefficient estimates may change erratically in response to small changes in the model or the data. Multicollinearity does not reduce the predictive power or reliability of the model as a whole, at least within the sample data themselves; it only affects calculations regarding individual predictors.

In the case of predicting student's future performance, we encounter a case of multi-regression in most of the scenarios. Therefore, the detection of multicollinearity issue was incorporated in all the prediction models. This was detected using "Variance Influence Factor" (VIF). It was formulated using the Eq (6).

$$tolerance = 1 - R^2$$
$$VIF = 1/tolerance$$

R *Indicates the correlation between predictors and predicted course*

Eq 6. Variance Influence Factor

A tolerance of less than 0.20 or 0.10 and/or a VIF of 2 or 5 and above indicates a multicollinearity problem (O'Brien *et al.* 2007). Once this issue was detected with a predictor course, that course was removed from the list of predictor courses and thus the final list of predictors for the model was derived.

Missing Data Imputation:

In machine learning, missing data, or missing values, occur when no data value is stored for the variable in the current observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.

It is quite common scenario in the case of prediction, as it is clear that every student cannot take all predictor courses before we derive a prediction for a particular course. This missing data problem was solved using imputation technique using package "Multi Iterated Chained Equations" using mice package in R (Buuren *et al.* 2006). This method regresses over predictors to impute appropriate values into data.

Thus, the prediction models were carefully built to encounter both multicollinearity and missing data problems.

Chapter 3

DEMONSTRATION OF PREDICTION MODEL DEVELOPMENT

All modeling approaches in R use the same basic structure of "Predicted variable ~ Predictor variables". This part of the dissertation demonstrates the development of linear regression model and quadratic regression with an example.

Consider the course "91.102 (Computing II)" which is one of the core CS courses usually taken in the first freshman year. From the list of predictors (Figure 8) from previous chapter, the predictors for Computing II are 92.131 (Calculus I), 91.101 (Computing I), 96.141 (Physics I Lab), 42.101 (College Writing I) and 42.102 (College Writing II). The following table (Figure 10) illustrates the correlation scores between 91.102 and all predictors of 91.102.

Predicted Course	Predictor	Correlation	No. of Students
91.102	42.101	0.368449	150
91.102	42.102	0.453128	167
91.102	96.141	0.625595	89
91.102	91.101	0.65812	211
91.102	92.131	0.509052	117

Fig 10. Correlation scores between 91.102 and its predictors

Once the predictors for a course was found, the data of the predictors and predicted course was merged into a single matrix for further analysis. A snapshot for the course "91.102" is as shown in Figure 11.

Gpa 91.102	Gpa 92.131	Gpa 91.101	Gpa 96.141	Gpa 42.102	Gpa 42.101
0	0	1	0	NA	1.7
2.7	2	3	2.3	3.3	3.3
3	NA	2.7	3	1.7	1.7
3	NA	2.7	0	1.7	1.7
0	0	2.3	NA	3.7	2.7
2.7	0	2.3	NA	3.7	2.7
0	1.7	2.3	NA	3.7	2.7
2.7	1.7	2.3	NA	3.7	2.7
2.3	NA	3.3	NA	3	3
4	NA	2.7	NA	NA	NA
0	2.7	3.3	NA	4	3
0	1	3.3	NA	4	3
1	NA	2.7	3	3.3	NA
0	NA	2.7	3	3.3	NA
3.7	NA	3.7	3	3.3	NA
0	2	0	NA	NA	NA
0	2	2	NA	NA	NA
2.7	0	3	NA	NA	3.3
2.7	0	3	NA	NA	0
0	3.3	3	NA	3	2.7
2	NA	3.3	NA	NA	NA
2.3	NA	2.7	2.7	4	2.7
4	4	4	3	4	3.7
3.7	NA	4	4	3.7	3.7
1.7	0	2	NA	2	1
3.3	3.3	NA	NA	2.3	4

Fig 11. Snapshot of merged data of predictors and data of 91.102

After merging the data, some of the records had missing data in them. The regression analysis usually cannot proceed until the missing values are imputed. Therefore, the missing data imputation was accomplished using "Mice" method in R. The snapshot of completed data set for 91.102 is as shown in Figure 12.

Gpa 91.102	Gpa 92.131	Gpa 91.101	Gpa 96.141	Gpa 42.102	Gpa 42.101
0	0	1	0	2.3	1.7
2.7	2	3	2.3	3.3	3.3
3	2.3	2.7	3	1.7	1.7
3	1.7	2.7	0	1.7	1.7
0	0	2.3	1	3.7	2.7
2.7	0	2.3	2.7	3.7	2.7
0	1.7	2.3	0	3.7	2.7
2.7	1.7	2.3	2	3.7	2.7
2.3	1	3.3	3	3	3
4	1.7	2.7	3.7	3	3
0	2.7	3.3	0	4	3
0	1	3.3	3	4	3
1	0	2.7	3	3.3	3
0	2.3	2.7	3	3.3	3.3
3.7	0	3.7	3	3.3	3
0	2	0	0	0	2
0	2	2	0	1	1
2.7	0	3	2.3	2	3.3
2.7	0	3	3	3	0
0	3.3	3	1	3	2.7
2	3	3.3	3	2.3	2.7
2.3	0	2.7	2.7	4	2.7
4	4	4	3	4	3.7

