**Table of Contents**

1. Project profile ……………………………………………………………………………………………………2
2. Objective …………………………………………………………………………………………………………..3
3. “Virtual labs” introduction ………………………………………………………………………………..6
4. Problem Statement …………………………………………………………………………………………..8
5. Challenges ………………………………………………………………………………………………………..9
6. Solution …………………………………………………………………………………………………………...11
7. Modules …………………………………………………………………………………………………………..13
8. Phases ……………………………………………………………………………………………………………..13
9. Footprint Gathering ………………………………………………………………………………………...14
10. Student Classification ………………………………………………………………………………………20
11. Markov models……………………………………………………………………………………………….. 21
12. Hidden markov model…………………………………………………………………………………….. 23
    1. Markov chain property…………………………………………………………………….. 23
    2. Examples of Hidden markov model…………………………………………………. 24
    3. 5 Properties of Hidden markov model ……………………………………………..25
    4. Canonical problems in Hidden markov model ……………………………….…29
13. Hidden markov classifier ………………………………………………………………………………….37
14. Analysis ……………………………………………………………………………………………………………43
    1. Category based performance analysis …………………………………………..…44
    2. Time based analysis …………………………………………………………………………46
    3. Sequence analysis ……………………………………………………………………………48
    4. Student classification………………………………………………………………………. 50
15. Evaluation………………………………………………………………………………………………………..52
16. System requirement ………………………………………………………………………………………..53
17. Technical specification ……………………………………………………………………………………..53
18. Future enhancement ……………………………………………………………………………………….58
19. Conclusion ……………………………………………………………………………………………………….59
20. Bibliography …………………………………………………………………………………………………….60
21. **Project Profile**

Project Title         : SMART V-LAB

Project Aim          : Smart E-Learning

Developed Tool  : “SMART V-LAB Student Classifier”

Institute Name   : Department of Computer Science, Gujarat University

Technologies       : PHP, JavaScript, ASP.NET with C#, MySQL

Other Tools        : Apache Tomcat, Internet Information Server (IIS)

Guide                  : Dr.Jyoti Pareek

Developed By    : Rinkesh Patel, Anand Solanki

1. **Objective**

There are two types of learning methods one is classroom based learning where the teacher teach subjects to students in classroom and second is E-learning using online platforms where students learn subjects using audio, video or text based tutorials on web portals.

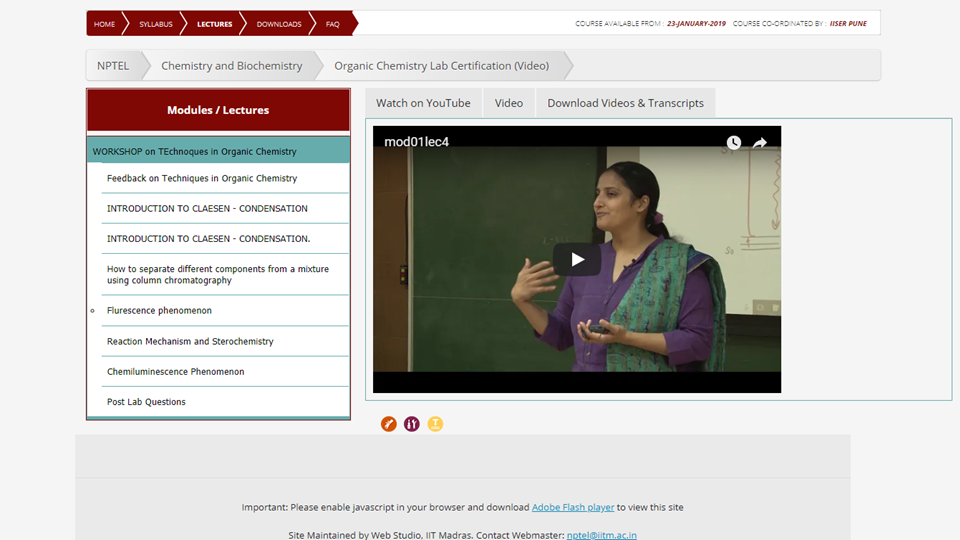
In classroom based learning a teacher can observe the ability of a student like whether a student is fast learner or slow learner, based on that a teacher can recommend different types of learning methods or can choose alternate teaching methods.

In E-learning systems like “Virtual Labs”, students learn their subjects. On such portals to provide such recommendation or to adopt personalized teaching/learning methods, we need to observe the type of student.

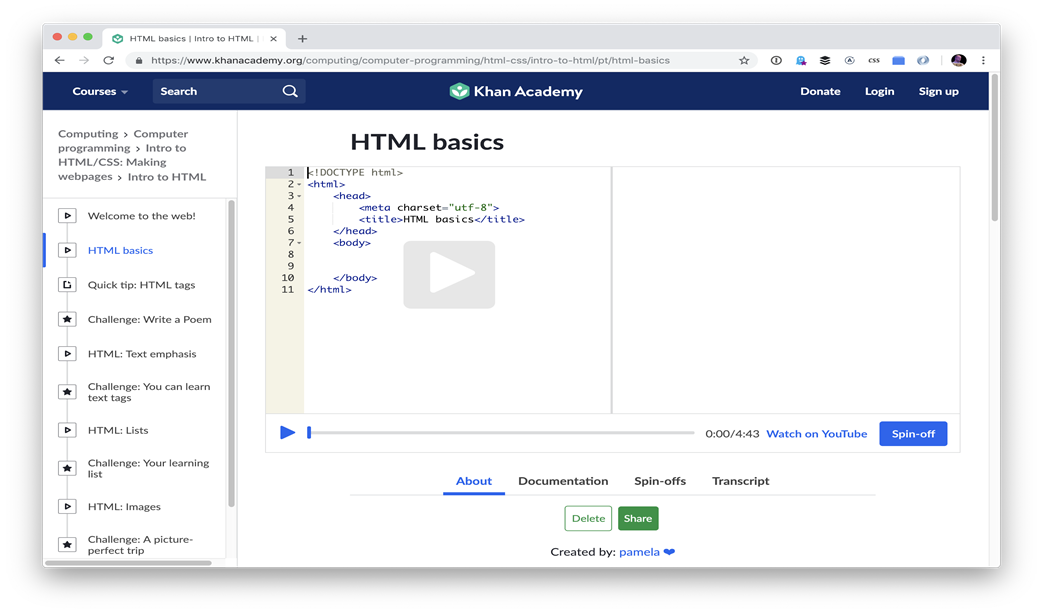
By observing the ability of student, we can provide personalized learning methods.

Examples of E-learning platforms:

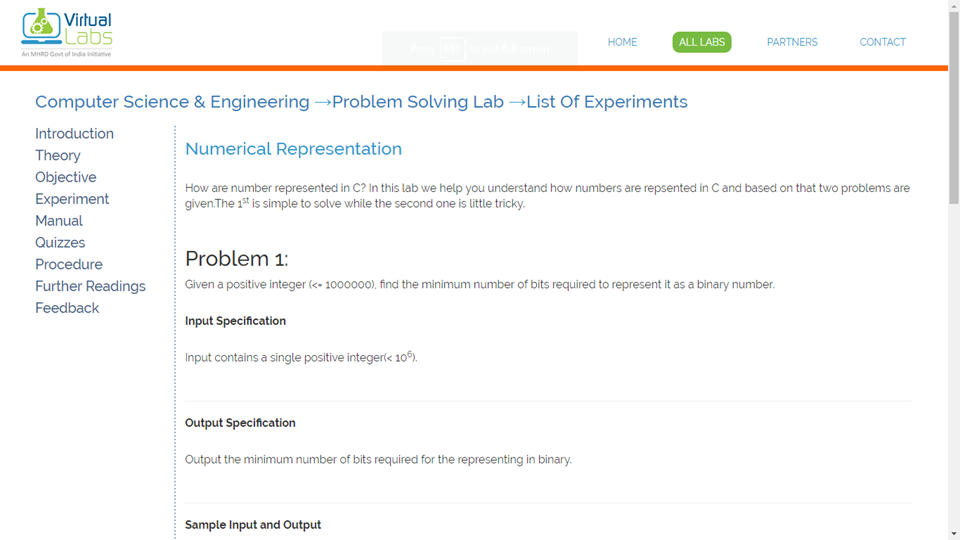
1. <https://nptel.ac.in/>



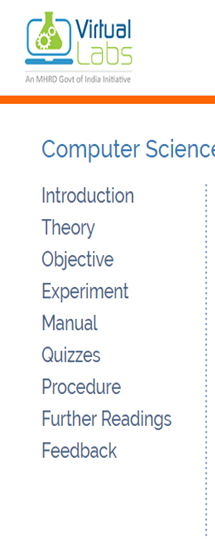
2. www.khanacademy.org



3. http://ps-iiith.vlabs.ac.in/



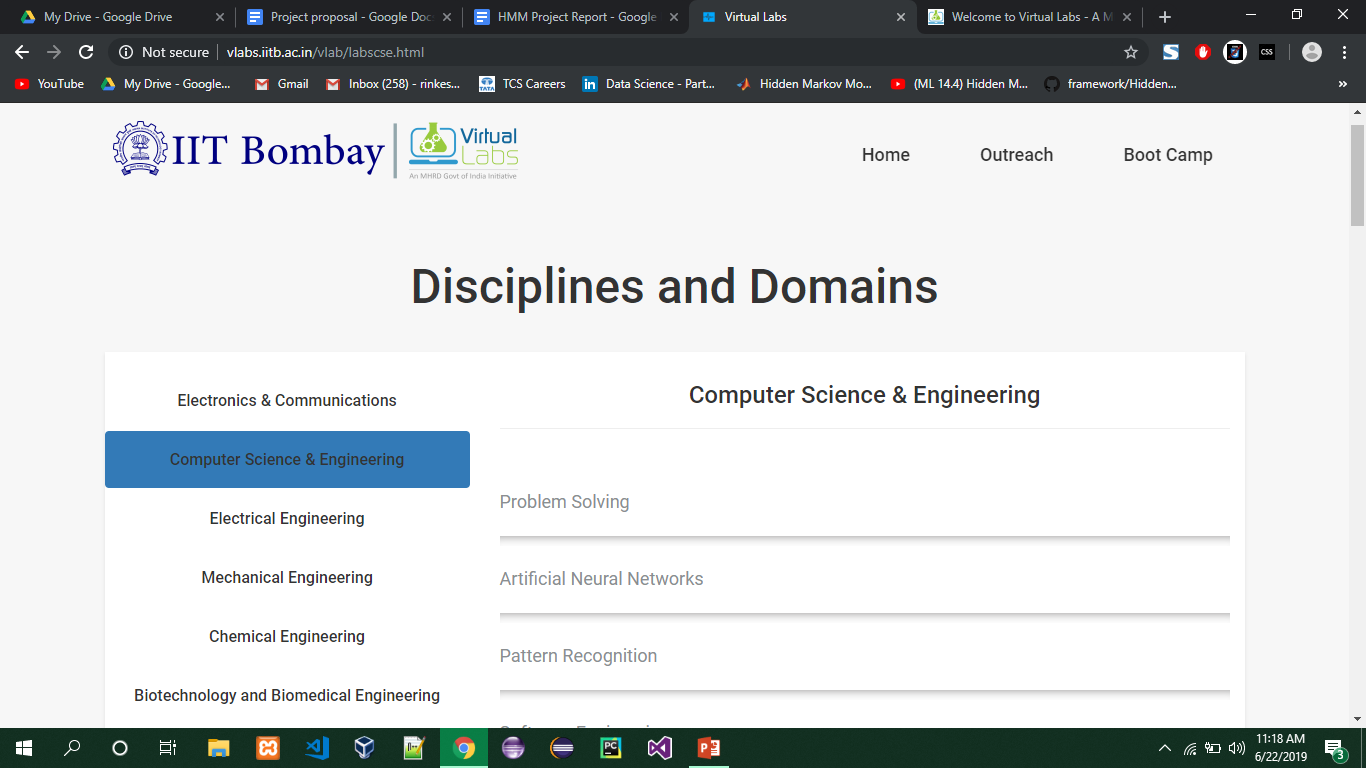
we can clearly see from the above images that all E-learning platforms are consist of different components like introductory part, theory contents, some manuals, audio-video tutorials, experimental coding area where students can perform different types of practicals, assignments and quizzes component where students can enhance their learning.



