

Government College of Engineering, Amravati

Department of Information Technology.

Third Year (6th Sem)

Subject: Minor Project Lab II

Topic: Student Placement Prediction

Submitted By:

Group: Batch N2

18007038: Makrand Bhonde

18007042: Kunal Wade

18007052: Anurag Shrirao

Guided By: Prof. S. R. Wankhade

CERTIFICATE

This is to certify that this report represents the project work done by **Makrand** , **Kunal** and **Anurag** during this project submission as a partial fulfilment of the requirement for Student Placement Prediction, of the Government College of Engineering Amravati
During academic year 2020-2021

Date: / /

Project Guidance by:

Prof. S.R Wankhade

DECLARATION

I hereby declare that the Minor Project work being presented in this report Entitled “***STUDENT PLACEMENT PREDICTON***” submitted in the Department of Information Technology, Government College of Engineering Amravati is the authentic work carried out by us under the guidance of **Prof S.R Wankhade** madam.

Date: / /

Makrand Bhonde

Kunal Wade

Anurag Shrirao

**Third Year Information
Technology**

ACKNOWLEDGEMENT

We are very thankful to our advisor **Prof. S.R Wankhade** for providing us an opportunity to do the project in domain *machine learning* and giving us all support and guidance, which made us complete the project duly, without whose constructive guidance this project would not have been success.

Her valuable advice and suggestions for corrections, modification, and improvement, enhanced the perfection of our job very well.

And last but not least I would like to say thanks to my group members for their encouragement and support for successfully completion of our project.

Department of Information Technology

Prof. S.R Wankhade

INDEX

SRNO.	TOPIC	PG NO
1	ABSTRACT AND PROBLEM DEF	6
2	MODULES USED	7
3	SOFTWARE REQUIREMENTS	8
4	SYSTEM FLOW	9
5	ARCHITECHTURE	10
6	DECISION TREE	11
7	IMPLEMENTATION	12
8	CONCLUSION	15
9	FUTURE SCOPE	16
10	REFERENCES	16

ABSTRACT:

Predicting the number of potential candidates is one of biggest challenges for the placement officer, since sometime a student with very good academic score doesn't get place and the one with the low academic score but good communication skills get place. Clearly, there are more than one factor influencing the placements. Finding the relation between factors can be even challenging for a trained statistician. Even companies visiting the college uses the criteria of aggregate to sort out students and sometime due to error of method they end up empty handed with no potential student. We are proposing a machine learning evaluation system which will consider many different factors like communication skills, technical skills, aggregate to determine the potential of the student. We are using Decision Tree Algorithm which gives excellent performance in classification problem with less data constraint.

PROBLEM DEFINITION:

The general Placement Prediction System considers only academic performances in order to predict whether a student can be placed or not. Judging the student based only on his academic performances would be unfair for the student, since a student could be having good aptitude, technical and communication skills but unfortunately might not be good in academic performances. It would wrong to judge a student based only on his academic performances, since Predicting the placement of a student needs a lot of parameters to be considered. But in order to get selected in campus interview, the student must be good in technical and aptitude skills. Of course, academic performances are important but don't hold the highest importance in the outcome of studentplacement.

PROPOSED SYSTEM:

- Proposed system follows the Decision Tree model of classification, which removes the problems caused by previous algorithms like negative prediction and over-value prediction (more than 100% prediction value).

Here we use Decision machine learning module to provide efficient and accurate results and we also provide a nice Graphical User Interface by Tkinter for easy interaction with the model

MODULES USED:

SCIKIT-LEARN

Defining scikit learn, it 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.

Scikit-learn was initially developed by David Cournapeau as a Google summer of code project in 2007. Later Matthieu Brucher joined the project and started to use it as a part of his thesis work. In 2010 INRIA got involved and the first public release (v0.1 beta) was published in late January 2010. The project now has more than 30 active contributors and has had paid sponsorship from INRIA, Google, Tinyclues and the Python Software Foundation.

