Personalized Recommendation System for Students by Grade Prediction

M.Tech Dissertation Report

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

by

Pavan Kotha
Roll No: 113050010

under the guidance of

Prof. Deepak B Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai
Declaration

I declare that this written submission represents my ideas in my own words and where other’s ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Pavan Kotha
(113050010)

Place: IIT Bombay, Mumbai
Date: 27\textsuperscript{th} June, 2013
Abstract

Education today is influenced by technology evolution on one side and requirements of society on other side. The main mission of our educational research is to solve the problems of society and give better education to everyone. To satisfy the increasing demand for technical education, computers and web are being harnessed. Currently there are many e-learning platforms which provide online education. No current e-learning platform is concentrating on improving the performance of student. This survey provides some key insights of improving the performance of students by predicting endsem marks and grades apriori and recommending personalized resources accordingly and also discusses about designing aspects for an online course to be effective.
# Contents

1 Introduction ........................................................................... 6
   1.1 Thesis Contributions .................................................. 7
   1.2 Organization of Thesis .................................................. 7

2 Enhancing Learning Through Online Materials ......................... 8
   2.1 Problem Based Learning ............................................... 8
   2.2 Opportunities For Improved Learning ............................. 8
      2.2.1 Active Learning .................................................. 9
      2.2.2 Explanations with voice and graphics ...................... 9
      2.2.3 Simulations ..................................................... 9
      2.2.4 Probe student thinking on multiple scales ................. 9
      2.2.5 Timely assessment of progress ............................... 10
      2.2.6 Peer interaction ................................................. 10
      2.2.7 Feedback to instructors ...................................... 10
   2.3 Challenges .................................................................. 10
   2.4 Do Diagrams Enhance Learning Always? ......................... 13
   2.5 Tutorial Systems ....................................................... 17

3 E-learning Content Visualization Module .................................. 19
   3.1 Introduction ............................................................ 19
   3.2 Mobile Device Recognition Methods ............................. 19
   3.3 Content Adaptation Approaches .................................. 20
      3.3.1 Server Based Approach ...................................... 20
      3.3.2 Proxy Based Approach ....................................... 20
      3.3.3 Client Based Approach .................................... 21

4 Human and Technological Issues in E-learning ......................... 22
   4.1 Human Issues .......................................................... 22
      4.1.1 Learners Types ............................................... 22
      4.1.2 Social and Cultural Factors ................................. 23
   4.2 Technology Issues ..................................................... 23
   4.3 E-Learning Environment ............................................. 23
      4.3.1 Functional Environment ..................................... 23
      4.3.2 Mobile Environment ......................................... 24
# Predicting Student Endsem Marks and Grades

## 5.1 Introduction

## 5.2 Pre-Processing

## 5.3 Predicting Endsem Marks

### 5.3.1 Linear model in single variable

### 5.3.2 Linear model in multiple variables

## 5.4 Classification

### 5.4.1 Motivation for Predictions

## 5.5 Decision Trees

### 5.5.1 Decision Tree Learning Algorithm

### 5.5.2 Grades Prediction using Decision Trees

## 5.6 Support Vector Machines

### 5.6.1 SVM algorithm for two classes

## 5.7 Nearest Neighbour Algorithm

## 5.8 Linear Discriminant Analysis Classifier (LDAC)

### 5.8.1 LDA for two classes

### 5.8.2 MultiClassLDA

## 5.9 Combination of classifiers

## 5.10 Comparison Among Different Classifiers

# Datamining and student e-learning profile

## 6.1 Sequential Pattern Analysis

# Theoretical Model for Personalized Recommendation System

## 7.1 Theoretical model

# Conclusion And Future Work
List of Figures

2.1 Video Controls explaining equilibrium conditions .......................... 11
2.2 Hints .................................................................................. 12
2.3 Scaffolding .......................................................................... 13
2.4 compareexercise ................................................................... 14
2.5 guidedsimulation showing motion of body ................................. 15
2.6 Did I get This Exercise .......................................................... 16
2.7 Molecular Diagrams ............................................................... 17

5.1 Schematic diagram of system ................................................... 25
5.2 Best line fit for data points ..................................................... 29
5.3 Training Dataset for playing tennis .......................................... 34
5.4 Decision Tree for dataset shown above .................................... 35
5.5 Partial Decision Tree for dataset shown above .......................... 37
5.6 Accuracy using Decision Trees .............................................. 38
5.7 Linearly Seperable Data ......................................................... 39
5.8 Non Linearly Seperable Data .................................................. 40
5.9 Linearly Seperable in Feature Space ....................................... 40
5.10 Accuracy using Linear Kernel Function ................................... 42
5.11 Accuracy using Gaussian Kernel Function .............................. 42
5.12 Example for Nearest Neighbour .......................................... 43
5.13 Accuracy using Nearest Neighbour ...................................... 44
5.14 Accuracy using Nearest Neighbour with Cluster ................... 44
5.15 Accuracy using LDA ............................................................ 46
5.16 Accuracy using Classifiers Combination ............................... 47

6.1 Sequence Database ............................................................... 50

7.1 Dependency Graph for C Language Course ............................... 53
7.2 DAG for Internal Score ......................................................... 53

8.1 Structure of Proposed System ............................................... 59
## List of Tables

5.1 Root Mean Square Error .................................................. 32
5.2 Accuracy using Decision Trees Classifier .......................... 38
5.3 Accuracy using SVM Classifier .......................................... 42
5.4 Accuracy using Nearest Neighbour Classifier ....................... 43
5.5 Accuracy using Nearest Neighbour Classifier with Cluster ...... 44
5.6 Accuracy using LDA Classifier .......................................... 46
5.7 Accuracy using Combination of Classifiers(Method2) .......... 47
5.8 Accuracy using Nearest Neighbour Classifier when Labels Reduced .......... 48
5.9 Comparison Among Different Classifiers .......................... 48

7.1 Probability Distribution of Difficulty Level ....................... 56
7.2 Probability Distribution of Intelligence Level .................... 56
7.3 Probability Distribution of Guessed given Intelligence ........ 56
7.4 Probability Distribution of Marks Secured given Difficulty, Intelligence .... 57
7.5 Probability Distribution of Internal Score given Marks Secured .... 57
Acknowledgments

I would like to thank my guide, Prof. DB. Phatak for the consistent directions towards my work.
Chapter 1

Introduction

With the advent of technology, there is lot of innovation in the educational field. In traditional classroom, the words and drawings of the instructor come only once, no instantaneous feedback for homeworks, no review questions in the class on the topic taught so that instructor can know whether student understood the topic or not, huge costs for higher education etc, these disadvantages has led to increase in demand for Online education. There are lot of advantages in an online course compared to traditional classrooms, hence it has changed the education system in both US schools and colleges. In US schools, video lectures are given to home so that student can learn at their own pace and the classroom time is used productively for hands-on instruction, problem solving etc. With the start of Coursera, online education has become pre-dominant in colleges also. Everything that happens in personal classroom is not replicated in online courses as of now. But it is possible to replicate with the help of technology. The advantages of online learning are timely assessments of progress, probe student thinking on multiple scales, instructors can see dashboard to see where students are struggling, log student interactions so that data mining, machine learning algorithms can be applied to study the students learning behaviour.

As traditional lecturer is replaced by video lecture in an online course, there are many challenges in online education. Hence video lectures, ematerials etc. provided in an online course should integrate multimedia, visual and stereo features to interpret focal and difficult points in teaching. We have a lot of data in an online course which can be used to study the students learning style. If we have one offering of the students marks data in a course, then we can predict his endsem marks, grades apriori before endsem in next offering of the same course. This prediction will help the instructors to identify the weak students and help them to score better marks in endsem by giving additional practise questions to students in the area in which they are weak. Giving personalized suggestions based on student characteristics is the hot research topic going on. Till date, no e-learning system has implemented personalized recommendation system.
1.1 Thesis Contributions

Our aim is to predict student endsem marks and grades apriori and suggest them personalized learning resources so that they can improve their performance in endsem and increase their grades. First, we formulated a simple model using linear function in single variable and minimized mean square error for predicting student endsem marks. Next, we increased the complexity of model by taking the linear function in multiple variables and minimized mean square error for predicting student endsem marks. These are the two approaches which we have used to solve the problem of endsem marks prediction. Once we have predicted the endsem marks, we have used classification algorithms like decision trees, nearest neighbour, support vector machines, linear discriminant analysis, combination of classifiers to predict student final grade. Then we calculated the accuracy of these classification algorithms and did a comparison among them. After solving the prediction problem, we have developed a theoretical model for personalized recommendation system which gives personalized learning resources to students according to their learning rate.

1.2 Organization of Thesis

The remainder of this thesis report is organized as follows. Chapter 2 discusses about enhancements needed in e-materials, advantages of incorporating multimedia into electronic teaching materials to meet the needs of students and also the challenges online learning materials are facing in promoting learning when problem solving is included in e-materials. Chapter 3 presents how e-learning content is visualized, and the existing methods for mobile device recognition, and different content adaptation approaches. Chapter 4 discusses about the human and technological issues which need to be incorporated for the e-learning system to be successful. Chapter 5 presents the way we have modelled and solved the problem of prediction of endsem marks and grades. It also presents the accuracy results of different classifiers achieved. Chapter 6 presents how data mining is used to study the learners behavior and then discover the patterns to characterize learners. Chapter 7 presents a basic theoretical model developed for solving the problem of personalized recommendation system. Finally, this thesis report concludes with Conclusions and Future Work that can be done.
Chapter 2

Enhancing Learning Through Online Materials

Paper teaching materials are not meeting the instructional needs, hence e-materials which can integrate multimedia, visual and stereo features etc. has become prominent. The electronic teaching material can interpret focal and difficult points in teaching through appropriate teaching media like graph, image, audio, video etc. This chapter deals with advantages of incorporating multimedia into electronic teaching materials to meet the needs of students and also the challenges online learning materials are facing in promoting learning when problem solving is tied to drawing and solving equations.

The online course should be divided into many modules. Each module should contain some text and variety of exercises and simulations. Student Learning should be supported through “Learn by Doing” activities which offer hints and feedback. These materials support to learn effectively. In traditional classroom, main activity outside of class is solving homework problems. The feedback for these homework problems is weak. Students are not graded homework immediately, hence the defects of students are not recognized immediately. Students would benefit if they are instantaneously assessed. The following sections deals with interactive online course materials which address above deficiencies.

2.1 Problem Based Learning

According to Socrates “Education is not filling of vessel but kindling of flame”. The way to learn is how to think and solve problem. Hence problem based learning should be introduced in electronic teaching materials. The learner’s characteristics like learning needs, learner’s features, learner’s psychological profile, cognition level should also be considered when preparing e-materials.

2.2 Opportunities For Improved Learning

The following are some of the benefits in e-learning which aim at improvement in learning.
2.2.1 Active Learning

Active Learning in a large classroom is difficult. But computer-based materials which can provide short review questions and give hints and feedback can promote higher levels of cognitive activity for students. By appropriately devising online materials, students can be actively engaged throughout their learning process by giving frequently small questions and checking their progress. In this way students can assimilate new information and they can check their understanding by tests with feedback in an online course.