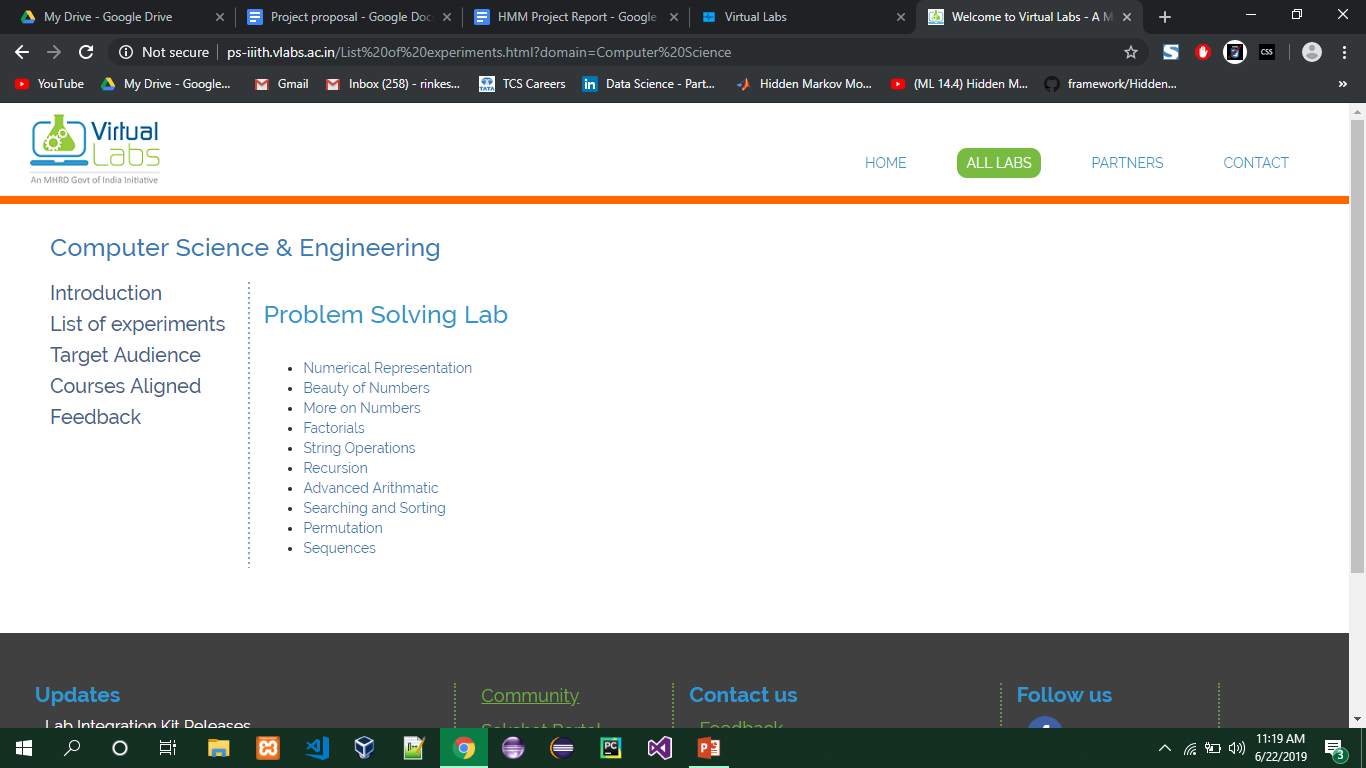
From above images we can see that all E-learning platforms have different components. For example “Virtual LAB” E-learning platform has components like introduction, theory, objective, experiment, manual, quizzes, procedure, further readings, feedback etc.

In order to learn any subject on such platforms, student visits all these components as and when required.

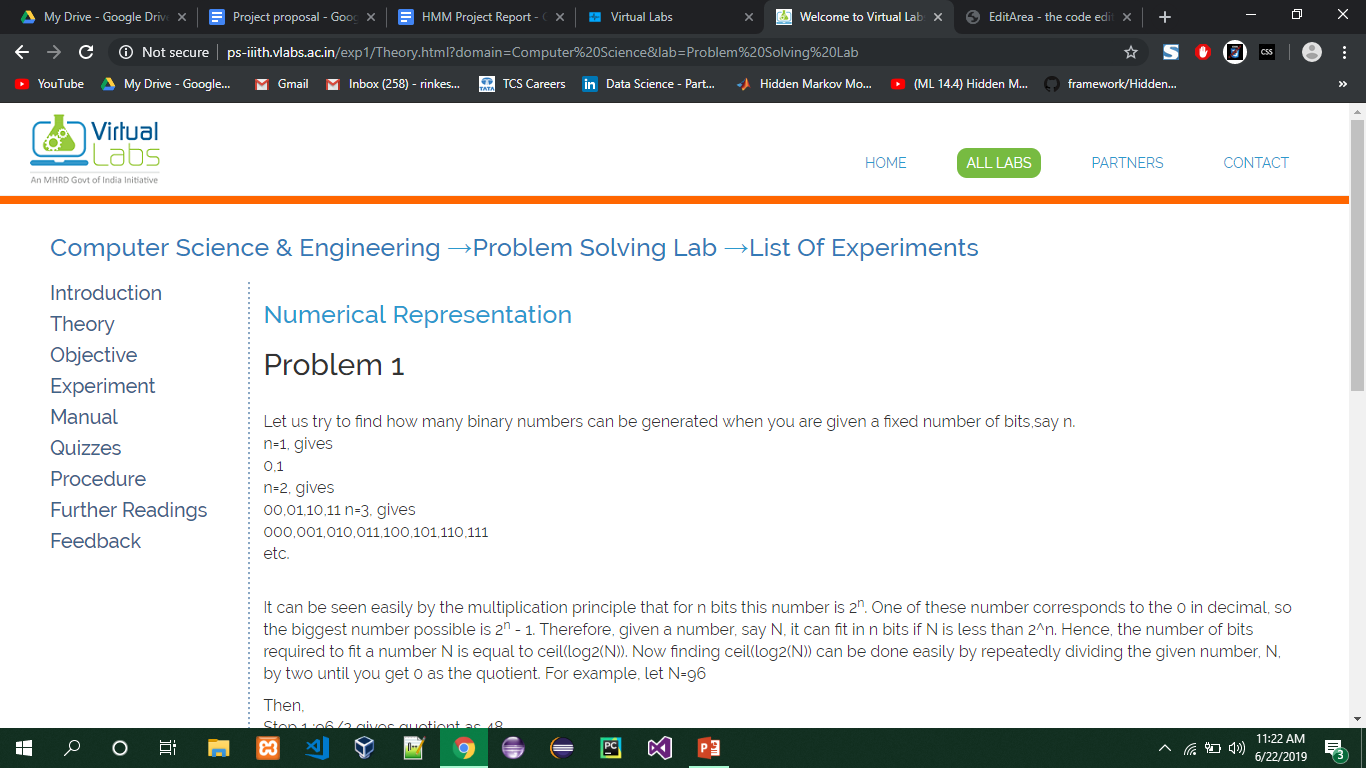
1. **”Virtual Labs” introduction**



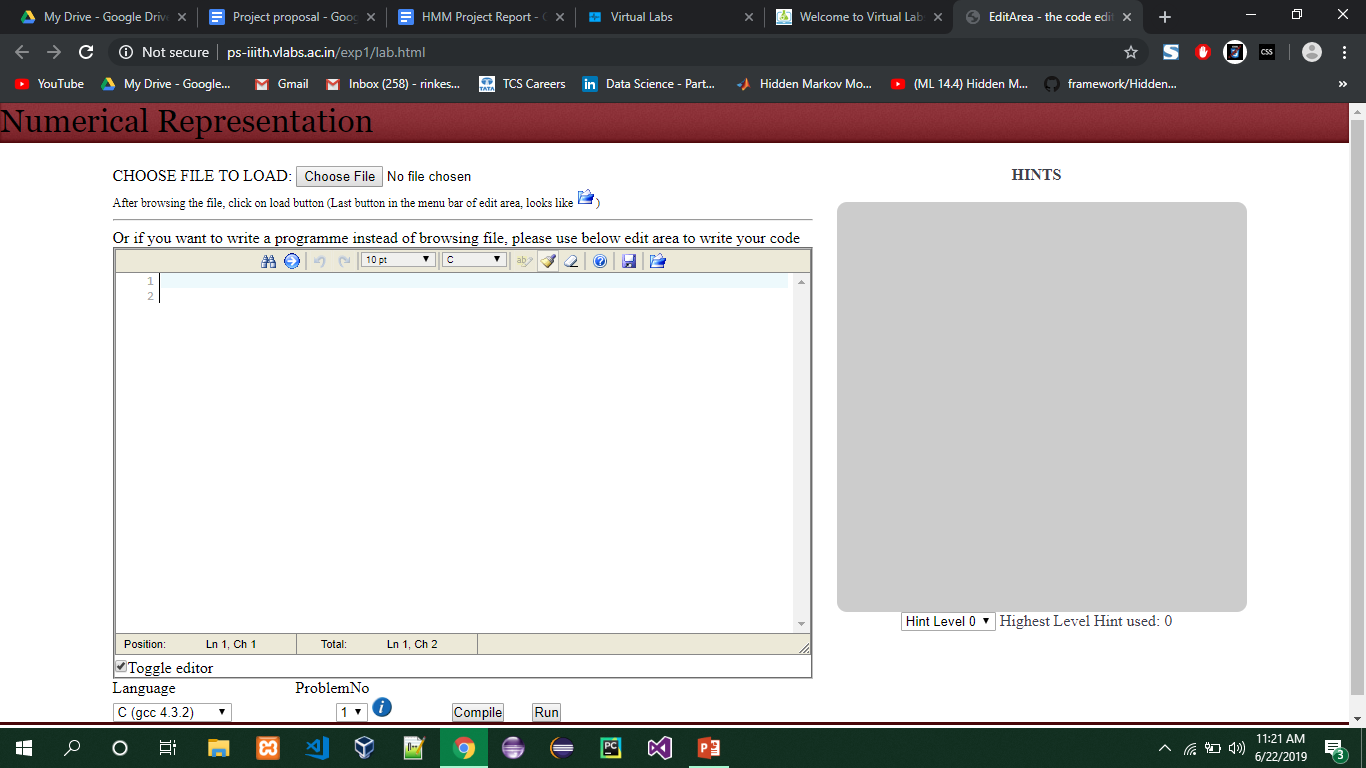
**List of experiments** on Problem solving lab in computer science & Engineering stream.



**Theory** page of problem solving Lab



**Experiment** page of Numerical representation topic



1. **Problem statement**

E-Learning platforms are growing faster than ever. With the help of E-Learning platforms user can learn any topic at anywhere and anytime.

But most E-Learning platforms are not as effective as traditional classroom based learning, because of one major factor.

In classroom based approach a teacher can teach with different methods and different pace to each student according to their grasping capabilities.

This type of smart teaching is not available in most of the E-Learning platforms, Mostly there is a same content for every type of student. This is the decisive factor in classroom based learning vs. E-Learning.

We can make E-Learning smart if we start to observe the student usage patterns and once we are able to identify the student capabilities we can provide recommendations and also whole new teaching model on the platform.

In this project we are using “Virtual Labs” E-learning platform. On the virtual labs students uses different experiments to learn topics. Our aim is to observe the student usage patterns and identify the student’s learning ability.

1. **Challenges**

There were several challenges we faced during the development of this project.

* **No historical data:**

For classification based on the page traversal sequence, we required a large amount of data of known students and sequence followed by them.

But this type of data was not available already. That’s why we have built a footprint recorder to gather page traversal sequence.

Footprint gathering in detail is discussed in earlier sections.

* **No access to live portal codebase:**

In order to record data we needed to embed our footprint gathering program into the V-LAB codebase.

But, we did not have access to the codebase.

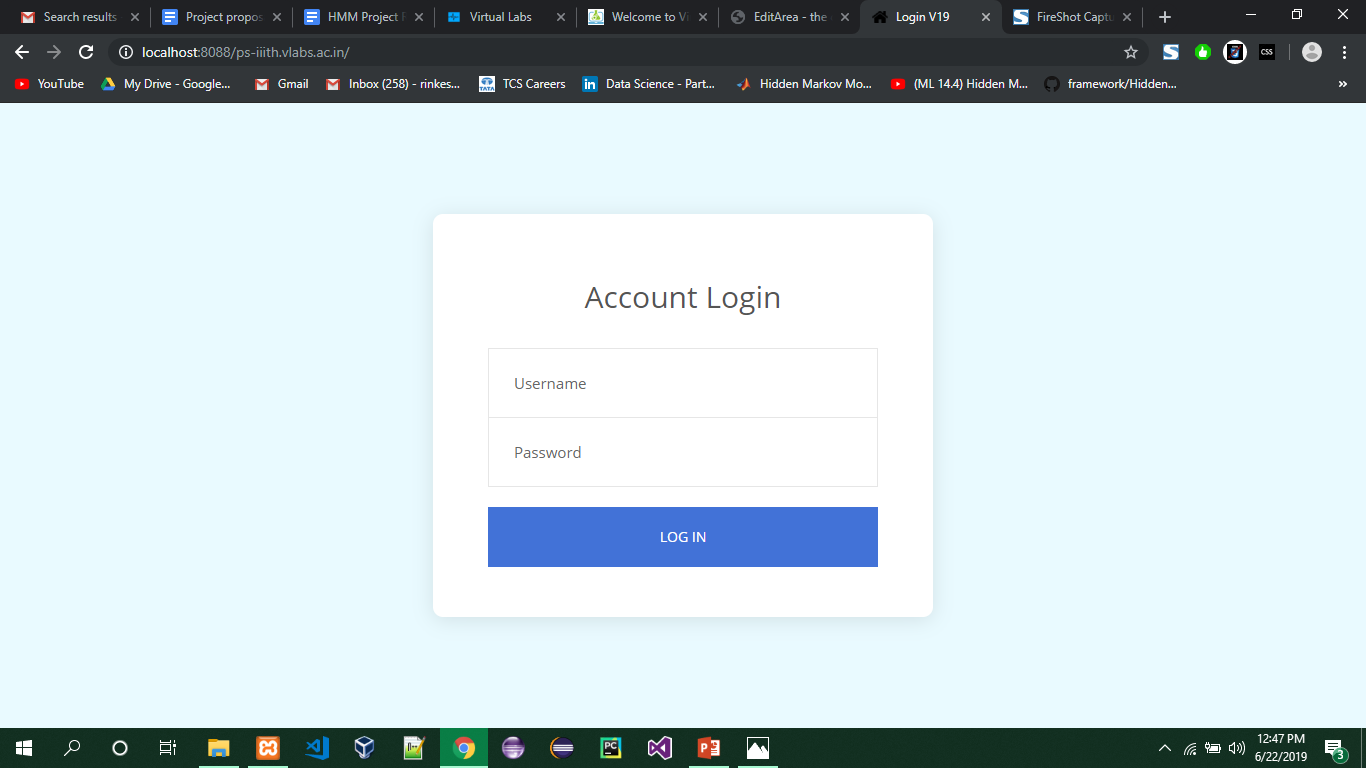
To overcome this problem we managed to replicate V-LAB platform on our local server and then embedded the footprint gathering program.

