

A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN ENGINEERING PROGRAM

**A Project Report submitted to Department of Computer Science and Engineering for partial
fulfillments of the requirements for the award of
the degree of**

BACHELOR OF TECHNOLOGY

**in
Computer Science and Engineering**

by

V. Harshini (1215316558)

A.S.Vinay (1215316501)

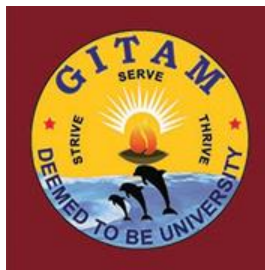
M.Vamsi (1215316535)

J. Bhagavath (1215316526)

Under the esteemed guidance of

Mr. Vikas B

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

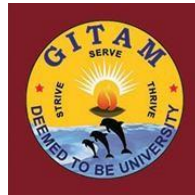
GITAM INSTITUTE OF TECHNOLOGY

GITAM

Visakhapatnam – 530045

2019-20

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM
INSTITUTE OF TECHNOLOGY
GITAM
(Deemed to be University)



DECLARATION

I/We, hereby declare that the project report entitled “**A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN ENGINEERING PROGRAM**” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration No(s).

Name(s)

Signature(s)

1215316558

V.HARSHINI

1215316501

A.S.VINAY

1215316535

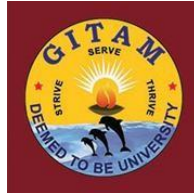
M.VAMSI

1215316526

J.BHAGAVATH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM
INSTITUTE OF TECHNOLOGY
GITAM

(Deemed to be University)



CERTIFICATE

This is to certify that the project report entitled “**A MACHINE LEARNING APPROACH FOR TRACKING AND PREDICTING STUDENT PERFORMANCE IN ENGINEERING PROGRAM**” is a bonafide record of work carried out by **V.Harshini(1215316558), A.S.Vinay(1215316501), M.Vamsi(1215316535), J.Bhagavath(1215316526)** students submitted in partial fulfillment of requirements for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide

Mr. B. Vikas

Assistant Professor

Head of the Department

Dr. K. Thammi Reddy

Professor

ACKNOWLEDGEMENT

It brings immense pleasure to put forth our training report. It is a great pleasure and an opportunity for us to express our deep sense of gratitude to all who have made it possible for us to accomplish this project.

Firstly, we would like to thank our project guide **Mr. VIKAS B, Assistant Professor** for his stimulating guidance and profuse assistance. We have learnt a lot, working under him and we will always be indebted to him for his value addition in us. We shall always cherish our association with him for his guidance, encouragement and valuable suggestions throughout the progress of this work. We consider it a privilege to work under his guidance and constant support.

We also extend our gratitude to project evaluator **Mr. G. APPARAO**, Professor and **Mr. N.SURESH KUMAR**, Assistant Professor, Dept. of CSE, for their guidance during the course of our project.

We are extremely grateful to **Dr. K. Thammi Reddy**, Professor and Head, Dept. of CSE for giving us an opportunity to explore the vast field of Data Mining through this project.

We extend our deep regards to our **Families** for their endless love, kindness and support which helped us to see the gold line in our dark cloud

We thank the staff of our department for their good wishes, help and constructive criticism which led to the successful completion of the first phase our project.

TABLE OF CONTENTS

1. Abstract
2. Introduction
 - 2.1 Need
 - 2.2 Basic Concept
 - 2.3 Applications
3. Review of Literature
 - 3.1 ID3 classification algorithms
 - 3.2 Gradient Descent algorithm
 - 3.3 Classification algorithms
 - 3.4 Logistic Regression
 - 3.5 Classification Algorithms
 - 3.6 Decision tree algorithm C4.5
 - 3.7 Genetic Programming Algorithm
 - 3.8 Data mining techniques
 - 3.9 ID3 and C4.5 classification algorithms
4. Report on the Present Investigation (Existing Systems)
5. Aim and Objectives
 - 5.1 Aim
 - 5.2 Objectives
6. Problem Statement
7. Proposed System for Project
8. Requirement Analysis (SRS)
 - 8.1 Functional Requirements
 - 8.2 Non-Functional Requirements
9. Scope (Feasibility of Project)
 - 9.1 Scope
 - 9.2 Feasibility
 - 9.2.1 Operational Feasibility
 - 9.2.2 Technical Feasibility
 - 9.2.3 Economic Feasibility

10. Methodology
 - 10.1 Gathering Data
 - 10.2 Data Preparation
 - 10.3 Choosing a model
 - 10.4 Training
 - 10.5 Evaluation
 - 10.6 Parameter Tuning
 - 10.7 Prediction
11. Hardware and Software Requirements
 - 11.1 Hardware Requirements
 - 11.2 Software Requirements
12. Testing
 - 12.1 Testing
 - 12.1.1 Unit Testing
 - 12.1.2 Integration Testing
 - 12.1.3 Functional Testing
 - 12.1.4 Performance Testing
 - 12.1.5 Load Stress Testing
 - 12.2 Test Cases
13. Models and Their Accuracy Checking
 - 13.1 Decision Tree Classifier
 - 13.2 Decision tree with entropy
 - 13.3 Gini Index
 - 13.4 Support Vector Machine
 - 13.5 Xgboost
14. Advantages and Limitations
 - 14.1 Advantages
 - 14.2 Limitations
15. Applications and Future Scope
 - 15.1 Applications
 - 15.2 Future Scope
16. Conclusion
17. Appendix

a. Concepts Related to Neural Networks

- i. What are Neural Networks made of?
- ii. How does a Neural Network learn?
- iii. How does Neural Network work in Practice?
- iv. What is Classification in Machine Learning?

17.1.5 List of Common Machine Learning Algorithms

18. Outputs

18.1 Probability of placement

18.2 T-distributed Stochastic Embedding

18.3 Principal Component Analysis

19. References

LIST OF FIGURES

Figure 7.1: System Architecture

Figure 10.1: Training Model

Figure 10.2: Parameter Tuning

Figure 13.1: Decision Tree Classifier

Figure 13.2: Decision tree with entropy

Figure 13.3: Gini Index

Figure 13.4: Support Vector Machine

Figure 13.5: Xgboost

Figure 18.1: Probability of placement

Figure 18.2: TSNE

Figure 18.3: PCA

1. ABSTRACT

Engineering students are skeptical about what they want to pursue after graduation. With wide options available, ranging from campus recruitments to Masters, students are perplexed, adding factors like salaries and different job opportunities makes it even worse. There aren't any reliable platforms where a student can predict the outcomes from the start of engineering and take actions to bridge this gap for a better future.

Students studying in engineering colleges feel the exigency to know where they stand in comparison to others, and what kind of placement they would get. The training and placement offices come in the picture when a student enters final year, but they are of no use to a student planning for future studies. Prediction about the student's performance is an integral part of an education system, as the overall growth of the student is directly proportional to the success rate of the students in their examinations and extra-curricular activities. Therefore, there are many situations where the performance of the student needs to be predicted, for example, in identifying weak performing students and taking actions for their betterment.

The students have no platform to check their current position and build on their strengths. The platforms currently available, have not been trained on real and complete data sets, and do not learn from their wrong predictions which reduces the accuracy, in the long term. To achieve a better accuracy and a system that learns with every wrong prediction it has made, we intend to use Neural Networks, which will cause a continuous accuracy growth. We aim to develop one, complete, robust platform, where students can check their current status, and the range of placements they would get, on an easy to use web application. To ensure effective results, the model will be trained on a real data set and a vast number of qualitative as well as quantitative parameters will be considered.

2. INTRODUCTION

Campus placement of a student plays a very important role in a college. Campus placement is a process where companies visit colleges and identify students who are talented and qualified, before they complete their graduation. Therefore, taking a wise career decision regarding the placement after completing a particular course is crucial in a student's life. An educational institution contains a large number of student records. Therefore, finding patterns and characteristics in this large pool of data, will help find parameters that are the most important for this placement procedure.

The prediction of engineering students, about where they can be placed, from the second year and onwards, will help to improve efforts of student for proper progress. [1] It will help teachers to take proper attention towards the progress of the student during the course of time. It will help to build reputation of the institute for having such a sophisticated system in place which helps the students to train and practice for campus placements. The present study concentrates on helping the students, bridging the gap between the industry and the curriculum, and showing them the path to a better future. We apply data mining and machine learning techniques using Artificial Neural Networks, in order to interpret the potential of the student.

Data Mining refers to extracting or mining useful patterns from a large database. It is knowledge discovery in large amount of data. Neural networks and Fuzzy Inference System is a part of Soft Computing, that works well with low level computing, gaining experience and knowledge from its mistakes, and the later works well with the irregularities and the incompleteness of the data. Neural Networks has various applications in number of sectors. Whilst data mining tools can be used to find patterns in the large set of data which can help understand, business requirements, market analysis and management. Learning using neural networks can be classified into three types, supervised learning and unsupervised learning. Supervised learning is so named because the data scientist acts as a guide to teach the algorithm what conclusions it should come up with. It's similar to the way a child might learn arithmetic from a teacher. Unsupervised machine learning is more closely aligned with what some call true artificial intelligence — the idea that a computer can learn to identify complex processes and patterns without a human to provide guidance along the way. We're going to use unsupervised learning techniques to provide guidance for the students.