2.2.2 Explanations with voice and graphics

The combination of voice and graphics offers advantage over textbooks as words are linked to relevant diagrams. An instructor can provide good explanation involving voice and graphics. Students can repeat selected portions of lecture multiple times if they do not understand concept by replaying a video file. The disadvantage is that video file does not have an altered way of explanation based on student query. Generally in any online course (e.g., courses in Coursera), the video lectures are 10 to 15 min duration only. The video lectures are short because researchers say that people concentration cannot stay on a topic more than 15 min continuously.

2.2.3 Simulations

Neither a static textbook nor an instructor can offer dynamic simulations, particularly in the simulations controlled by user parameters. Online learning materials can include this aspect of simulations. Example: In Statics, it can be shown clearly in online materials, how motion of object varies with applied force.

2.2.4 Probe student thinking on multiple scales

Progress in learning is not only assessed by big problems such as found in textbooks, frequent short questions on fine-grained issues (e.g., simply with yes or no answers or multiple-choice questions or conceptual questions which requires explanation to be written and which can be compared by the student to an expert answer immediately) are also appropriate. In online environment, this is feasible than with traditional homework. We can also pose conditional questions which depend on the answers to the previous questions.

Example: Coursera became a successful online learning platform because of implementation of these techniques in teaching. Many videos contain simple multiple-choice questions (on the topic taught in that video) embedded within them. Some videos also contain reflection exercises where the student is asked to think about something, write down a response, and then share it later in the forums. These exercises concentrate on fine-grained issues. Apart from these exercises, there are quizzes, midsem, endsem etc. which concentrate at coarse-grained issues.
2.2.5 Timely assessment of progress

With little feedback and the delay in returning homework, makes student unaware that they are lagging behind in subject in an offline course. Computer-based learning materials helps the students to recognize progress instantaneously. Students can then choose to repeat selected exercises to get more practice. Students do not lose focus if feedback is immediate and they can quickly self-correct themselves. Personalized feedback improves their learning.

2.2.6 Peer interaction

Using discussion forums, students can interact with other students to clarify their doubts on subjects. Example: Piazza is a forum which has many good features for making peer interaction better.

2.2.7 Feedback to instructors

We can log student online interactions and then data-mining techniques can be used to track progress and also the detailed paths taken by individual students, then find the topics where students are finding difficulty. Instructor can then use his class time more productively to discuss those challenging problems.

2.3 Challenges

Despite these benefits, there are many challenges that are being faced by computer based online materials for learning. The materials should engage the student in as many kinds of interactions as possible. For eg. in Statics course, we should be able to ask students to choose from several forces, to choose points where forces act, or to move a slider to an appropriate point where support should be. Students can also be asked to enter free-form explanations, the computer can save this input for later inspection by the instructor. Student should be able to view an expert answer after submitting. Another challenge is good user interface design. If students want to engage with the system without any external intervention, it is important to signal to users where they are and what is expected at each instant. The following are some examples in which an online course should be designed so that students can understand better.

Complex explanations are difficult to understand with written text and diagrams. In these scenarios, both aural and visual ways of explanations can be combined so that diagrams and voice go in synchrony and makes the users attention focussed. By using video controls, user has ability to pause, stop, rewind, repeat and make him to learn at his own pace. Figure 2.1 uses Video Controls explaining equilibrium conditions and the effect of different choices of co-ordinate axes.
Online materials can also help the students by giving hints and feedback on wrong answers. These are called as “Learn By Doing” exercises. Figure 2.2 asks the student to solve for equilibrium conditions and determine the unknown support on their own and offers scaffolding, if they need help. So users can solve the problem on their own and enter the answer. User is taken through a sequence of steps and expected to perform one step at a time.

In problem shown in Figure 2.2, user is taken through a sequence of steps to solve the problem, with one step at a time. Figure 2.3 shows 4th step of scaffolding given to student in writing algebraic equation with drop down menus. The ultimate goal to follow step by step procedure and giving hints to student at each step is to make student solve problem on his own.
Figure 2.2: Hints

[1]

Figure 2.4 shows how student can compare his answer with an expert answer. In case of dynamic simulations, these online learning materials can explain the concepts in a better way than static textbook, because motion of body depends on applied force. Static textbook can only show single picture whereas these materials show motion of body depending on user entered parameters. Figure 2.5 shows how student can observe the motion of body based on the forces he applied.

Student progress can be assessed with Did I Get This Exercise. Figure 2.6 shows how student is asked to choose points where forces are acting. If the student chooses wrong answer, feedback to student explains the principles of free body diagrams. With this instantaneous feedback student can understand his mistakes immediately.

Andrew Ng, founder of Coursera also followed many of above principles when teaching his Machine Learning course, because the above way of teaching clearly explains the fundamentals of the subject and also tests his level of understanding in each topic. According to student understanding level, we can classify him and suggest customized exercises and references for him. This section explained the benefit in integrating multimedia, visual and stereo features in online materials. So the enhancements need to be included in online
2.4 Do Diagrams Enhance Learning Always?

In the previous section, we have included many diagrams for better explanation of concepts. But the question is do diagrams enhance the learning always? Generally, there are large learning gains when e-materials include diagrams and verbal explanations, but diagrams do not always lead to improved outcomes. This section describes when diagrams can enhance learning.

The need of diagram in a particular learning situation depends on learning objective, design of visual representation and cognitive processing of learner. Pictures only help if they are relevant to current instruction and exclude interesting but extraneous information. A series of studies was conducted in chemistry classrooms to test whether molecular-level diagrams (Figure 2.7) would enhance conceptual understanding of chemical equilibrium. Some participants are given materials consisting text+diagrams and some
other are given materials with only text and test was conducted for both sets of people. The result showed that students who learnt through diagrams+text did not outperform students who learnt this concept with only text. The reason for this is, students who learnt using text+diagrams mapped the features of diagrams into text and did not read the text thoroughly. The self explanation here is negatively correlated with performance because of extraneous information i.e here diagrams are un-necessary. They perceived information incorrectly as molecular level diagrams has extraneous information.

The current models of multimedia learning do not specify what features of diagrams lead to enhanced learning. Researchers suggested that three factors that determine the effectiveness of diagram are a) the specific learning objectives, b) how the
Figure 2.5: guided simulation showing motion of body

[1]

diagram makes key information salient and c) the learners cognitive processing (student must select meaningful information from the diagram for processing) and prior knowledge. **a) Specific Learning Objective**: The diagram should convey correct mental model of the system. For eg., if students want to construct mental model of a mechanical system (such as a bicycle pump), then learner must identify the parts of the system and the relationships between these parts in the diagram. The mental model here consists of process of air being pushed through the pump. A series of diagrams enhances the development of the mental model because misinterpretation might happen if only single picture used. **b) Design of diagram**: Diagram should guide to important information. Every representation makes certain aspects salient while suppressing other details. For instance, black and white line drawings make the parts and overall shape of an object salient but suppress details about color and texture. If the parts of a mechanism are important and the color is irrelevant, then black and white line drawings may be more effective than
c) Cognitive Processing of Learner: Student must select meaningful information from the diagram for processing. Processing ability and prior knowledge of the student influence the information extracted from a diagram. For eg. in learning about a bicycle pump, the student must understand that pushing down the handle of the pump causes air to flow out of the open valve at the bottom because air is an entity that has mass and decreasing the volume of air results in greater pressure on the valve. The diagrams created using above 3 principles improve learning.
2.5 Tutorial Systems

This section describes types of tutorial systems which need to be incorporated in e-learning. General and Domain Specific are two types of tutorial systems.

a) General Tutorial Systems These are also called as Help-type systems. These are designed to support teaching activities. Help type systems in the software product appear as specific command menu. Their purpose is to assist the user when he needs information about using the application.

i) Text Type Help: Information is present as text only. It displays the explanation about option chosen by the user. Read me files also come under this category.

ii) Hyper link help: Displays the information of selected terms by clicking the term.

iii) Help with related topics: It is used to access information related to data present on main screen.

iv) What’s This type: It offers help regarding buttons and commands on screen.

v) Offline Tutorial Help: These contain tutorial fragments that can be accessed without internet connection. These contain short tutorials to guide the student in step by step
v) **Online Tutorials:** These contain tutorials which can be accessed using internet connection.

b) **Domain Specific Learning Systems** These systems are designed based on its working domain. Example: Learning Software for Economics uses graphics to explain financial concepts. Learning Software for Mathematics provides contents according to age groups. For 12-15 years children it explains basic algebra and for age > 15 it explains advanced algebra.
Chapter 3

E-learning Content Visualization Module

Till now, we have seen what content to be put up in e-learning materials. In this chapter, we see how e-learning content is visualized, and the existing methods for mobile device recognition, and different content adaptation approaches.

3.1 Introduction

The evolution of technology has led to wide variety of devices like desktop computers, notebook computers, tablets, personal digital assistants (PDAs), cell phones, mobile phones etc. Each device differs from other in web browser they use, supported mark up languages(HTML, XHTML, cHTML, WML etc), supported script languages, file formats, screen resolution etc. These different characteristics of the devices shows the need to develop an adaptation method for correct visualization of electronic content.

The solution for problem of correct content visualization on different devices is very important for e-Learning because same education content need not be developed again.

Example: The existing courses in the e-Learning Shell Software Platform are available only for desktop computers. They cannot be used with mobile devices. Hence there is need for developing new content visualization subsystem which delivers content to users that use any device like PC’s, tablets, mobile devices etc.

3.2 Mobile Device Recognition Methods

First device has to be recognised for content adaptation. Currently request header field in http protocol is used by servers to identify the device. Alternative methods to identify device are:

HTTP USER AGENT HEADER:Browsers use http protocol to transfer information on web. The server decides the kind of information to send depending on device profile.
Http request sends Accept Header which indicates the types of data the browser accepts. The client also sends User Agent Header for identifying client device and contains information about browser, operating system and hardware information. The drawback here is, the information in user agent header is not sufficient as number of different kind of devices are growing up.

**Composite Capabilities/Preferences Profiles (CC/PP):** World Wide Web Consortium has standard way for specifying CC/PP. The standard says, the devices should transmit their configuration details and abilities like screen resolution, audio characteristics, frequency band etc. to web servers. CC/PP is universal profile that describes the devices characteristics.

**WAP User Agent Profile:** The device is identified using user agent profile. When mobile device sends request to server, it also sends an URL address to its mobile profile by adding X-Wap-Profile Header in request. This header indicates the server where to find the device profile. The content server extracts the necessary information about the client from device profile repository and can store it, so that it can be used later. Information is present in XML, hence using XSLT information can be transformed to the type which device can recognize.