* **Dynamic portal implementation with PHP and MySQL**

To record footprint we created dynamic platform using PHP that store user footprint in MySQL database.

* **“Virtual LAB” is free to use without LOGIN authentication**

As “Virtual LAB” is free and open system without login authentication, we have embedded “**LOGIN**” module, in order to record single user specific session.

****

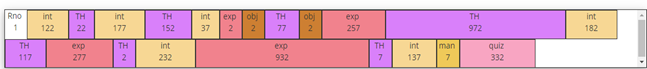
* **Heartbeat system to record user session.**

As our platform is web client server based system, once a page is loaded in client browser there will not be any track of time spent on page if user directly closes the window or somehow window closed due to a power failure.

To overcome this problem we are using “**Heartbeat System**”. In this system a JavaScript code invokes database query every 5 seconds, and these query updates database record that a particular user is still using a particular page.

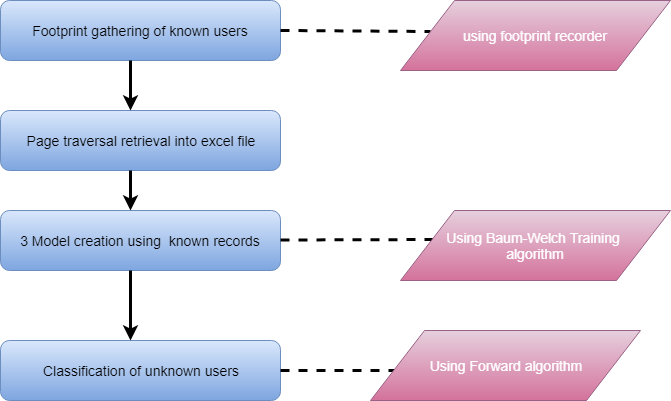
1. **Solution**
   1. In order to observe students ability, we need to capture user footprints on E-learning platform, like in which sequence of pages user traverse on platform, how much time he/she spent on each page.

For example, in below image it shows the page traversal sequence followed by the student on “Virtual lab” platform and also the time spent in seconds on each page. Here, student opens introduction page and uses it for 122 seconds and then traverses to theory page and reads theory for 22 seconds and again traverses the introduction page and so on.



* 1. By matching this sequence to predefined sequences of different types of students, we can classify the current student like fast learner or slow learner.

To define such sequences we are using Hidden Markov Model.



1. **Modules**

* Login
* Data gathering
* Sequence generation
* Model creation and training, classifying user
* Lab analysis tool

1. **Phases**
   * Footprint Gathering
   * Student Classification
2. **Footprint Gathering**

Requirement

For student classification we need to create predefined sequences of different types of students. For that we need training set of records. The requirement for footprint gathering phase is to record this training set data.

In this experiment to record student’s page traversal sequences, we are using “Virtual LAB” E-learning platform.

Implementation

As “Virtual Labs” is a live portal, we have created dynamic platform using PHP, JavaScript and MySQL.

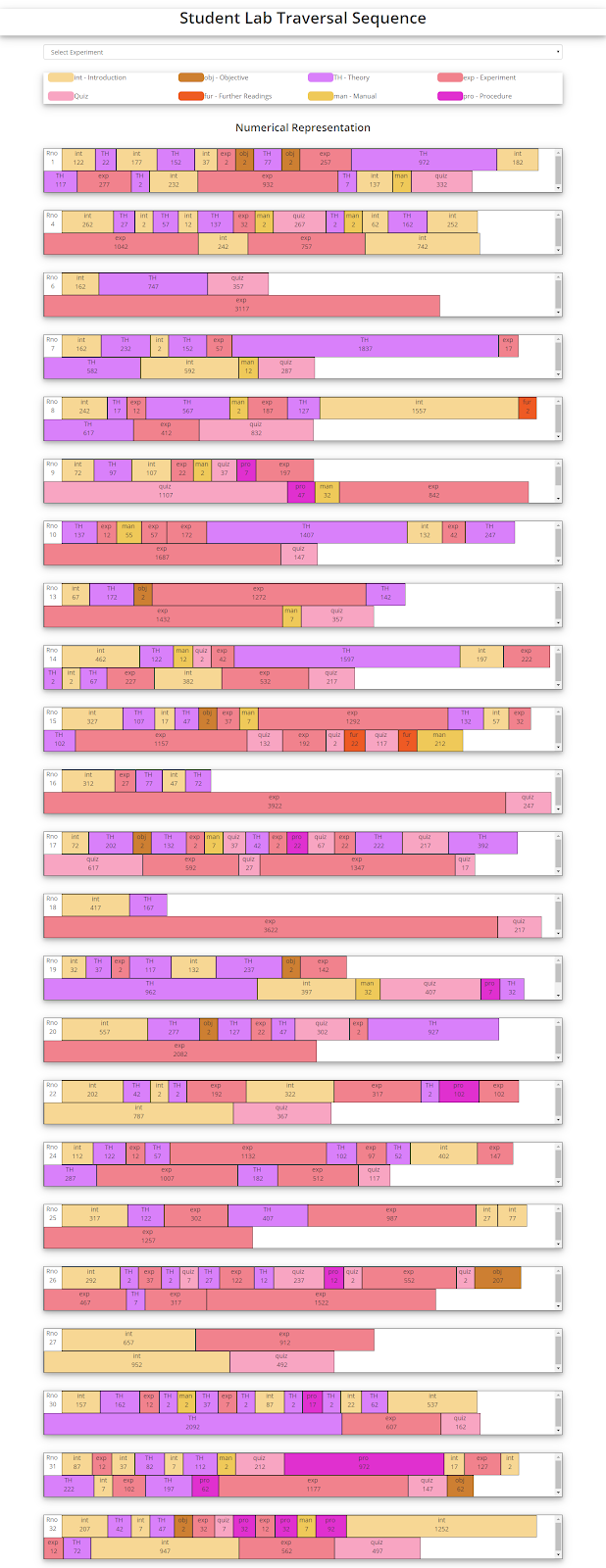
This platform can record students footprint in the form of page traversal sequences and time spent in seconds on each page while they use the platform to learn any topic.

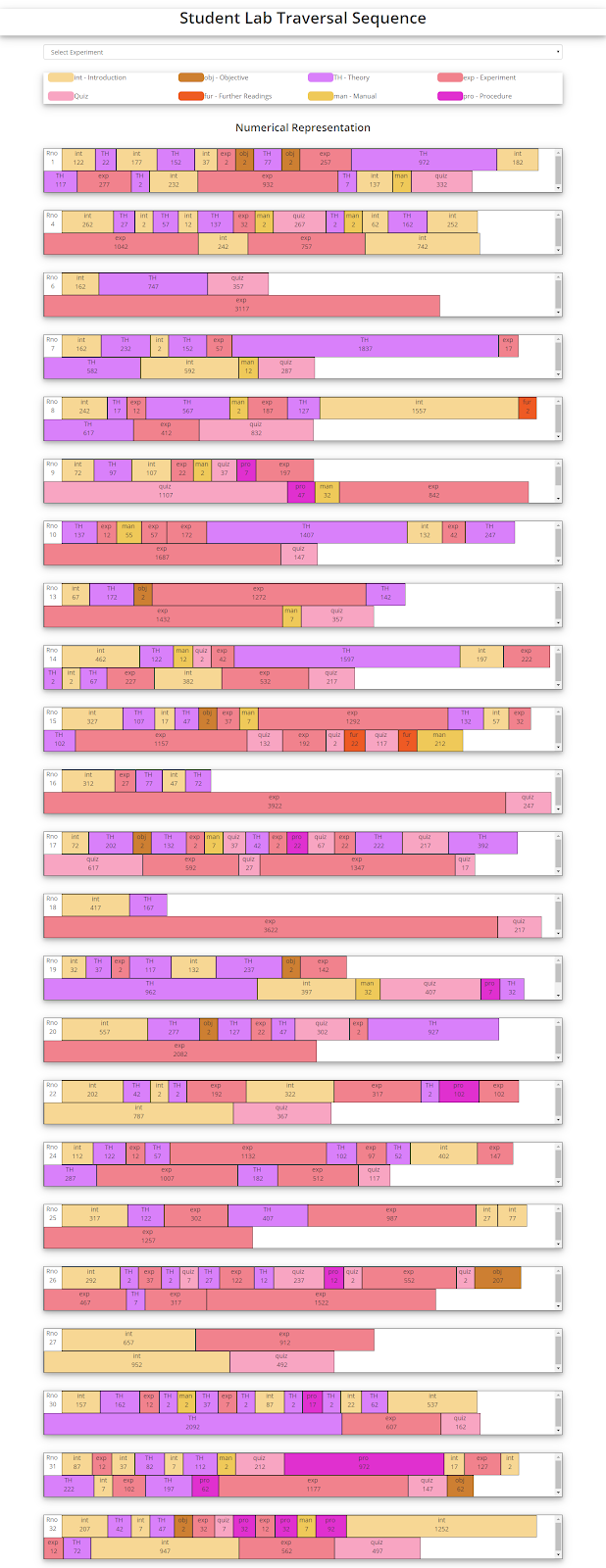
Experiments

In order to record genuine data records, we conducted experiments in department’s labs where we asked students of MCA to use “Virtual LAB” platform.

As a result of an experiment we have recorded students lab traversal sequences with time spent on each page.

The time relevant students page traversal sequences are shown in the image on the next page.

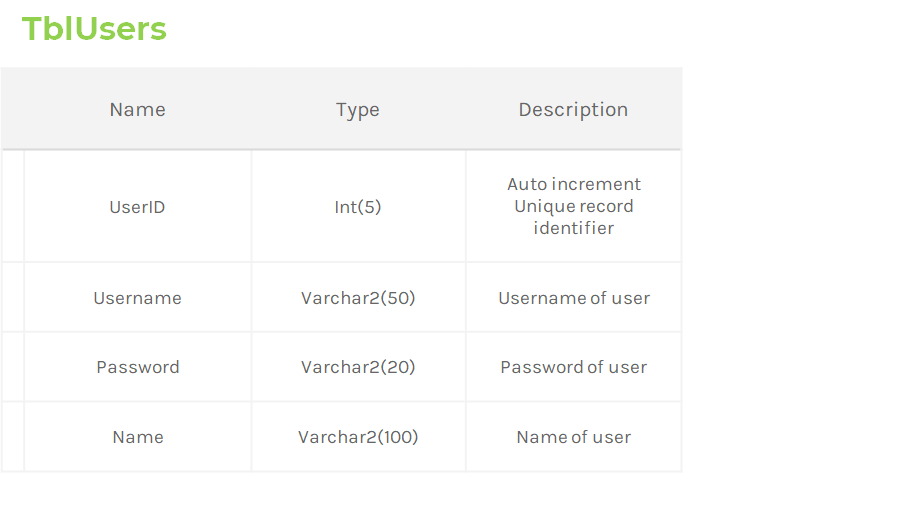




In footprint gathering phase, to store sequences we are using MySQL relational database. In which we are using the following tables:

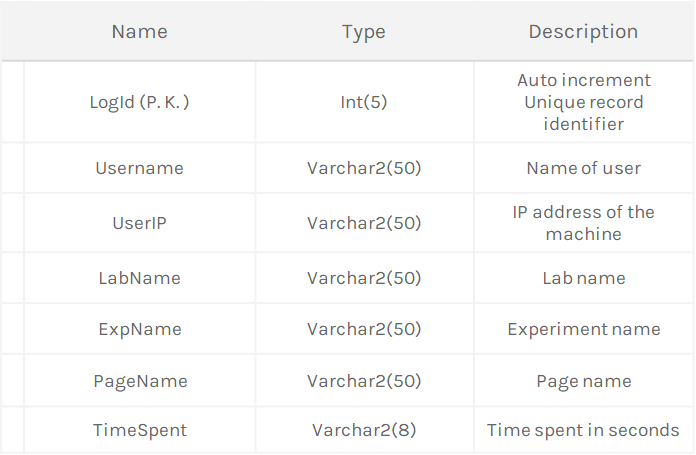
**TblUsers**

This table is used to store the user’s information like username, password, and full name.



