A
Project Report
On

STUDENT PERFORMANCE PREDICTION AND VISUALIZATION

BTech-sem VII

Prepared By
Parashar Parikh(IT-070)
Kaivan Shah(IT-104)

DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF TECHNOLOGY,
DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001

December,2018

A
Project Report
On

STUDENT PERFORMANCE PREDICTION AND VISUALIZATION

BTech-sem VII

In partial fulfillment of requirements for

Bachelor of Technology

in

Information Technology

Submitted By:

1.Parashar Parikh(IT-070)
2.Kaivan Shah(IT-104)

Under the Guidance of

Prof.(Dr.) H.B.Prajapati

Prof.(Dr.) M.M.Goswami

DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF TECHNOLOGY, DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001

CANDIDATE'S DECLARATION

We declare that pre-final semester report entitled "STUDENT_PERFORMANCE PREDICTION AND VISUALIZATION" is our own work conducted under the supervision of the guide Prof. (Dr.) H.B.Prajapati and Prof.(Dr.) M.M.Goswami

We further declare that to the best of our knowledge the report for B.Tech. VII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Signature
Candidate's Name: Parashar Parikh
Student ID:15ITUOS051

Candidate's Signature:
Candidate's Name: Kaivan Shah
Student ID: 15ITUOS123

Submitted To:
Prof.(Dr.) H.B.Prajapati
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad Gujarat.

Submitted To:
Prof. (Dr.) M.M.Goswami
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad Gujarat.

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of Software Design Project, entitled "STUDENT PERFORMANCE PREDICTION AND VISUALIZATION" and recorded in this report is a bonafide report of work of

1) Parashar Parikh Roll No. IT-070 ID No: 15ITUOS051
2) Kaivan Shah Roll No. IT-104 ID No: 15ITUOS123

of Department of Information Technology, semester VII. They were involved in Project work during academic year 2018 -2019

Prof.(Dr). M.M.Goswami

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date:

Prof.(Dr.).V.K.Dabhi,

Head , Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad.

Date:

ACKNOWLEDGEMENT

On completion of this project we would like to express our sincere thanks to all those who have guided, advised, inspired and supported during our project work.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. We, hereby, avail this opportunity to express our heartfelt gratitude to a number of people who extended their valuable time, full support and cooperation in developing this project.

We are heartily thankful to the qualified staff of the university and especially to our project guide Prof.(Dr.)H.B.Prajapati & Prof.(Dr.)M.M.Goswami.

We believe that there computer expertise, problem solving abilities and valuable guidance have made it possible to present such a great project.

We are heartily thankful to our peers who have helped and guided us whenever needed and also provided necessary help if needed.

We are thankful to Dept. Of Information Technology of Dharamsinh Desai University,Head of the Department Prof.(Dr).V.K.Dhabi and also all the faculty members for giving us their valuable guidance.

Yours Sincerely,

Parashar Parikh(IT – 070)
Kaivan Shah(IT -104)

Table of Contents PAGE NO

ABSTRACT

An educational institution needs to have an approximate prior knowledge of enrolled students to predict their performance in future academics. This helps them to identify promising students and also provides them an opportunity to pay attention to and improve those who would probably get lower grades. As a solution, we have developed a system which can predict the performance of students from their previous performances using concepts of data mining techniques under Classification. We have analyzed the data set containing information about students, such as gender, marks scored in the board examinations of classes X and XII, marks and rank in entrance examinations and results in first year of the previous batch of students. By applying the ID3 (Iterative Dichotomiser 3) and C4.5 classification,Random tree Generator algorithms on this data, we have predicted the general and individual performance of freshly admitted students in future examinations.

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Every year, educational institutes admit students under various courses from different locations, educational background and with varying merit scores in entrance examinations. Moreover, schools and junior colleges may be affiliated to different boards, each board having different subjects in their curricula and also different level of depths in their subjects. Analyzing the past performance of admitted students would provide a better perspective of the probable academic performance of students in the future. This can very well be achieved using the concepts of data mining.

For this purpose, we have analysed the data of students enrolled in first year of engineering. This data was obtained from the information provided by the admitted students to the institute. It includes their full name, gender, application ID, scores in board examinations of classes X and XII, scores in entrance examinations, category and admission type. We then applied the ID3 and C4.5 algorithms after pruning the dataset to predict the results of these students in their first semester as precisely as possible.

1.1 Problems In Nowadays Performance predictors

Clearly nowadays systems which predict student performances don't precisely take in measures of there 11,12th marks as percentile system exist in India and some other countries . Percentile are based on all marks considering English and computer which may not play a relevant role in predicting the marks of student in his/her college curriculum. Therefore to articulately and in concise manner to know the performance of a student their PCM marks should be know so that knowing it and relating it to subjects offered by universities in first semester one can predict the performance of student. Right now seeing current scenario most of colleges don't have a homogenous curriculum due to which applying this system in omnipresent manner is a cumbersome task anyway.

1.2 Objective Of The System

Basically in 21st century where education hold paramount importance in everyone's life it is always good to know where one needs to work what are his or strengths and weakness so he or she can improve. In this application we strive to predict the marks of student based on his/her previous performance in relevant subjects and from data gathered by previous year students.

Once the marks of relevant subjects are obtained we train the data set in random manner so that we can achieve higher accuracy rate & based on predicted marks faculty or

mentor can guide student in better way.Moreover faculty can know strength and weakness of student and can inform his parents also, and which can be helpful in molding a way better path for future studies.

1.3 Scope Of The System

This system can be used at various institutes around the world who needs to improvise the performance of student under condition that it has previous years of data about student available. Moreover in university faculty can use it gauge the performance of student and can know which boards of students are lagging back and which area of students do well , or even students in which subject scoring less tend to score less in other subject . In school also it can be used so teachers can know where student can be improved and how he or she can perform better in competitive exams which are held on global level

Various MNC can use this so they can know that which student performing bad in given subjects is not a perfect fit for company and can filter them out on PI Basis . aSuppose a company or firm wants a student with good quantitative skills so they can predict there performance using previous year of data available and filter them out if needed

1.4 Technology and Literature Review

1.4.1 Python

Python is an interpreted high-level programming language for generalpurpose programming. Created by Guido van Rossum and first released in
1991, Python has a design philosophy that emphasizes code readability,
notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation

1.4.2 DASH

Dash is a productive Python framework for building web applications.Written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python. It's particularly suited for anyone who works with data in Python.