### 3.3 Content Adaptation Approaches

Content adaptation is a process of selecting, generating or modification of content (text, images, audio, video), so that it can be presented to different devices. If the web page is accessed using desktop computer then it needs no modification. If the same web page is accessed via mobile device, images must be resized and compressed, text must be formatted and video is presented as text or as an image depending on available bandwidth. Hence content adaptation is needed. Different Content Adaptation Approaches are as follows.

#### 3.3.1 Server Based Approach

In this approach, web server has modules for content adaptation. The same content is stored in different versions on the server, but only the content, which coincides with the clients profile is presented. General approach used for content delivery for different devices is to store the information in XML format and then use XSLT to transform to the client preferable markup language.

#### 3.3.2 Proxy Based Approach

In this approach, a proxy server analyses and transforms the content before sending it to the client. Proxy server also caches the adapted content so that it can be used later. Proxy server and the web server should know type of device to send the appropriate content.
Eg: AvantGo is a service that uses a proxy-based approach and delivers mobile websites to consumers’ personal digital assistants (PDAs) and smartphones.

### 3.3.3 Client Based Approach

In this approach, the necessary transformation is done by the consumer’s device. Eg: Opera Software uses client-based adaptation technology called Small-Screen Rendering (SSR) which transforms the content according to the consumer’s device type. Another method is to use XHTML and different CSS for each device type.
Chapter 4

Human and Technological Issues in E-learning

Many e-learning systems fail to satisfy the learners needs and requirements due to the absence of human and technological issues such as social and cultural factors, quality components and pedagogy requirements, technological issues such as the learning environment. E-learning systems are complex web applications as they should transfer traditional pedagogy methods and practices to electronic environment.

4.1 Human Issues

The development of a successful learning system is based on understanding the learners needs and behaviour and incorporating them in learning process. This will help achieve the pedagogical goals. When developing an e-learning system, we need to classify the learners needs and behaviour. General user requirements for E-learning system in perspective of web-design process are

a) Keep navigation clicks to minimum
b) Keep scrolling to minimum
c) Have contents both for low and high-connection speeds
d) Have a consistent user interface

4.1.1 Learners Types

The learners are classified into following types:

**Traditional Learner:** Focuses his effort on reading the required material.

**Achiever:** Focuses his effort on completing the quizzes and review questions.

**The Interactive Learner:** Focuses his effort on interacting with peers and tutors

**The Struggler:** Studies less frequently than all other students.

Clearly, with above classification we can see some learners are passionate, some are exam based and so on. These different types of learners should be given different kinds of
learning resources and different types of learning support to suit their individual needs. To do so, we need to have detailed information of each student which is possible in an online course only. For example, in an online course we can know whether student is reading required material and solving practise questions before attempting exam or just reading and attempting exam or without reading attempting exam.

4.1.2 Social and Cultural Factors

Learner behaviour is shaped by social and cultural factors also.

Demographics: The Web engineer must specify and design the E-Learning application based on the targeted population.

Social characteristics: The developer should examine the educational system, the literacy level, and the languages spoken in the country.

4.2 Technology Issues

Identifying the technology level of each targeted country helps the Web engineer to decide on the type of technology and resources to use. Countries with advanced technologies and high web usage are excellent candidates for an e-learning application. Countries new in the Internet arena have basic technologies, hence need to design E-Learning systems with low bandwidth and capabilities due to poor communications. The e-learning platform should have quality factors like usability, functionality, efficiency, reliability and maintainability.

4.3 E-Learning Environment

E-Learning environment is separated into two sub-environments a) Functional Environment and b) Mobile Environment

4.3.1 Functional Environment

E-Learning system should have following operations: Class announcements, access to course material, assignments and case studies, online quizzes with a timer and a feedback mechanism, a virtual classroom with collaborative study groups. In addition, a Personal Preferences section where the system provides progress report, a homework submission utility, access to classmates public information, and should be able to view other courses enrolled by student. Other facilities to be provided are online help with access to the helpdesk administrator, a calendar for scheduling, a customization utility for specific preferences such as, the change of language, a requirements analyzer tool for suggesting new requirements, and external links to online databases and related material.
4.3.2 Mobile Environment

Mobile education is defined as the dissemination of pedagogical material through the use of wireless networks and devices. With technology revolution e-learning is migrating to mobile environment. Technologies used for providing wireless access are WLAN, WAP, Short Message Service(SMS), and UMTS. WLAN is usually only available at the institution premises, whereas WAP, SMS and voice-technologies are more widely available.

Till now, we have discussed about design aspects of an online course. In the next chapter, we discuss about prediction of marks and grades using learning algorithms.
Chapter 5

Predicting Student Endsem Marks and Grades

Classification is one type of prediction where predicted variable is binary or categorical (means takes some discrete value) variable. Classification methods like decision trees, nearest neighbour algorithms etc can be applied on the educational data for predicting the students behavior, performance in examination etc. Classification is nothing but dividing data tuples into different classes. This prediction will help the instructors to identify the weak students and help them to score better marks in endsem by giving additional practise questions to students in the area in which they are weak (finding overall weakness of all students which is possible in an online course and suggesting accordingly) and also give customized references (finding where each student is weak and suggesting accordingly) to them in an online course. We have used linear function (with respect to parameters) and minimised mean square error to predict endsem marks and then used decision tree, k-nearest neighbour, support vector machines, linear discriminant analysis techniques in predicting student grades. This chapter explains different classification algorithms and how it can be applied in educational field to predict grades of the student.

![Figure 5.1: Schematic diagram of system](image)
5.1 Introduction

Machine learning is a branch of artificial intelligence which deals with study of systems that can learn from data. Machine learning focuses on prediction based on known properties learned from the training data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam messages. In the same way, we can learn from one offering of CS101 data and then predict endsem marks and grades in another offering so that weak students can be identified earlier and appropriate measures can be taken to improve their performance in endsem.

A learning problem usually considers a set of $p$-dimensional (each dimension represents an attribute) samples of data and tries to predict properties of unknown data. We are given training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. $x_i$ denotes tuple representing different attributes of $i$th object. $y_i$ represents the class to which tuple $x_i$ (i.e $i$th object) belongs and are called as labels. In classification problem, the learner approximates a function mapping a vector $x$ into classes $y$ by looking at input-output examples of the training data. The goal during training is to learn function $f(x) \rightarrow y$ such that prediction error on unseen instances should be small.

What function to choose is the biggest challenge as the space of possible functions is extremely large. We should limit the set from which $f$ is chosen. The problem of which function class to choose is called model selection problem. We need to ensure that $f$’s performance on $D$ should generalize to unseen instances. Generalization here refers to ability of an algorithm to perform accurately on new, unseen examples after having trained on a learning data set. The training examples come from unknown probability distribution (generally) and the learner has to extract something more general, something about that distribution from them such that it allows us to produce useful predictions in new cases.

5.2 Pre-Processing

Before applying machine learning algorithm for classification, the given data must undergo pre-processing operations because the data given to us is raw-data, which consists of many attributes. All the attributes do not contain necessary information which is needed for our prediction. The attributes given in original dataset are Serialnumber, Roll No, Name, Batch, Day, TA, Sub Batch number, Assignment lab 3, Assignment lab 4, Assignment lab 5, Assignment lab 7, Assignment lab 9, Quiz, Project Stage I, Project Stage II, Midsem, Endsem, Total, Grades. These are the attributes present in original raw data. Based on problem we are solving, we select subset of attributes which are useful to us from given attributes. For the problem of endsem marks prediction, we have selected assignment, quiz, midsem attributes. For the problem of grade prediction, we have selected assignment, quiz, midsem, project and predicted endsem marks. Machine Learning can also be applied
to select the students for formation of group in a project. Normally all the top grade
students form a group and they don’t allow least grade students to join their group, so
learning for least grade students will not happen properly. So automatic formation of
groups is needed. If we are solving this problem of grouping the students for project, then
we also need batch number, day attributes because we need to form groups among the
same batch members on that day. Hence the attributes that should be selected depends
on problem we are solving.

The target variable grades have values AA, AB, etc. We need to convert them to
numerical values. In the given marks dataset, absent students are marked as “AB” and one
of the grades is also marked as “AB”. The values of marks columns should be numerical,
hence absent marked AB need to be converted into value 0. The problem arises if we run
script directly to convert grade to value because absent marks(AB) will also be converted
into value 9. So we first converted absent student marks entry to 0 from “AB” and then
grade AB to its corresponding value using shell scripts. Grades are given in another file.
Hence shell scripts are written to combine both grades and marks. Lot of operations
in excel are done to find the weightage of different attributes because the weights of
attributes should be same in prediction of final grade in both training and test sets. For
example, in 2011 offering, endsem is given 30% weightage and the endsem is conducted for
50 marks, whereas in 2010 offering, endsem is given 35% weightage and conducted for 70
marks. We should normalize the values in both offerings to their respective weightages(i.e
finding weighted endsem marks after normalization)

Before applying any machine learning algorithm, we need to ensure that the attributes
in training and test are in same order and should contain same weightage in determining
target attribute. The exact weightage need not be known to us but their weightage should
be same. The algorithms itself learns the amount of weight it should give to attribute
from training data so that it can predict target attribute properly. The biggest stumbling
block we have faced is, data is not correctly given. In 2010 CS101, assignments are given
5% weightage and no quiz exam, project was given 40% weightage, midsem is given 20%
weightage, endsem is given 35% weightage. In 2011 CS101, assignments are given 15%
weightage and quiz exam was given 10% weightage, project was given 25% weightage,
midsem is given 20% weightage, endsem is given 30% weightage. Clearly, the weights of
these features are given differently in 2011 and 2010 and there was no attribute of quiz
in 2010. Hence we have taken 2011 marks as training data and found the relationship
among attributes. Using these relationships and using a linear model, we predicted quiz
marks. We have changed weightage of remaining attributes to same as that of 2011 data.
We found the distribution of grades in 2010 data and re-assigned the grades to students
with the same distribution by calculating total and created a new datset for 2010. We
have used linear model for prediction of quiz marks and minimized mean square error as
it gives good approximate predictions for our problem domain. Instead of above way of
prediction for quiz marks, we can also combine attributes and do but we don’t get proper
results. For example, if we combine quiz+project marks as one feature then we don’t get
proper accuracy because project is team work, all individuals don’t contribute equally

27
and project marks are given 40% weightage in 2010, so even less intelligent person gets
good marks in this new attribute, and if we predict endsem marks using this attribute
we get wrong prediction of endsem marks. In this way from the raw data given, we
have organised them into proper format and selected appropriate attributes and formed
training(2011 CS101 offering) and testing datsets(2010 CS101 offering).

5.3 Predicting Endsem Marks

Predicting a continuous value target variable is known as regression. From training data,
we have to capture relationship between different attributes and then develop a model
which can predict the value target attribute takes. The training data in our problem
consists of CS101 data. The attributes consists of assignment, quiz, midsem, project,
endsem, total, grades.