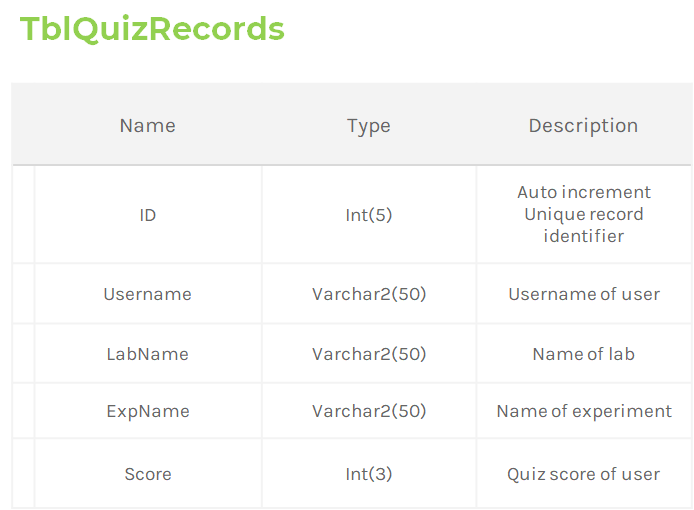
**TblPageUseLog**

This table is used to store students lab traversal sequences with time spent on each page.



**TblQuizRecords**

This table is used to store quiz score recorded when students attempt a quiz on any experiments on lab.



1. **Student Classification**

After recording the student’s sequences (usage patterns) using footprint gathering stage, we can define sequences of different types of students like sequence model for Excellent performer, Good performer and Below average performer. The sequence collected can be tagged on the basis of their academic record. The tagged sequences can be used to train the model and expected sequence for all the three models namely excellent learner, average learner and below average learner is generated.

These sequences of different types of students can be used to classify new student (whose learning capability is not known) like whether a student is Excellent learner, Good learner and Below average learner.

In order to define such sequences for classification, we are using Hidden Markov Model.

1. **Markov Models**

In probability theory, a Markov model is a stochastic model used to model randomly changing systems.

It is assumed that future states depend only on the current state, not on events that occurred before it (that is, it assumes the Markov property). Generally, this assumption enables reasoning and computation with the model that would otherwise be intractable.

* Set of states: {S1,S2,...Sn}
* Process moves from one state to another generating a sequence of states :  {Si1,Si2,..,Sik}
* **Markov chain property:**

Probability of each subsequent state depends only on what was the previous state:

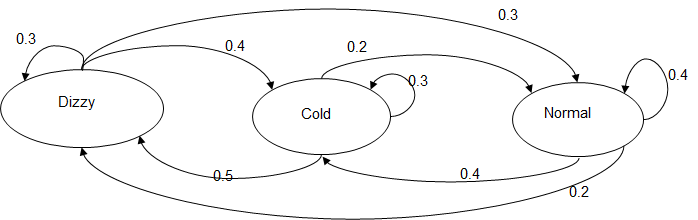
P (Sik|Si1,Si2,...Sik-1) = P(Sik|Sik-1)

To define Markov model, the following probabilities have to be specified:

Transition Probability: aij = P(Si|Sj)

Initial Probability: π = P(Si)

* **Example of markov model:**



* States: Dizzy, Cold, Normal

* Transition Probabilities: P(Dizzy | Dizzy) = 0.3, P(Dizzy | Cold) = 0.4,P(Dizzy | Normal) = 0.3

* Initial Probabilities: say P(Dizzy) = 0.8, P(Cold) = 0.1, P(Normal) = 0.1
* **Calculation of sequence probability:**

By Markov chain property, probability of state sequence can be found by the formula:

P(Si,Si2,...Sik) = P(Sik| Si1,Si2,...,Sik-1) \* P(Si1,Si2,...,Sik-1)

= P(Sik|Sik-1) \* P(Si1,Si2,...,Sik-1)

= P(Si1) \* P(Si2|Si1) \* P(Sik|Sik-1) \* P(Sik-1|Sik-2)

Suppose we want to calculate the probability of a sequence of states in our example, {Dizzy, Cold, Cold, Normal}.

      P ({Dizzy, Cold, Cold, Normal}) =

P (Dizzy) P (Dizzy| Cold) P (Cold| Cold) P (Cold| Normal) =

          = 0.8\*0.4\*0.3\*0.2

= 0.0192

1. **Hidden Markov Model**

Hidden Markov Model (HMM) is a variant of a finite state machine having

▹Set of hidden states,

▹Set of observations,

▹Initial state probabilities,

▹Transition probabilities,

▹Emission probabilities.

Hidden markov models are the extended version of markov models with the addition of hidden states.

* Hidden markov models holds the following components:
* Set of States : {S1,S2,...Sn}
* Process move from one state to another generating a sequence of states {Si1,Si2,..,Sik}

12.1 **Markov chain property**:

Probability of each subsequent state depends only on what was the previous state.

Set of observations: States are not visible, but each state randomly generates one of M observations (or visible states) {V1, V2, … , VM}.

To define a hidden markov model, the following probabilities have to be specified.

1. Matrix of transition probabilities:

A = (aij) , aij = P(Si | Sj)

1. Matrix of observation probabilities:

B = (bi(Vm)), bi(Vm) = P(Vm| Si)

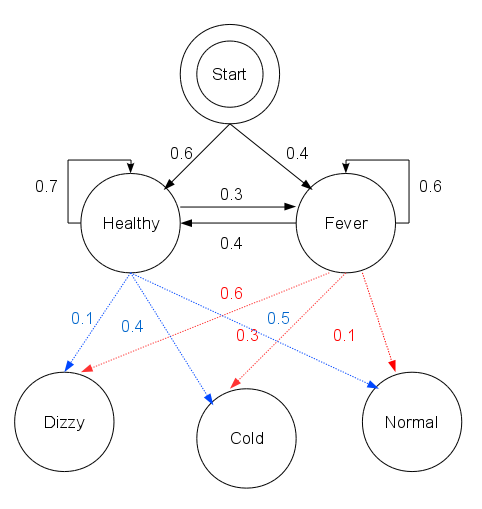
1. A vector of initial probabilities:

π = (πi), πi = P(Si).

The Model is represented by

ƛ = (A, B, π).

**12.2 Examples of Hidden Markov Model**

****

Observations = (Normal, Cold, Dizzy)

States = (Healthy, Fever)

Initial probabilities = (Healthy: 0.6, Fever: 0.3)

Transition Probabilities =

Healthy: (Healthy: 0.7, Fever: 0.3)

Fever: (Healthy: 0.4, Fever: 0.6)

Emission Probabilities =

Healthy: (Normal: 0.5, Cold: 0.4, Dizzy: 0.1)

Fever: (Normal: 0.1, Cold: 0.3, Dizzy: 0.6)

**12.3 5 Properties of Hidden Markov Model**

* Observations:

In our case Observations are the student category we want to classify.

We took 3 types of student categories as observations:

1. Excellent Performer
2. Good Performer
3. Below average Performer

* Hidden States:

Hidden states are the states which can affect the observation.

We can say that hidden states are the decisive factors of a particular observation.

In our case we have used virtual lab as a platform, so our hidden states are going to be:



These are all the pages on each topic of virtual lab.

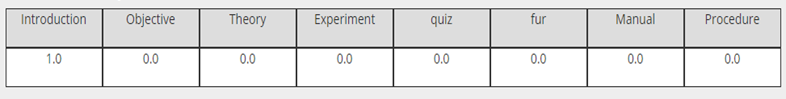
A student can go through all above pages to learn a particular topic.

* Initial Probabilities:

Initial probabilities represents ‘in which state the HMM is in when the user first opens a particular lab on the platform ’.

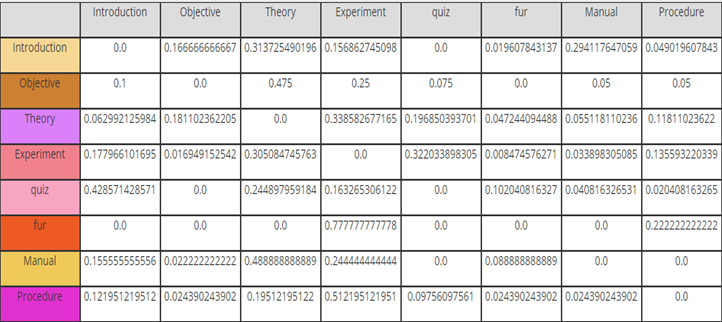
In our case the student is greeted with introduction page every time he/she wants to learn any topic.

That’s why the initial probability distribution would look like this:



* Transition Probabilities:

Transition probability is the probability of transitioning from one state to another.



This is an example table where we have calculated transition probabilities.

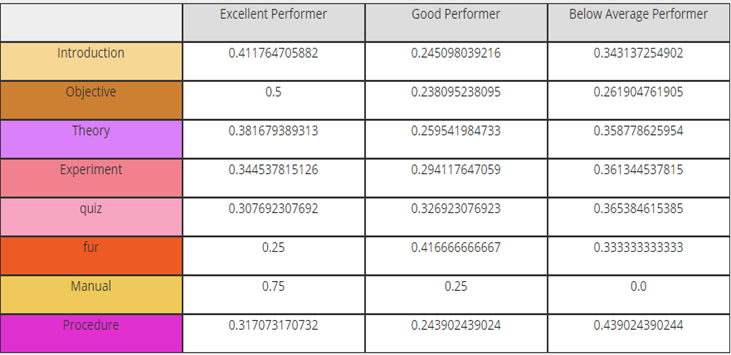
Example:

P(Xt | Xt-1)

P(Quiz | Theory) = 0.1968

* Emission Probabilities:

Emission probabilities gives us the probability of getting an observation based on a particular hidden state



Example:

P( Excellent Performer | Objective ) = 0.5

* **Calculation of observation sequence probability:**

Suppose we want to calculate the probability of a sequence of observations in our example, {Normal, Cold}.

Consider all possible hidden state sequences:

P( {Normal, Cold} ) = P( {Normal, Cold} , {Healthy, Healthy}) + P( {Normal, Cold} , {Health, Fever}) + P( {Normal, Cold} , {Fever, Healthy}) + P( {Normal, Cold} , {Fever, Fever})

Where the first term is:

P( {Normal, Cold} , {Healthy, Healthy})=

P( {Normal, Cold} | {Healthy, Healthy})  P({Healthy, Healthy}) =

P(Normal | Healthy) P(Cold | Healthy) P(Healthy) P(Healthy| Healthy)

= 0.5\*0.4\*0.6\*0.6\*0.7

This example gave us the probability of observation Normal to Cold happening over the state transition Healthy to Healthy.

Using the same method, we can get probability of observation Normal to Cold happening over the state transition Healthy to Fever, Fever to Fever & Fever to Healthy.

Then, calculate all the probabilities and whichever state transition probability turns out to be the highest we can determine observation sequence probability on particular state transition.

**12.4 Canonical Problems in Hidden Markov Model**

There are three canonical problems in sequence analysis:

* 1. Evaluation Problem
  2. Decoding Problem
  3. Learning Problem

**1. Evaluation Problem**

Given a model and an output sequence, what is the probability that the model generated that output?

To answer this, we consider all possible paths through the model

A solution to this problem gives us a way of scoring the match between an HMM and an observed sequence

We have a model ƛ = (A, B, π) and a sequence of observations {O1, O2, … , OM}, and  P(O | ƛ) must be found.

We can calculate this quantity using simple probabilistic arguments. But this calculation involves number of operations in the order of NT.

This is very large even if the length of the sequence, *T* is moderate. Therefore we have to look for another method for this calculation. Fortunately there exists one which has a considerably low complexity and makes use an auxiliary variable, ***ɑt(****i****)*** called *forward variable*.

To calculate *forward variable* we have to use forward algorithm.

* **Forward Algorithm:**

Forward Algorithm is used to calculate forward probabilities.

Forward probabilities are written as:

***P(St | O1:t)***

Here,

St is the set of hidden states.

O1:t are the sequence of observations.

This formula represents The forward probability for i at time t, ɑt(i), is the probability that an HMM will output a sequence O1:t and end in state si.

Recursive formula for forward probabilities:

***ɑt(xt) = P(Ot | Xt) xt-1P(Xt | Xt-1) ɑt-1 (xt-1)***

Here,

***P(Ot | Xt)*** is the emission probability

***P(Xt | Xt-1)*** is the transition probability

***ɑt-1 (xt-1)*** is a recursive call to the function

Now, let’s see further explanation of all this individual mathematical figures.

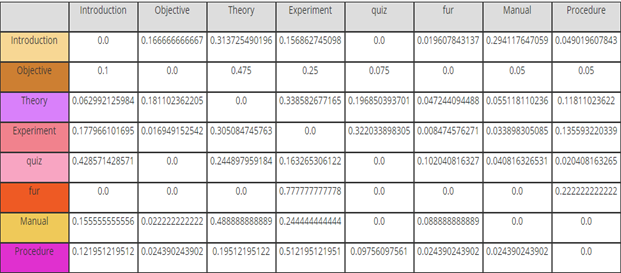
***P(Xt | Xt-1):***

This is the transition probability of transitioning from    state Xt-1 to state Xt.