Through a couple of simple patterns, Dash abstracts away all of the technologies and protocols that are required to build an interactive web-based application. Dash is simple enough that you can bind a user interface around your Python code in an afternoon.

Dash apps are rendered in the web browser. You can deploy your apps to servers and then share them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently crossplatform and mobile ready.

There is a lot behind the framework. To learn more about how it is built and what motivated Dash, watch our talk from Plotcon below or read our announcement letter.

Dash is an open source library, released under the permissive MIT license. Plotly develops Dash and offers a platform for easily deploying Dash apps in an enterprise environment

1.4.3 SCIKIT LEARN

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy.The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from INRIA took leadership of the project and made the first public release on February the 1st 2010.Of the various scikits, scikit-learn as well as scikit-image were described as "well-maintained and popular" in November 2018.

2.PROJECT MANAGEMENT

2.1 Feasibility Study

2.1.1 Technical Feasibility

The project is technically feasible since there will not be much difficulty in getting the resources required for the development and maintaining the system as well. All the resources needed for the development as well as the updation is easily available on internet like python , Dash , sk-learn tools etc thus We are using the resources which are easily available.

2.1.2 Time Schedule Feasibility

The project has simple working and the basic requirement like uploading previous year data, updating records , training data sets etc. can be satisfied within the allotted time period. So, the project is feasible can be completed before the deadline.

2.1.3 Operational Feasibility

Single kind of user would use the system i.e. faculty. Files can be easily uploaded in system from local personal computer or cloud , in a single click also faculty just.To enter the student details which can easily be entered from database maintained by university an can have the desired output.

2.1.4 Implementation Feasibility

The system can be easily implemented as it uses PYTHON and DASH framework which is very common in web development. System will use the basic coding standards and implementation rules and logic. We must check that system is reliable and easy going for all users.

2.2 Project Planning

2.2.1 Project Development Approach and Justification:

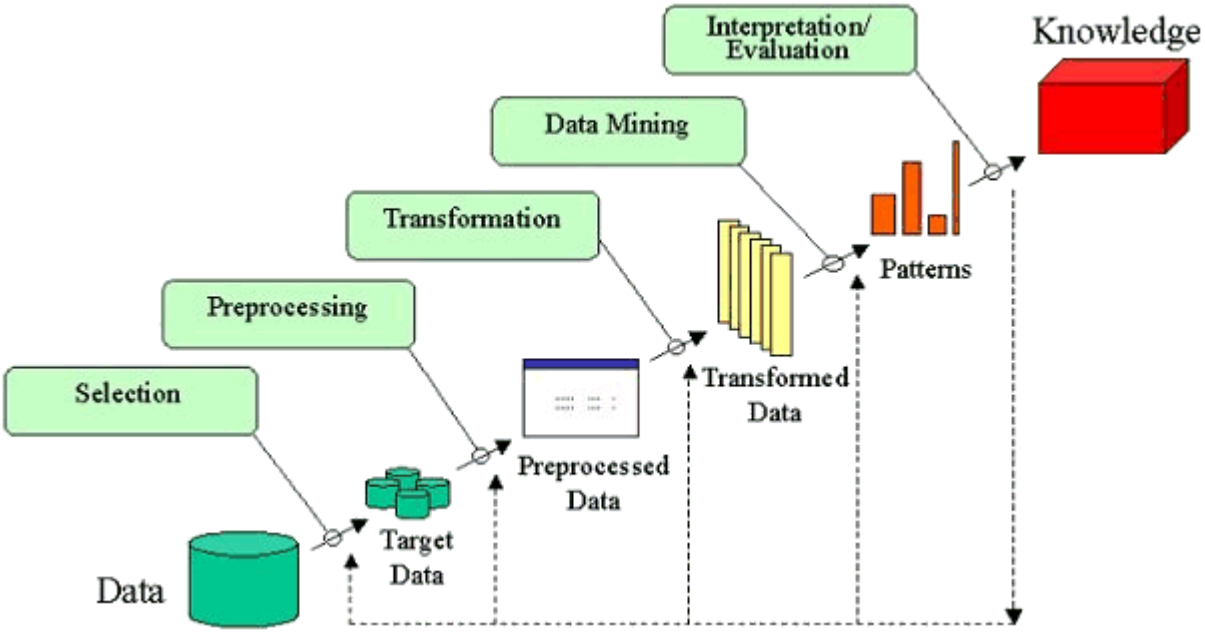


Figure 1 KDD steps

What is the KDD Process?

The term Knowledge Discovery in Databases, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "highlevel" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.

The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required preprocessing, subsampling, and transformations of that database.The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1) Developing an understanding of

- the application domain
- the relevant prior knowledge
- the goals of the end-user

2)Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.

3) Data cleaning and preprocessing.

- Removal of noise or outliers.
- Collecting necessary information to model or account for noise.
- Strategies for handling missing data fields.
- Accounting for time sequence information and known changes.

4) Data reduction and projection.

- Finding useful features to represent the data depending on the goal of the task.
- Using dimensionality reduction or transformation methods to reduce the
- effective number of variables under consideration or to find invariant representations for the data.

5) Choosing the data mining task.

- Deciding whether the goal of the KDD process is classification, regression, clustering, etc.

6) Choosing the data mining algorithm(s).

- Selecting method(s) to be used for searching for patterns in the data
- Deciding which models and parameters may be appropriate.
- Matching a particular data mining method with the overall criteria of the KDD process.

7) Data mining.

Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.

8) Interpreting mined patterns.

9) Consolidating discovered knowledge

Justification:

After feasibility study as the functional requirements were almost clear, but UI related requirements were not clear. Here we have decomposed the system into modules. That is why we decided to use iterative waterfall model which is most suitable model here i.e. if we find any difficulty in coding and testing a modification in design can be done easily.

2.2.2 Project Plan

After feasibility study as the functional requirements were almost clear which were decided by our project lead. After analyzing and thoroughly understanding the requirements of the application we planned the project.

3-tier architecture is used for this System. Here we have decomposed the system into modules. Also the internals of the individual modules are designed in greater details. Coding and Unit Testing phase is required to translate the software design into source code. Also during this phase each module is unit tested to determine the correct working of all the individual modules. Integration and System Testing phase consists of the integration of the modules in a planned manner. Here during each integration step we have tested the partially integrated system. Finally, when all the modules were successfully integrated and tested, system testing was carried out successful.