For the classification problem, grades (discrete variable) is the target variable. For regres-
sion problem, endsem marks (continuous variable) is the target variable. For predicting
endsem marks, we consider assignment, quiz, midsem attributes only. We don’t consider
project marks because project is a team effort. Project marks doesn’t show individual
caliber in CS101 as the group size is big and hence all the people need not contribute
equally. We have observed data, many persons who got less marks in exams have scored
good marks in project because of large group. By observing data, we have found people
who got 0 marks and very very few marks in exams also secured 20 marks in project
because of group effect. It happens in CS101 because of large team size. In other courses,
just 3 or 4 people per team and hence each has to work out. In CS101 as group size is 7
to 10 members, only 3 or 4 people work out and remaining also get good marks. Hence
we have excluded project attribute in predicting endsem marks.

5.3.1 Linear model in single variable

We don’t have any information about the distribution of data. But we can find mean,
variance and correlation between attributes from the given training data. So the only
information which we have about the data is mean and variance. There is relationship
between the attributes. For example, if a person gets good marks in midsem then he is
most likely to get good marks in endsem also. We need to capture such a relationship
among the attributes from training data and develop a model which can predict target
attribute(endsem marks) in test data. Hence we have used a linear function of the form
\[ Y = a + bX \] which can best predict the target variable endsem marks($Y$). Initially $Y$
refers to known endsem marks, $X$ refers to any one of known assignment, quiz or midsem
marks when we are finding parameters $a, b$ from training data. When test data is given as
input, $X$, $a$, $b$ are knowns and $Y$ is unknown and refers to endsem marks to be predicted.

To obtain the best linear function which can predict endsem values(target variable),
we need to choose ’a’ and ’b’ such that prediction error is minimized. Error between
actual value and predicted value is ($Y - (a + bX)$) where $Y$ here denotes actual value(i.e
actual endsem marks in training data), $a + bX$ denotes predicted value of endsem marks using our fitted model.

Minimize $E[\{(Y - (a + bX))^2\}]$ (known as mean square error) where $E$ refers to Expectation. We have squared inner term to make the error function convex. If a function is convex, then it is differentiable and hence we can find optimum values of ‘a’ and ‘b’ by differentiating the above equation.

$$E[\{(Y - (a + bX))^2\}] = E[Y^2 - 2aY - 2bXY + a^2 + 2abX + b^2X^2]$$

Taking partial derivatives

$$\frac{\partial}{\partial a} E[\{(Y - a - bX)^2\}] = -2E[Y] + 2a + 2bE[X]$$
$$\frac{\partial}{\partial b} E[\{(Y - a - bX)^2\}] = -2E[XY] + 2aE[X] + 2bE[X^2]$$

After equating above equations to 0 and solving for a and b, we get

$$b = \frac{E[XY] - E[X]E[Y]}{E[X^2] - (E[X])^2}$$
$$= \frac{Cov(X,Y)}{\sigma_x^2}$$
$$= \rho \frac{\sigma_y}{\sigma_x}$$

where $\rho$ refers to correlation between X and Y attributes, $\sigma_y$ refers to standard deviation of Y, $\sigma_x$ refers to standard deviation of X.

$$a = E[Y] - bE[X]$$

**Example**

![Figure 5.2: Best line fit for data points](image)
The following is an example in two dimensional space to understand the above linear model.

Consider four \((x, y)\) data points. \((1, 6), (2, 5), (3, 7)\) and \((4, 10)\) (shown in red in figure 5.2). We need to find a line \(y = a + bx\) that best fits these four points(training data). The assumed function is linear in the parameters to be estimated. In other words, we want to find the parameters ‘a’ and ‘b’ such that mean square error (=\(E[(\hat{Y}_i - Y_i)^2]\), \((\hat{Y}_i - Y_i)\) is known as residual which is shown as vertical distances in figure 5.2) is minimized where \(\hat{Y}_i\) denotes predicted value for \(i\)th point and \(Y_i\) denotes actual value for \(i\)th point. Substituting above points in line equation results in following equations.

\[
\begin{align*}
a + b &= 6 \\
a + 2b &= 5 \\
a + 3b &= 7 \\
a + 4b &= 10 \\
\end{align*}
\]

We have four equations and two unknowns i.e number of data points used for fitting exceeded the number of unknown parameters. This is an overdetermined system of linear equations. We cannot solve these equations directly. The best approximation is we need to minimize the sum of squared differences(i.e error) between the predicted data values and their corresponding actual values.

Error function \(S(a, b) = [6 - (a + b)]^2 + [5 - (a + 2b)]^2 + [7 - (a + 3b)]^2 + [10 - (a + 4b)]^2\)

The minimum is determined by calculating the partial derivatives of \(S(a, b)\) with respect to \(a, b\) and setting them to zero

\[
\begin{align*}
\frac{\partial S}{\partial a} &= 0 = 8a + 20b - 56 \\
\frac{\partial S}{\partial b} &= 0 = 20a + 60b - 154 \\
\end{align*}
\]

This results in a system of two equations in two unknowns. After solving we get

\(a = 3.5, b = 1.4\)

For the given set of points, the best fit line is \(y = 3.5 + 1.4x\). The resulting fitted model can be used to predict unobserved values from the same system.

The above example is an application in two-dimensional space. In this manner using our training data of CS101, we predicted endsem marks(Y) if the system gives midsem marks(X). Similarly we predicted endsem marks(Y) if quiz marks(X) is given and predicted endsem marks(Y) given assignment marks(X). We can predict like this because there is relationship between attributes of endsem, midsem, quiz, assignment. We captured that relationship and taken average of 3 endsem marks obtained using quiz, midsem, assignment. This is the basic model we used for prediction of endsem marks. Instead of taking average, we can take weighted average which is still more good model. Hence we have used linear model with multiple variables which is discussed in next section.
5.3.2 Linear model in multiple variables

In the previous model, we have predicted endsem marks by taking

\[ \text{endsem} = f(\text{midsem}) \]

\[ \text{endsem} = f(\text{assignment}) \]

\[ \text{endsem} = f(\text{quiz}) \]

and then calculated average of 3 endsem marks obtained. In previous model, my problem domain is in two dimensional space. In this model, problem domain is in four dimensional space.

In this model, the relationship being captured is complicated.

\[ \text{endsem} = f(\text{assignment}, \text{quiz}, \text{midsem}) \]

Writing mathematically using linear function for prediction.

\[ \sum_{j=1}^{3} (X_{ij}b_j + b_0) = y_i, \quad (i = 1, 2, \ldots, m) \]

where \( i \) refers to \( i \)th training example, \( j \) refers to \( j \)th attribute. \( X_{ij} \) refers to \( j \)th attribute value of \( i \)th training example. \( m \) refers to total number of training examples.

This is an overdetermined system with \( m \) linear equations and 4 unknowns \((b_0, b_1, b_2, b_3)\).

The above can be written in matrix form as

\[ \mathbf{X} \mathbf{b} = \mathbf{y} \]

where

\[ \mathbf{X} = \begin{bmatrix}
1 & X_{12} & X_{13} & X_{14} \\
1 & X_{22} & X_{23} & X_{24} \\
\vdots & \vdots & \vdots & \vdots \\
1 & X_{m2} & X_{m3} & X_{m4}
\end{bmatrix} \]

\( \mathbf{X} \) contains training dataset i.e \( X_{12} \) denotes \( i \)th person assignment marks, \( X_{13} \) denotes \( i \)th person quiz marks, \( X_{14} \) denotes \( i \)th person midsem marks

\[ \mathbf{b} = \begin{bmatrix}
b_0 \\
b_1 \\
b_2 \\
b_3
\end{bmatrix} \]

\( \mathbf{b} \) denotes vector of unknown parameters

\[ \mathbf{y} = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_m
\end{bmatrix} \]

\( \mathbf{y} \) denotes value of target attribute (i.e here endsem marks). When used for training data to obtain \( \mathbf{b} \), \( \mathbf{y} \) denotes training endsem marks (known values). Once \( \mathbf{b} \) is obtained, the above model is used for prediction of endsem marks, at that instant \( \mathbf{y} \) denotes endsem marks of test data to be predicted.

This is overdetermined system as number of equations greater than number of unknowns and hence has no solution. So the goal is to find the coefficients \( \mathbf{b} \) which fit the equations best in the sense of minimising mean square error \((=E[(\text{actual value} - \text{predicted value})^2])\).
\[ \hat{b} = \arg \min_b S(b), \] 

\( \hat{b} \) refers to optimal value of \( b \)

where the objective function \( S \) is given by

\[ S(b) = \sum_{i=1}^{m} |y_i - \sum_{j=1}^{3}(X_{ij}b_j + b_0)|^2 = \|y - Xb\|^2. \]

The objective is to minimize

\[ S(b) = \|y - Xb\|^2 = (y - Xb)^T(y - Xb) = y^T y - 2b^T X^T y + b^T X^T X b. \]

Differentiating with respect to \( b \) and equating to zero gives

\[ -X^T y + (X^T X)\hat{b} = 0 \]

\[ \hat{b} = (X^T X)^{-1} X^T y \]

Using the above equation, the vector \( b \) (which contains unknown parameters) is found. In this way unknown parameters are estimated with the help of training data. Once the unknown parameters are found, we can predict endsem marks for test data as

\[ \text{endsemmarks} = b_0 + b_1 \text{assignment} + b_2 \text{quiz} + b_3 \text{midsem} \]

<table>
<thead>
<tr>
<th>Endsem Prediction</th>
<th>Root Mean Square Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Model in Single Variable</td>
<td>5.09</td>
</tr>
<tr>
<td>Linear Model in Multiple Variable</td>
<td>5.1</td>
</tr>
</tbody>
</table>

Table 5.1: Root Mean Square Error

Table 5.1 shows root mean square error \( (\sqrt{E[(\text{predictedendsem - actualendsem})^2]]) \) in prediction of endsem marks. The good value for the root mean square error depends on data we are trying to model or estimate. Analysis result says that 5% to 20% error is good estimate. The errors we have got is nearly 5%. So we have achieved good estimates of endsem marks using our model.

### 5.4 Classification

An algorithm that implements classification is known as a classifier. It refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category. Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems. Determining a suitable classifier for a given problem is more an art than a science. There are different types of classifier like probabilistic classifier, discriminative classifier, distance based classifier etc. The type of classifier to be chosen depends on problem we are trying to solve.