2.1 NEED

The questions this work can provide the solutions to, can be given as follows:

1. What are the types of students the college has according to their academic scoring?
2. Predict in advance, the placement status of the pre - final year students.
3. What should the students learn next, to have a better chance of placement?
4. What are the clusters of domains, students in a college fall into?
5. Provide info to the hiring companies and prompt questions and proof of the student details and help companies graphs and visuals to filter students.

So, we aim to fix the above vulnerabilities and answer those questions in our system, using Artificial neural networks.

2.2 BASIC CONCEPT

In this technical world many different techniques are used by humans for different purposes. There are many softwares and applications that are developed for reducing human effort. Data mining techniques and neural networks can be used to find patterns in large databases, and to guide decisions about future activities. It's expected that by using neural networks, the model will learn on its own, and work efficiently even with minimal input from the user to recognize. The model can be useful to understand the unexpected and provide an analysis of data followed by decision-making which is examined and it ultimately leads to strategic decisions and business intelligence. The simplest word for knowledge extraction and exploration of volume data is very high and the more appropriate term is, "Exploring the hidden knowledge of the database." This process includes preparation and interpretation of results.

2.3 APPLICATION

There are various applications of this system, few of them being

1. Student's will have an idea of where they stand and what to do next to bridge the gap and become better.
2. Student's will have a clear option which will help reduce the ambiguity in their mind.
3. The college will have the statistics of all the students and what are the different domains they fall into.
4. The college, will be able to take decisions to improve students and have better, insights of the students.
5. The student's will get their resume based on the data they feed.

6. The corporate end, will be able to filter the students and download the resumes of the students, according to their needs.
7. The corporate end and the college end will be able to post there, requirements and send messages directly to the student, or maybe even globally.
8. The corporate end there will be interview questions that will be prompted, different for different students based on the student resume.

3.REVIEW OF LITERATURE

3.1 ID3 CLASSIFICATION ALGORITHM

Hitarthi Bhatt identified relevant attributes based on quantitative and qualitative aspects of a student's profile such as CGPA, academic performance, technical and communication skills and designed a model which can predict the placement of a student using ID3[1].

3.2 GRADIENT DESCENT ALGORITHM

One of the most prominent work on prediction of placement for students has been cited by Ajay Shiv Sharma, Swaraj Prince, Shubham Kapoor and Keshav Kumar where they presented the development of placement predictor system (PPS) using logistic regression model. They used Machine learning technique to design and implement a logistic classifier that predicts the probability of the student to get placed along with Gradient Descent algorithm. The results are generated from an open source GNU Octave programming tool. The developed model has been applied to predict the placement of students at training and placement office (TPO) [5].

3.3 CLASSIFICATION ALGORITHM

S. Taruna and Mrinal Pandey implemented an empirical analysis on predicting academic performance by using classification techniques or mapping of data items into predefined groups and classes using supervised learning. They compared five classification algorithms namely Decision Tree, Naïve Bayes, Naïve Bayes Tree, K-Nearest Neighbour and Bayesian Network algorithms for predicting students' grade particularly for engineering students using a four-class prediction problem [6].

State of the art regression algorithms Kotsiantis and Pintelas, 2005 predicted the student marks (pass and fail classes) using the regression methods and available previous data. The scope of this work compares some of the state of the art regression algorithms in the application domain of predicting students' marks. A number of experiments have been conducted with six algorithms, which were trained using datasets provided by the Hellenic Open University [7].

3.4 LOGISTIC REGRESSION

Saha and Goutam applied logistic regression method on the examination result data and analyzed the data under the University Grant Commission sponsored project entitled - Prospects and Problems of Educational development (Higher Secondary Stage) in Tripura - An In-depth Study [8].

3.5 CLASSIFICATION ALGORITHMS

Syed Tanveer Jishan, Raisul Islam Rashu, Naheena Haque and Rashedur M Rahman predicted student's performance using attributes such as Cumulative Grade Point Average, Quiz, Laboratory, Midterm and Attendance marks. Also, in order to improve the prediction model, they introduced some preprocessing techniques so that the prediction model provides with more precise results by applying three classification algorithms, e.g., Naïve Bayes, Decision Tree and Neural Network [9].

3.6 DECISION TREE ALGORITHM C4.5

Zhiwu Liu and Xiuzhi Zhang used decision tree algorithm C4.5 to establish a classification rule and an analysis-forecasting model for students' marks. They described how the analysis-forecasting result can be used to find out the factors which can affect students' marks, so some negative learning habits or behaviors of students can be revealed and corrected in time and the teaching effect of the teacher can be checked, the teaching management can also be assisted [10].

3.7 GENETIC PROGRAMMING ALGORITHM

Carlos Márquez-Vera, Alberto Cano, Cristóbal Romero and Sebastián Ventura predicted the student's failure at school using genetic programming algorithm and different data mining approaches using cost sensitive classification in order to resolve the problem of classifying imbalanced data. A genetic programming algorithm and different data mining approaches were proposed for solving the problems using real data about 670 high school students from Zacatecas, Mexico [11].

3.8 DATA MINING TECHNIQUES

The initial results from Kabakchieva and Dorina's implemented research project aimed to show the high potential of data mining applications for university management. This paper is focused on the implementation of data mining techniques and methods for acquiring new knowledge from data collected by universities. The main goal of the research is to reveal the high potential of data mining applications for university management [12].

3.9 ID3 AND C4.5 CLASSIFICATION ALGORITHMS

Kalpesh Adhatrao, Aditya Gaykar, Amiraj Dhawan, Rohit Jha and Vipul Honrao 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 and C4.5 classification algorithms on this data, they have predicted the general and individual performance of freshly admitted students in future examinations[13].

4. Report on the Present Investigation (Existing Systems)

A lot of research has been done on the topic of placement prediction in the past decade. Different researcher used different methods to produce intended results. Naik et. al. (2012) used classification algorithm to predict final result and placement of the students[2]. They used data mining techniques for producing knowledge about students of Master of Computer Application (MCA) course before admitting them to the course. The overall error occurred to classify validation data using MCA result prediction classification tree was 38.46% while for validating placement prediction classification tree it was 45.38%. Sharma et.al. (2014) used logistic regression model to create a Placement Predictor System (PPS)[5]. They generated results from an open source GNU Octave programming tool which brings about 83.33% accuracy. Another approach for placement prediction is taken by Bhatt et. al. (2015) where they used ID3 Decision Tree Algorithm[1]. While predicting the placement they incorporated both qualitative and quantitative parameters of a student to achieve better results. Giri et. al. (2016) used machine learning model of K-Nearest Classifier to predict probability of a undergrad student getting placed in an IT company[3]. They compared the results of the same against the results obtained from other models like Logistic Regression and SVM and proved that KNN produces better results.

Based on above research, we proposed usage of Artificial Neural Network for placement guidance which will provide higher accuracy compared to other algorithms. Though attempts were made to create such system taking into consideration both qualitative and quantitative parameters; amount of qualitative factors considered for the same was very less which we intend to change by using more than fifty qualitative parameters which constitutes an important role in placement of a student consequently improving the accuracy of the system.

5. AIM AND OBJECTIVES

5.1 AIM

Our project aims to create placement guidance system which will use the concept of Artificial Neural Networks. We intend to combine both qualitative and quantitative parameters for the decision making process. To do so we consider the academic history of the student as well as their skill set like, programming skills, communication skills, analytical skills and teamwork, which are tested by the hiring companies during the recruitment process. Though many research has been done previously on placement prediction using different methods, none of them gave consideration to qualitative parameters to a large extent, which plays a vital role in placement of any student. Thus, by taking this into account our aim is to achieve a system with greater than 85% of accuracy.

5.2 OBJECTIVES

Predicting the placement of a student gives an idea to the placement office as well as the student on where they stand. Not all companies look for similar talents. If the strengths and weaknesses of the students are identified it would benefit the student in getting placed. The placement Office can work on identifying the weaknesses of the students and take measures of improvement so that the students can overcome the weakness and perform to the best of their abilities. Thus the key lies in assessing the capabilities of the student in the right areas and subjecting them to the right training which is essentially our objective behind creating such system.

6. PROBLEM STATEMENT

Students studying in Engineering colleges feel the exigency to know where they stand in comparison to others, and what kind of placement they would get. The training and placement offices come in the picture when a student enters final year, but they are of no use to a student planning for future studies. The students have no platform to check their current position and build on their strengths.

The platforms currently available, have not been trained on real and complete data sets, and do not learn from their wrong predictions which reduces the accuracy, in the long term.

Planning for future role constitutes an important role in any Engineering student's life. This necessitates a system to assist the academic planners to design a strategy to improve the performance of students that will help them in getting placed at the earliest.

In all of the previous systems placement prediction of a student was done in terms of binary values i.e. 0 and 1 which does not represent clear picture to the user. Creating a system which will guide a user in a better way like showing probability ranging from 0 to 1 is essential in such cases.

During placement prediction various attributes of a student plays vital role in whether that particular student will get selected or not. These attributes constitute both qualitative and quantitative parameters. Previous systems considered only qualitative parameters of a student overlooking personal aspects of a candidate such as his confidence, ability to work on a problem etc. Taking this into consideration a system incorporating both parameters will provide a better guidance to the students.