Fig 12. Snapshot of completed data set for course 91.102

The initial formula for linear regression for 91.102 was constructed as:

"Gpa_91.102 ~ Gpa_92.131 + Gpa_91.101 + Gpa_96.141 + Gpa_42.102 + Gpa_42.101"

The "stats" package in R has a method "lm" which is generally used to fit regression models in R. Thus "lm" was used to fit a regression model for the initial formula. The model lists only the weight of each predictor in the model and is as shown in Figure 13. The summary of the model lists the weight of each predictor, t-score of each predictor in the model and their significant code denoted by their probability of significance and is as shown in Figure 14.

```
> init_model

Call:
lm(formula = eval(parse(text = temp_str)), data = prediction_dat_matrix_2)

Coefficients:
(Intercept)   Gpa_92.131   Gpa_91.101   Gpa_96.141   Gpa_42.102   Gpa_42.101
   -0.77206      0.05987      0.42369      0.39554      0.18167      0.10968
```

Fig 13. Initial Linear Regression model for 91.102

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.77206    0.22013  -3.507 0.000522 ***
Gpa_92.131   0.05987    0.04170   1.436 0.152061
Gpa_91.101   0.42369    0.07987   5.305  2.2e-07 ***
Gpa_96.141   0.39554    0.04435   8.919  < 2e-16 ***
Gpa_42.102   0.18167    0.04648   3.908 0.000115 ***
Gpa_42.101   0.10968    0.05748   1.908 0.057318 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig 14. Summary of initial linear regression model for course 91.102

After removing lesser significance predictors by solving multicollinearity issue using VIF factor, the final linear regression model for 91.102 was derived. The final regression model and its summary are as shown in Figure 15 and Figure 16.

```
> get("Final_model_L_91.102")

Call:
lm(formula = Gpa_91.102 ~ Gpa_91.101 + Gpa_96.141 + Gpa_42.102 +
    Gpa_42.101, data = prediction_dat_matrix_2)

Coefficients:
(Intercept)   Gpa_91.101   Gpa_96.141   Gpa_42.102   Gpa_42.101
    -1.1698       0.5161       0.2928       0.2174       0.2317
```

Fig 15. Final Linear Regression model for 91.102

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.16980    0.22539  -5.190 3.88e-07 ***
Gpa_91.101   0.51608    0.07962   6.482 3.72e-10 ***
Gpa_96.141   0.29279    0.04833   6.058 4.13e-09 ***
Gpa_42.102   0.21742    0.04523   4.807 2.43e-06 ***
Gpa_42.101   0.23168    0.05895   3.930 0.000105 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig 16. Summary of final regression model for course 91.102

Thus, the final linear models for every course were derived and their coefficients were used for prediction. Subsequently, the above procedure to build linear model was followed to build quadratic regression models as well.

The initial formula for "91.102" using quadratic regression was constructed as:

"Gpa_91.102 ~ Gpa_92.131 + Gpa_92.131_square + Gpa_91.101 + Gpa_91.101_square + Gpa_96.141 + Gpa_96.141_square + Gpa_42.102 + Gpa_42.102_square + Gpa_42.101 + Gpa_42.101_square"

The initial model (Figure 17), summary of initial model (Figure 18), final model (Figure 19) and summary of final model (Figure 20) using quadratic regression for the course "91.102" are as shown below.

```
> get("Init_model_91.102")

Call:
lm(formula = eval(parse(text = temp_str)), data = prediction_dat_matrix_2,
    na.action = na.omit)

Coefficients:
      (Intercept)         Gpa_92.131  Gpa_92.131_square         Gpa_91.101
         -0.06729           -0.02356            0.03125           -0.16017
Gpa_91.101_square         Gpa_96.141  Gpa_96.141_square         Gpa_42.102
          0.14859           -0.02680            0.03837            0.48871
Gpa_42.102_square         Gpa_42.101  Gpa_42.101_square
         -0.06336           -0.08751            0.05946
```

Fig 17. Initial quadratic regression model for 91.102

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.06729    0.38743  -0.174  0.86223
Gpa_92.131        -0.02356    0.13188  -0.179  0.85836
Gpa_92.131_square  0.03125    0.03650   0.856  0.39256
Gpa_91.101        -0.16017    0.24455  -0.655  0.51299
Gpa_91.101_square  0.14859    0.05221   2.846  0.00474 **
Gpa_96.141        -0.02680    0.15815  -0.169  0.86556
Gpa_96.141_square  0.03837    0.04442   0.864  0.38831
Gpa_42.102         0.48871    0.15397   3.174  0.00166 **
Gpa_42.102_square -0.06336    0.03730  -1.699  0.09042 .
Gpa_42.101        -0.08751    0.24292  -0.360  0.71894
Gpa_42.101_square  0.05946    0.05229   1.137  0.25646
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig 18. Summary of initial quadratic regression model for course 91.102

```
> get("Final_model_91.102")

Call:
lm(formula = Gpa_91.102 ~ Gpa_92.131_square + Gpa_91.101_square +
    Gpa_96.141_square + Gpa_42.102 + Gpa_42.102_square + Gpa_42.101_square,
    data = prediction_dat_matrix_2, na.action = na.omit)

Coefficients:
      (Intercept)  Gpa_92.131_square  Gpa_91.101_square  Gpa_96.141_square
         -0.34010            0.02665            0.11847            0.03136
       Gpa_42.102  Gpa_42.102_square  Gpa_42.101_square
          0.44945           -0.05533            0.04119
```