The training data in our problem consists of CS101 data. The attributes consists of assignment, quiz, midsem, project, endsem, total, grades. For predicting grade, we consider assignment, quiz, midsem, project, endsem, total attributes. We do consider project marks in determining grade because this attribute influences final grade of the student.
5.4.1 Motivation for Predictions

We have used classification techniques like decision tree, nearest neighbour, support vector machines, linear discriminant analysis to predict the grade of student in CS101 so that student can improve their performance in endsem and improve their grade. In foreign universities, marks of one person is not shown to another. In an online course also, marks of one person is not shown to another. In any course, showing marks of one student to other is not good and students cannot predict their grade as course like CS101 has huge strength, students also do not know marks of all other people and they do not know distribution which professor follows in grading. Through historical information we can build the model. So for a student who is yet to write endsem this year can approximately know his grade before writing endsem, which is predicted with the marks of exams conducted till then. Based on his predicted grade, we can also provide him customized references so that he can read those materials and can improve in endsem. This model will work per professor because each professor paper difficulty level, exam criterion, grading strategy etc are different. These models will work very well in an online course because the same professor will offer the course many times. For example, in Coursera, Machine Learning is always offered by Andrew Ng, Probabilistic Graphical Models is always offered by Daphne Koller. In an online course, we can get huge data as we can track each and every click of user, whether they are studying suggested materials or not etc. Using this data, we can get many inferences about the student and then we can do customized suggestions accordingly. Till date, no online course has achieved customized suggestions.

5.5 Decision Trees

Statistical data is represented in terms of tuples. The data consists of many attributes and there is target attribute which we have to predict. For example, in the dataset shown in Figure 5.3, the target attribute to be predicted is PlayTennis. This attribute shows whether we can play tennis or not on a particular day i.e there are 2 classes, yes or no. This problem is called as classification problem because we are classifying the dataset into 2 classes which says whether we can play tennis or not on a particular day. The PlayTennis attribute is predicted depending on the values other attributes like Temperature, Humidity etc. take on that particular day. For prediction, here we use Decision Tree technique.

Decision tree is a technique used for classification of instances. It is a method for predicting discrete-valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be represented by if-then rules. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node and then moving down the tree branch corresponding to the value the attribute takes in given example. This process is then repeated for the subtree rooted at the new
node. Figure 5.4 shows a learned decision tree from dataset shown in Figure 5.3. This decision tree classifies whether morning of a day is suitable for playing tennis or not. For example, if the morning of a day has these attributes (Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong), then this instance would traverse via leftmost branch of decision tree shown in Figure 5.4 and would therefore be classified as a negative instance (i.e. tree predicts that PlayTennis = no).

Decision trees can be represented as a disjunction of conjunctions of constraints on the attribute values of instances. Every path from the tree root to a leaf corresponds to a conjunction of attribute tests and the tree itself is a disjunction of these conjunctions. For example, the decision tree shown in Figure 5.4 corresponds to the expression (Outlook = Sunny AND Humidity = Normal) OR (Outlook = Overcast) OR (Outlook = Rain AND Wind = Weak). This expression says whether to play tennis or not on a particular day which is dependent on attributes of that day. The attributes we considered are climatic conditions.

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 5.3: Training Dataset for playing tennis
[Tom Mitchell TextBook]

Fig 5.3 shows data set which has target attribute PlayTennis. It can have values yes or no for different mornings. Whether a person has to play Tennis or not depends on attributes like Outlook, Temperature etc. Given a new tuple we predict whether he can play Tennis or not by traversing decision tree.
5.5.1 Decision Tree Learning Algorithm

Basic algorithm in decision tree is called as ID3. It learns decision trees by constructing it top-down fashion. Each attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used to test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute and training examples are splitted to the branch corresponding to the example’s value for this attribute. The entire process is then repeated using the training examples associated with each descendant node so as to select the best attribute to test at that point in the tree.

The attribute to be tested at a node in the tree for classifying examples depends on a statistical property called information gain of attribute. Information gain measures how well a given attribute separates the training examples with respect to their target classification. ID3 uses this information gain measure to select among the candidate attributes at each step while growing the tree.

Given a collection $S$, and if the target attribute has $c$ different values, then the entropy of $S$ with respect to this $c$-wise classification is defined as $\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$.
where $p_i$ is proportion of $S$ belonging to class $i$. Information gain, $\text{Gain}(S, A)$ of an attribute $A$ for a collection of examples $S$, is defined as

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where $\text{Values}(A)$ is the set of all possible values for attribute $A$ and $S_v$ is the subset of $S$ for which attribute $A$ has value $v$.

In the given dataset, $S$ is a collection of 14 examples. Attribute Wind has the values Weak or Strong. There are 9 positive (i.e., play tennis = yes) and 5 negative examples denoted as [9+, 5-]. Of these 14 examples, 6 of the positive and 2 of the negative examples have Wind = Weak. The information gain from attribute Wind is calculated as follows:

$\text{Values(Wind)} = \text{Weak, Strong}$

$S = [9+, 5-]$

$S_{\text{weak}} = [6+, 2-]$

$S_{\text{strong}} = [3+, 3-]$

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - \sum_{v \in \text{Weak, Strong}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - (8/14) \text{Entropy}(S_{\text{weak}}) - (6/14) \text{Entropy}(S_{\text{strong}})$$

$$= 0.940 - (8/14)0.811 - (6/14)1.00$$

$$= 0.048$$

Information gain is used to select the best attribute at each step in constructing the tree. So information gain is calculated for each attribute. The information gain values for all four attributes are

$\text{Gain}(S, \text{Outlook}) = 0.246$

$\text{Gain}(S, \text{Humidity}) = 0.151$

$\text{Gain}(S, \text{Wind}) = 0.048$

$\text{Gain}(S, \text{Temperature}) = 0.029$

where $S$ denotes the collection of training examples from dataset shown in Fig 5.3

According to information gain, the Outlook attribute best classifies the target attribute, PlayTennis, over the training examples. So, Outlook is selected as the decision attribute for the root node, and branches are created below the root for each of its possible values it takes i.e, Sunny, Overcast, and Rain. The resulting partial decision tree is shown in Figure 5.5. The training examples are split to the branch corresponding to the example’s value for this attribute. Every example for which Outlook = Overcast is a positive example of PlayTennis. Therefore, this node of the tree becomes a leaf node with the classification PlayTennis = Yes. The descendants corresponding to Outlook = Sunny and Outlook = Rain do not have all positive or all negative. So the decision tree procedure is repeated for these nodes.

The process of selecting a new attribute and splitting the training examples is now repeated for each descendant node. Now only the training examples associated with that node are used. Attributes that are incorporated higher in the tree are excluded as any given attribute should appear at most once along any path through the tree. This process repeats for each new leaf node until either of the conditions is met: (1) every attribute has already been included along this path in the tree (2) all the training examples associated with this leaf node takes same target attribute value. Final decision tree is shown in
Figure 5.4

![Decision Tree Diagram]

Figure 5.5: Partial Decision Tree for dataset shown above

Tom Mitchell TextBook

5.5.2 Grades Prediction using Decision Trees

Decision tree algorithm is applied on CS101 past performance data to generate the model and this model is used to predict the students grades in CS101 of first year engineering students. This will enable to identify the students who are likely to fail in advance, who have shown poor performance etc so that they can improve their performance in endsem exam and push up their grades.

This section describes the model that predicts the grades in CS101 course using decision trees. The features selected for creating the model are based on students past performance in CS101 data(training data) consisting of marks and grades in CS101.
The model is built by analyzing the data tuples from training data which has assignment, quiz, midsem, project, endsem, total, grades as attributes. For each tuple in the training data, the value of target class attribute (grades) is known. Using this data we build the model using decision tree technique. Next, test data is used to check the accuracy of the model. If the accuracy of the model is acceptable then the model can be used to classify the data tuples for which the class label is not known (i.e., we predict freshmen grades of test data).

Table 5.2 summarizes the accuracy using decision tree classifier in predicting final grade. Accurate accuracy (Single Variable) column refers to exact grade predictions when endsem marks are predicted using linear model in single variable. Approx. Accuracy (Single Variable) column refers to prediction where + or -1 grade deviation from original grade is also assumed to be correct prediction, when endsem marks are predicted using linear model in single variable. Accurate accuracy (Multi Variable) column refers to exact grade predictions when endsem marks are predicted using linear model in multiple variable. Approx. Accuracy (Multi Variable) column refers to prediction where + or -1 grade deviation from original grade is also assumed to be correct prediction, when endsem marks are predicted using linear model in multiple variable.

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy (Single Variable)</th>
<th>Approx. Accuracy (Single Variable)</th>
<th>Accurate Accuracy (Multi Variable)</th>
<th>Approx. Accuracy (Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DecisionTree</td>
<td>65.32</td>
<td>98.24</td>
<td>65.14</td>
<td>98.42</td>
</tr>
</tbody>
</table>

Table 5.2: Accuracy using Decision Trees Classifier

![Decision Tree Accuracy](image.png)

Figure 5.6: Accuracy using Decision Trees
5.6 Support Vector Machines

Support Vector Machines (SVM) is based on decision planes that define decision boundaries. A decision plane separates objects which are having different classes. A data point is represented as a p-dimensional vector (p denotes number of attributes). If there exists (p - 1) dimensional hyperplane that separates the points, then it is called as linear classifier. Many hyperplanes exist that classify the data. SVM tries to find a hyperplane that has the maximum separation (known as margin) between the two classes i.e it chooses a hyperplane such that distance from hyperplane to the nearest data point on either side is maximized.

**Example:** In the training data shown in figure 5.7, objects belong either to class green or red. SVM learns from the data and finds a decision boundary (a line in our case) which separates green and red objects. On the right side of line, all objects are green and to the left, all objects are red. Any new object falling to the right of line is labeled i.e, classified as green. If it falls to the left of the separating line, it is classified as red. There are many decision boundaries that separate red and green groups but SVM tries to separate different classes by a wide gap as much as possible. This gap is called as margin (distance between lines 2 and 3).

![Linearly Separable Data](image)

Figure 5.7: Linearly Separable Data

The above is an example of a linear classifier i.e, a classifier that separates a set of objects into their respective groups (green and red groups in our example) with a line. In reality, most classification tasks, are not that simple and complex decision boundaries are needed to make an optimal separation in order to correctly classify new objects (test set) on the basis of the examples that are available (train set). In example shown in figure 5.8, it is clear that for good separation of the green and red objects requires a curve which is more complex than a line. Classification tasks based on drawing linear decision boundaries to distinguish between objects of different classes are known as hyperplane classifiers.
Figure 5.8: Non Linearly Separable Data

Figure 5.9 shows the idea behind Support Vector Machines. The original objects, left side of the diagram are mapped to objects in different space (shown in right side of diagram) using a set of mathematical functions known as kernels. The mapped objects on the right side of the diagram are linearly separable. So instead of a complex decision boundary as in left diagram, we map objects into different feature space and find an optimal line that separates the green and the red groups in new feature space. This is the working procedure of support vector machines.

\[ \text{Input Space} \rightarrow \text{Feature Space} \]

5.6.1 SVM algorithm for two classes

Support Vector Machine (SVM) is a classifier that performs classification tasks by constructing hyperplanes in multidimensional space and separates data of different class labels with as much separation as possible.

Training Data: \( \mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\} \) where \( \mathbf{x}_i \) is input vector containing \( p \) attributes, \( y_i \) denotes class label. We need to find a hyperplane that separates two classes and should maximize the margin.

The equation of hyperplane (eg. line1 in Fig 5.7) is of form \( \mathbf{w} \cdot \mathbf{x} - b = 0 \) where \( \cdot \) denotes dot product and \( \mathbf{w} \) is normal vector to the hyperplane.