Some of the earlier systems used decision trees such as ID3 to provide placement prediction. But such methods are computationally too heavy and bound to break when large number of datasets are provided. A self-adaptive Artificial Neural Network will overcome this important drawback of previous systems.

7. PROPOSED SYSTEM FOR PROJECT

Students are most benefited by this application. The students can manage their profile and give tests about programming languages, logic building and other such topics. The college has the student's quantitative data like CGPA, marks, internships, projects and certifications. The test data which gives the qualitative parameters and the quantitative parameters aid the predictive model that uses maximization of entropy. Once the prediction graph is generated, we have to fit a curve to map the data and apply the entropy maximization algorithm so that prediction can be done accurately. The students get the statistical data that will help with analytics and knowing how to improve themselves to get a better package. Statistical analytics also help the TPO to verify the data and if incorrect, TPO can change the data to maintain the accuracy.

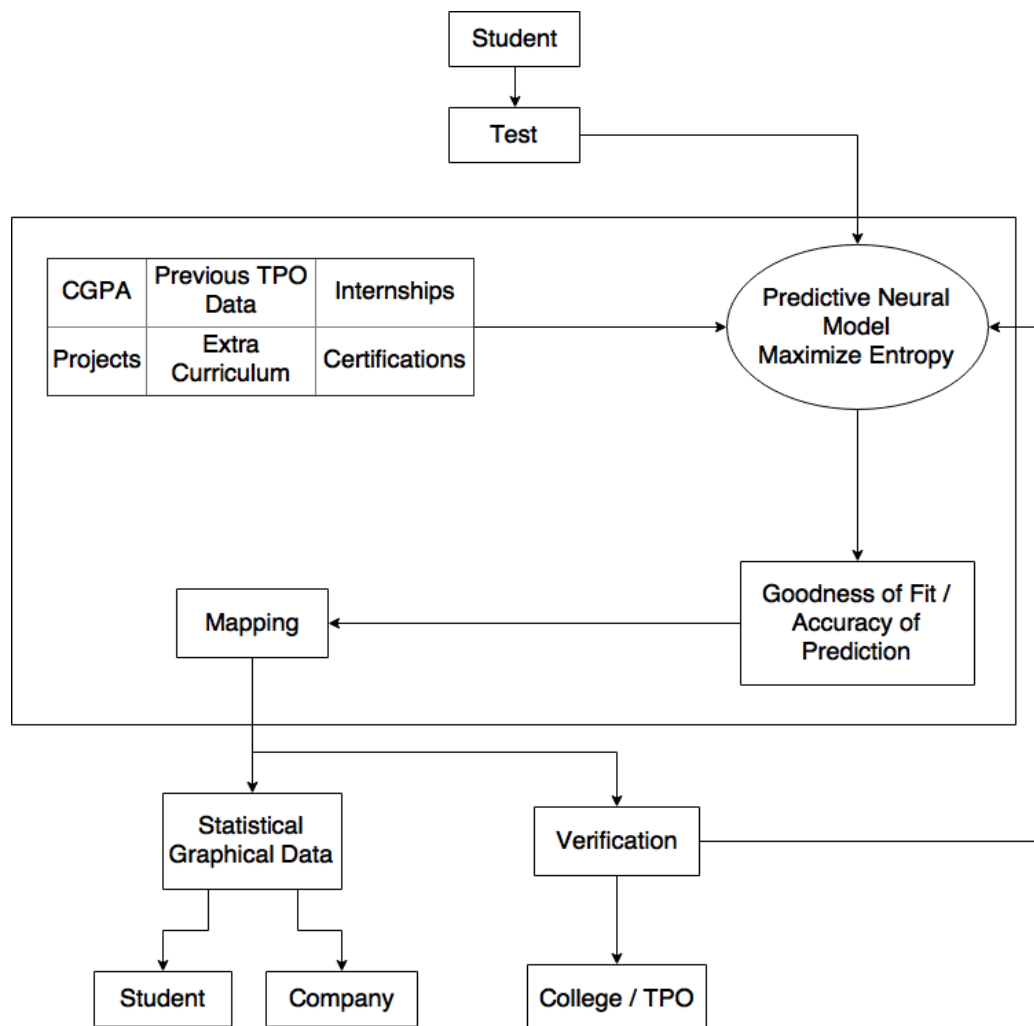


Figure 7.1: System Architecture

8. REQUIREMENT ANALYSIS (SRS)

8.1 FUNCTIONAL REQUIREMENTS

This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Our Web Application has three modules which are design for three different users. First is Students Portal where they can login through their portal page or registered themselves. Students can create their profile using personal dashboard. An Administrator Portal can login into his account and he/she will send emails regarding placement and companies and verify the details and apply filters on the data. View placement prediction analysis reports generated by our model .The Company Portal where interviewer can view student profile while taking interview and will be prompted questions based on the student resume. The main backend of our system is our Logistic Model, a mathematical model/machine which continually learns with every student's test data from database and it will process this information and will give final numeric value/probability of success or getting placed.

8.2 NON-FUNCTIONAL REQUIREMENTS

A description and, where possible, target values of associated non-functional requirements. Non-functional requirements detail constraints, targets or control mechanisms for the new system. They describe how, how well or to what standard a function should be provided. For example, levels of required service such as response times; security in the form Login Authentication for Student and Admin and access requirements; technical constraints; required interfacing with users' and other systems. Service level requirements are measures of the quality of service required, and is crucial to capacity planning and physical design. Identify realistic, measurable target values for each service level. These include service hours, service availability, and responsiveness due to UI design, throughput and reliability regarding the display of results on searching about company or students. Access restrictions should deal with what data needs protected; what data should be restricted to a particular user role; and level of restriction required, e.g. physical, password, view only. Non-functional requirements may cover the system as a whole or relate to specific functional requirements.

9. SCOPE (FEASIBILITY OF PROJECT)

9.1 SCOPE

1. The work currently deals with predicting the results based on last 5 year's training and placement data, examination department's data, and the extra curricular data of the students.
2. The present work will be using a regression model which is training itself, so the accuracy of the system will increase over time, making the system more reliable over time.
3. Even if the parameters change or the system will adapt to it over time.
4. In future, this system will be extended on the web end to give better outputs and competitive insights and also providing more statistics to the corporate end.
5. This system can be used for many years, as the system is adapting with the data, and the accuracy is increasing over time.

9.2 FEASIBILITY

9.2.1 Operational Feasibility

Operational Feasibility is the ability to utilize, support and perform the necessary tasks of a system such as model training and mapping the different types of parameters. It includes everyone, from who creates, operates or uses the system. To be operationally feasible, the system must fulfill a need required by the business.

9.2.2 Technical Feasibility

Technical Feasibility, involves development of a working model of the product or a service. The concept of Predictive analysis is used for finding the chances of prediction and other guiding parameters using Artificial Neural Network algorithm. It is not necessary that the initial materials and components of the working model represent those that actually will be used in the finished product or the service.

9.2.3 Economic Feasibility

Economic feasibility is the cost and logistical outlook for a business project or endeavor. Prior to embarking on a new venture, most businesses conduct an economic feasibility study, which is a study that analyzes data to determine whether the cost of the prospective new venture will ultimately be profitable to the college since they would be able to have more information about the ability of their students. College can conduct seminars and extra training sections to improve the caliber of the students. Economic feasibility is sometimes determined within the organization, while other times companies hire an external company, who has an expertise in this domain, to do the task for them.

10. METHODOLOGY

Methodology being used here is Agile Methodology. This method promotes continuous iteration of development and testing throughout the software development lifecycle of the project. During the life cycle of the product iterations were built simultaneously providing efficient and quality output.

Implementation Plan is comprised of following major steps:

1. Gathering Data
2. Data Preparation
3. Choosing a model
4. Training
5. Evaluation
6. Hyper-parameter tuning
7. Prediction

10.1 GATHERING DATA:

This step is very important because the quality and quantity of data that you gather will directly determine how good your predictive model can be. In this case, data we collected consisted student's marks across all semester.

10.2 DATA PREPARATION:

Data preparation, where we load our data into a suitable place and prepare it for use in our machine learning training. This is also a good time to do any pertinent visualizations of your data, to help you see if there are any relevant relationships between different variables you can take advantage of, as well as show you if there are any data imbalances. This step comprised of converting data from different formats to Excel and perform different data visualisation techniques to get the insights about the features.

We'll also need to split the data in two parts. The first part, used in training our model, will be the majority of the dataset. The second part will be used for evaluating our trained model's performance. We don't want to use the same data that the model was trained on for evaluation.

10.3 CHOOSING A MODEL:

The next step in our workflow is choosing a model. There are 60+ predictive modelling algorithms to choose from. We must understand the type of problem and solution requirement to narrow down to a select few models which we can evaluate. The algorithms we considered –

1. Logistic Regression
2. Support Vector Machine (SVM)
3. K-Nearest Neighbour (KNN)
4. Decision Tree
5. Artificial Neural Network (ANN)

After getting confidence score from each model we ranked our evaluation of all the models to choose the best one for our problem. We decided to go forward with ANN because it can handle much more variability as compared to traditional models.

10.4 TRAINING:

In this step, we will use our data to incrementally improve our model's ability to predict the probability of a student being placed. The training model consists of Weights (W) and biases (b) where weights is nothing but a collection of features.