Fig 19. Final quadratic regression model for 91.102

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.34010    0.17201  -1.977  0.04893 *
Gpa_92.131_square  0.02665    0.01255   2.124  0.03451 *
Gpa_91.101_square  0.11847    0.01934   6.126 2.85e-09 ***
Gpa_96.141_square  0.03136    0.01448   2.166  0.03114 *
Gpa_42.102         0.44945    0.14468   3.106  0.00208 **
Gpa_42.102_square -0.05533    0.03567  -1.551  0.12188
Gpa_42.101_square  0.04119    0.01409   2.924  0.00372 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig 20. Summary of final quadratic regression model for course 91.102

Thus, the prediction models were built and tested for their accuracies. Finally, the model with high accuracy was further used as a part of recommendation model which is described in the next chapter.

Testing the prediction models:

There were 129 records in the test-set. The test set was tested with both linear and quadratic prediction models. The evaluation technique used was "RMSE-Root Mean Square Error" (Shani *et al.* 2010). The RMSE value for linear model was 0.2844 and 0.30 which indicated that the quadratic model did not fit the data well due to over-fitting problem.

Confidence in each prediction:

In statistical inference, specifically predictive inference, a prediction interval is an estimate of an interval in which future observations will fall, with a certain probability, given what has already been observed (Prediction Interval 2013). This was calculated as part of "lm" method in R and was formulated using the Eq 7. The prediction interval was flat distribution between the upper and lower values.

$$\textbf{\textit{Prediction interval}} = y' \pm t * \sqrt{1 + \frac{1}{n} + \frac{(x^* - x_{mean})^2}{(n-1) * s_x^2}}$$

y' *Indicates the prediction value*

x^* *Indicates the current x value using which prediction y' is made*

t *Indicates t_test score for x and y values used for prediction model*

n *Indicates the sample size used for prediction model*

x_{mean} *Indicates the mean value of x values*

s_x *Indicates the standard deviation of x values*

Eq.7 Prediction interval in R

More width of the interval for each prediction leads to lesser confidence in prediction which was further sorted to output the most confident predictions. Predictions and corresponding confidence scores for the student "64841" is as shown by Figure 21.

Course	Actual Grade	Predicted Grade	Confidence
College Writing II (42.102)	4	No_prev_courses	0
General Psychology (47.101)	4	No_prev_courses	0
Computing I (91.101)	3.7	No_prev_courses	0
Calculus I (92.131)	4	No_prev_courses	0
Turning Fiction into Film (42.232)	4	0	0
Computing II (91.102)	3.7	2.6988361	1.847329568
Honors Calculus II (92.142)	3.3	0	0
Physics I (95.141)	4	2.138734086	1.65364751
Physics I Lab (96.141)	4	2.040919597	1.834018254
Computing III (91.201)	2.7	3.072364317	2.462596707
Computer Org (91.203)	3	2.234152689	1.830213304
Calculus III (92.231)	3	0	0
Discrete Structures I (92.321)	4	2.538093419	1.992276287
Logic Design (16.265)	4	3.214625681	1.948225234
Oral & Written Communication for CS (42.220)	4	3.233802817	3.99
Computing IV (91.204)	4	2.912132559	1.938993866
Physics II (95.144)	2.7	2.595525493	1.672004686
Physics II Lab (96.144)	4	3.158156398	1.776835647
Introduction to Ethics (45.203)	4	0	0
Economics I (49.201)	3.7	0	0
Org Prog Lang (91.301)	3.3	2.919796618	1.121204793
Computer Architecture (91.305)	2.3	2.613864718	1.741022907
Discrete Structures II (92.322)	3	3.300287802	2.02146382
Operating Systems (91.308)	2.7	2.583490754	1.254358362
Database I (91.309)	4	2.833333333	3.99
Algorithms (91.404)	4	3.434320785	1.499646031
Probability and Statistics I (92.386)	4	2.601575234	2.776887598

Fig 21. Predictions and confidence scores for the student "64841"

Therefore due lesser RMSE value, the linear model was chosen as it was the best fit for prediction. Thus the predictions from the linear model were used as part of the recommendation system.

Chapter 4

RECOMMENDER SYSTEM

Recommender systems or recommendation systems are a subclass of information filtering systems that seek to predict the unknown, which they had not yet considered, using a pre-built prediction model.

In our scenario, the course recommender system was built to predict the future performance of a student based on the past data of the student and convert them into valuable recommendations for automatic advising. The course recommender is as shown by Figure 22.