2.2.3 Milestones and Deliverables

Timely directions are always required to run a project successfully. Milestones tell the developers how far he has reached and also tell him what things are still left and how to fulfill them. Milestones may be the short report of achievement in project activity that are used by the project manager to check project progress but which are not delivered to the Clients. The deliverables are the project results that are provided to the customer. It is usually delivered at the end of some major project phases.

Table 2.1 Milestones and Deliverables

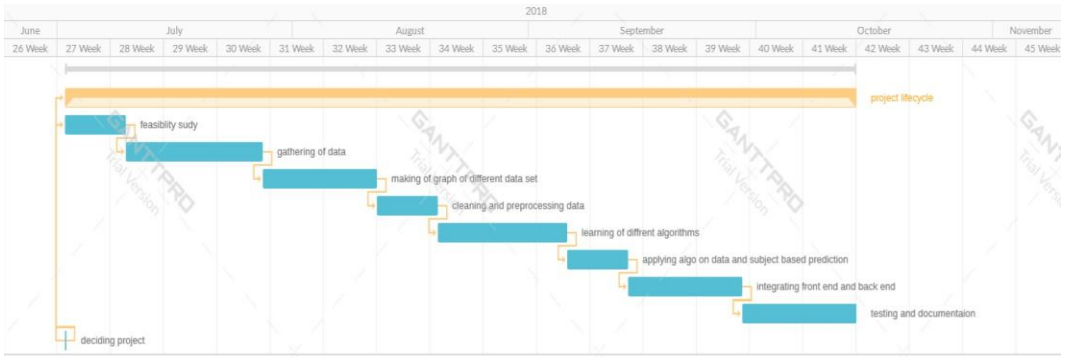
MILESTONES	DELIVERABLES	PURPOSE
Software Installation and Understanding of Technology.	Had complete knowledge of Python and dash framework	To be familiar with python programing
Gather data which was necessary for program to work	Data obtained from various places had to be pre-processed which was done over here	It gives exact understanding of which data is relevant for system and which is not.
Selecting the proper algorithm which can be used to implement on given data set and provides high accuracy	Applied random forest classifier algorithm which was providing highest accuracy of all implemented algorithms	Provides precise prediction of CPI and marks of students
Coding and Unit Testing and corrections if any.	Individually Tested and Functional Modules.	It gives the required Module.
Integration and System Testing.	The output obtained for the required functionality after implementing and doing various types of testing.	Integrated System is Ready.

2.2.4 Roles and Responsibilities

As only two members were involved in the whole team each of them had to perform all the tasks as the project proceeded through its different phases. This helped each one to develop all kinds of skills in all the phases.

2.2.5 Project Scheduling

Scheduling the project tasks is an important project planning. activity. It involves deciding which tasks should be taken up and when. In order to schedule the project activities; a software project manager needs to do the following:



Task name	Start date	End date	Duration (week)
Total estimate	02/07/2018 ...	13/10/2018 ...	14.86
project lifecycle	02/07/2018 0...	13/10/2018 0...	14.86
feasiblity sudy	02/07/2018 0...	09/07/2018 0...	1.14
gathering of data	10/07/2018 0...	27/07/2018 0...	2.57
making of gra...	28/07/2018 0...	11/08/2018 0...	2.14
cleaning and p...	12/08/2018 0...	19/08/2018 0...	1.14
learning of diff...	20/08/2018 0...	05/09/2018 0...	2.43
applying algo ...	06/09/2018 0...	13/09/2018 0...	1.14
integrating fro...	14/09/2018 0...	28/09/2018 0...	2.14
testing and do...	29/09/2018 0...	13/10/2018 0...	2.14
deciding project	02/07/2018 0...	02/07/2018 1...	0.02

Figure 2 Project scheduling

3. STUDY OF REQUIREMENTS

3.1 Study of Current System

There are many system currently available which allows prediction of performance of student currently pursuing their under-graduation in various parts of world. But all those systems are confined to some college or some curricula and can't be applied for all institutions.

3.2 Problems and weakness of current system

Current system can only predict the CPI of student precisely, but when it boils down to prediction of Marks, Subject wise then it may fail abysmally as data available is not enough so model trained has not much accuracy. Moreover system doesn't take in consideration the area of student and other relevant factors it only takes in consideration HSC,SSC marks and previous Semester CPI Of the student. System is viable only for student pursuing Btech or BE in Gujrat as data obtained is From ACPC which is admission committee for admission in Gujrat only.

3.3 User Characteristics

The major User classes in the System would be:

1. Administrator users (Faculty)

   Admin has to login using Id and password he or she may be provided with. Admin can enter the marks or CPI of students of previous semester which he or she obtained and can submit it, on clicking submit button admin is provided with the predicted value of CPI or marks of student.

3.4 Hardware and Software Requirement

3.4.1 Hardware Requirements

Processor:-Pentium Dual-Core
RAM:-2 GB
Display Monitor
Enabled Internet connection

3.4.2 Software Requirements

Server Side:

Operating system: Any operating system which has a graphical user interface
Web Server: AWS

Client Side:

The user's browser should support HTML5 and cookies must be enabled for a satisfactory user experience.

3.5 Constraints

3.5.1 Hardware limitations

There are no hardware limitations for this system because once the complete system is developed care would be taken while deploying system so necessary pre-requisites are met.

3.5.2 Interface to other applications

There are no other systems that use this application as an interface.

3.5.3 Reliability Requirements

The application does demand much reliability and it is fully assured that the particular information about the user should be secured and flow is maintained and accessed according to the rights.

3.5.4 Criticality of the Application

The application deals with the user's personal tasks so the task and respective details should be highly confidential and in proper flow.

3.5.5 Security and Safety considerations

The system provides a tight security to every students' records. students' records are secured by unique id are stored to database.

3.6 Assumptions and Dependencies

Assumptions are described as follows:

- User has sufficient privileges to access internet.
- Server is running smoothly.
- Database updates are giving expected and accurate results.
- Admin is provided with necessary Id and password.
- Admin has sufficient knowledge of accessing PC.

4.SYSTEM ANALYSIS

4.1 Requirements of system

4.1.1 System will authenticate Admin.

Input: User Credentials.

Output: Home Page as per user or error message.
Description: There is one type of user. Based on user's role in the system, system will validate the user and show homepage or an error message.

4.1.2 System will provide facility to filter students

Input: Student Details or Marks or CPI.