In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry (aka multivariate data), it is said to have several attributes or **features**.

TRAINING AND TESTING SET

Machine learning is about learning some properties of a data set and then testing those properties against another data set. A common practice in machine learning is to evaluate an algorithm by splitting a data set into two. We call one of those sets the **training set**, on which we learn some properties; we call the other set the **testing set**, on which we test the learned properties.

LEARNING AND PREDICTING

In the case of the digits dataset, the task is to predict, given an image, which digit it represents. We are given samples of each of the 10 possible classes (the digits zero through nine) on which we *fit* an estimator to be able to *predict* the classes to which unseen samples belong.

In scikit-learn, an estimator for classification is a Python object that implements the methods `fit(X, y)` and `predict(T)`.

NUMPY

NumPy is a library for the Python programming language, adding support for large,

multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

The Python programming language was not initially designed for numerical computing, but attracted the attention of the scientific and engineering community early on, so that a special interest group called matrix-sig was founded in 1995 with the aim of defining an array computing package. Among its members was Python designer and maintainer Guido van Rossum, who implemented extensions to Python's syntax (in particular the indexing syntax) to make array computing easier.

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. **Pandas** is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python. Additionally, it has the broader goal of becoming **the most powerful and flexible open source data analysis / manipulation tool available in any language**. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels

SOFTWARE REQUIREMENTS:

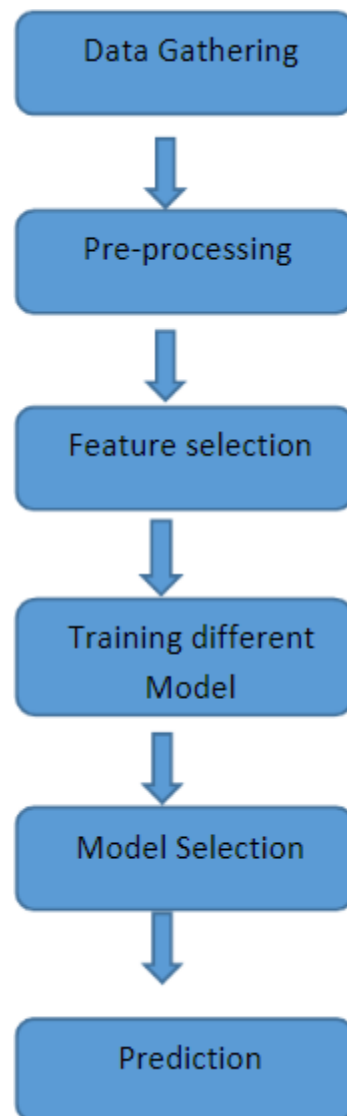
The Software Requirements in this project include:

Anaconda (spyder IDE)