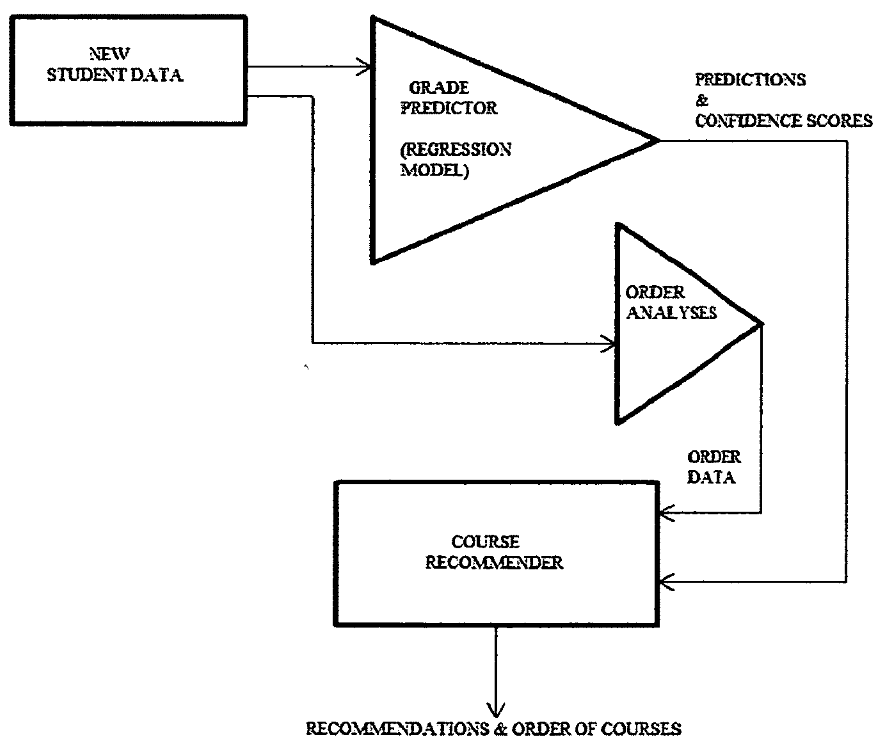

Fig 22. Course Recommender system

The input to the course recommender system was the past courses' data of the student for whom, the recommendations are computed. Then the courses for which the predictions are to be computed are decided as follows:

For every course in the past data of the student, if the course is a predictor for any "other course", then the "other course" is added to list for which predictions are to be computed. Note: All the CS courses prescribed by the university curriculum are used for predictions. They include core CS courses, supporting courses (Natural Science Electives and Math Courses) and General ethics courses.

Then, the predictions and the confidence scores for the listed courses are determined using the pre-built regression model. If one of the predictors for the course prediction is missing, then the corresponding predictor value is imputed with an average score to compute the prediction and confidence score. After which these predicted courses are sorted based on confidence scores. Other courses are listed under suggestions section, if there are no predictors in the past data.

After predictions and suggestions, the "order of courses" which was earlier computed using only the enrollment order was used to ensure that the course selection adheres to the rules of the curriculum including the pre-requisite structure. Thus, the recommendations are combination of predictions, suggestions and order of courses. Demonstration of recommendation can be found in the next chapter.

Chapter 5

RESULTS

The prediction model predicts the student's future performance and the recommender system converts them into valuable recommendations. The accuracy of prediction model was evaluated using "RMSE" value (Shani *et al.* 2010) and thus the linear model with RMSE of 0.2844 was chosen.

The prediction results for every semester are divided into four sections: Predictions for Computer Science Courses, Predictions for Supporting Courses which includes Natural Science Electives and Ethics Courses and finally an order in which courses have to be taken. This order is deduced from data patterns which includes the hard and soft pre-requisites. This pattern is followed till the last semester the student pursued courses as CS major.

Please note that some of the early courses could be the predictors for courses that the students are not yet eligible to take which is shown in list of predictors derived in Chapter 2. These courses are filtered based on the order analyses in recommendation system and thus only courses that the student is eligible to take next are displayed. This order indicates that, if the student takes the courses which are recommended then he has to follow the order displayed.

The results of the prediction model are shown by demonstrating three types of students, the first student being an average CS student because his average grade is 3.0 in all CS courses, second student being a strong CS student because his average grade is more than 3.3 and third student being a weak CS student as his average grade is less than 2.7.

1. Student with ID – 212193 – This student is an example for an average CS student. This student pursued CS courses for five semesters and obtained an average score of 3.0 in required basic CS courses.

Demonstration of results of recommender system output and the actual choice of courses for a freshman student with ID – 212193 is given as follows:

Predictions for student with ID - 212193 for semester 1:

Most probable courses in First semester are: College Writing I (42.101), Computing I (91.101), Calculus I (92.131)

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
College Writing I (42.101)	2.7	No_prev_courses	0
History of Crime and Social Control (43.308)	2.7	No_prev_courses	0
Computing I (91.101)	3	No_prev_courses	0
Calculus I (92.131)	2.3	No_prev_courses	0

Predictions for student with ID - 212193 for semester 2:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing II (91.102)	2.2180114	1.8389667
Computer Org (91.203)	2.283746	1.8075466
Computing III (91.201)	2.1703885	2.4468936