Example:

Xt: Theory

Xt-1: Introduction



Here probability of transitioning from introduction to theory is 0.3137.

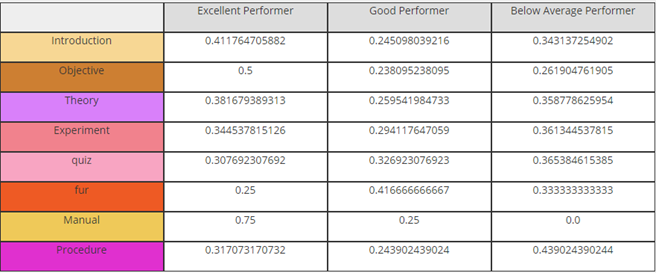
***P(Ot | Xt):***

This is the emission probability of state Xt happening over observation Ot.

Example:

Ot: Excellent Performer

Xt: Theory



Here the probability of state Theory happening over observation Excellent performer is 0.3816.

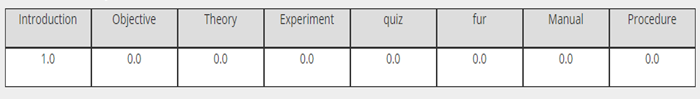
***ɑt-1 (xt-1):***

This is a recursive call to the same function and decreases the value of t on each call.

Once the value of t becomes 1 it takes the value from initial probability of Xt.

Example:

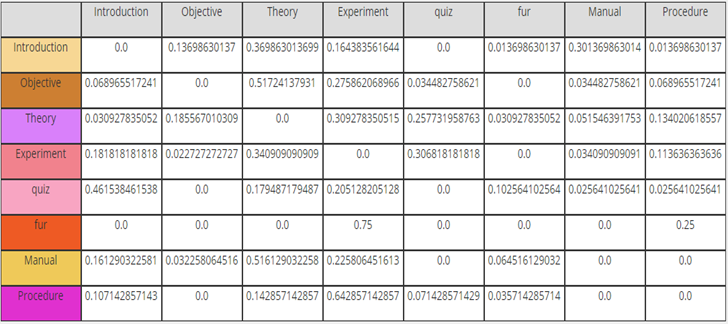
Xt: Introduction



Here the initial probability of introduction is 1.0.

Finally we would get a forward probability matrix, which would be similar to transition matrix but with newly calculated values with the help of forward algorithm.

Forward probability matrix would look like this:



1. **Decoding Problem:**

Given the HMM ƛ = (A,B,π) and the observation sequence  O=o1,o2 ..., oK , calculate the most likely sequence of hidden states si that produced this observation sequence.

Solution of decoding problem helps us to find the most likely state sequence from given observation sequence.

Decoding problem can be solved with the help of Viterbi algorithm.

* **Viterbi Algorithm:**

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden state called the Viterbi path, which results in a sequence of observed events.

**Input:**

Observation sequence O={O1, O2, … , ON}

State Sequence S={S1,S2,...Sn}

Vector of initial probabilities πi = P(Si)

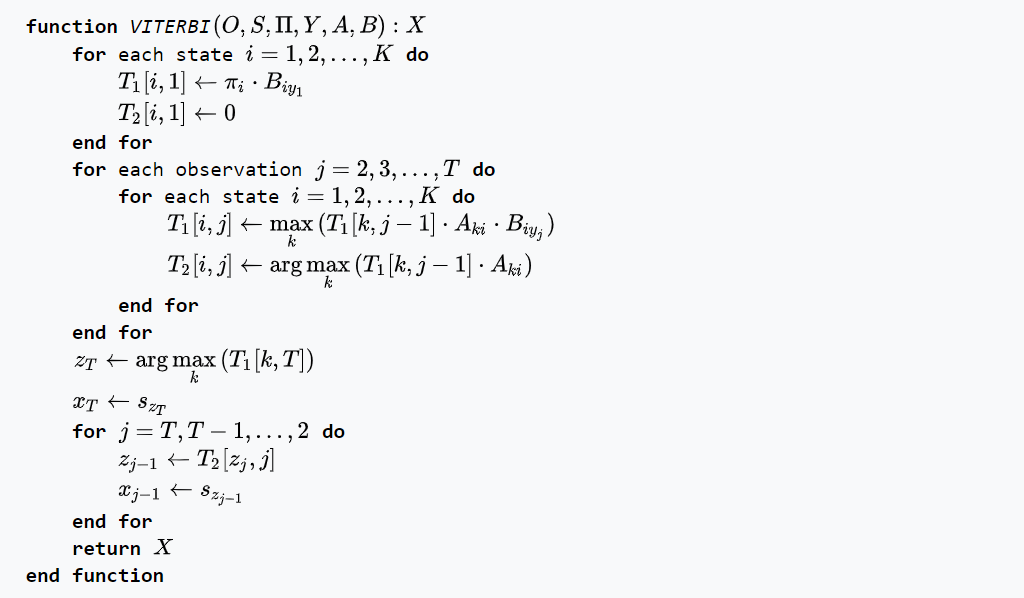
Sequence of observations Y={Y1, Y2, … , YN}

Transition Matrix A = (aij) , aij = P(Si | Sj)

Emission Matrix B = (bi(Vm)), bi(Vm) = P(Vm| Si)

**Output:**

The most likely hidden state sequence X={X1, X2, … , XN}

**Pseudo code: **

Consider a village where all villagers are either healthy or have a fever and only the village doctor can determine whether each has a fever. The doctor diagnoses fever by asking patients how they feel. The villagers may only answer that they feel normal, dizzy, or cold.

The doctor believes that the health condition of his patients operate as a discrete Markov chain. There are two states, "Healthy" and "Fever", but the doctor cannot observe them directly; they are hidden from him. On each day, there is a certain chance that the patient will tell the doctor he/she is "normal", "cold", or "dizzy", depending on their health condition.

obs = ('normal', 'cold', 'dizzy')

States = ('Healthy', 'Fever')

start\_p = {'Healthy': 0.6, 'Fever': 0.4}

trans\_p = {

  'Healthy' : {'Healthy': 0.7, 'Fever': 0.3},

  'Fever' : {'Healthy': 0.4, 'Fever': 0.6}

  }

emit\_p = {

  'Healthy' : {'normal': 0.5, 'cold': 0.4, 'dizzy': 0.1},

  'Fever' : {'normal': 0.1, 'cold': 0.3, 'dizzy': 0.6}

  }

The function viterbi takes the following arguments:

* obs is the sequence of observations, e.g. ['normal', 'cold', 'dizzy'];
* States is the set of hidden states
* start\_p is the start probability
* trans\_p are the transition probabilities
* emit\_p are the emission probabilities.

Output will show that, Observation sequence Normal-Cold-Dizzy were most likely generated by states ['Healthy', 'Healthy', 'Fever'].

In other words, given the observed activities, the patient was most likely to have been healthy both on the first day when he felt normal as well as on the second day when he felt cold, and then he contracted a fever the third day.

1. **Training Problem:**

Given some training sequences O=o1, o2,...,oK and general structure of HMM (numbers of hidden and visible states), determine HMM parameters M=(A, B, )   that best fit the training data, that maximizes P(O | M) .

This problem can be solved with the help of Baum-Welch algorithm.

* **Baum-Welch Algorithm:**

Begin with some model   (perhaps random, perhaps preselected)

• Run O through the current model to estimate the expectations of each model parameter.

• Change the model to maximize the value of the paths that are used a lot (while still respecting the stochastic constraints).

• Repeat, hoping to converge on optimal values for the model parameters  ƛ.

Basically Baum-Welch algorithm takes historical data as input and it will try to create models according to the available data.

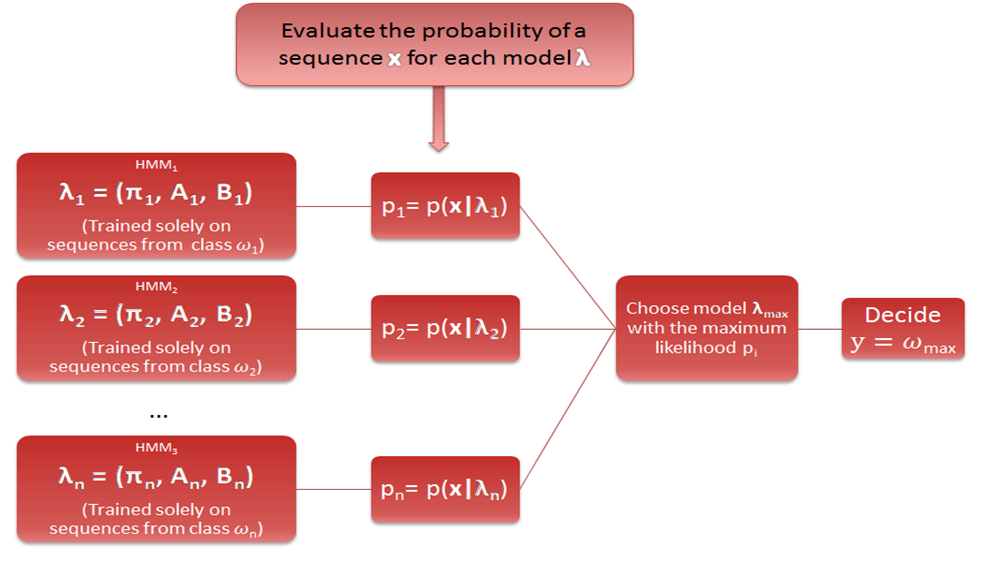
* + - 1. It will calculate forward probabilities with the help of forward algorithm.
      2. Apply new changes in the previous model.
      3. Check if log likelihood has reached the threshold limit or not.
      4. If not then, again calculate the forward probabilities and repeat the process.

This way we would have three models representing Excellent, Good and below average performer.

1. **Hidden Markov Classifier:**

Hidden Markov Classifier will help us to classify any student on the basis of state traversal sequence of the student.

* Generate multiple models for each type of sequence using Baum-Welch training.
* In our case 3 models will be prepared, Excellent, Good and Below Average Performer.
* Finally the classifier will attempt to select the class with maximum probability given the sequence.



Here,

A = transition matrix

B = Emission matrix

π = Initial matrix

ƛ = class of input sequence

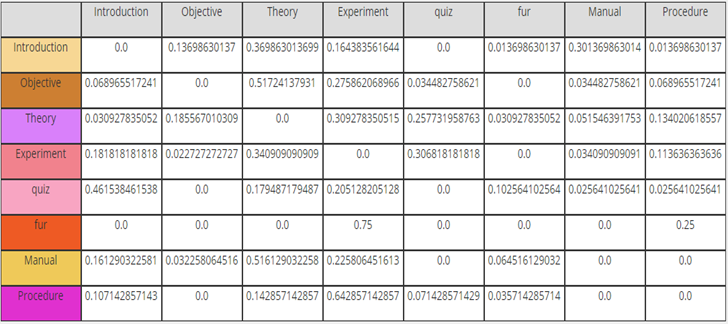
Model will be generated with the help of Baum-Welch algorithm.

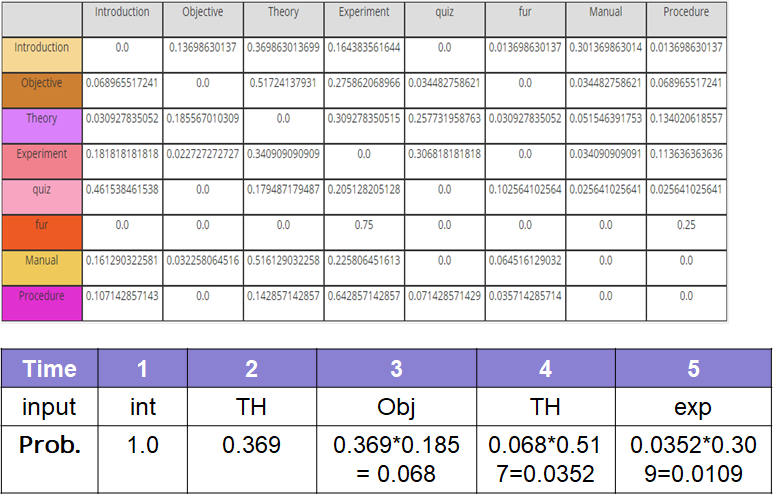
And to evaluate the probability of a sequence X over each model ƛ we would require to use forward algorithm.

Forward algorithm will give us the forward probability matrix for each model.

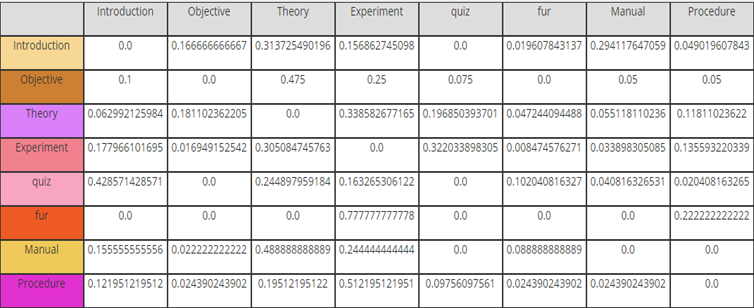
Here is the example of forward probability matrix for the state sequence **Introduction-Theory-Objective-Theory-Experiment** for each model.