Output: Tuples of student data.

Description: Admin can view students data based on filters applied and data provided.

4.1.3 System will provide facility to predict marks or CPI

Input:-Student HSC,SSC,SEM1,SEM2,SEM3,SEM4 CPI

Ouput:-SEM 5 CPI

Description: Admin can provide marks of respective semester and predict marks of further semester based on data provided

4.1.4 System will provide facility to visualize marks or CPI

Input:-Student Data file containing all relevant data

Output:-Bar graph, Pie chart, Line graph

Description: Admin can provide whole file of data which when executed will dynamically provide output of different kinds of graphs

5.METHOD OF IMPLEMENTATION

5.1 Developing an understanding of requirements :-

We need to analyze the requirements of the user and customers what he specifically needs from the system. Only terse understanding can be then used in development of the project.If one doesn't carry out this step properly then it may cause plethora of mistakes in system.

5.2 Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.

We have selected data sets provided by ACPC which is admission committee responsible for providing merit ranks and admission is specific colleges. Here data which was obtained was not perfectly homogenous, data set was also taken form college institute.Dataset which was relevant was only selected out of the data available.

roll no sem5	StudentID	student name	SSC %-all subjects	SSC Board	HSC %-all subjects	HSC Board	dip college,university	DOB(DD/MM/YY)	1-SPI	2-SPI	2-CPI	3-SPI	3-CPI	4-SPI	4-CPI	5-SPI	5-CPI	6-SPI	6-CPI
IT001	15ITUON035	ISHAN BHATT	9.40	CBSE	80	CBSE		30/12/1997	7.2	6.55	6.88	6.16	6.16	6.2	6.18	6.42	6.26	7	6.44
IT002	15ITUBS073	JAYDEV PATEL MAHENDRABHAI	8.8	CBSE	83	CBSE		30/04/1997	7.29	6.92	7.1	6.68	6.68	6.67	6.67	7.35	6.9	7.4	7.02
IT003	15ITUOS063	KEERTHANA K	95.00	CBSE	92%	CBSE		4/11/97	8.96	8	8.48	9.24	9.24	8.75	8.99	8.69	8.89	8.2	8.72
IT004	15ITUOS077	KRUPAL PRADIPKUMAR PATEL	91.33	ICSE	92.8	ISC		18/02/97	8.84	9.1	8.97	9	9	9.49	9.25	9.46	9.32	9.4	9.34
IT005	15ITUOS068	MONISH SHAILESHBHAI SHAH	10	CBSE	92.6	CBSE		18/04/1997	8.43	8.73	8.58	8.8	8.8	9.29	9.05	8.77	8.95	9.2	9.01
IT006	15ITUOS114	ROHAN RAMESH RUDANI	9	CBSE	91.2	CBSE		11/8/97	7.49	7.43	7.46	4.52	NaN	7.41	6.97	6.92	6.95	7.6	7.11
IT007	15ITUOS071	ATAL NISHANT MANISH	9.8	CBSE	92.6	CBSE		19/02/98	7.76	7.63	7.69	7.08	7.08	5.82	NaN	6.85	6.97	7	6.98
IT008	15ITUOS096	BHATT DARSH HITEN	87.16	GSHSEB	73.8	GSHSEB		28/06/1997	6.1	6.12	6.11	4.2	NaN	5.08	NaN	6.27	6.24	5.4	NaN
IT009	15ITUBS093	BHUTAIYA ARUSHI KIRITBHAI	84.83	GSEB	72	GSEB		12/3/98	6.76	6.39	6.57	5.36	NaN	5.9	NaN	6.46	6.44	6.8	6.53
IT010	15ITUOF076	BODARA GAURAV BHARATBHAI	90.83	GSEB	87.8	GHSEB		26/09/97	8.16	7.45	7.81	7.44	7.44	8.9	8.18	8.12	8.16	8.4	8.22

Figure 3 Raw dataset

5.3 Data cleaning and preprocessing.

Once we had details of all the students, we then segmented the training dataset further, considering various feasible splitting attributes, i.e. the attributes which would have a higher impact on the performance of a student. For instance, we had considered 'location' as a splitting attribute, and then segmented the data according to students' locality.

A snapshot of the student database is shown in Figure 2. Here, irrelevant attributes such as students residential address, name, application ID, etc. had been removed. For example, the admission date of the student was irrelevant in predicting the future performance of the student. The attributes that had been retained are those for merit score or marks scored in entrance examination, gender, percentage of marks scored in Physics, Chemistry and Mathematics in the board examination of class XII and admission type.

- Removal of noise or outlierCollecting necessary information to model or account for noise.
- Strategies for handling missing data fields.
- Accounting for time sequence information and known changes.

SSC Board	HSC Board	1SPI_	2SPI_	3SPI_	4SPI_	5SPI_	HSC %-all s	SSC%all su	Results
0	0	7.2	6.55	6.16	6.2	6.42	80	84.6	0
0	0	7.29	6.92	6.68	6.67	7.35	83	79.9	1
0	0	8.96	8	9.24	8.75	8.69	92	95	2
2	2	8.84	9.1	9	9.49	9.46	92.8	91.33	3
0	0	8.43	8.73	8.8	9.29	8.77	92.6	90	2
0	0	7.49	7.43	4.52	7.41	6.92	91.2	81	0
0	0	7.76	7.63	7.08	5.82	6.85	92.6	88.2	0
1	1	6.1	6.12	4.2	5.08	6.27	73.8	87.16	0
1	1	6.76	6.39	5.36	5.9	6.46	72	84.83	0

Figure 4 preprecessed dataset

5.4 Data reduction and projection.

- Finding useful features to represent the data depending on the goal of the task.
- Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.

5.5 Choosing the data mining task.

- Deciding whether the goal of the KDD process is classification, regression, clustering, etc.

5.5.1 Classification

Classification is a data mining technique that maps data into predefined groups or classes. It is a supervised learning method which requires labelled training data to generate rules for classifying test data into predetermined groups or classes [2]. It is a two-phase process. The first phase is the learning phase, where the training data is analyzed and classification rules are generated. The next phase is the classification, where test data is classified into classes according to the generated rules. Since classification algorithms require that classes be defined based on data attribute values, we had created an attribute “class” for every student, which can have a value of either “Pass” or “Fail”.