The training process involves initializing some random values for W and b and attempting to predict the output with those values. We can compare our model's predictions with the output that it should produced, and adjust the values in W and b such that we will have more correct predictions. This process then repeats. Each iteration or cycle of updating the weights and biases is called one training "step".

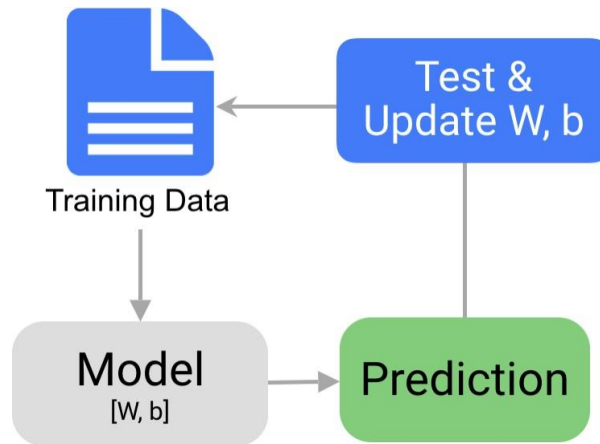


Figure 10.1: Training Model

10.5 EVALUATION:

Once training is complete, it's time to see if the model is any good, using evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen.

10.6 PARAMETER TUNING:

Further improvement of the model is done using Parameter tuning. There were a few parameters we implicitly assumed when we did our training, and in this step we go back and test those assumptions and try other values.

One example is how many times we run through the training dataset during training. We can “show” the model our full dataset multiple times, rather than just once. This leads to higher accuracies.

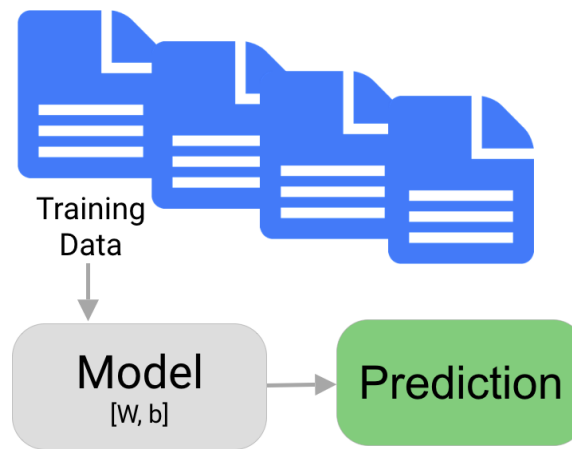


Figure 10.2: Parameter Tuning

Another parameter is “*learning rate*”. This defines how far we shift the line during each step, based on the information from the previous training step. These values all play a role in how accurate our model can become, and how long the training takes.

10.7 PREDICTION:

Machine learning is using data to answer the questions. So Prediction, or inference, is the step where we get to answer some questions. In this step we used our model to predict probability of a student getting placed.

11. HARDWARE AND SOFTWARE REQUIREMENTS

11.1 HARDWARE REQUIREMENTS

1. 1GB of RAM
2. Processor: i3 or Higher
3. Internet Connection: 512 Kb/s or above.
4. Screen Resolution: 1020 x 768 (or above)
5. Disk Storage for Database : 10 GB

11.2 SOFTWARE REQUIREMENTS

1. Browser (preferably Chrome or Firefox)
2. Operating System : Windows, Linux, OSX.
3. Text Editor : Sublime / Visual Studio Code / Atom
4. Development Environment: SQL, Node.JS, Python 3.0, PostgreSQL, Numpy, Pandas, Scikit-Learn, Plotly.js, Matplotlib, Seaborn

12. TESTING

12.1 TESTING

System testing is a critical phase implementation. Testing of the system involves hardware device implementation and debugging of the computer programs and testing information processing procedures. Testing can be done with text data, which attempts to stimulate all possible conditions that may arise during processing. If structured programming Methodologies have been adopted during coding the testing proceeds from higher level to lower level of program module until the entire program is tested as unit. The testing methods adopted during the testing of the system were unit testing and integrated testing.

12.1.1 UNIT TESTING

Unit testing focuses first on the modules, independently of one another, to locate errors. This enables the tester to detect errors in coding and logical errors that is contained within that module alone. Those resulting from the interaction between modules are initially avoided. In this methodology of the testing every single element in the project pipeline has been individually tested.

1. Data Cleaning Script - Whether the excel exportation is correct or not.
2. Dimensionality Reduction - Whether the parameters have their co-relations maintained
3. Exploratory Data Analysis - Whether the captured dependancies and graphs are actually efficient
4. Prediction - Whether the prediction is accurate or not

12.1.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time to uncover the errors associated with interfacing. The objective is to take unit- tested module and build a program structure that has been detected by designing. It also tests to find the discrepancies between the system and its original objectives. Subordinate stubs are replaced one at time actual module. Tests were conducted at each module was integrated. On completion of each set another stub was replaced with the real module. The entire project pipeline was treated as one and the entire process from insertion of data to training the model was done and the interactions between them were captured.

12.1.3 FUNCTIONAL TESTING

The testing of functionalities or individual features in a module is known as functional testing. In this methodology of testing we have incorporated various parameters to be supplied to the modules to check the outcome. Functional testing was done on graph generation and student tracking. Also checking whether the centralised record created was right or not. All the functionalities were tested before integrating them in a pipeline.

12.1.4 PERFORMANCE TESTING

Expected Results:

- The Accuracy was expected to be somewhere between 80 - 85 percent
- The Classification was expected to be the best in Decision Tree.

Observed Results:

- The Accuracy for the best model was 98 percent.
- The Best model for this classification was ID3 Decision Tree.

12.1.5 LOAD STRESS TESTING

Expected Result:

- Response time will be affected by the number of data points and sample size.
- The introduction of the more number of data points should not break the model.

Observation

- The speed of query execution was fine even when the more students were added

12.2 Test Cases

Test cases are the scenarios where we try to evaluate the performance of the system and test all the modules in different aspects and scenarios. The different models we have tuned for the guidance of placement are all classification models and it is a multi-class classification. These models need to be evaluated based on how they perform when they are exposed to outliers or average data or data from different sigma ranges. There are a set of tests that need to be performed while doing the same.

The test cases for the same that we have considered are as follows:

1. False Positives and False Negatives :

An evaluation of how many unplaced candidates were suggested to be placed and how many placed candidates were predicted to be unplaced. These two scenarios are known as False Positives and False Negatives, where in the negative results are predicted to be positive and the positive results are predicted to be negative.

2. Confusion Matrix :

A 2 x 2 matrix with True Positive, False Positive, True Negative, False Negatives, which helps in the evaluation of the actual accuracy rather than the hoax accuracy

3. Accuracy Paradox :

The numbers mentioned in the accuracy are just numbers it is on how many false negatives and positives are present in the different models, we find the actual accuracy rather than the numeric accuracy

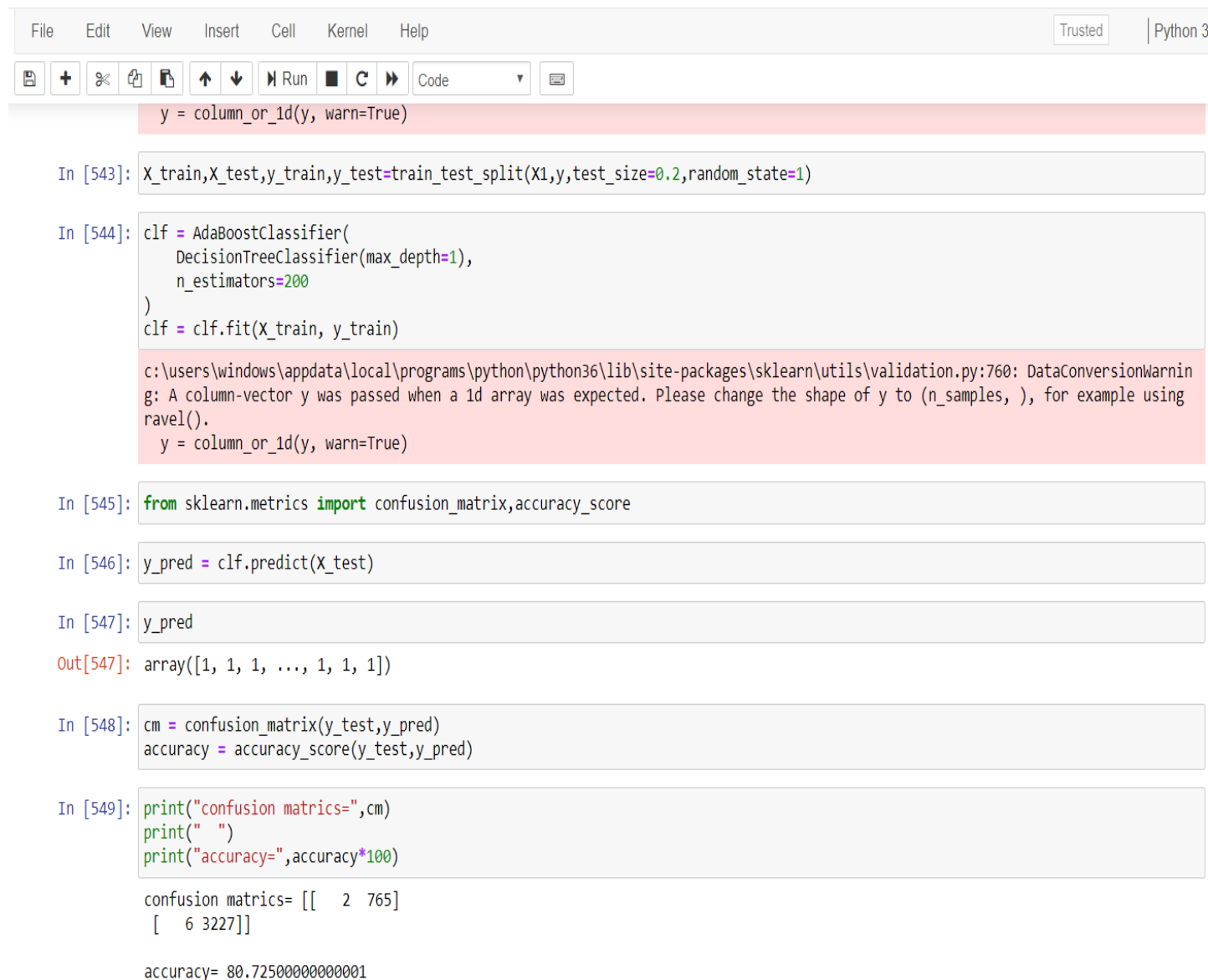
4. CAP Curve and Analysis :