* **Excellent Performer forward probabilities:**

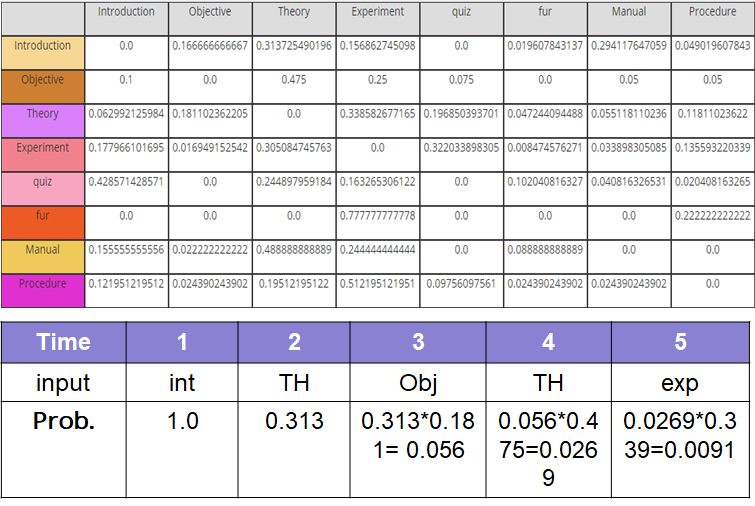
****

Sequence probability calculation:            

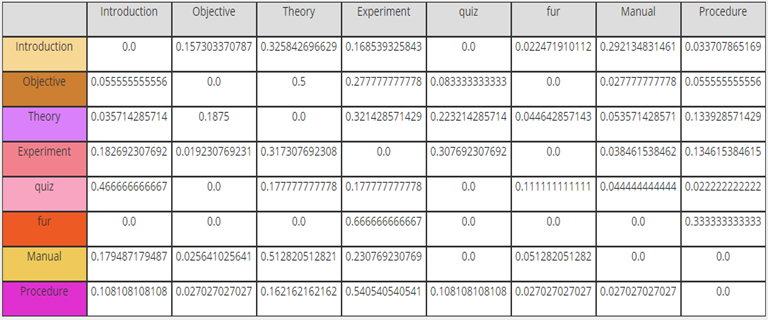
* **Good Performer forward probabilities:**



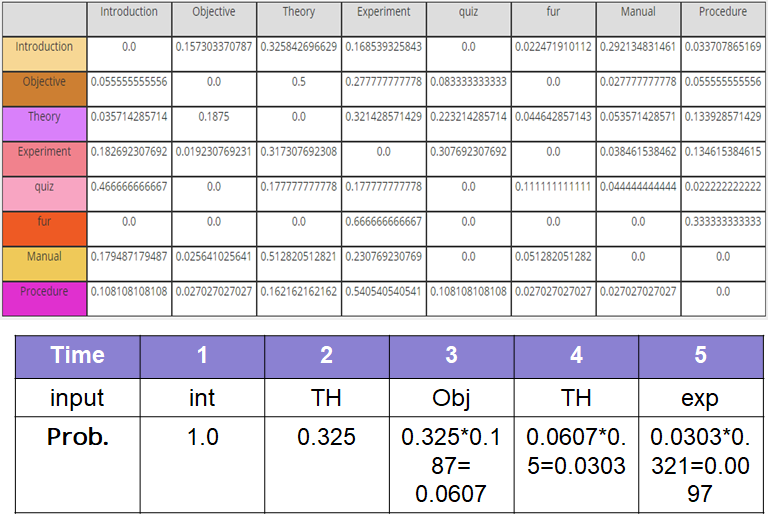
Sequence probability calculation:



* **Below Average Performer forward probabilities:**

****

Sequence probability calculation:



* **Classification:**
* Input sequence to classify

int TH obj TH exp

* Probability :

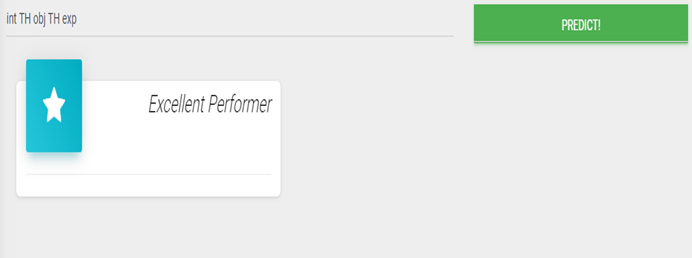
Excellent performer HMM= 0.0109

Good performer HMM      = 0.0091

Below average performer = 0.0097

* **Wmax = 0.0109 (Excellent Performer)**

As you can see the classifier predicts that this sequence would be generated by excellent performer category of student.



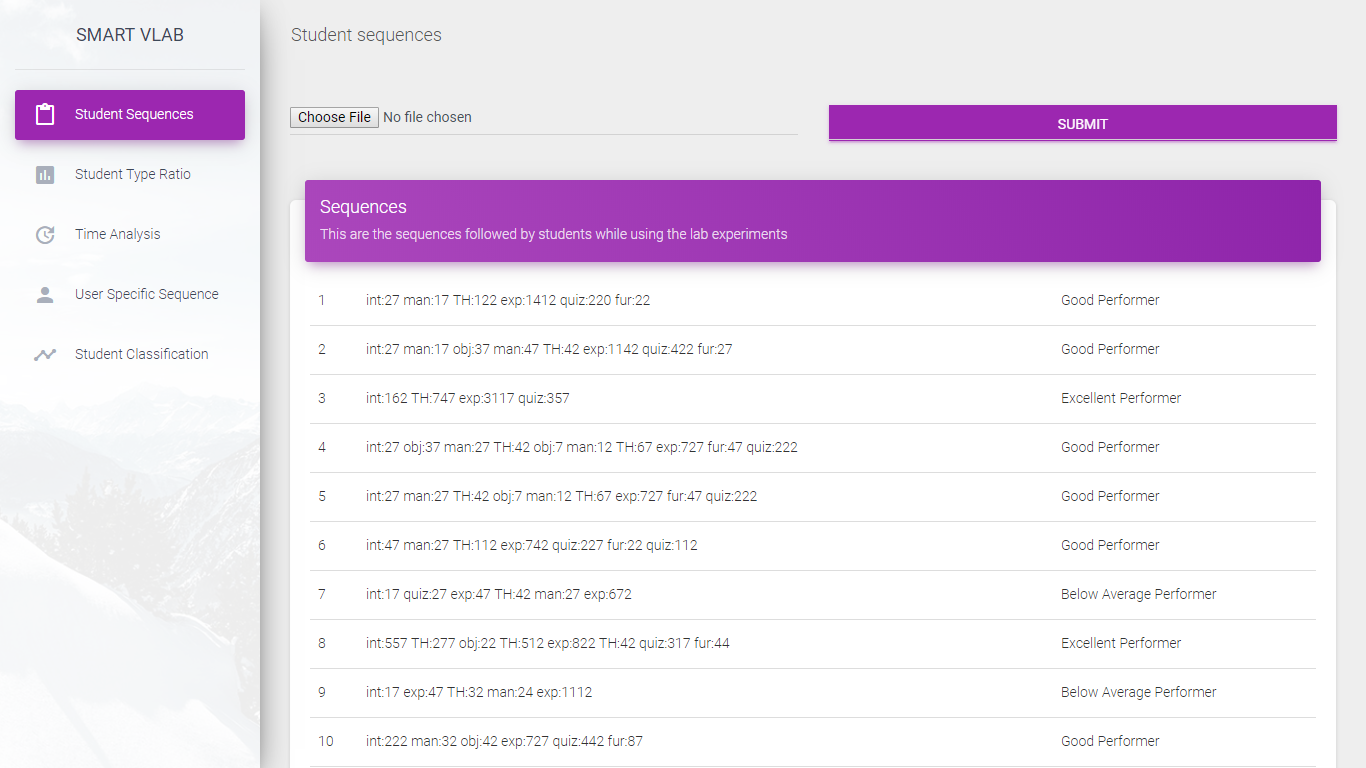
Above is a screenshot where the classifier predicts that the student would be an Excellent Performer.

1. **Analysis**

From footprint gathering and student classification there are multiple types of analysis we can do to improve the E-Learning experience.

For the analysis part we have developed a web based application using

* ASP.NET framework for core development.
* Accord.NET framework for Hidden Markov Classifier.



This is a screenshot of the first page where we require to upload an excel file containing footprints of students (Historical data).

**Analysis we can perform based on gathered data:**

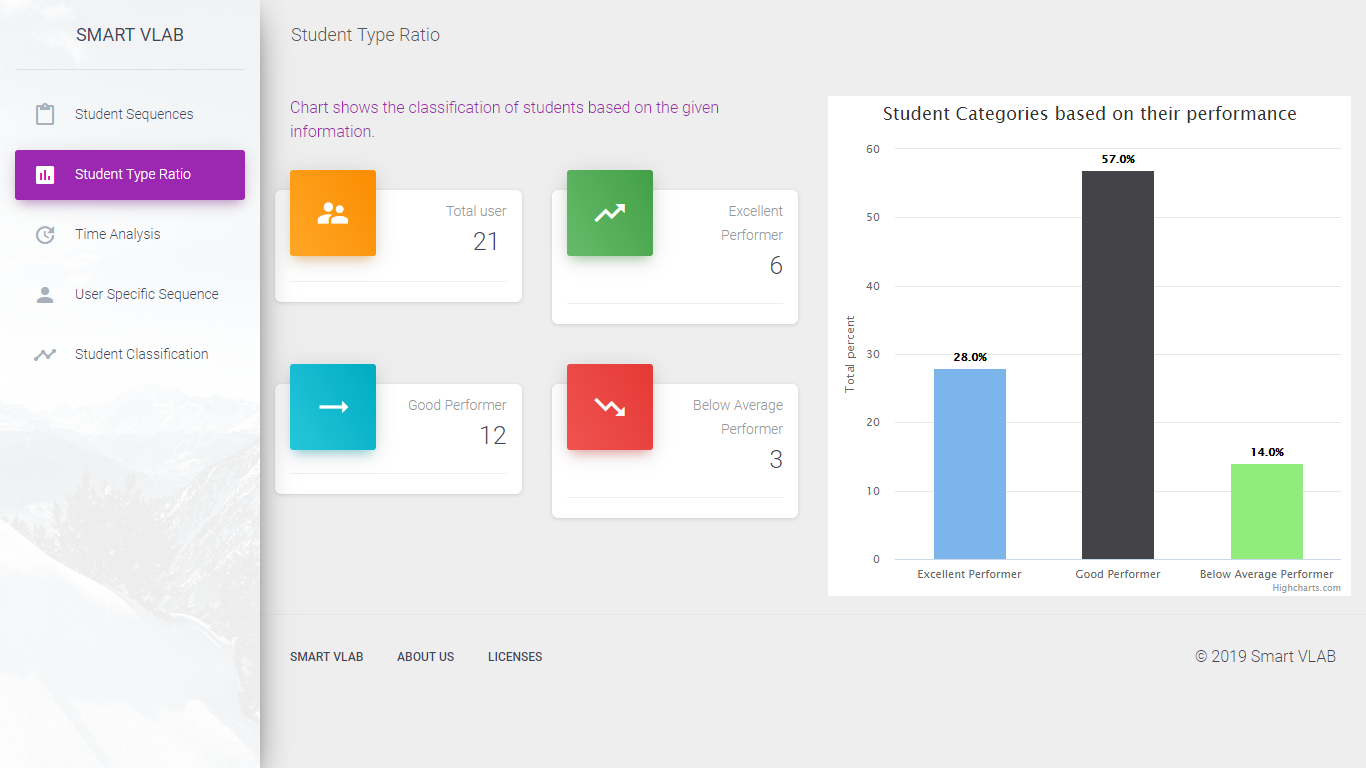
* Category based performance analysis
* Time based analysis
* Sequence analysis
* Student classification

**14.1 Category based performance analysis:**

In Footprint gathering section we recorded student lab traversal sequence along with that we already knew their corresponding academic score to classify them.

We performed footprint gathering on multiple experiments of V-LAB.

After footprint gathering we can classify each sequence by the category of that student’s performance in academics.



As we can see in student based performance analysis we can see count of total users along with that count of each category of students.