5.5.2 Clustering

Clustering is the process of grouping a set of elements in such a way that the elements in the same group or cluster are more similar to each other than to those in other groups or clusters [1]. It is a common technique for statistical data analysis used in the fields of pattern recognition, information retrieval, bioinformatics, machine learning and image analysis. Clustering can be achieved by various algorithms that differ about the similarities required between elements of a cluster and how to find the elements of the clusters efficiently. Most algorithms used for clustering try to create clusters with small distances among the cluster elements, intervals, dense areas of the data space or particular statistical distributions.

5.5.3 Selecting Classification over Clustering

In clustering, classes are unknown apriori and are discovered from the data. Since our goal is to predict students' performance into either of the predefined classes - "Pass" and "Fail", clustering is not a suitable choice and so we have used classification algorithms instead of clustering algorithms.

5.6 Choosing the data mining algorithm(s).

- Selecting method(s) to be used for searching for patterns in the data.Deciding which models and parameters may be appropriate.
- Matching a particular data mining method with the overall criteria of the KDD process.
    - **Random forest classifier**

        - One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. I will talk about random forest in classification, since classification is sometimes considered the building block of machine learning.Random Forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, you don't have to combine a decision tree with a bagging classifier and can just easily use the classifier-class of Random Forest. Like I already said, with Random Forest, you can also deal with Regression tasks by using the Random Forest regressor**.** Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

- **KNN**
  - A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor. It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.
- **Neural network**
  - Artificial Neural Network algorithms are inspired by the human brain. The artificial neurons are interconnected and communicate with each other. Each connection is weighted by previous learning events and with each new input of data more learning takes place. A lot of different algorithms are associated with Artificial Neural Networks and one of the most important is Deep learning. An example of Deep Learning can be seen in the picture above. It is especially concerned with building much larger complex neural networks.

5.7 Data mining.

Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.

5.8 Interpreting mined patterns.

Once pattern are mined then we need to make sure that mined patterns are providing the useful result and is relevant. It may happen that patterns once mined may not be useful or even futile. Interpreting pattern in different scenario is also important aspect of KDD steps

5.9 Consolidating discovered knowledge.

Once pattern are interpreted they are needed to be consolidated with the system. And should be integrated in a way so that it can be used with other interfacing systems also.

#######CODE SNIPPET ############

```
rf = RandomForestClassifier(n_estimators = 500, verbose = 10, n_jobs = 1)

rf.fit(X_train,y_train)

filename='finalized_model_CPI.sav'pickle.dump(rf,
open(filename,'wb'))

pkp=loaded_model.predict(X_test)
```

6.TESTING

6.1 Testing Plan

The testing technique that is going to be used in the project is black box testing. In black box testing the expected inputs to the system are applied and only the outputs are checked.

6.2 Testing Strategy

The development process repeats this testing sub-process a number of times for the following phases.

a) Unit Testing.
b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

6.3 Testing Methods

Black box and White box testing:
In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the.

structural system coverage in white box testing. For this purpose tools or compilers that can insert test probes into the programs can be used.

6.4 Test Cases

CPI based model

Table 1 Accuracy if different algorithms for CPI model

Algorithm applied	Prediction Accuracy
1)K-nearest neighbour	52.83%
2)Random forest classifier	76.19%
3)Neural Network	26.19%

```
In [31]: rf.score(X_test,y_test)

[Parallel(n_jobs=8)]: Done   2 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done   9 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  16 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  25 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  34 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  45 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  56 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  69 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  82 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  97 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 112 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 129 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 146 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 165 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 184 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 205 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 226 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 249 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 272 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 297 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 322 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 349 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 376 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 405 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 434 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 465 tasks       | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 500 out of 500 | elapsed:    0.0s finished
Out[31]: 0.7619047619047619
```

Figure 5 Accuracy of random forest classifier

```
In [32]: from sklearn.neural_network import MLPClassifier
         clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                             hidden_layer_sizes=(5, 2), random_state=1)

         clf.fit(X_train,y_train)

Out[32]: MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9,
                beta_2=0.999, early_stopping=False, epsilon=1e-08,
                hidden_layer_sizes=(5, 2), learning_rate='constant',
                learning_rate_init=0.001, max_iter=200, momentum=0.9,
                nesterovs_momentum=True, power_t=0.5, random_state=1, shuffle=True,
                solver='lbfgs', tol=0.0001, validation_fraction=0.1, verbose=False,
                warm_start=False)

In [33]: clf.score(X_test,y_test)

Out[33]: 0.2619047619047619
```

Figure 6 Accuracy of Neural Network

```
In [43]: from sklearn.neighbors import KNeighborsClassifier
         neigh = KNeighborsClassifier(n_neighbors=2)
         neigh.fit(X_train, y_train)
         neigh.score(X_test,y_test)

Out[43]: 0.5238095238095238
```

Figure 7 Accuracy of nesrest neighbour

Subejct based Model

Table 2 Accuracy of different algorithms for Subject model

Algorithm Applied	Prediction accuracy
1)K-nearest neighbour	18%
2)Random forest classifier	18%
3)Neural Network	14%

```
In [128]: rf.score(X_test,y_test)
[Parallel(n_jobs=8)]: Done   2 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done   9 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  16 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  25 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  34 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  45 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  56 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  69 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  82 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done  97 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 112 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 129 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 146 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 165 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 184 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 205 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 226 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 249 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 272 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 297 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 322 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 349 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 376 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 405 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 434 tasks      | elapsed:    0.0s
[Parallel(n_jobs=8)]: Done 465 tasks      | elapsed:    0.1s
[Parallel(n_jobs=8)]: Done 500 out of 500 | elapsed:    0.1s finished

Out[128]: 0.18
```

Figure 8 Accuracy of Random forest classifier(subject)

```
In [47]:
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=2)
neigh.fit(X_train, y_train)
neigh.score(X_test,y_test)

Out[47]: 0.18
```

Figure 9 Accuracy for Neural network algorithm(subject)

```
In [28]:
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                    hidden_layer_sizes=(5, 2), random_state=1)

clf.fit(X_train,y_train)

Out[28]: MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(5, 2), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=1, shuffle=True,
       solver='lbfgs', tol=0.0001, validation_fraction=0.1, verbose=False,
       warm_start=False)

In [29]:
clf.score(X_test,y_test)

Out[29]: 0.14
```

Figure 10 Accuracy of K Nearest Neighbour Algorithm(subject)

7.USER MANUAL

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system.

It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies.

User guides are most commonly associated with electronic goods, computer hardware and software.