Cumulative Accuracy Profile (CAP) Curve—It is a more robust method to assist our machine model. The idea here is to compare the model to a random scenario and evaluate the results.

13. MODELS AND THEIR ACCURACY CHECKING

Availability of numerous models for prediction makes it difficult to select appropriate model for your system. To overcome the challenge we trained our dataset on different models to check their accuracy and suitability for our system.

13.1 DECISION TREE CLASSIFIER–



```
File Edit View Insert Cell Kernel Help Trusted Python 3
[Icons] Run Code

y = column_or_1d(y, warn=True)

In [543]: X_train,X_test,y_train,y_test=train_test_split(X1,y,test_size=0.2,random_state=1)

In [544]: clf = AdaBoostClassifier(
            DecisionTreeClassifier(max_depth=1),
            n_estimators=200
          )
          clf = clf.fit(X_train, y_train)

c:\users\windows\appdata\local\programs\python\python36\lib\site-packages\sklearn\utils\validation.py:760: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using
ravel().
y = column_or_1d(y, warn=True)

In [545]: from sklearn.metrics import confusion_matrix,accuracy_score

In [546]: y_pred = clf.predict(X_test)

In [547]: y_pred

Out[547]: array([1, 1, 1, ..., 1, 1, 1])

In [548]: cm = confusion_matrix(y_test,y_pred)
          accuracy = accuracy_score(y_test,y_pred)

In [549]: print("confusion matrices=",cm)
          print(" ")
          print("accuracy=",accuracy*100)

confusion matrices= [[ 2 765]
 [ 6 3227]]

accuracy= 80.72500000000001
```

Figure 13.1: Decision Tree Classifier

13.2 DECISION TREE WITH ENTROPY

Decision tree with entropy

```
In [550]: clf_entropy =DecisionTreeClassifier(criterion = "entropy", random_state = 10)
          clf_entropy.fit(X_train, y_train)

Out[550]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                                max_depth=None, max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort='deprecated',
                                random_state=10, splitter='best')

In [551]: X_train,X_test,y_train,y_test=train_test_split(X1,y,test_size=0.2,random_state=10)

In [552]: entropy_y_pred=clf_entropy.predict(X_test)

In [553]: cm_entropy = confusion_matrix(y_test,entropy_y_pred)

In [554]: entropy_accuracy = accuracy_score(y_test,entropy_y_pred)

In [555]: print("confusion matrices=",cm_entropy)
          print(" ")
          print("accuracy=",entropy_accuracy*100)

confusion matrices= [[ 696  129]
                    [ 147 3028]]

accuracy= 93.10000000000001
```

Figure 13.2: Decision Tree with entropy

13.3 GINI INDEX

gini index

```
In [571]: clf_gini =DecisionTreeClassifier(criterion = "gini", random_state = 10)
          clf_gini.fit(X_train, y_train)

Out[571]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                                max_depth=None, max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort='deprecated',
                                random_state=10, splitter='best')

In [577]: X_train,X_test,y_train,y_test=train_test_split(X1,y,test_size=0.2,random_state=100)

In [578]: gini_y_pred=clf_gini.predict(X_test)

In [579]: cm_gini = confusion_matrix(y_test,gini_y_pred)

In [580]: gini_accuracy = accuracy_score(y_test,gini_y_pred)

In [581]: print("confusion matrices=",cm_gini)
          print(" ")
          print("accuracy=",gini_accuracy*100)

confusion matrices= [[ 629  123]
                    [ 139 3109]]

accuracy= 93.45
```

Figure 13.3: gini index

13.4 SUPPORT VECTOR MACHINE

SVM (Support vector machine) classifier

```
In [556]: from sklearn import svm

In [557]: clf = svm.SVC()
clf.fit(X_train, y_train)

c:\users\windows\appdata\local\programs\python\python36\lib\site-packages\sklearn\utils\validation.py:760: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

Out[557]: SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)

In [558]: X_train,X_test,y_train,y_test=train_test_split(X1,y,test_size=0.2,random_state=1)

In [559]: svm_y_pred = clf.predict(X_test)

In [560]: svm_cm = confusion_matrix(y_test,svm_y_pred)
svm_accuracy = accuracy_score(y_test,svm_y_pred)

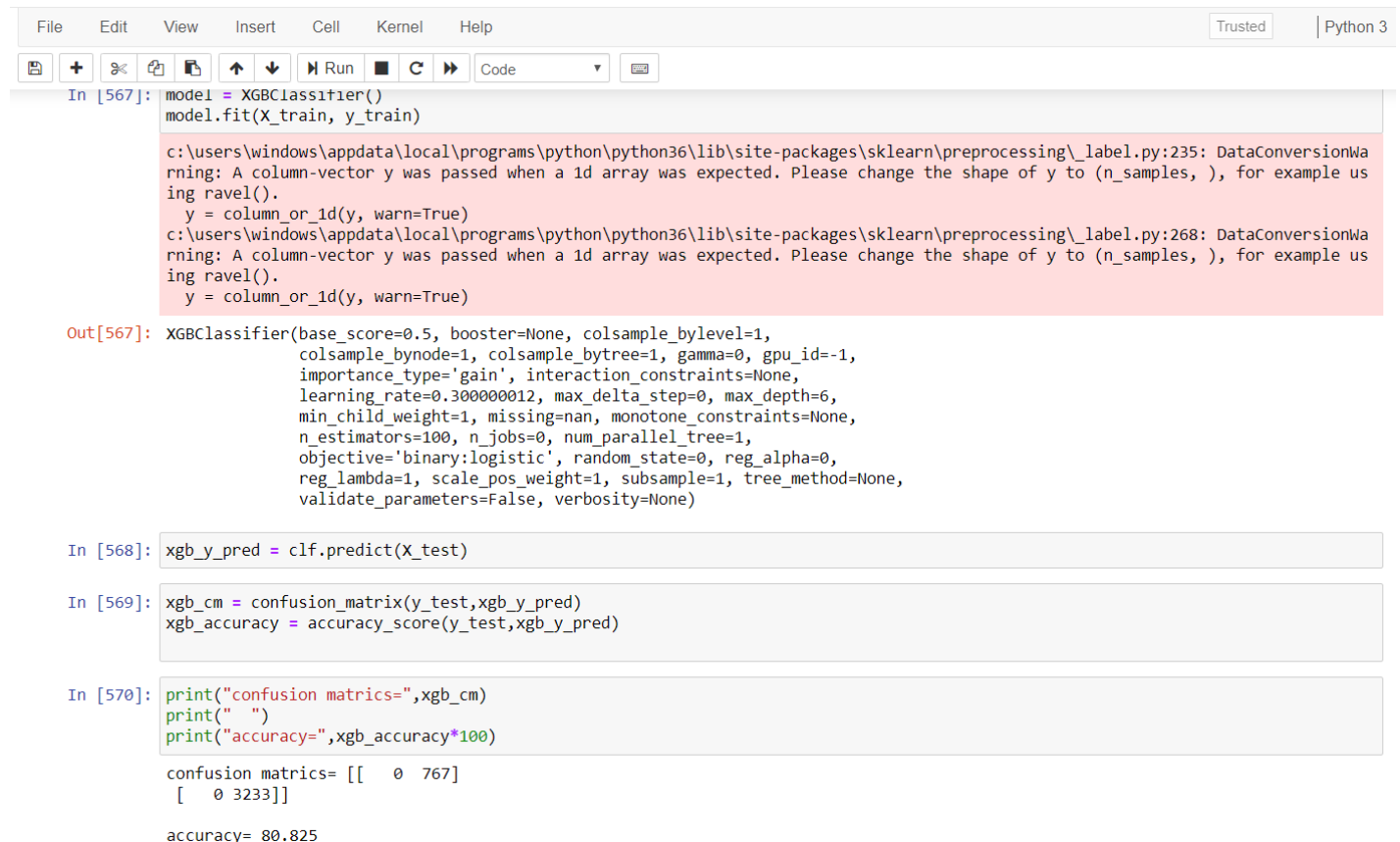
In [561]: print("confusion matrices=",svm_cm)
print(" ")
print("accuracy=",svm_accuracy*100)

confusion matrices= [[ 0 767]
 [ 0 3233]]

accuracy= 80.825
```