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	1.730635926	1.637331911
Physics I Lab (96.141)	2.182900166	1.825000745
Discrete Structures I (92.321)	2.409825159	1.982363315
Calculus II (92.132)	1.46700509	2.239734786
College Writing II (42.102)	2.4313532	2.407594

The order recommended is:
91.102 >>> 91.201
92.102 >>> 91.203

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
College Writing II (42.102)	3.3	2.431353161	2.407594
Intro to Philosophy (45.201)	3	0	0
Calculus II (92.132)	1	1.46700509	2.2397348

Predictions for student with ID - 212193 for semester 3:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing II (91.102)	2.3414652	1.8401276
Computing III (91.201)	2.1703885	2.4468936
Computer Org (91.203)	2.1508523	1.8098739
Computing IV (91.204)	3.1178749	1.9311341

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	1.654155	1.63837
Physics II (95.144)	1.485055	1.6480009
Physics II Lab (96.144)	2.332252	1.7521184
Physics I Lab (96.141)	2.10743	1.8279409
Discrete Structures I (92.321)	2.336553	1.983379
Sustainable Development (57.211)	2.745471	2.0879254

The order recommended is:
91.102 >>> 91.201
92.102 >>> 91.203
91.201 >>> 91.204

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Drawing – Form & Space (70.255)	3	0	0
Earth and Environmental Systems I (87.201)	3	2.7	3.99
Earth And Environmental Systems Laboratory (87.203)	3.3	2.777777778	3.99
Computing II (91.102)	3	2.341465194	1.8401276
Discrete Structures I (92.321)	3.3	2.336553431	1.983379

Predictions for student with ID - 212193 for semester 4:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing III (91.201)	2.520255059	2.450886115
Computer Org (91.203)	2.263123542	1.810876005
Computing IV (91.204)	3.143825911	1.9322503
Org Prog. Lang (91.301)	2.410925152	1.039120042

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	1.65415538	1.638369952
Physics II (95.144)	1.749762497	1.652538999
Physics II Lab (96.144)	2.552251202	1.751179992
Physics I Lab (96.141)	2.347830111	1.827952024
Discrete Structures II (92.322)	2.212291984	2.007493
Logic Design (16.265)	1.696803193	1.93719627
Probability & Statistics (92.386)	1.842848443	2.765436117
Sustainable Development (57.211)	2.74547143	2.08792541

The order recommended is:
91.201 >>> 91.204
91.201 >>> 92.386
91.201 >>> 91.301
91.203 >>> 91.301
16.265 >>> 91.301

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Computin III(91.201)	3	2.520255059	2.4508861
Comp. Org (91.203)	2.7	2.263123542	1.810876
Disc.Struct.II(92.322)	4	2.212291984	2.007493
Physics I (95.141)	2.3	1.65415538	1.63837
Physics Lab (96.141)	3	2.347830111	1.827952

Predictions for student with ID – 212193 for semester 5:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing IV (91.204)	2.971624147	1.932300821
Org Prog. Lang (91.301)	2.317329693	1.05311315
Computer Architecture (91.305)	2.253969839	1.727905744
Foundations (91.304)	2.81645035	1.739123632

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Life Science II (83.102)	3.08790352	0.898357589
Life Science I (83.101)	2.875837652	1.299779773
Life Science I Laboratory (83.103)	3.980790577	1.432208162
Physics II (95.144)	1.988230922	1.648906022
Chemistry I (84.121)	1.622378615	1.652270377
Physics II Lab (96.144)	2.691378802	1.751552367
Logic Design (16.265)	2.17166642	1.937599657
Chemistry I Laboratory (84.123)	3.020883618	2.422785943
Probability & Statistics (92.386)	1.88832784	2.765489088
Sustainable Development (57.211)	3.49332012	2.13006707

The order recommended is:
91.204 >>> 91.304
16.265 >>> 91.304
16.265 >>> 91.305
16.265 >>> 91.301
91.204 >>> 91.305

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Logic Design (16.265)	2	2.17166642	1.9375997
3D Animation I (70.376)	3.7	0	0
Life Science I (83.101)	2	2.875837652	1.2997798
Life Science I Lab (83.103)	3	3.980790577	1.4322082
Computing IV (91.204)	2	2.971624147	1.9323008
Probability and Statistics I (92.386)	1	1.88832784	2.7654891

2. Student with ID – 64841 - This student is an example for a strong CS student. This student pursued CS courses for six semesters and obtained an average score of 3.3 in required CS courses.