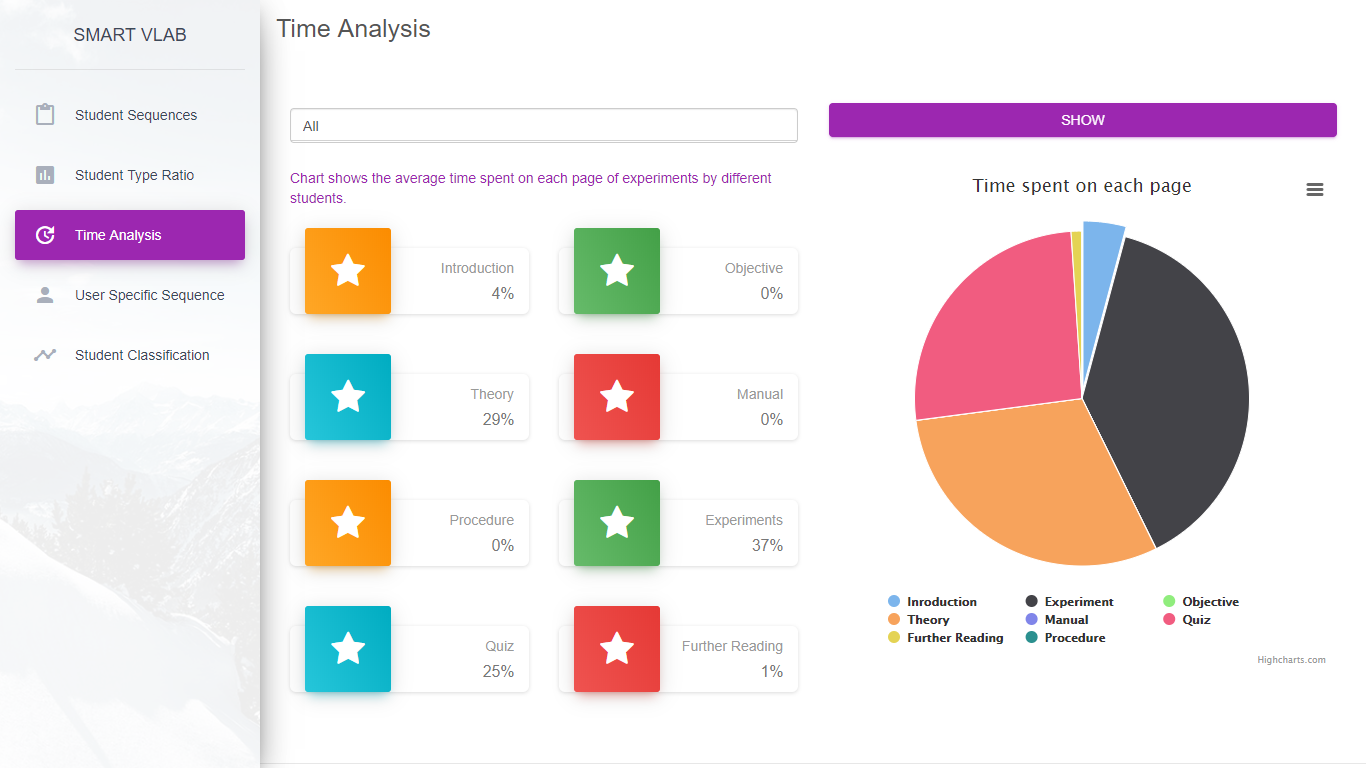
Along with the textual information there is a graphical representation displaying the student ratio in bar graph.

From this analysis the analytics team can identify the ratio of students on each type of topic and improve the platform according to that.

**14.2 Time based analysis:**

In footprint gathering section along with the page traversal sequence we recorded the time spent on each page by the user.

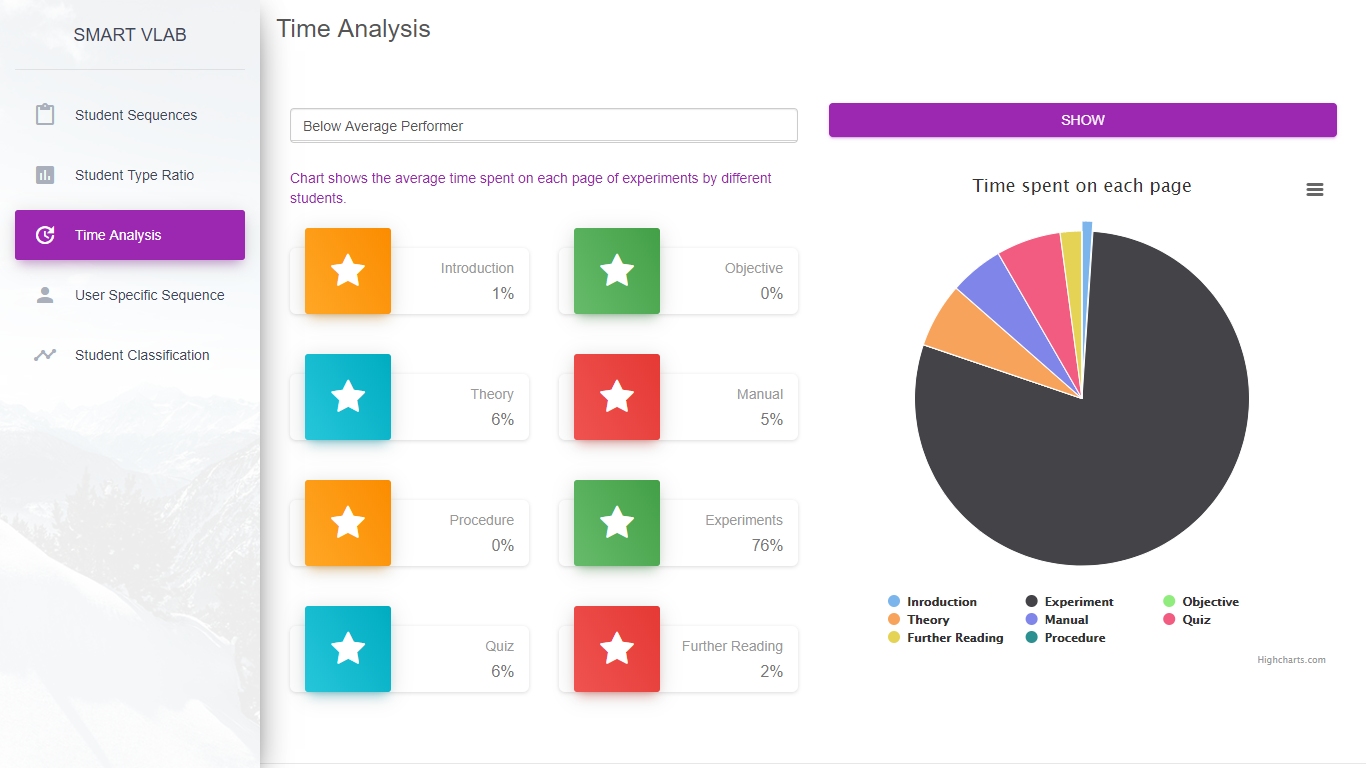
This way we got the time spent by all the category of users to analyze how much a user spent on a particular page.



As we can see from above page the average time spent by all the students on each page.

For example we can see that students tend to focus more on experiment section. From total time a student spends its 37% time only on experiment page.

This way the analytics team can focus on to improve the important pages.



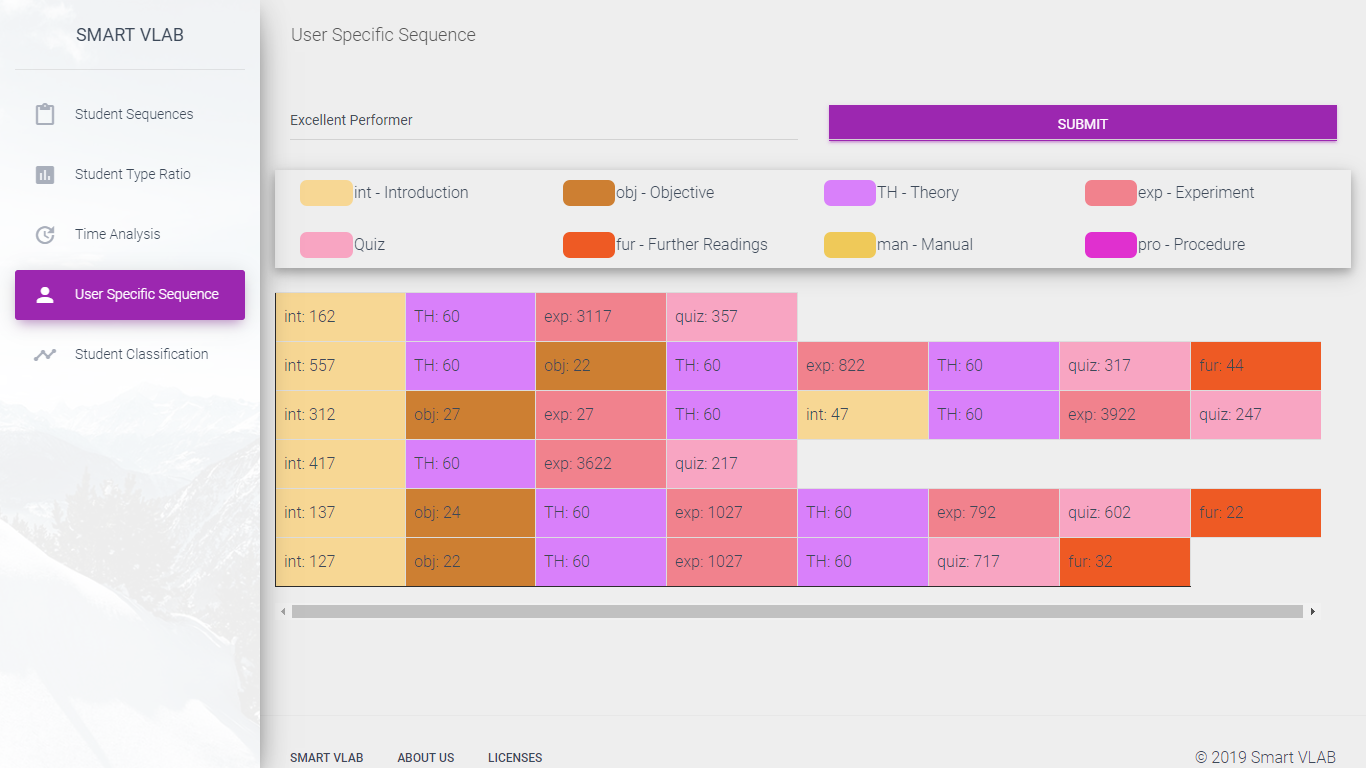
This is a screenshot of analyzing the time for below average performer.

As we can identify that below average tends to focus way too much time on experiments and not on other theoretical parts as well as quiz.

This way analytics can give according suggestions to the students from this data.

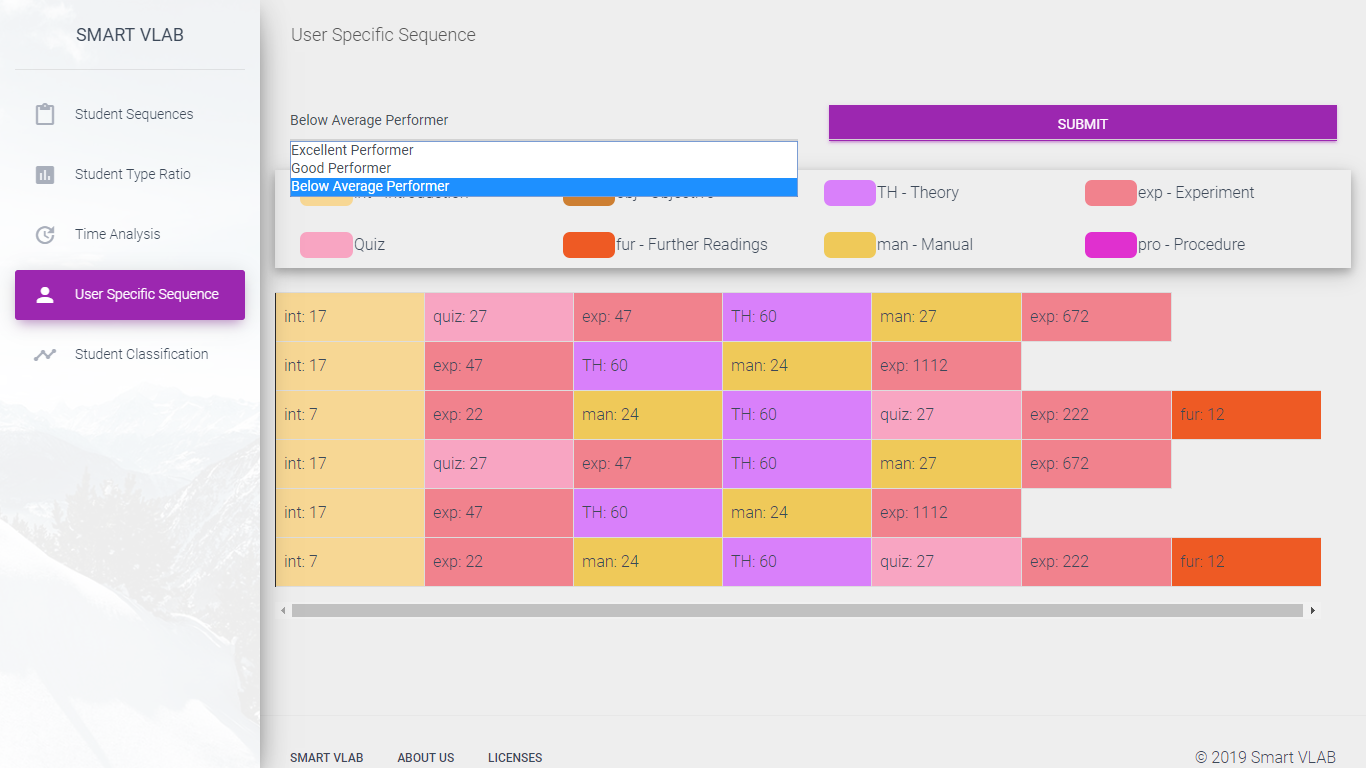
**14.3 Sequence analysis:**

Sequence analysis to look at the sequences followed by each category of the student.