Figure 13.4: svm

13.5 XGBOOST



```
File Edit View Insert Cell Kernel Help Trusted Python 3

In [567]: model = XGBClassifier()
model.fit(X_train, y_train)

c:\users\windows\appdata\local\programs\python\python36\lib\site-packages\sklearn\preprocessing\_label.py:235: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
c:\users\windows\appdata\local\programs\python\python36\lib\site-packages\sklearn\preprocessing\_label.py:268: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

Out[567]: XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
  colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
  importance_type='gain', interaction_constraints=None,
  learning_rate=0.300000012, max_delta_step=0, max_depth=6,
  min_child_weight=1, missing=nan, monotone_constraints=None,
  n_estimators=100, n_jobs=0, num_parallel_tree=1,
  objective='binary:logistic', random_state=0, reg_alpha=0,
  reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None,
  validate_parameters=False, verbosity=None)

In [568]: xgb_y_pred = clf.predict(X_test)

In [569]: xgb_cm = confusion_matrix(y_test,xgb_y_pred)
xgb_accuracy = accuracy_score(y_test,xgb_y_pred)

In [570]: print("confusion matrices=",xgb_cm)
print(" ")
print("accuracy=",xgb_accuracy*100)

confusion matrices= [[ 0 767]
 [ 0 3233]]

accuracy= 80.825
```

Figure 13.5: xgboost

13.6 NAÏVE BAYES

Naive Bayes Classifier

```
In [94]: X_train, X_test, y_train, y_test = train_test_split(X1, y, test_size=0.2, random_state=1)

In [95]: from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)

c:\users\windows\appdata\local\programs\python\python36\lib\site-packages\sklearn\naive_bayes.py:206: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  y = column_or_1d(y, warn=True)

Out[95]: GaussianNB(priors=None, var_smoothing=1e-09)

In [96]: gnb_y_pred = gnb.predict(X_test)

In [97]: cm_gnb = confusion_matrix(y_test, gnb_y_pred)

In [98]: gnb_accuracy = accuracy_score(y_test, gnb_y_pred)

In [99]: print("confusion matrices=", cm_gnb)
print(" ")
print("accuracy=", gnb_accuracy*100)

confusion matrices= [[ 0 767]
 [ 0 3233]]

accuracy= 80.825
```

Figure 13.6: Naïve Bayes

ALGORITHMS WITH THEIR ACCURACY

ALGORITHM	ACCURACY
DECISION TREE CLASSIFIER	80.725
DECISION TREE WITH ENTROPY	93.1
GINI INDEX	93.45
SVM	80.825
XGBOOST	80.825
NAÏVE BAYES	80.825

14. ADVANTAGES AND LIMITATIONS

14.1 ADVANTAGES

1. Models have been trained on real dataset
2. To solve the issue for training models a centralised database was created
3. Result analysis easier
4. College can get easy insights and generate graphs easier.
5. By training multiple models we have identified the pros and cons of all the algorithms giving the freedom to find the best possible algorithm and leverage the best parts of all the algorithms.
6. The model being trained on actual student data captures the anomalies in real scenario and adapts to it over time.
7. Classification problem can be converted to a regression problem by predicting salaries instead of whether being placed or not thus gaining best of both worlds.

Some models have accuracy as high as 99 percent.

14.2 LIMITATIONS

8. The data on which the model was trained on was of the previous few years and hence there were limitations that had occurred while training the model.
9. The project for now doesn't have a front end to be accessed from, and adding the front end will give the students to access insights and basically the scripts will come to life
10. The problem that we have treated is primarily a classification and there are better ways to convert this problem into a regression problem and solve it efficiently.

15. APPLICATIONS AND FUTURE SCOPE

15.1 APPLICATIONS

This project can find its applications in various places:

1. For students who want to look at their overall growth and understand what their strengths and weaknesses are
2. For students to get a fair idea about the subjects they are going to face and how difficult will it be
3. The college can find the insights of all the population and see the growth of individual people as well.
4. Focus on the curriculum and see the changes that occur.
5. Train students as per the latest trends in technology
6. The company can have a look at this insight for the recruitment procedure as well.
7. Whenever there is an educational leap, this project can be turned into an application.

15.2 FUTURE SCOPE

8. When the research work and the different models combined together and given a web application to be accessed the project will be of use to all the people.
9. The continual additions of data will cause the models to work more efficiently but fine tuning of the models with more qualitative parameters is necessary.
10. The models are exposed to the risk of overfitting in the future and hence the parameters can change their co relations and hence retraining the models is important.
11. A complete application where in a centralised database and the internship data of the student can be kept for mining processes to better understand the students and provide valuable guidance to students.
12. The project is open for using the models that can be available to the mankind in near future (Capsule Networks).

16. CONCLUSION

As we have seen throughout our studies, that the problem statements we have approached are student, college, and corporate centric. The solution to all of these problem statements, is based on the model we are going to build, the output of which will be a number between 0-1, which will determine, the prediction of a percentage of students being placed. During this process, a lot of other dependent variables will be predicted which will help solve the problem statements. The expected outputs of the system for student end, is the prediction about their placement, and the statistics of how they can fair well. College end will have the analysis of every student, and will have the opportunity to focus more on the improvement of students. Also because of the system, the college will have one platform to manage the data of the students, thus solving another issue. The corporates will be able to apply filters, compare students, and download resume of the students. With the help of this percentage we ensure that the percentage of students will be placed.

17. APPENDIX

17.1 CONCEPTS RELATED TO NEURAL NETWORKS

17.1.1 What are Neural Networks made of?

A typical neural network has anything from a few dozen to hundreds, thousands, or even millions of artificial neurons called units arranged in a series of layers, each of which connects to the layers on either side. Some of them, known as input units, are designed to receive various forms of information from the outside world that the network will attempt to learn about, recognize, or otherwise process. Other units sit on the opposite side of the network and signal how it responds to the information it's learned; those are known as output units. In between the input units and output units are one or more layers of hidden units, which, together, form the majority of the artificial brain. Most neural networks are fully connected, which means each hidden unit and each output unit is connected to every unit in the layers either side. The connections between one unit and another are represented by a number called a weight, which can be either positive (if one unit excites another) or negative (if one unit suppresses or inhibits another). The higher the weight, the more influence one unit has on another. (This corresponds to the way actual brain cells trigger one another across tiny gaps called synapses.)

17.1.2 How does a Neural Network learn?

Information flows through a neural network in two ways. When it's learning (being trained) or operating normally (after being trained), patterns of information are fed into the network via the input units, which trigger the layers of hidden units, and these in turn arrive at the output units. This common design is called a feedforward network. Not all units "fire" all the time. Each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections they travel along. Every unit adds up all the inputs it receives in this way and (in the simplest type of network) if the sum is more than a certain threshold value, the unit "fires" and triggers the units it's connected to (those on its right).

For a neural network to learn, there has to be an element of feedback involved—just as children learn by being told what they're doing right or wrong. In fact, we all use feedback, all the time. Think back to when you first learned to play a game like ten-pin bowling. As you picked up the heavy ball and rolled it down the alley, your brain watched how quickly the ball moved and the

line it followed, and noted how close you came to knocking down the skittles. Next time it was your turn, you remembered what you'd done wrong before, modified your movements accordingly, and hopefully threw the ball a bit better. So you used feedback to compare the outcome you wanted with what actually happened, figured out the difference between the two, and used that to change what you did next time ("I need to throw it harder," "I need to roll slightly more to the left," "I need to let go later," and so on). The bigger the difference between the intended and actual outcome, the more radically you would have altered your moves.

Neural networks learn things in exactly the same way, typically by a feedback process called backpropagation (sometimes abbreviated as "backprop"). This involves comparing the output a network produces with the output it was meant to produce, and using the difference between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input units—going backward, in other words. In time, back propagation causes the network to learn, reducing the difference between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should

17.1.3 How does Neural Network work in Practice?

Once the network has been trained with enough learning examples, it reaches a point where you can present it with an entirely new set of inputs it's never seen before and see how it responds. For example, suppose you've been teaching a network by showing it lots of pictures of chairs and tables, represented in some appropriate way it can understand, and telling it whether each one is a chair or a table. After showing it, let's say, 25 different chairs and 25 different tables, you feed it a picture of some new design it's not encountered before—let's say a chaise longue—and see what happens. Depending on how you've trained it, it'll attempt to categorize the new example as either a chair or a table, generalizing on the basis of its past experience—just like a human. That doesn't mean to say a neural network can just "look" at pieces of furniture and instantly respond to them in meaningful ways; it's not behaving like a person. Consider the example we've just given: the network is not actually looking at pieces of furniture. The inputs to a network are essentially binary numbers: each input unit is either switched on or switched off. So if you had five input units, you could feed in information about five different characteristics of different chairs using binary (yes/no) answers. The questions might be 1) Does it have a back? 2) Does it have a top? 3) Does it have soft upholstery? 4) Can you sit on it comfortably for long periods of time? 5) Can you put lots of things on top of it? A typical chair would then present as Yes, No, Yes, Yes, No or 10110 in

binary, while a typical table might be No, Yes, No, No, Yes or 01001. So, during the learning phase, the network is simply looking at lots of numbers like 10110 and 01001 and learning that some mean chair (which might be an output of 1) while others mean table (an output of 0).