Demonstration of results of recommender system output and the actual choice of courses for a freshman student with ID – 64841 is given as follows:

Predictions for 64841 for semester 1:

Most probable courses in First semester are: College Writing I (42.101), Computing I (91.101), Calculus I (92.131)

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
College Writing II (42.102)	4	No_prev_courses	0
General Psychology(47.101)	4	No_prev_courses	0
Computing I (91.101)	3.7	No_prev_courses	0
Calculus I (92.131)	4	No_prev_courses	0

Predictions for student with ID - 64841 for semester 2:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing II (91.102)	2.6988361	1.847329568
Computing III (91.201)	2.736982109	2.46600193
Computing IV (91.204)	3.142104115	1.931133801
Computer Org (91.203)	1.772550668	1.830916692

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	2.138734086	1.65364751
Physics II (95.144)	1.25697649	1.692638898
Physics II Lab (96.144)	2.681534427	1.769896551
Physics I Lab (96.141)	2.040919597	1.834018254
Discrete Structures I (92.321)	2.635812405	1.989479173
Calculus II (92.132)	2.053398444	2.255288451
Sustainable Development (57.211)	2.92935928	2.0989427

The order recommended is:
91.102 >>> 91.201
91.102 >>> 91.203
91.201 >>> 91.204
92.132 >>> 91.204

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Turning Fiction to Film (42.232)	4	0	0
Computing II (91.102)	3.7	2.6988361	1.84732957
Honors Calc. II (92.142)	3.3	0	0
Physics I (95.141)	4	2.138734086	1.65364751
Physics I Lab (96.141)	4	2.040919597	1.83401825

Predictions for student with ID - 64841 for semester 3:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing III (91.201)	3.072364317	2.462596707
Computer Org (91.203)	2.234152689	1.830213304
Computing IV (91.204)	2.410359349	1.936863937

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics II (95.144)	2.284275908	1.668144936
Chemistry I (84.121)	2.200555175	1.687531033
Physics II Lab (96.144)	3.073340642	1.779835389
Discrete Structures I (92.321)	2.538093419	1.992276287
Calculus II (92.132)	2.986570205	2.263373139
Logic Design (16.265)	2.726577753	1.947726219
Chemistry I Laboratory (84.123)	3.130006951	2.439575301
Sustainable Development (57.211)	2.92935928	2.0989427

The order recommended is:
91.201 >>> 91.204
92.132 >>> 91.204

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Computing III (91.201)	2.7	3.072	2.46259671
Computer Org (91.203)	3	2.234	1.8302133
Calculus III (92.231)	3	0	0
Discrete Structures I (92.321)	4	2.538	1.99227629

Predictions for student with ID - 64841 for semester 4:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing IV (91.204)	2.912132559	1.938993866
Org Prog Lang (91.301)	2.510908381	1.061746829
Computer Architecture (91.305)	2.507246009	1.73790818
Foundations (91.304)	3.033018599	1.749756191

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Life Science II (83.102)	3.08790352	0.898357589
Physics II (95.144)	2.595525493	1.672004686
Chemistry I (84.121)	2.200555175	1.687531033
Physics II Lab (96.144)	3.158156398	1.776835647
Discrete Structures II (92.322)	2.90954598	2.020985404
Calculus II (92.132)	2.986570205	2.263373139
Logic Design (16.265)	3.214625681	1.948225234
Chemistry I Lab (84.123)	3.130006951	2.439575301
Probability & Statistics (92.386)	2.021400255	2.770749293
Sustainable Development (57.211)	3.13420158	2.1003176

The order recommended is:
92.132 >>> 91.204
16.265 >>> 91.301
92.132 >>> 92.386
92.132 >>> 92.322
91.204 >>> 91.304
16.265 >>> 91.304
92.322 >>> 91.304
91.204 >>> 91.305
16.265 >>> 91.305

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Logic Design (16.265)	4	3.215	1.94822523
Oral & Written Comm. in CS (42.220)	4	3.234	3.99
Computing IV (91.204)	4	2.912	1.93899387
Physics II (95.144)	2.7	2.596	1.67200469
Physics II Lab (96.144)	4	3.158	1.77683565

Predictions for student with ID - 64841 for semester 5:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Org Prog Lang (91.301)	2.9197966	1.1212048
Computer Architecture (91.305)	2.6138647	1.7410229
Foundations (91.304)	3.1655217	1.7490043

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Life Science II (83.102)	3.657497915	0.939327558
Chemistry I (84.121)	2.401112501	1.691136366
Discrete Structures II (92.322)	3.300287802	2.02146382
Calculus II (92.132)	2.986570205	2.263373139
Chemistry I Lab (84.123)	3.130006951	2.439575301
Probability & Statistics (92.386)	2.601575234	2.776887598
Sustainable Development (57.211)	3.46296264	2.14371821

The order recommended is:
92.322 >>> 91.304
92.132 >>> 91.305
92.132 >>> 91.301
92.132 >>> 92.386
92.132 >>> 92.322

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Intro to Ethics (45.203)	4	0	0
Econo. I (49.201)	3.7	0	0
Org Prog. Lang (91.301)	3.3	2.92	1.12120479
Computer Arch. (91.305)	2.3	2.614	1.74102291
Discrete Struct. II (92.322)	3	3.3	2.02146382

Predictions for student with ID - 64841 for semester 6:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Operating Systems (91.308)	2.583490754	1.254358362
Foundations (91.304)	3.220957849	1.747144134
Data Comm. I (91.413)	3.774926869	1.346976259
Compiler Construction (91.406)	3.424287408	2.452927399

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Life Science II (83.102)	3.657497915	0.939327558
Chemistry I (84.121)	2.401112501	1.691136366
Calculus II (92.132)	2.986570205	2.263373139
Chemistry I Lab (84.123)	3.130006951	2.439575301
Probability & Statistics (92.386)	2.601575234	2.776887598
Sustainable Development (57.211)	3.46296264	2.14371821

The order recommended is:
92.132 >>> 91.304
92.386 >>> 91.308
92.132 >>> 92.386
92.386 >>> 91.413

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Operating Systems (91.308)	2.7	2.583	1.25435836
Database I (91.309)	4	2.833	3.99
Algorithms (91.404)	4	3.434	1.49964603
Probability and Statistics I (92.386)	4	2.602	2.7768876

3. Student with ID - 141217 – This student is an example for a weak CS student. This student pursued CS courses for three semesters and obtained an average score of less than 3.0 in required CS courses.