Our user guides contain both a written guide and the associated images. In the case of our application, it is usual to include screenshots of how the program should look. The language used is matched to the intended audience.

Data depicting page

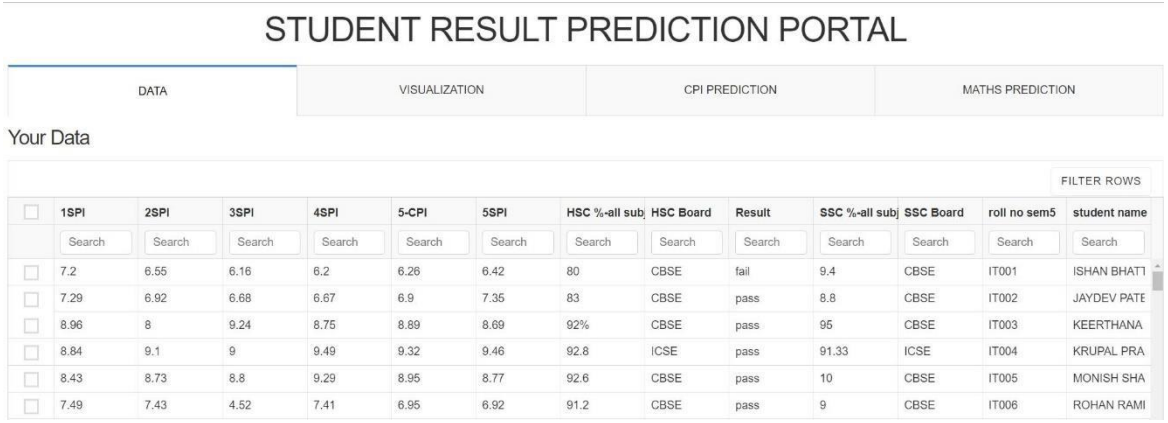


Figure 11 First tab

Home Page

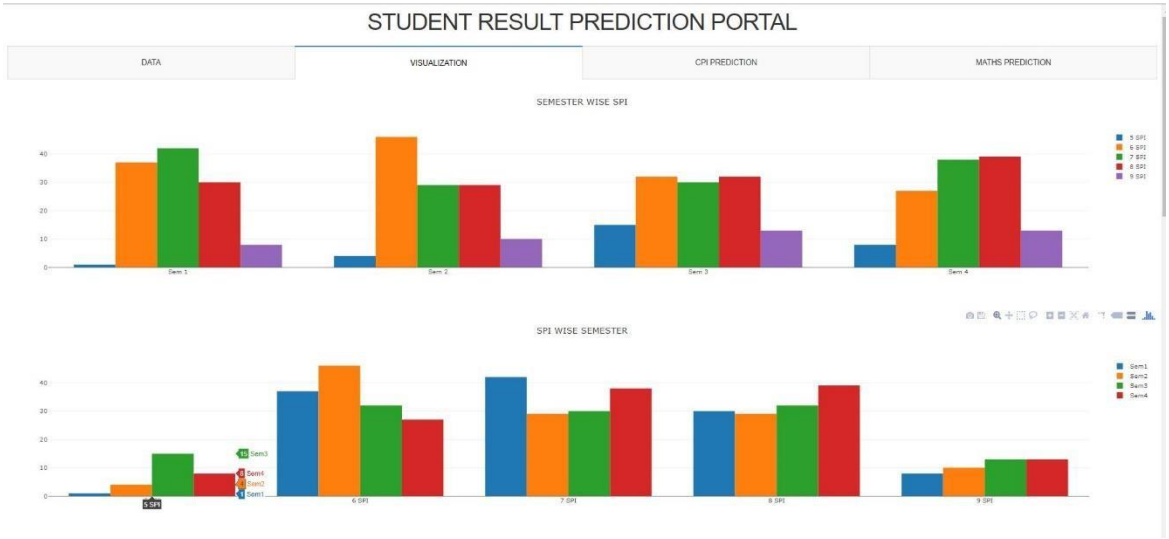


Figure 12 Home page

Visualization tab

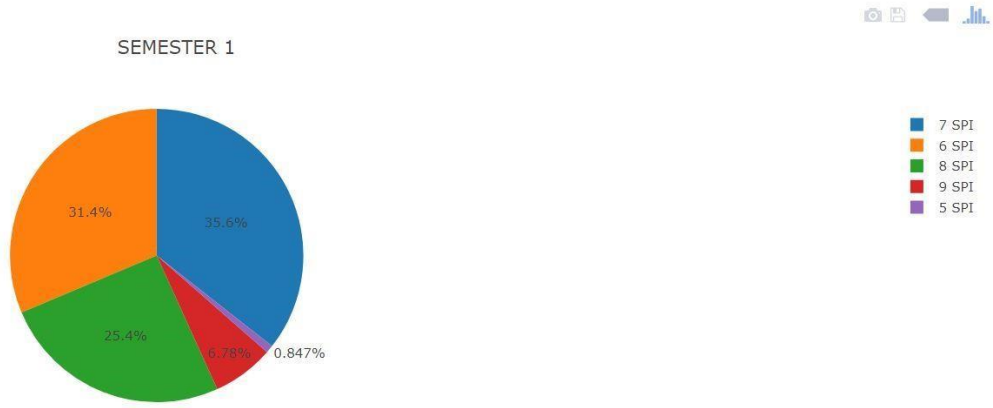


Figure 13 visualization Page

Visualization tab

Figure 14 visualization Page

Visualization tab

Figure 15 visualization Page Sem 3

Visualization tab

Figure 16 visualization Page sem 4

Add details Page

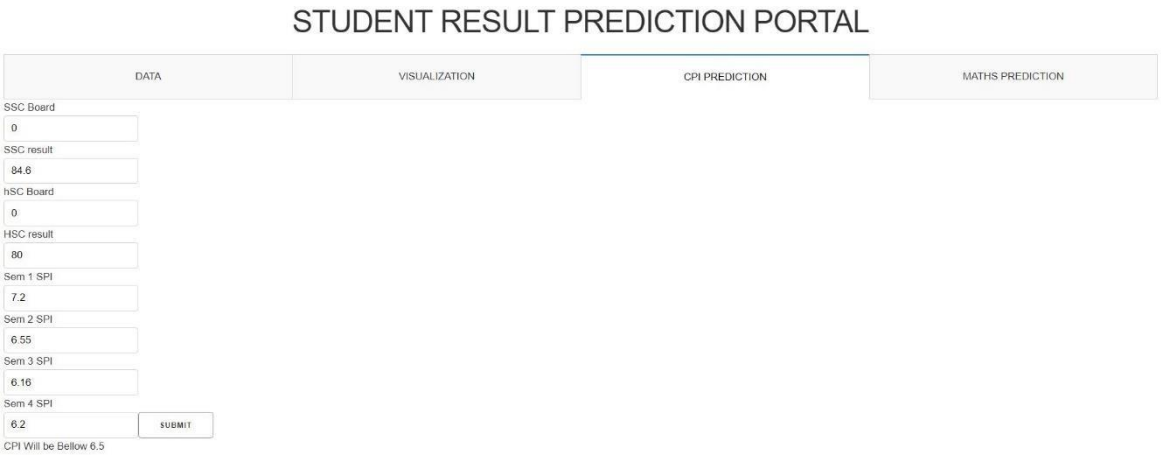


Figure 17 CPI Prediction page

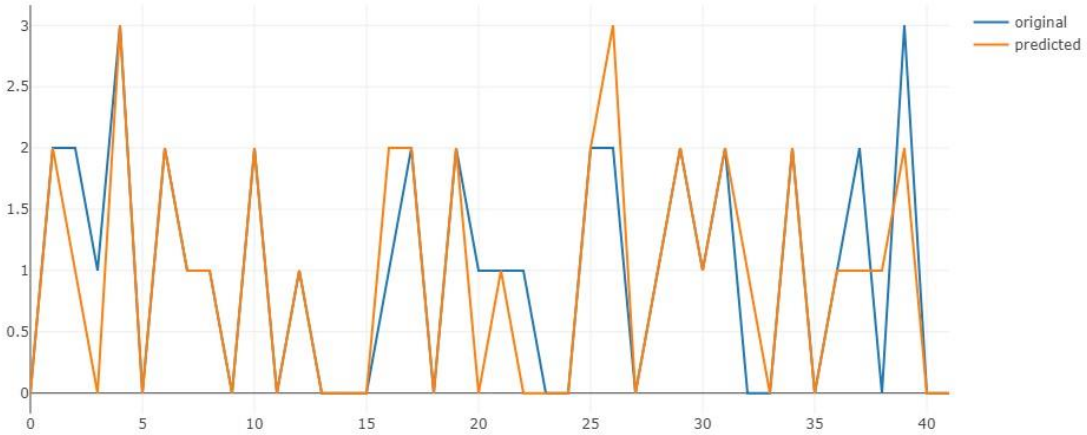


Figure 18 CPI predicted vs Original

Subject Wise prediction

Figure 19 Subject wise prediction Page

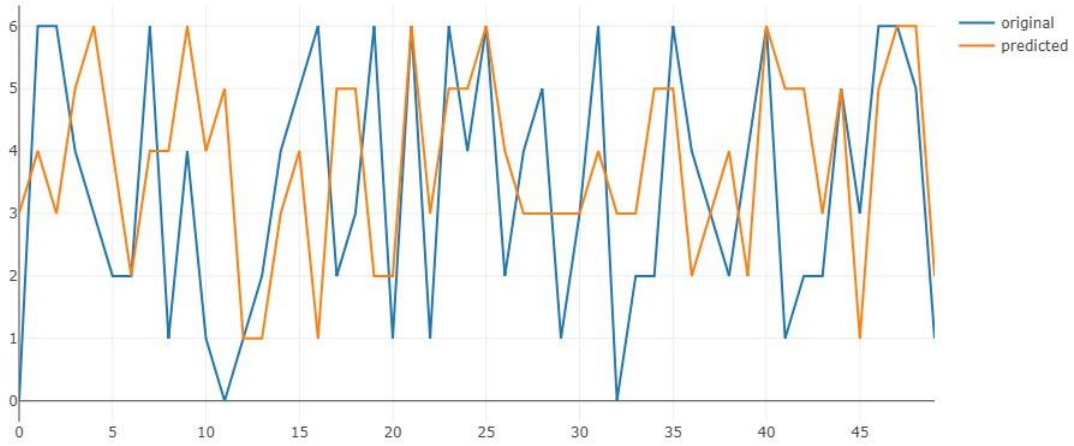


Figure 20 predicted maths marks vs original marks

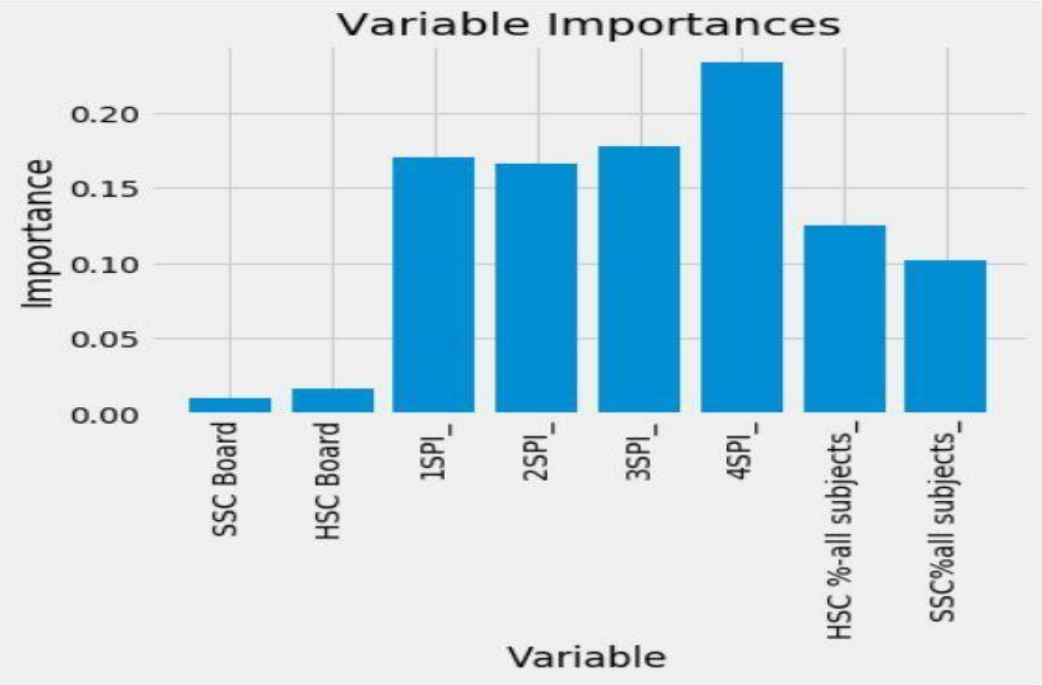


Figure 21 variable importance

```
feature_list = list(tp.columns)

# Get numerical feature importances
importances = list(rf.feature_importances_)
# List of tuples with variable and importance
feature_importances = [(feature, round(importance, 2)) for feature, importance in zip(feature_list, importances)]
# Sort the feature importances by most important first
feature_importances = sorted(feature_importances, key = lambda x: x[1], reverse = True)
# Print out the feature and importances
[print('Variable: {:20} Importance: {}'.format(*pair)) for pair in feature_importances];
```

```
Variable: 4SPI_               Importance: 0.23
Variable: 3SPI_               Importance: 0.18
Variable: 1SPI_               Importance: 0.17
Variable: 2SPI_               Importance: 0.17