The equation of hyperplane2 in Fig 5.7 is \( \mathbf{w} \cdot \mathbf{x} - b = 1 \) (if we assume green objects has class
(1) and

The equation of hyperplane 3 in Fig 5.7 is \(\mathbf{w} \cdot \mathbf{x} - b = -1\) (if we assume red objects has class label=-1).

The distance between two hyperplanes 2 and 3 in Fig 5.7 is \(\frac{2}{\|\mathbf{w}\|}\). For maximum margin, we need to minimize \(\|\mathbf{w}\|\). As training set is linearly separable in Fig 5.7, we add the constraints
\[
\begin{align*}
\mathbf{w} \cdot \mathbf{x}_i - b &\geq 1 & \text{for } \mathbf{x}_i \text{ belongs to green class} \\
\mathbf{w} \cdot \mathbf{x}_i - b &\leq -1 & \text{for } \mathbf{x}_i \text{ belongs to red class.}
\end{align*}
\]

Combining the above two equations, we get \(y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \text{ for all } 1 \leq i \leq n\).

The optimization objective for SVM then becomes
\[
\begin{align*}
\text{Minimize } & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\
\text{subject to } & (\text{for any } i = 1, \ldots, n) \text{ constraints } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1
\end{align*}
\]

Here we tried to fit training data with linear kernel (i.e no kernel is used). If we fit training data with complicated function (non linear function), then we need to use other than linear kernel and we need to transform from given input space to feature space. When we try to fit using non linear function for linearly non separable data, then the optimization objective changes. In this case SVM chooses a hyperplane that splits the training examples as cleanly as possible, and also tries to maximize the margin as much as possible.

In this method we introduce non-negative slack variables, \(\xi_i \geq 0\), one slack variable for each training point. It measures the degree of misclassification of the data point \(x_i\). The value of slack variable is 0 for data points that are on or inside the margin boundary and \(\xi_i = |y_i - (\mathbf{w} \cdot \mathbf{x}_i - b)|\) for other points. The data point that is on decision boundary have \(\mathbf{w} \cdot \mathbf{x}_i - b = 0\), hence these points have \(\xi_i = 1\) and for points which have \(\xi_i \geq 1\) are misclassified.

As data is linearly non seperable we use kernel function to transform to feature space.

The optimization objective then becomes
\[
\begin{align*}
\text{minimize } & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i=1}^{N} \xi_i \\
\text{subject to constraints } & y_i(\mathbf{w}^T\Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \text{ and } \xi_i \geq 0, i = 1, \ldots, N
\end{align*}
\]

where \(C\) is constant, \(\mathbf{w}\) is the vector coefficients of size \(d(d\) denotes number of attributes), \(b\) is a constant. The index \(i\) denotes the \(i\)th training instance. The kernel \(\Phi\) is used to transform data from the input space to the feature space. The reason behind using \(\mathbf{w}^T\mathbf{w}\) instead of just \(\mathbf{w}\) is to make it convex function. If a function becomes convex, it is differentiable and we can derivate it and find the optimal \(\mathbf{w}\).

**Example:** \(w = (x_1, x_2, x_3, \cdots x_n)\) then \(\mathbf{w}^T\mathbf{w}\) becomes \(x_1^2 + x_2^2 + x_3^2 + \cdots + x_n^2\) which is convex.

Kernel function represents a dot product of input data points mapped into the higher dimensional feature space by transformation \(\Phi\). \(K(X_i, X_j) = \Phi(X_i) \cdot \Phi(X_j)\)

Linear Kernel Function: \(K(X_i, X_j) = X_i \cdot X_j\)

Radial Basis Function (or Gaussian Kernel): \(K(X_i, X_j) = exp(-\gamma |X_i - X_j|^2)\)

This algorithm is applied to CS101 training set for SVM to learn, and then we predict the grades of student from test set. The accuracy achieved is shown in table 5.3.
### Table 5.3: Accuracy using SVM Classifier

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy(Single Variable)</th>
<th>Approx Accuracy(Single Variable)</th>
<th>Accurate Accuracy(Multi Variable)</th>
<th>Approx Accuracy(Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMLinear</td>
<td>64.44</td>
<td>98.59</td>
<td>64.79</td>
<td>98.94</td>
</tr>
<tr>
<td>SVMRbf</td>
<td>61.8</td>
<td>97.89</td>
<td>66.37</td>
<td>98.42</td>
</tr>
</tbody>
</table>

Figure 5.10: Accuracy using Linear Kernel Function

Figure 5.11: Accuracy using Gaussian Kernel Function
5.7 Nearest Neighbour Algorithm

Nearest neighbour algorithm is a non-parametric method for classifying objects based on closest training examples in the feature space. An object is assigned to the class of its nearest neighbour. Objects in the training set form the neighbours as the correct classification is known for these objects. The training examples are vectors in a multidimensional feature space, each with a class label. In training phase, the algorithm stores the feature vectors and class labels of the training samples. An unlabeled vector i.e, test point is classified by assigning the label which is closest to that point in the training examples. Here we are considering euclidean distance as distance metric.

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy(Single Variable)</th>
<th>Approx. Accuracy(Single Variable)</th>
<th>Accurate Accuracy(Multi Variable)</th>
<th>Approx. Accuracy(Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbour</td>
<td>66.37</td>
<td>97.89</td>
<td>66.02</td>
<td>98.59</td>
</tr>
</tbody>
</table>

Table 5.4: Accuracy using Nearest Neighbour Classifier

![Figure 5.12: Example for Nearest Neighbour](image)

In Figure 5.12, there are blue and red points (i.e, 2 classes). A test point (green point) is classified as red class as it is nearer to red point than green point. This algorithm is applied to CS101 training set for the classifier to learn, and then we predict the grades of student from test set. The accuracy achieved is shown in table 5.4. The reason behind using nearest neighbour classifier is those people who are getting similar grades, those marks will be nearer.

The drawback of this algorithm is mis-classification occurs when the class distribution is skewed. As distribution of grades is not uniform, some grades like BB and BC dominates. That is, examples of a more frequent class dominates the prediction of the new example, because they tend to be common among the nearest neighbours due to their large number. So we reduced the data set by replacing a cluster of similar grades, regardless of their density in the original training data with single point which is its cluster center. Nearest Neighbour is then applied to the reduced data set. The values of cluster
centers of adjacent grades are very close to each other. The difference between values of each attribute of adjacent cluster centers (i.e., between adjacent classes) is just 1 or 2 marks difference. The accuracy has slightly increased as mis-classification did not reduce much because all the points are close together. The accuracy achieved is shown in table 5.5

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy (Single Variable)</th>
<th>Approx Accuracy (Single Variable)</th>
<th>Accurate Accuracy (Multi Variable)</th>
<th>Approx Accuracy (Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbour Cluster</td>
<td>66.55</td>
<td>98.77</td>
<td>63.56</td>
<td>98.94</td>
</tr>
</tbody>
</table>

Table 5.5: Accuracy using Nearest Neighbour Classifier with Cluster
5.8 Linear Discriminant Analysis Classifier (LDAC)

Linear discriminant analysis (LDA) is used to find a linear combination of features which separates two or more classes of objects. The resulting combination obtained is a linear classifier.

5.8.1 LDA for two classes

Consider a set of observations \( \vec{x} \) with known class label “y”. These samples forms the training set. The classification problem is then to find a good predictor for the class “y” given only an observation \( \vec{x} \)

LDAC solves the problem by assuming that the conditional probability density functions \( p(\vec{x}|y=0) \) and \( p(\vec{x}|y=1) \) are both normally distributed with mean and covariance parameters \( (\vec{\mu}_0, \Sigma_{y=0}) \) and \( (\vec{\mu}_1, \Sigma_{y=1}) \) respectively. Covariance is a \( d \times d \) matrix where \( d \) refers to number of attributes. LDAC also assumes covariance of both classes are same.

\[
P(\vec{x}|y=0) \sim N(\vec{x}, \vec{\mu}_0, \Sigma)
\]

where \( \vec{\mu}_0 \) is a vector containing means of \( d \)-attributes, taking those samples whose target label is 0.

\[
P(\vec{x}|y=1) \sim N(\vec{x}, \vec{\mu}_1, \Sigma)
\]

where \( \vec{\mu}_1 \) is a vector containing means of \( d \)-attributes, taking those samples whose target label is 1.

To classify any new observation \( \vec{x} \), we assign label \( y=0 \) for this observation if

\[
P(y = 0|\vec{x}) > P(y = 1|\vec{x})
\]

\[
= (P(y = 0, \vec{x})|P(\vec{x})) > (P(y = 1, \vec{x})|P(\vec{x}))
\]

\[
= (P(y = 0)P(\vec{x}|y = 0)) > (P(y = 1)P(\vec{x}|y = 1))
\]

Applying logarithm both sides

\[
= \log(P(y = 0)P(\vec{x}|y = 0)) > \log(P(y = 1)P(\vec{x}|y = 1))
\]

\[
= \log(P(y = 0) + \log(P(\vec{x}|y = 0)) > \log(P(y = 1) + \log(P(\vec{x}|y = 1)
\]

\[
= \log(P(y = 0) - \frac{1}{2}(\vec{x} - \vec{\mu}_0)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_0) > \log(P(y = 1) - \frac{1}{2}(\vec{x} - \vec{\mu}_1)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_1)
\]

\[
= \log(P(y = 0) - \frac{1}{2}\vec{x}^T \Sigma^{-1}\vec{x} + \vec{\mu}_0^T \Sigma^{-1}\vec{\mu}_0 > \log(P(y = 1) - \frac{1}{2}\vec{x}^T \Sigma^{-1}\vec{x} + \vec{\mu}_1^T \Sigma^{-1}\vec{\mu}_1 - \frac{1}{2}\vec{\mu}_1^T \Sigma^{-1}\vec{\mu}_1)
\]

Solving the above equation, we get

\[
= (\vec{\mu}_0^T \Sigma^{-1}\vec{x} - \vec{\mu}_1^T \Sigma^{-1}\vec{x}) - \frac{1}{2}\vec{\mu}_0^T \Sigma^{-1}\vec{\mu}_0 - \frac{1}{2}\vec{\mu}_1^T \Sigma^{-1}\vec{\mu}_1 + (\log(P(y = 0) - \log(P(y = 1)) > 0
\]

The above equation is clearly linear in \( x \). Hence we get linear decision boundary. This means that whether input \( \vec{x} \) belongs to class “y” or not is purely a function of linear combination of the known observations. If the above inequality satisfies for \( \vec{x} \), then it belongs to class \( y=0 \) else \( y=1 \)
5.8.2 MultiClassLDA

The above section explained about linear discriminant analysis when there are two classes. The same algorithm can be extended when multiple classes are present. We have partitioned the classes, where the points from one class(say AA) are put in one group, and everything else in the other group(all other grades) and then applied LDA. This will result in C classifiers(i.e, C linear decision boundaries, where C is total number of classes, here C=10 as 10 grades are present) whose results are combined. This algorithm is applied to CS101 training set for LDA to learn, and then we predict the grades of student from test set. The accuracy achieved is shown in table 5.6

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy(Single Variable)</th>
<th>Approx Accuracy(Single Variable)</th>
<th>Accurate Accuracy(Multi Variable)</th>
<th>Approx Accuracy(Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAC</td>
<td>63.73</td>
<td>98.06</td>
<td>64.26</td>
<td>98.59</td>
</tr>
</tbody>
</table>

Table 5.6: Accuracy using LDA Classifier

![Figure 5.15: Accuracy using LDA](image)

5.9 Combination of classifiers

Till now, we have used only single classifiers like decision tree, nearest neighbour, linear discriminant analysis, support vector machines for prediction of grades. We have used many classifiers because it is not possible to come up with a single classifier that can give good results. The optimal classifier is dependent on problem domain. In this section, we used combination of classifiers to predict the target variable(grade) rather than single classifier. There are different ways of combining multiple classifiers(CMC).