17.1.4 What is Classification in Machine Learning?

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into spam non-spam classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

17.1.5 List of Common Machine Learning Algorithms

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naive Bayes
6. K - NN
7. K - Means
8. Random Forest
9. Dimensionality Reduction Algorithms
10. Gradient Boosting algorithms
11. GBM
12. XGBoost
13. LightGBM
14. CatBoost

18. OUTPUTS

G:\new project\studentplacement\studentplacement\DimensionalityReduction.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
DimensionalityReduction.py x
1 from mpl_toolkits.mplot3d import Axes3D
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from sklearn.manifold import TSNE
6 from sklearn.decomposition import PCA
7
8
9 # import file
10 data = np.loadtxt("Book1.txt",encoding="utf8", delimiter='\t', skiprows=2)
11 #data = pd.read_excel('Book1.xlsx',encoding="ut8", delimiter='\t', skiprows=1)
12 # 0 Coding Skills
13 # 1 Aptitude Skills
14 # 2 Technical Skills
15 # 3 Communication Skills
16 # 4 Core Knowledge
17 # 5 Presentation Skills
18 # 6 Academic Performance
19 # 7 Puzzle Solving skills
20 # 8 English Proficiency
21 # 9 Programming Skills
22 # 10 Management Skills
23 # 11 Projects
24 # 12 Internships
25 # 13 Training
26 # 14 Backlog
27 # 15 Placed
28
29 # X Axis 0,2,7,9,11,12,13,14
30 # Y Axis 1,3,4,5,6,8,10,14
31
32 # Splitting data for averaging skills to represent in 3D
33 w1 = data[:,0]
34 w2 = data[:,1]
35 w3 = data[:,2]
36 w4 = data[:,3]
37 w5 = data[:,4]
38 w6 = data[:,5]
39 w7 = data[:,6]
40 w8 = data[:,7]
```



```

40 w8 = data[:,7]
41 w9 = data[:,8]
42 w10 = data[:,9]
43 w11 = data[:,10]
44 w12 = data[:,11]
45 w13 = data[:,12]
46 w14 = data[:,13]
47 w15 = -10*data[:,14]
48 w16 = data[:,15]
49
50 # X Axis 0,2,7,9,11,12,13,14
51 # Y Axis 1,3,4,5,6,8,10,14
52 x = (w1 + w3 + w8 + w10 + w12 + w13 + w14 + w15)/8
53 y = (w2 + w4 + w5 + w6 + w7 + w9 + w11 + w15)/8
54 z = w16
55
56 # Show plot obtained by averaging
57 fig = plt.figure()
58 ax = fig.add_subplot(111,projection='3d')
59 three_d = ax.scatter(x,y,z, marker='o', c=z, cmap='jet')
60 plt.colorbar(three_d)
61
62 ax.set_xlabel('Qualitative Skills in %')
63 ax.set_ylabel('Quantitative Skills in %')
64 ax.set_zlabel('Probability Of Placement in %')
65 ax.set_title('Average of Qualitative & Quantitative vs Probability of Placement')
66 plt.show()
67
68 # Implementing TSNE for plotting 16D data into 2D
69 X_embedded = TSNE(n_components=2).fit_transform(data[:,:])
70
71 plt.figure()
72 two_d_tsne = plt.scatter(X_embedded[:, 0], X_embedded[:, 1], c=z)
73 plt.colorbar(two_d_tsne)
74 plt.xlabel('Skills')
75 plt.ylabel('Probability of being placed')
76 plt.title('Represented Higher Dimensions to 2D using TSNE')
77 plt.show()

```