Variable: HSC %-all subjects_ Importance: 0.12
Variable: SSC%all subjects_   Importance: 0.1
Variable: HSC Board           Importance: 0.02
Variable: SSC Board           Importance: 0.01
```

Figure 22 Numerical data

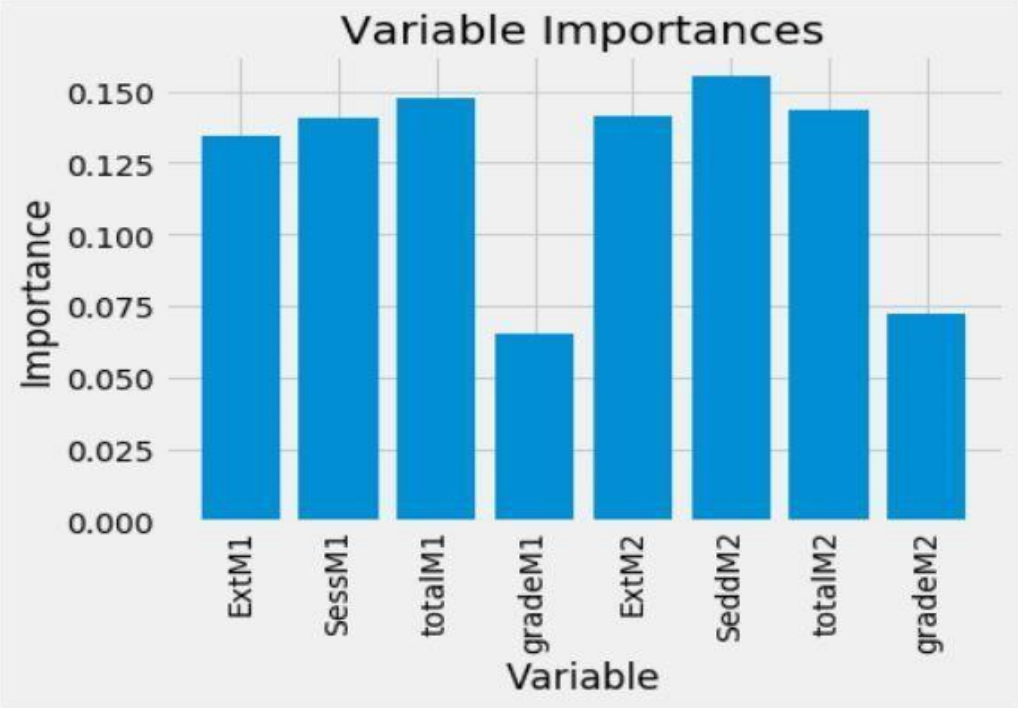


Figure 23 variable importance (maths)

```
In [144]: feature_list = list(train.columns)

          # Get numerical feature importances
          importances = list(rf.feature_importances_)
          # List of tuples with variable and importance
          feature_importances = [(feature, round(importance, 2)) for feature, importance in zip(feature_list, importances)]
          # Sort the feature importances by most important first
          feature_importances = sorted(feature_importances, key = lambda x: x[1], reverse = True)
          # Print out the feature and importances
          [print('Variable: {:20} Importance: {}'.format(*pair)) for pair in feature_importances];

          Variable: totalM2              Importance: 0.16
          Variable: totalM1              Importance: 0.15
          Variable: ExtM2                Importance: 0.15
          Variable: ExtM1                Importance: 0.14
          Variable: SessM1               Importance: 0.14
          Variable: SeddM2               Importance: 0.14
          Variable: gradeM1              Importance: 0.07
          Variable: gradeM2              Importance: 0.07
```

Figure 24 variable importance numerical

8.LIMITATION AND FUTURE ENHANCEMENT

8.1 Limitation

1. It can not be applied in curricula of other institution.
2. It can’t predict marks of subject precisely.
3. It doesn’t validate with database if cpi entered are legit or not.
4. Subject wise accuracy is low because of unavailability

8.2 Future Enhancement