The above image represents the sequence of pages visited by the user of type Excellent Performer.

As we can clearly identify that excellent performer do not jump right onto the experiment page or quiz page instead they tend to focus on introduction and then read theory first.



This is the screenshot of sequence analysis for below average performer students.

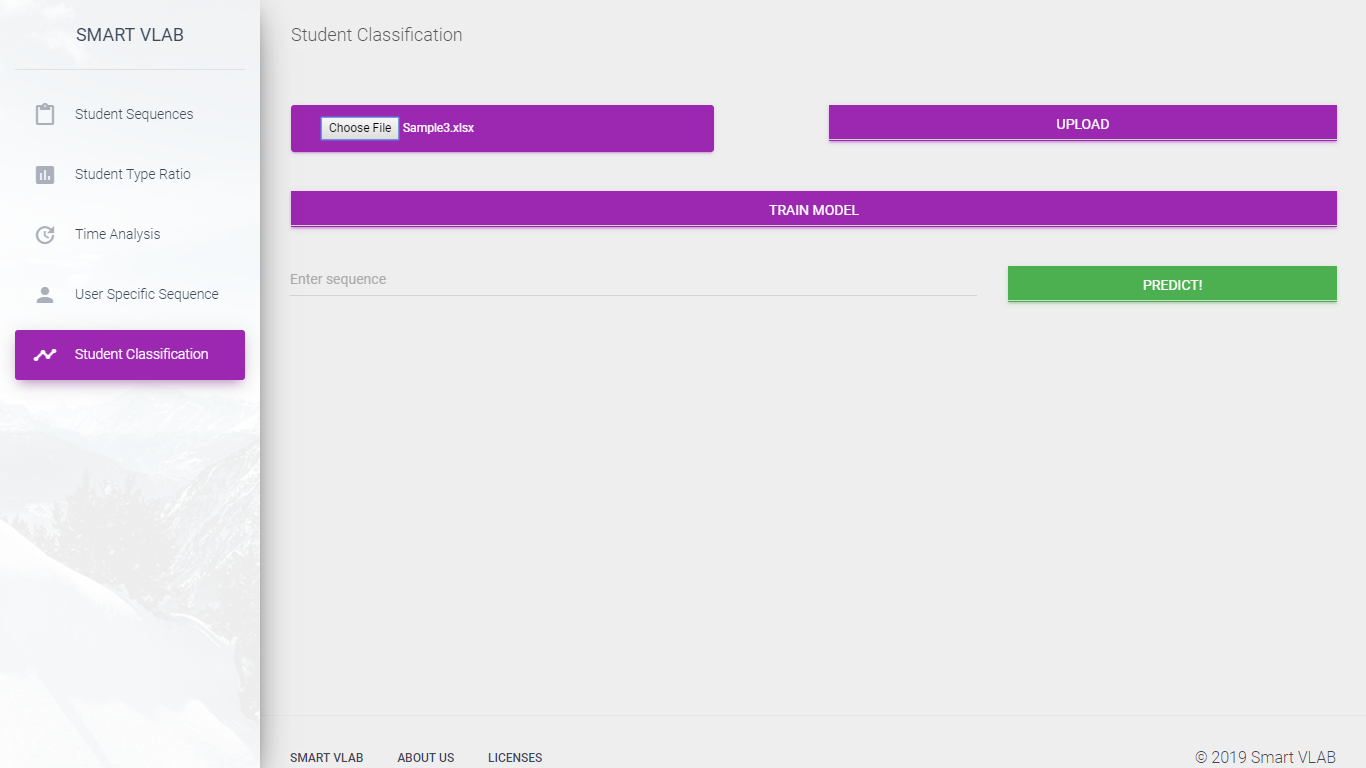
Here we can see that almost every student directly focuses on practical part and quiz without understanding about the topic in theory section.

**14.4 Student classification:**

With the help of historical data we gathered in footprint section we can use the hidden markov classifier to classify the student from the student’s sequence pattern in real time.

Student classification has been explained in detail in the previous section.

With the help of classification of student in real time the analytics team can instantly provide recommendations to that particular student.



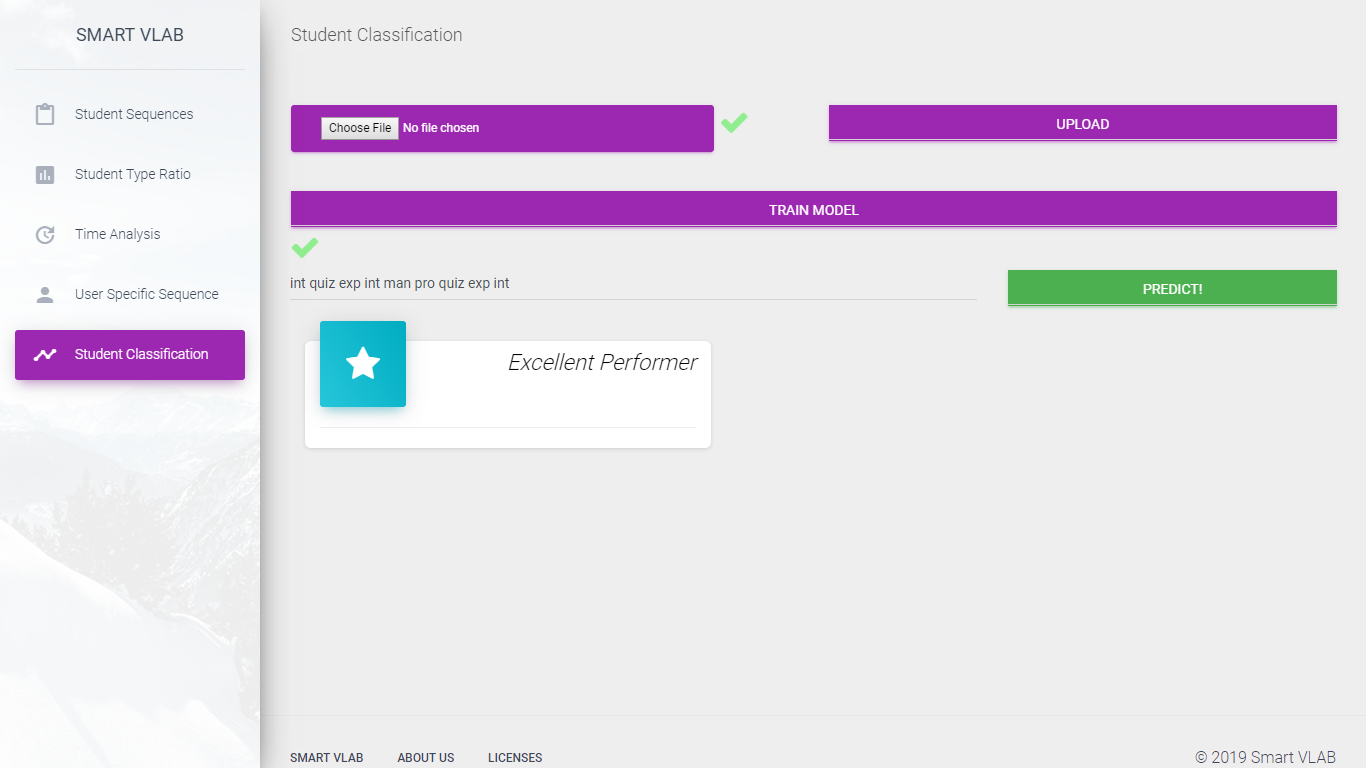
This is a screenshot of the student classification page.

Here we need to upload excel file containing footprint of students (Historical data).

After uploading the file, we would require to train the model, internally model training would be done by the Baum-Welch algorithm which we have discussed in previous section.

After training of model, we need to give traversal sequence to classify that sequence to category of student.

And finally the model would predict the category of student with the help of hidden markov classifier.



From the above screenshot we can see that, we gave a page traversal sequence and clicked on the predict button and model gave an output as ‘Excellent Performer’

1. **Evaluation**

For classification we have used 47 genuine records.

In this test we used 47 records of student’s page traversal sequence with time spent on each page.

From these 47 records we used 37 records as training sets and 10 records as test sets.

We did two types of classification:

* Classification considering time factor:

In this method the model takes sequence as well as time spent on each page into consideration for classification.

Using this approach the classification accuracy was around 60-70%.

* Classification without considering time factor:

In this method the model takes student’s page traversal sequence into consideration for classification.

Using this approach the classification accuracy was around 80-90%.

It is important to mention that we had only 47 genuine records, which is considered as very less for training a model.

In future enhancement we are planning to test the accuracy with large amount of dataset for better accuracy results.

We are also planning to modify the algorithm parameters to increase accuracy with time consideration approach.

1. **System Requirement**

Server Side:

                       Operating System: Windows

                        Server : Apache HTTP Server, IIS

                        Ram : 2 GB

                        HDD : 20 GB

Client Side:

             Web Browser with Internet Connection

Development Tools:

Notepad++, XAMPP, MS Visual Studio

1. **Technical Specification**

* **An introduction to PHP**

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user. You add, delete, and modify elements within your database through PHP.

* **An introduction to MySQL**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Twitter, Flickr, and YouTube.

* **An introduction to  Apache HTTP Server**

The Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software, released under the terms of the Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from authentication schemes to supporting server-side programming languages such as Perl, Python, Tcl and PHP.

* **An introduction to JavaScript**

JavaScript was released by Netscape and Sun Microsystems in 1995. However, JavaScript is not the same thing as Java.

It is programming and interpreted language. It is object-based programming and widely used and supported language. JavaScript code is typically embedded in the HTML, to be interpreted and run by the client’s browser.

Uses of JavaScript:

* Use it to add multimedia elements.
* Create pages dynamically.
* Interact with the user.
* To run JavaScript on your browser, you need JavaScript-enabled browser.

* **An introduction to .NET Framework**

.NET Framework (pronounced as "*dot net"*) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named as Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages.

Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named the Common Language Runtime (CLR). The CLR is an application virtual machine that provides services such as security, memory management, and exception handling. As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework.

* **An introduction to C#**

C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# was developed by Anders Hejlsberg and his team during the development of .Net Framework.

C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows the use of various high-level languages on different computer platforms and architectures.

It is a modern, general-purpose programming language. It is object oriented. It is component oriented. It is easy to learn. It is a structured language. It produces efficient programs.  It can be compiled on a variety of computer platforms. It is a part of .Net Framework.

* **An introduction to ASP.NET**

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform.

ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework. The ASP.NET application codes can be written in any of the languages like C#, Visual Basic.Net, Jscript, J#.

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

* **An introduction to Internet Information Server (IIS)**

Internet Information Services (IIS) is a flexible, general-purpose web server from Microsoft that runs on Windows systems to serve requested HTML pages or files.

An IIS web server accepts requests from remote client computers and returns the appropriate response. This basic functionality allows web servers to share and deliver information across local area networks, such as corporate intranets, and wide area networks, such as the internet.

A web server can deliver information to users in several forms, such as static web pages coded in HTML; through file exchanges as downloads and uploads; and text documents, image files and more.

* **An introduction to Accord.NET**

The **Accord.NET Framework** is a .NET machine learning framework combined with audio and image processing libraries completely written in C#. It is a complete framework for building production-grade computer vision, computer audition, and signal processing and statistics applications even **for commercial use**.

It contains classes related to Hidden Markov Models and their learning algorithms. Offers support for both discrete and continuous-density models, as well as Markov classifiers and threshold models for sequence rejection.

1. **Future Enhancement**

Till now we have classified student based on his/her way of using the V-Lab platform.

After classification it is up to the analytics team to give proper recommendations.

In future we can apply another machine learning model which can provide automatic recommendations once the user is classified based on the previous recommendations.

In that case the model will match the new sequence with already observed sequences and also checks the given recommendation in past for that particular student.

And according to that the system will classify the student as well as give recommendations based on previous recommendations.

1. **Conclusion**

E-Learning platforms are growing faster than ever. With the help of E-Learning platforms user can learn any topic at anywhere and anytime.

But most E-Learning platforms are not as effective as traditional classroom based learning, because of one major factor.

In classroom based approach a teacher can teach with different methods and different pace to each student according to their grasping capabilities.

This type of smart teaching is not available in most of the E-Learning platforms, Mostly there is a same content for every type of student.

This is the decisive factor in classroom based learning vs. E-Learning.

We can make E-Learning smart if we start to observe the student usage patterns and once we are able to identify the student capabilities we can provide recommendations and also whole new teaching model on the platform.

In this project we have tried making V-Lab platform smart which observes user behavior to give recommendations and adjusts the new content on the basis of student capabilities.

Hidden Markov Classifier helped us to classify a student, and the factor of classification was the page usage sequence and time spent on each page.

1. **Bibliography**

[https://en.wikipedia.org](https://en.wikipedia.org/)

[https://dotnet.microsoft.com/learn/web/](https://dotnet.microsoft.com/learn/web/what-is-aspnet)

[http://accordframework.net/docs/html/N\_Accord\_Statistics\_Models\_Markov.htm](http://accord-framework.net/docs/html/N_Accord_Statistics_Models_Markov.htm)

<https://www.codeproject.com/Articles/541428/Sequence-Classifiers-in-Csharp-Part-I-Hidden-Marko>