**Method1:** Out of many classifiers, we choose that classifier which has least error rate on given dataset. It has better performance than individual classifier. Out of the many
classifiers we have used, Nearest Neighbour algorithm with Clustering similar classes (refer Table 5.9) has given good accuracy i.e least error rate.

**Method2:** Each classifier votes to what class (grade) student belongs to. The class getting the maximum votes from the individual classifiers is assigned to student. For example, if many classifiers predict that the student fails, then we assign “fail” label to student.

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy (Single Variable)</th>
<th>Approx Accuracy (Single Variable)</th>
<th>Accurate Accuracy (Multi Variable)</th>
<th>Approx Accuracy (Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifiers Combination</td>
<td>63.38</td>
<td>97.89</td>
<td>65.49</td>
<td>98.59</td>
</tr>
</tbody>
</table>

Table 5.7: Accuracy using Combination of Classifiers (Method2)

![Figure 5.16: Accuracy using Classifiers Combination](image)

We tried various ways to improve accuracy, but we always used to get 63 to 67% accuracy only. The reason behind it is, when number of target classes is 11, then it is considered to be difficult classification problem. In this scenario, even best methods achieve around 40% errors on test data [18]. Hence we have achieved good accuracy using the classifiers in solving the problem.

The only way to improve accuracy is to reduce the number of class labels. We can group the students according to their final grades in several ways.

**Example:** We grouped them into four classes, “high” representing grades AB, AA, AP, “middle” representing grades CC, BC, BB, “low” representing grades DD, CD, and “fail” those got FR and FF. Table 5.8 shows the accuracy using Nearest Neighbour classifier when the target grades are grouped as above. As the target classes are reduced, the accuracy is improved to 89% as compared to 66% when there are 10 class labels. It clearly depicts that accuracy depends on number of target classes.

47
Table 5.8: Accuracy using Nearest Neighbour Classifier when Labels Reduced

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accurate Accuracy(Single Variable)</th>
<th>Approx Accuracy(Single Variable)</th>
<th>Accurate Accuracy(Multi Variable)</th>
<th>Approx Accuracy(Multi Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>65.32</td>
<td>98.24</td>
<td>65.14</td>
<td>98.42</td>
</tr>
<tr>
<td>SVMLinear</td>
<td>64.44</td>
<td>98.59</td>
<td>64.79</td>
<td>98.94</td>
</tr>
<tr>
<td>SVMRbf</td>
<td>61.8</td>
<td>97.89</td>
<td>66.37</td>
<td>98.42</td>
</tr>
<tr>
<td>Nearest Neighbour</td>
<td>66.37</td>
<td>98.89</td>
<td>66.02</td>
<td>98.59</td>
</tr>
<tr>
<td>Nearest Neighbour Cluster</td>
<td>66.55</td>
<td>98.77</td>
<td>63.56</td>
<td>98.94</td>
</tr>
<tr>
<td>Mac</td>
<td>63.73</td>
<td>98.06</td>
<td>64.26</td>
<td>98.59</td>
</tr>
<tr>
<td>Classifiers Combination</td>
<td>63.38</td>
<td>97.89</td>
<td>65.40</td>
<td>98.59</td>
</tr>
</tbody>
</table>

Table 5.9: Comparison Among Different Classifiers

5.10 Comparison Among Different Classifiers

Table 5.9 compares different classifiers performance in predicting final grades when end-semester marks are predicted using linear model in single variable and when predicted using linear model in multiple variables.

In this chapter, we have discussed about various classification algorithms. For these algorithms to be applied, we need to have training data for which target labels is known. These type of problems come under category of machine learning. In the next chapter, we discuss how in an online course data mining can be used to extract patterns which are useful to study the behaviour of students.
Chapter 6

Datamining and student e-learning profile

Educational data mining is used to study the data available in the educational field and bring out the hidden knowledge from it. Instructors can first receive a detailed record of a learners behavior and then data mining algorithms can be employed to discover patterns to characterize learners. By observing how learners behave during their online self-study, we can understand learners learning style and can then make suggestions to learners. So we can understand student behaviour much better if the course is offered online.

6.1 Sequential Pattern Analysis

A sequence is an ordering of events and each event in the ordered list is called item. A sequence x is a subsequence of another sequence y, if x can be formed from y by leaving out some events without disturbing the relative positions of the remaining events. For example, if x = < C, E, D, B > and y = < C, A, E, D, E, G, C, E, B, G >, then x is a subsequence of y. Given a set of sequences S, the support of a sequence x in S is the number of sequences containing x, denoted as support(x) . Given a percentage min_sup as the minimum support threshold, a sequence x is said to be sequential pattern observed in S if support(x) >= min_sup

Example: Figure 6.1 shows a Sequence Database which contains items A, B, C, D, E, F and say it has min_sup=3. S1, S2, S4, S5 contain subsequence s = <A, D, B, D>. So s is a sequential pattern of length 4. <A, C> is not a sequential pattern as it is supported only by two sequences S1 and S4. These sequential patterns are useful to evaluate learners activities, understand his behaviour and accordingly we can provide customized resources.

To profile learners based on their learning behavior, student actions are logged by the system to extract patterns from it. The learning logs involve a complex series of low-level events spaced along a time dimension. Through a log parsing mechanism, sequence of
temporally ordered learner actions are generated. These sequences of learner actions are then fed into the sequential mining algorithm to discover patterns across the learning logs. With these patterns we can study student learning style and we can evaluate him.

**Example:** Suppose for each video lecture, let us say there is an assignment. If the student doesn’t get good marks in assignment then assume the e-learning system gives personalized references for the student to study and then go to next lesson. Then through this pattern analysis we can know whether student is reading those customized references and going to next lesson or not. As the data is logged, we can know detailed actions done by the student like how much effort he has put into subject etc.

In an online course, we can use many features in predicting their final grade based on features extracted from logged data.

Some of the features that can be used for classification are

1. Total number of correct answers.
2. Getting the problem right on the first try vs. those with high number of tries. A student who gets all correct answers need not be in successful group because they might take an average of 4 tries per problem. This feature helps to identify such people.
3. Total number of tries for doing homework.
4. Total time taken from the first attempt to getting correct solution. Time at which student submitted the problem relative to due date.
5. Total time spent on the problem irrespective of getting the correct answer or not.
6. Reading the given material before attempting homework vs. attempting the homework first and then reading it.
7. Submitting many attempts in a short amount of time without reading material in between, vs. those giving it one try and reading up.
8. Giving up problem vs. students who are trying till the deadline.

With these attributes, we can build a model in the same way as discussed earlier and predict the final grade of student.
Chapter 7

Theoretical Model for Personalized Recommendation System

No two students are identical, they learn at different rates, come from different educational backgrounds, and have different intellectual capabilities, different modes of learning. So we need to design a real-time recommendation system which captures the characteristics of each student. We need to ensure that every student progresses through the course material so that he maximizes his learning. But it is an immensely complex task.

The biggest drawback in present education system is no personal care on students. Professor teaches lessons, gives the assignments, conducts the exams and the course gets completed. There are no measures taken for improvement of the student. A student who is intelligent, can read and think on his own. But many students fall into category of average or weak. For such students some personal attention, some personalized suggestions is needed. Personal attention in an offline course is difficult as professor personally cannot check each student strengths and weaknesses in each topic. But it is possible in an online course with the help of technology. Giving personalized suggestions based on student characteristics is the hot research topic going on. Andrew Ng(Stanford professor, founder of Coursera) and his team is trying to implement this recommendation system. Till date, no e-learning system has implemented it.

7.1 Theoretical model

Before extracting useful information from the data for customization, lot of pre-processing operations need to be done on data. Many students register the course and they do not start the course work also. For example, in Coursera many students just register the course so that they can download video lectures, assignments, quizzes later. Such student data should be removed before applying any algorithm on data so that we can get correct inferences about the learning behaviour of the students.

To give customized suggestions to student we need to know student mastery level on the topic. We cannot give customized references just based on total marks secured in the topic. For example, in an exam there are 4 questions, out of which 2 are tough
and 2 are easy, both of each are 5 marks. Student A answered 2 tough questions and
got 5 marks. Student B answered 2 easy questions and got 5 marks. Student A has
mastered the subject than B. But if we look at total marks and provide customized
suggestions then we are suggesting him wrongly. So we need to give different internal
score for the student which says about his mastery level in the topic. Test questions do
not have uniform characteristics. So we need to model student ability using question-
level performance instead of aggregate-test-level performance. Instead of assuming that
all questions contribute equally to understand students abilities, we need to extract the
information each question provides about a student. The probability of a correct response
to a test question is a mathematical function of parameters such as a persons abilities
and question characteristics (such as difficulty, guessability, and specific to topic). This
model help us to better understand how a students performance on tests relates to his
ability. The developed system should continuously learn about the user and update its
parameters about the user so that we can suggest him properly and make him to achieve
his learning objectives.

We need to constantly learn from student performance data. Upon completion of a
given activity, the system should direct the student to the next activity. For example,
when a student struggles with a particular set of questions, our system should know
where that particular students weaknesses lie in relation to the concepts assessed by those
questions. Our system should then deliver content to increase the students proficiency on
those concepts. Our system should track the information about what student knows and
how he learns best. The student profile gets better if the student does many courses on our
platform. It then provides good insight how the student understands the material, what
they truly grasp and don’t grasp, their misunderstandings and misconceptions. When we
learn about student strength and weakness we can help them in better way. Using the
dashboard, we can see how students are performing in individual subject areas, which
segments of material are the most and least challenging for students.