Demonstration of results of recommender system output and the actual choice of courses for a freshman student with ID – 141217 is given as follows:

Predictions for student with ID - 141217 for semester 1:

Most probable courses in First semester are: College Writing I (42.101), Computing I (91.101), Calculus I (92.131)

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
College Writing I (42.101)	2.7	No_prev_courses	0
The Modern World (43.106)	2.3	No_prev_courses	0
Computing I (91.101)	2.3	No_prev_courses	0
Preparation for Calculus (92.127)	2.3	No_prev_courses	0

Predictions for student with ID-141217 for semester 2:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing II (91.102)	1.884826	1.8413609

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	1.4582935	1.6367118
Physics I Lab (96.141)	2.1829002	1.8250007
Calculus II (92.132)	1.316295	2.2431845
College Writing II (42.102)	2.4313532	2.407594

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
College Writing II (42.102)	2.7	2.431353161	2.407593975
United States History to1877 (43.111)	1	0	0
Computing II (91.102)	2	1.884826038	1.841360931
Calculus I (92.131)	0	1.45625599	2.405763126

Predictions for student with ID-141217 for semester 3:

Predictions for Computer Science Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Computing III (91.201)	1.9503871	2.4514156
Computer Org (91.203)	2.5193027	1.8168259
Computing IV (91.204)	3.1633439	1.9315146

Predictions for Supporting Courses:		
Predicted courses	**Predicted Grades**	**Confidence**
Physics I (95.141)	1.230154439	1.64257712
Physics II (95.144)	1.702625874	1.66401588
Physics II Lab (96.144)	2.007427616	1.74770387
Physics I Lab (96.141)	2.136639293	1.82567704
Discrete Structures I (92.321)	2.183837912	1.98357938
Calculus II (92.132)	1.347699491	2.24408867
Logic Design (16.265)	1.341337087	1.9388315
Sustainable Development (57.211)	2.5878533	2.0847746

The order recommended is:
91.201 >>> 91.204
92.132 >>> 91.204

Student's actual choice:			
Course	**Actual Grade**	**Predicted Grade**	**Confidence**
Life Science I (83.101)	2.3	2.69439713	1.299536544
Computing III (91.201)	0	1.950387112	2.451415627
Calculus I (92.131)	0	1.45625599	2.405763126

The important aspect here is that the order of courses is derived based on predictions. The order of suggestions could also be displayed based on needs. Thus, the recommender system could be a valuable addition to advising system to better address the needs of the student based on his past academic performances.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The predictive models and the correlation coefficients built here are expected to change, as curriculum changes are implemented, and as the dataset grows over time as well. Given the dynamic nature of the curriculum, it would be unreasonable to expect a static set of predictive models and other statistical relationships using the approach described above. Instead, we envision an evolution of these relationships as curriculum changes occur and the size of the dataset increases. After initial analyses, these models could be updated on semester basis and then used in the recommender system.

The recommender model could also include a more efficient filter to remove lower level courses when already an upper level course in the same subject has been taken. Using this recommender system, advisors can help focus student effort on right choice of courses and improve student's success.

The prediction model used here yielded an accuracy of 72%, it is possible that more robust non-linear models could be fit to get better results. Imputation error could be reduced by implementing clustering technique (Oyelade *et al.* 2010) for imputing data based on similar students rather than regressing over the entire data set.

It is also possible to build more sophisticated model using not only previous course data but also a complete assessment of each course which could provide more insightful details. To gain these additional benefits, we suggest an ongoing data collection strategy where course assessment data for each course is collected and archived every semester. This approach would ensure that complete data is available for each student, providing the opportunity to include assessment predictor variables in the predictive models rather than limiting the predictor variables to previous course grades.

It would also be interesting to determine whether or not it is possible to predict probable success or failure in the major based on performance in the early courses in the major. These predictions could simply predict "Yes" or "No" for graduation with a computer science degree. These models could also use SAT scores and high school scores to tackle cold start problem (Cold Start Problem 2013) rather than using most probable courses, as this could prove a potential research to cross major advising.

REFERENCES

[1] Linear Regression, URL = http://en.wikipedia.org/wiki/Linear_regression; 2013

[2] Correlation and dependence, URL = http://en.wikipedia.org/wiki/Correlation_and_dependence; 2013

[3] Prediction Interval, URL= http://en.wikipedia.org/wiki/Prediction_interval; 2013

[4] Parameswaran, Aditya; Venetis, Petros and Garcia-Molina, Hector (2011) "Recommendation systems with complex constraints: A course recommendation perspective", *ACM Transactions on Information Systems (TOIS)*, v.29 n.4, p.1-33.