```

79 # Implementing PCA to reduce dimensions
80 pca = PCA(n_components=2)
81 pca_result = pca.fit_transform(data)
82
83 plt.figure()
84 two_d_pca = plt.scatter(pca_result[:, 0], pca_result[:, 1], c=z)
85 plt.colorbar(two_d_pca)
86 plt.xlabel('Skills')
87 plt.ylabel('Probability of being placed')
88 plt.title('Represented Higher Dimensions to 2D using PCA')
89 plt.show()

```

1. Averaging the values of the parameters and represent it as a 3D plot. *Graph is as shown:*

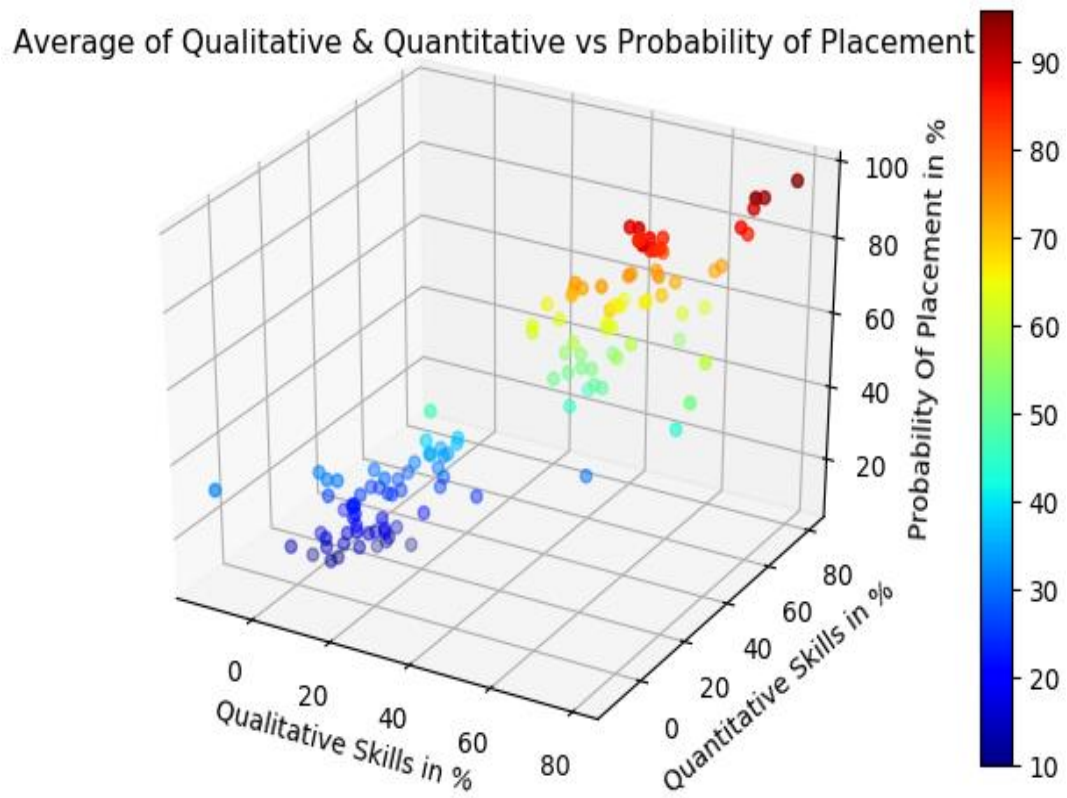


Figure 18.1: Probability of placement

2. T-Distributed Stochastic Neighbor Embedding (TSNE) is used to visualize Higher Dimensional data into Lower Dimensions. In this method, the parameters are combined by using TSNE method (`sklearn.manifold.TSNE`) to obtain a

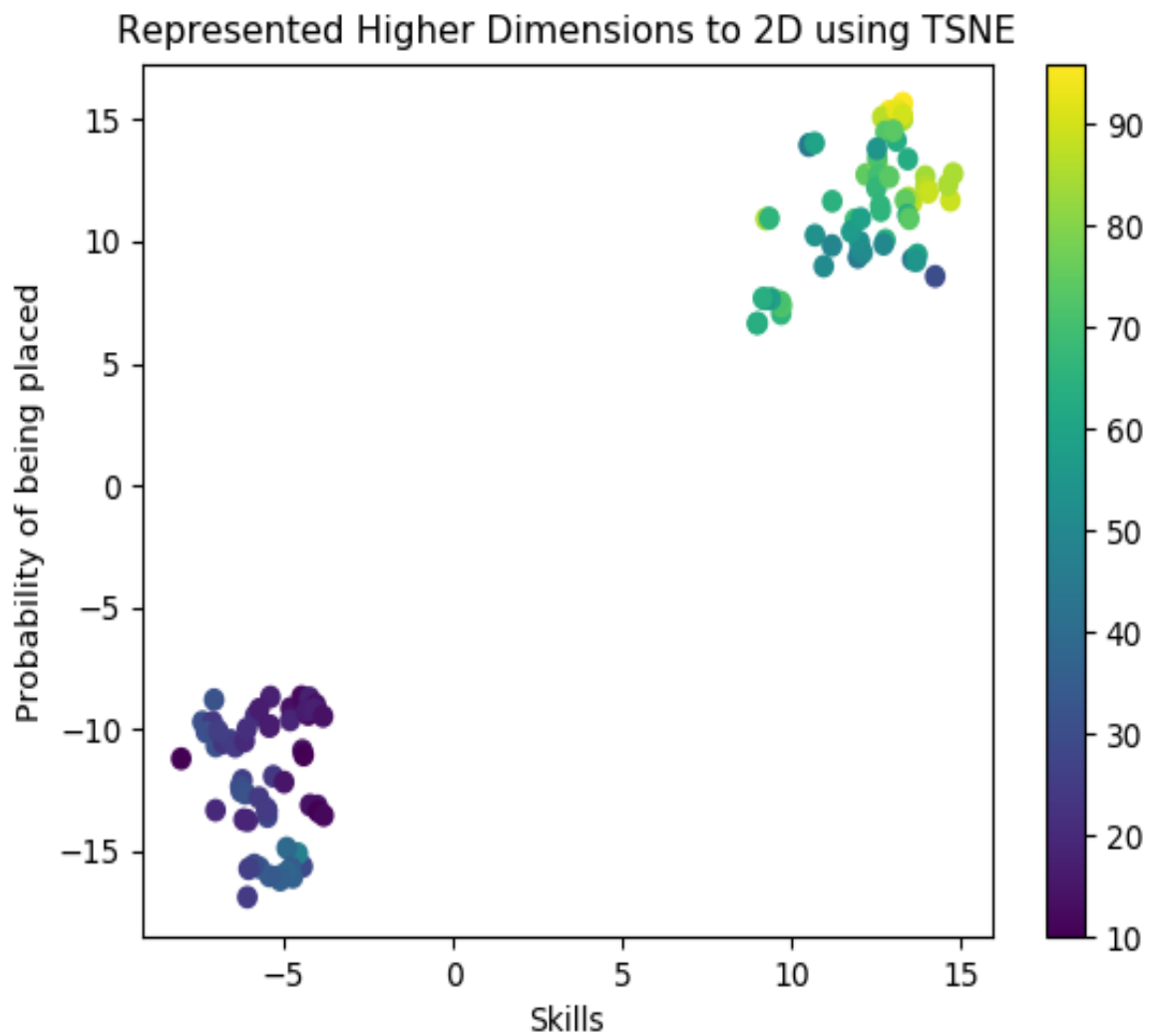


Figure 18.2: TSNE

3. Principal Component Analysis (PCA) is used to visualize Higher Dimensional data into Lower Dimensions. In this method, the parameters are combined by using TSNE method (`sklearn.decomposition.PCA`) to obtain a 2D plot. *Graph is as shown:*

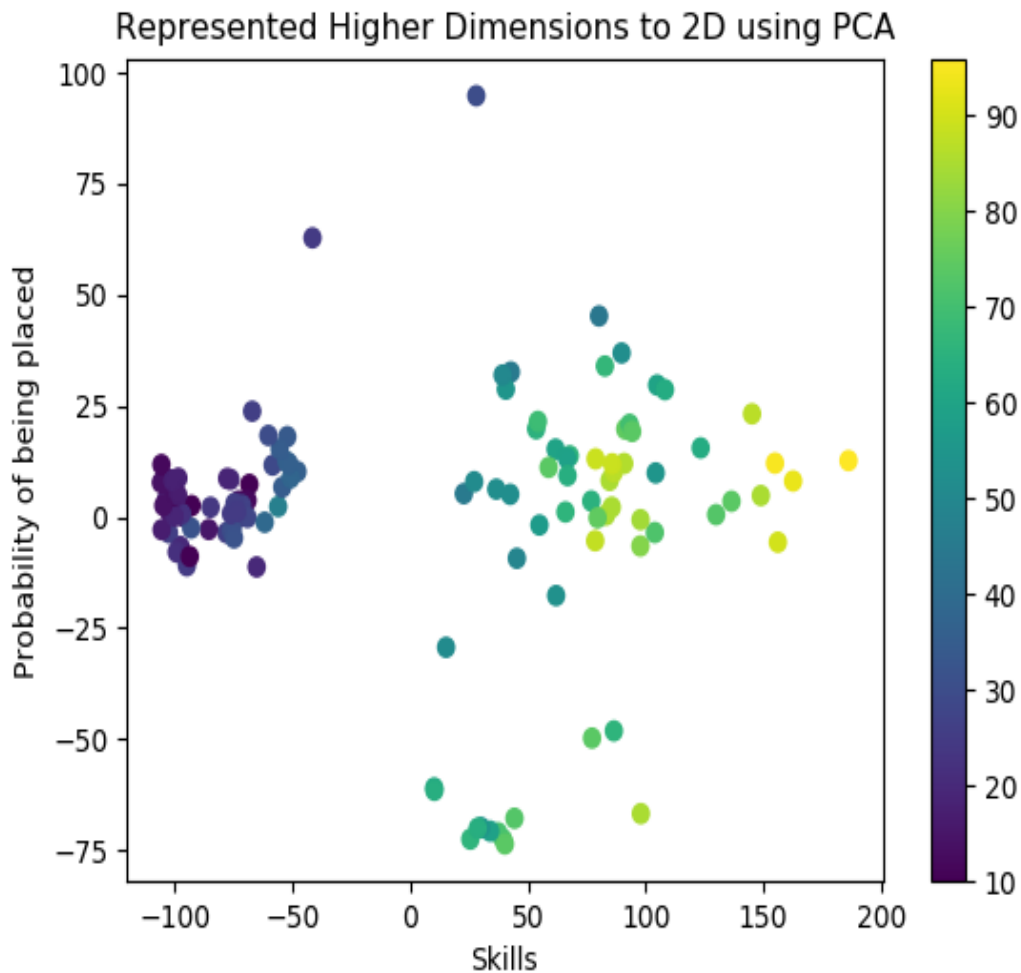


Figure 18.3: PCA

The only difference is that the TSNE preserves the distances and it works better when there are huge number of dimensions.

19. REFERENCES

1. Use of ID3 Decision Tree algorithm for Placement Prediction.
Bhatt, H., Mehta, S., & D'mello, L. R. (2015). Use of ID3 Decision Tree Algorithm for Placement Prediction. *International Journal of Computer Science and Information Technologies*, 4785-4789.
2. Prediction of Final Result and Placement of Students using Classification Algorithm
Naik, N., & Purohit, S. (2012). Prediction of Final Result and Placement of Students using Classification Algorithm. *International Journal of Computer Applications*, 56(12).
3. A Placement prediction system using K-Nearest Neighbors Classifier
Giri, A., Bhagavath, M. V. V., Pruthvi, B., & Dubey, N. (2016, August). A Placement Prediction System using k-nearest neighbors classifier. In *Cognitive Computing and Information Processing (CCIP), 2016 Second International Conference on* (pp. 1-4). IEEE.
4. Student Placement Prediction Using ID3 Algorithm.
Namita Puri, Deepali Khot, Pratiksha Shinde, Kishori Bhoite, , , , ."Student Placement Prediction Using ID3 Algorithm", Volume 3, Issue III, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: , ISSN : 2321-9653, www.ijraset.com.
5. PPS - Placement Prediction System using Logistic Regression.
Sharma, A. S., Prince, S., Kapoor, S., & Kumar, K. (2014, December). PPS—Placement prediction system using logistic regression. In *MOOC, Innovation and Technology in Education (MITE), 2014 IEEE International Conference on*(pp. 337-341). IEEE.
6. An Empirical Analysis of Classification Techniques for Predicting Academic Performance
S.Taruna , Mrinal Pandey ,”An Empirical Analysis of Classification Techniques for Predicting Academic Performance” in 2014 IEEE International Advance Computing Conference (IACC).
7. Predicting students marks in hellenic open university
Kotsiantis, Sotiris B., and Panayiotis E. Pintelas, "Predicting students marks in hellenic open university", in Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on, pp. 664-668. IEEE, 2005.

8. Applying logistic regression model to the examination results data
Saha, Goutam, "Applying logistic regression model to the examination results data.,in Journal of Reliability and Statistical Studies 4, no.2(2011):1-13.
9. Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique
Syed Tanveer Jishan, Raisul Islam Rashu, Naheena Haque and Rashedur M Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," in Decision Analytics (2015) 2:1 DOI 10.1186/s40165-014-0010-2(Springer Journal).
10. Prediction and Analysis for Students' Marks Based on Decision Tree Algorithm
Zhiwu Liu and Xiuzhi Zhang, "Prediction and Analysis for Students' Marks Based on Decision Tree Algorithm" in 2010 Third International Conference on Intelligent Networks and Intelligent Systems.
11. Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data
Carlos Márquez-Vera, Alberto Cano, Cristóbal Romero, Sebastián Ventura, "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data" in Appl Intell (2013) 38:315–330 DOI 10.1007/s10489-012-0374-8.
12. Predicting student performance by using data mining methods for classification
Kabakchieva, Dorina. "Predicting student performance by using data mining methods for classification", in Cybernetics and Information Technologies 13, no. 1 (2013): 61-72.
13. Predicting Student Performance using Artificial Neural Network Analysis
Van Heerden et. al designed a system for placement prediction in the University of Pretoria Medical School, using artificial neural networks, which had 99 input parameters out of which 80 parameters were qualitative. They tested their system on some students and found out that for those students where all the parameters were available the prediction of the system was almost 100% accurate, and a 90% accuracy was obtained.