**Theoretical Example:** Figure 7.1 is a dependency graph, whose nodes represent
concepts and edges represent relationship between the nodes(concepts). Each node is a
function of its parents. The entire course module should be converted into dependency
graph. Figure 7.1 is a sample graph build by considering few concepts in C language. As
discussed, personalized suggestions cannot be done just based on total marks. We need
to consider many attributes or parameters for each question like difficulty level of it, time
taken to answer it, number of attempts to get correct answer for it, marks he secured
for it, probability of guessing the answer etc. Difficulty level of question is a categorical
attribute having values easy, medium, tough. The attribute Time taken to answer takes
continuous values. Number of attempts to get correct answer, marks secured for each
question are also categorical attributes. Attribute Probability of guessing the answer
takes continuous values but it can be converted to categorical attribute having values yes
guessed, dont know, not guessed for attribute value ranging between 0 to 0.3, 0.4 to 0.6,
0.7 to 1 respectively.

Example:If less intelligent person answers a difficult question then mostly it might be
guess work. This attribute captures these scenarios. Using these we can calculate internal score of student for the entire topic (there are many questions in the topic) which determines his mastery level in the topic. For simplicity, we have taken just few attributes and drawn the directed acyclic graph and calculated the internal score.

Figure 7.1: Dependency Graph for C Language Course

Figure 7.2: DAG for Internal Score
Figure 7.2 is directed acyclic graph (DAG) whose nodes are random variables and edges represent direct influence of one node on another. This DAG contains information about student caliber in overall topic (i.e., random variables in nodes represent overall topic, not individual questions) and is present at each node in dependency graph (Fig 7.1). The topic difficulty and student intelligence in prior topics are independent and hence no edge between them. The answers answered by the student can be guessed or he might know the answers. In reality, it depends on topic difficulty and student intelligence in prior topics, but for simplicity, we have taken only intelligence level in prior topics. Marks secured by the student in the topic depends on both difficulty in topic and intelligence level of student in prior topics. In our model, Internal score given to students depends on answers guessed or not, marks secured in the topic, difficulty of topic, intelligence level of student in prior topics.

The attribute Difficulty of Topic refers to overall difficulty of the topic. Each question in the exam conducted in a particular topic can be classified as easy, medium, tough. From each question we take its difficulty and calculate overall difficulty of the topic. The attribute intelligence level in prior topics is a score given to a person in a topic and varies from topic to topic. After each topic this attribute is updated. This attribute denotes cumulated intelligence in the topics covered till then. The reason behind the updation of this attribute is that to answer a question in a topic, the student must know the topics present in parent nodes also (clearly shown the dependencies in dependency graph). For example, to traverse an array, student must know using for loop which is its parent node in dependency graph. This attribute should also be updated if the student reads the personalized reference material given to him and solved the problems. The attribute Guessed answer refers to overall probability of guessing the answer for the student for that particular topic. This attribute is derived from probability of guessing attribute which is present for each question. The attribute Marks Secured denotes overall marks in that topic. In this model, even if two persons got same marks, Internal score will be high for those students who answered tough questions than that of person who answered easy.

Using this graph we calculate internal score of the student in that topic which says about his proficiency or mastery on the topic. Now at each node in dependency graph we have attributes like mastery level on topic, intelligence level of student, difficulty level of topic etc. The attribute intelligence level of student here refers to overall intelligence level not topic level intelligence. This attribute can be obtained from registered information by which we can know his achievements in academics, olympiads, rank of entrance exams like JEE, AIEEE etc. This attribute will be updated after completion of each course so that updated value can be used in further course. If we do not update then we are giving importance to historical information and not considering his present performance. In general, people who topped in entrance tests need not be toppers of the class in engineering. We are accounting this fact by updating above attribute. We have mentioned only few attributes, in real modelling situations many attributes comes into picture. Now classification algorithms can be applied on this data and we can classify him as weak student or average student or intelligent student at topic level and accordingly suggest
him personalized learning resources. By this classification at topic level, we can give him new video tutorials on this topic, personalized materials, personalized problem sets etc. In this way even if weak student passionate in education, can show improvement by this personalized recommendation.

Updation of attribute Intelligence Level in prior topics: From registered information, we give a value for the attribute intelligence level of student which lies between 0 and 1. This attribute value is assigned to attribute Intelligence Level in prior topics when student is doing the first topic in course(say C Declarations). After completing this topic, system gives him internal score for the topic.

Now attribute Intelligence Level in prior topics(after completion of first topic) is updated as follows:

\[
\text{Intelligence Level in prior topics} = (1-\alpha) \ast \text{intelligence level of student} + \alpha \ast \text{Internal score in topic}
\]

where \(\alpha\) is different for different topics and it denotes the weightage that should be given to the topic in determining student intelligence level in that topic. (Here we want student intelligence level in topic. Do not confuse with the attribute internal score in topic which says about his mastery level in the topic) The parameter \(\alpha\) is decided by statisticians using statistical estimates. For example, if the topic is very easy and historical information says that many persons get good marks in that topic, then \(\alpha\) is set to low value because it doesn’t play much role in determining intelligence level in topic. The above equation is updated such that we are taking care of historical information(like his achievements in olympiads etc which is contained in attribute intelligence level of student) and also present level performance(which is contained in attribute Internal score in topic). This new value of attribute Intelligence Level in prior topics is used for next topic(say Decision Control Structures). For further topics it is updated as follows:

\[
\text{Intelligence Level in prior topics} = (1-\alpha) \ast \text{Intelligence Level in prior topics} + \alpha \ast \text{Internal score in topic}
\]

Here \(\text{Intelligence Level in prior topic}\) on RHS contains all cumulated information i.e about historical achievements, his intelligence level in the topics covered so far.

**Calculation of Internal Score for a particular topic:**

Example: We have 5 random variables in DAG shown in Figure 7.2. Let difficulty of topic take values easy, hard which is represented as \(d^0, d^1\) respectively, intelligence level in prior topics take values low intelligence, high intelligence which is represented as \(i^0, i^1\) respectively. Marks secured in the topic depends on both difficulty level in topic, intelligence level in prior topics. Hence probability distribution of marks secured depends on these two attributes. Let the attribute marks secured takes categorical values good marks, average marks, less marks represented by \(s^1, s^2, s^3\) respectively. We have different distribution for each assignment of values of \(i, d\) which is shown in Table 7.4. For example an high intelligent person in an easy topic gets good marks with 0.9 probability, average marks with 0.08 probability, less marks with 0.02 probability which is represented in 3rd row of Table 7.4. The attribute Guessed answer depends only on intelligence level in prior topics(by our model) and it takes values guessed and not guessed which is represented as \(g^0, g^1\). We have different distribution for different values of \(i\) which is shown in Table 7.3.

The attribute Internal score depends only on marks secured in topic and let it takes
values low score and high score which is represented as is0, is1. We have different distribution for different values of s which is shown in Table 7.5. A student who got good marks falls into category of low score with 0.1 probability and falls into category of high score with 0.9 probability which is represented in 1st row of Table 7.5. We have considered only simple model so that we can explain in simple tables. If complicated model is taken then we need to write distribution for each assignment of values of its corresponding dependencies. These tables shown below are not realistic tables. We have assigned meaningful random values to entries in the tables to explain the concept. These tables should be obtained using statistical estimation techniques from raw data.

After writing exam for the topic we have values of all attributes except internal score. Let the attributes have values i1, d0, s2, g1

\[ P(i_1, d_0, s^0, g_1) = P(i_1)P(d_0)P(s^0|i_1, d_0)P(g_1|i_1)P(is^0|s^2) = 0.3 \times 0.6 \times 0.08 \times 0.8 \times 0.4 = 0.004608 \]

\[ P(i_1, d_0, s^2, g_1) = P(i_1)P(d_0)P(s^2|i_1, d_0)P(g_1|i_1)P(is^1|s^2) = 0.3 \times 0.6 \times 0.08 \times 0.8 \times 0.6 = 0.006912 \]

Clearly the probability that student belongs to high internal score has greater value. Hence student is assigned internal score as high.

<table>
<thead>
<tr>
<th>d0</th>
<th>d1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 7.1: Probability Distribution of Difficulty Level

<table>
<thead>
<tr>
<th>i0</th>
<th>i1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 7.2: Probability Distribution of Intelligence Level

<table>
<thead>
<tr>
<th>g0</th>
<th>g1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>0.05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 7.3: Probability Distribution of Guessed given Intelligence
Table 7.4: Probability Distribution of Marks Secured given Difficulty, Intelligence

<table>
<thead>
<tr>
<th>$s^0, d^0$</th>
<th>$s^1$</th>
<th>$s^2$</th>
<th>$s^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i^0, d^0$</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>$i^0, d^1$</td>
<td>0.05</td>
<td>0.25</td>
<td>0.7</td>
</tr>
<tr>
<td>$i^1, d^0$</td>
<td>0.9</td>
<td>0.08</td>
<td>0.02</td>
</tr>
<tr>
<td>$i^1, d^1$</td>
<td>0.5</td>
<td>0.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 7.5: Probability Distribution of Internal Score given Marks Secured

<table>
<thead>
<tr>
<th>$i s^0$</th>
<th>$i s^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$s^2$</td>
<td>0.4</td>
</tr>
<tr>
<td>$s^3$</td>
<td>0.99</td>
</tr>
</tbody>
</table>

If we want to solve the problem of customization then we need to have much complicated models. We need to have realistic data for many offerings of course and update the developed model after each offering. This much complicated is this topic if we look at fine grained level. That’s the reason till now no e-learning platform[15], [17] has developed customization. Many e-learning platforms are re-directing to another video tutorial if the student is struck in solving the problem. This is not customization. This is just giving alternative way of explanation for the concept. Just the basic model which we have developed is so complicated to implement. So we have solved this problem at a coarse grained level i.e at the course level instead of topic level.
Chapter 8

Conclusion And Future Work

By predicting endsem marks, grade of student apriori and classifying them (at course level) into four groups as high, middle, low, fail and providing customized learning materials for these groups definitely improve their performance in endsem and improve their grade.

Different learners have different learning styles. Different people build process and store knowledge in different ways. So, different people will relate to a particular learning resource in different ways. Human instructors can learn which style of presentation suits which learner and adjust their mode of presentation accordingly. Learners have different backgrounds and previous experience, so different learners may need to focus on different material to achieve the same learning objective. But the current e-Learning systems do not allow for diversification and presents the same sequence of learning modules to every user of the system.

The proposed learning system gives learning support based on individual learning characteristic. Different learning proposals are provided to students in feedback according to learners learning rate. A test is provided after completion of each chapter and system gives corresponding learning proposals according to test result. The proposed system is shown in Figure 8.1

Personalized resource recommendation can be applied at many granularities, recommendation at problem level, topic level, course level etc. Suppose if a student is unable to solve a problem, then if he clicks on the help button then we can redirect him to a short segment video lecture (recommendation at problem level). So, after listening this the student can solve the problem. Similarly we can provide personalized learning materials at topic level and course level. Finally, these customized suggestions definitely aim at improvement of student.
Figure 8.1: Structure of Proposed System
Bibliography


[17] www.it.iitb.ac.in/ekshiksha/(referred on 31st Aug 2012)

[18] The Elements of Statistical Learning by Friedman, Hastie, Tibshirani printed in 2001, pg 85