System can be further improved by providing model sufficient amount of data for predicting marks. More graphs can be made which can provide better visualization of data and can also provide a better perspective. If system is provided with all other attributed like English, Computer subject marks there can be more relevance of prediction. Website can be made more aesthetic if time permits. Further facility of uploading files and connecting system with proper database scan be provided.

9.CONCLUSION AND DISCUSSION

9.1 Conclusion

Project summarizes by predicting the CPI or marks of student under the condition the condition that information provided is mannered way and is valid,If invalid information is provided then system will prompt to admin to enter proper information or will show a haphazard output.

9.2 Discussion

9.2.1 Problems Encountered and Possible Solutions

1. Major problem encountered was to get data set of students of with their HSC,SSC marks and their respective percentile.
2. Which algorithm should be applied to data set was a big question as under different conditions different algorithms are applied.
3. Data available for prediction of Subject wise model is in Moot amount so accuracy of prediction is very low.

9.2.2 Summary of Project Work

This project taught us lot of things say from learning whole new language to advance algorithms of Machine learning.Doing project made us familiar with various Practical facets of which we only had theoretical base. It provided various insights Machine learning algorithms and made us Familiar with visualization of data in dash Framework.

REFERENCES

Web Resources:

1. www.w3schools.com
2. www.google.com
3. www.stackoverflow.com
4. www.vidyaAnalytics.com
5. http://scikit-learn.org/
6. https://dash.plot.ly/