[5] Siegfried, R. M.; Wittenstein, A. M. & Sharma, T. (2003), "An automated advising system for course selection and scheduling", J. Comput. Small Coll. 18(3), 17-25.

[6] D'Agostino, J. V. and Bonner,S. M. (2009) "High School Exit Exam Scores and University Performance", Educational Assessment, 14(1):25--37.

[7] Donnelly, P. J. (2010) "Examining Pre-College Academic Variables: Investigating Future College Success", PhD thesis.

[8] Oyelade, O.J.; Oladipupo, O.O. & Obagbuwa, I.C. (2010), "Application of k means clustering algorithm for prediction of students' academic performance", International Journal of Computer Science and Information Security (IJCSIS), pp. 292–295.

[9] Salter, J. and Antonopoulos, N. (2006) "Cinema Screen Recommender Agent: Combining Collaborative and Content-Based Filtering", Intelligent Systems Magazine, vol. 21, issue 1, pages 35-41.

[10] Young, J. R. (2011). The Netflix effect: When software suggests students' courses

[11] Cold Start Problem, URL = http://en.wikipedia.org/wiki/Cold_start; 2013

[12] T-test, URL = http://en.wikipedia.org/wiki/Student's_t-test; 2013

[13] Shani,G. and Gunawardana, A.(2010), " Evaluating recommendation systems", Recommender Systems Handbook, pages 257--297.

[14] Buuren ,Van S., Oudshoorn CGM (2006) "MICE: Multivariate Imputation by Chained Equations. R package version 1.16"

APPENDIX A

UMASS
LOWELL

IRB Administrator
Phone 978.934.4134
Fax 978.934.6012

2nd Floor Wannalancit
Lowell, MA 01854
Email: IRB@uml.edu

Office of Institutional Compliance

June 1, 2012

Dr. Fred Martin
Department of Computer Science

IRB Protocol #: 12-069 **Study Status:** Expedited

Protocol Title: Analyzing Historical Student Course Records for More Effective Advising

Approval Period: 6/1/2012 to 5/31/2013

This letter is to notify you that the above referenced Protocol has been reviewed and approved by the UMass Lowell IRB. A copy of the signed Informed Consent Form is attached if applicable.

Please notify the IRB Office of any change in your research protocol, unexpected adverse events, and also, if you decide not to continue or postpone your research project.

If your project exceeds one year in duration, an Annual/Continuing Review Form must be submitted to the IRB Office. Please allow sufficient time so that your project may be reviewed and receive IRB approval prior to the anniversary date of the research project. Upon completion of this project, it is your responsibility to notify the IRB, using the Final Report form, available on the website. Our best wishes for continued success with your research!

For up-to-date forms and information go to www.uml.edu/Research/OIC/

APPENDIX B

List of CS courses and Natural Electives

Course Number	Course Name
91.101	Computing I
92.131	Calculus I
16.265	Logic Design
92.132	Calculus II
92.321	Discrete Structures I
92.386	Probability and Statistics I
92.322	Discrete Structures II
91.404	Analysis of Algorithms
91.102	Computing II
91.304	Foundations of Computer Science
91.561	Computer & Network Security I
91.201	Computing III
91.203	Computer Org
91.305	Computer Architecture
91.427	Computer Graphics I
91.204	Computing IV
91.308	Operating Systems
91.450	Robotics I
91.461	Graphical User Interface Programming I
91.462	Graphical User Interface Programming II
91.428	Computer Graphics II
91.451	Robotics II
91.301	Organization of Programming Languages
91.309	Database I
91.413	Data Communications I
91.414	Data Communications II
91.310	Database II
91.411	Software Engineering I
91.406	Compiler Construction I
91.420	Artificial Intelligence
91.421	Data Mining
91.412	Software Engineering II
57.211	Sustainable Development
59.395	Computers in Society
45.335	Ethical Issues in Technology

45.341	Science, Ethics, and Society
45.334	Engineering and Ethics
45.401	Bioethics and Genetic Research
42.101	College Writing I
42.102	College Writing II
42.220	Oral & Written Communication for Computer Science
81.111	Principles of Biology I
81.117	Experimental Biology I
81.112	Principles of Biology II
81.118	Experimental Biology II
83.101	Life Science I
83.103	Life Science I Laboratory
83.102	Life Science II
83.104	Life Science II Laboratory
84.121	Chemistry I
84.123	Chemistry I Laboratory
84.122	Chemistry II
84.124	Chemistry II Laboratory
87.201	Earth and Environmental Systems I
87.203	Earth And Environmental Systems Laboratory
87.202	Earth And Environmental Systems II
87.204	Earth And Environmental Systems Laboratory
89.215	Forensic Geology
89.314	Hydrogeology
89.315	Environmental Geochemistry
95.141	Physics I
96.141	Physics I Lab
95.144	Physics II
96.144	Physics II Lab

APPENDIX C

Computer Science
Course Prerequisite Chart

Courses with bold outlines are required for all CS majors.

Courses grouped within dashed outlines are required for CS minors.

Updated April 18, 2012