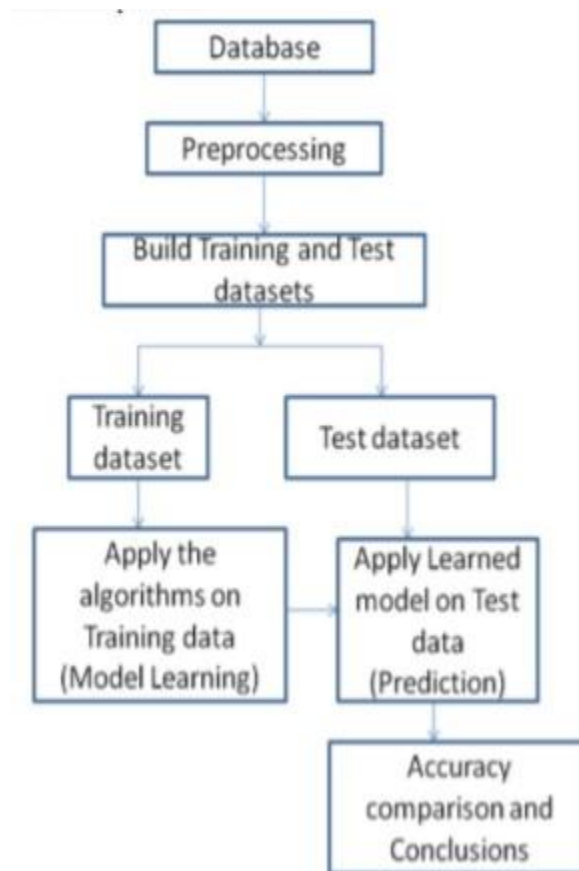
Tkinter , Pandas, numpy , modules

Machine Learning algorithms

SYTEM FLOW:



PLACEMENT PREDICTION SYSTEM ARCHITECTURE:



DECISION TREE:

A tree has many analogies in real life, and turns out that it has influenced a wide area of **machine learning**, covering both **classification and regression**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning.

How can an algorithm be represented as a tree?

For this let's consider a very basic example that uses titanic data set for predicting whether a passenger will survive or not. Below model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).

A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/**internal node**, based on which the tree splits into branches/**edges**. The end of the branch that doesn't split anymore is the decision/**leaf**, in this case, whether the passenger died or survived, represented as red and green text respectively.

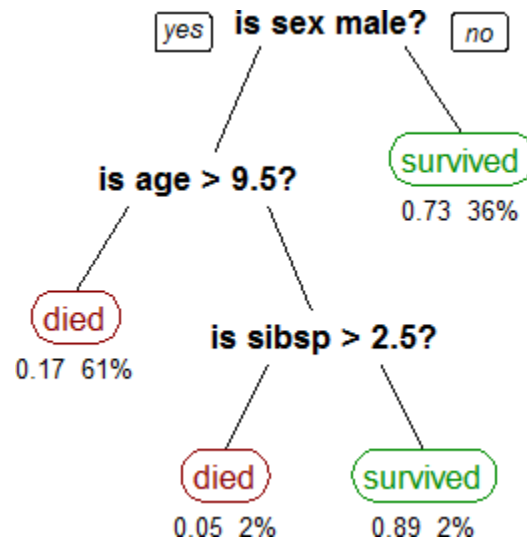


Fig.5.1

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The **feature importance** is clear and relations can be viewed easily. This methodology is more commonly known as **learning decision tree from data** and above tree is called **Classification tree** as the target is to classify passenger as survived or died. **Regression trees** are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.


So, what is actually going on in the background? Growing a tree involves deciding on **which features to choose** and **what conditions to use** for splitting, along with knowing when to stop. As a tree generally grows arbitrarily, **you will need to trim it down** for it to look beautiful. Lets start with a common technique used for splitting.

IMPLEMENTATION

OUTPUT TKINTER WINDOW:

Placement Prediction

STUDENT PLACEMENT PREDICTION




Name	<input type="text"/>	Technical Skills	None
Roll no	<input type="text"/>	Communication Skills	None
Aggregate	<input type="text"/> %	Backlogs	<input type="text"/>

PREDICT

SELECTION:

Placement Prediction

STUDENT PLACEMENT PREDICTION



Name	Anurag	Technical Skills	8
Roll no	18007052	Communication Skills	7
Aggregate	78%	Backlogs	1


PREDICT

You might get Selected

REJECTION:

Placement Prediction

STUDENT PLACEMENT PREDICTION



Name	Swamy	Technical Skills	5
Roll no	058	Communication Skills	6
Aggregate	80 %	Backlogs	0


PREDICT

You might NOT get Selected

VALIDATION:

Placement Prediction

STUDENT PLACEMENT PREDICTION



Name	<input type="text" value="Harshal"/>	Technical Skills	<input type="text" value="8"/>
Roll no	<input type="text" value="036"/>	Communication Skills	<input type="text" value="6"/>
Aggregate	<input type="text" value="85"/> %	Backlogs	<input type="text"/>

please fill every field

PREDICT

CONCLUSION:

Student Placement Prediction is a system which predicts student placement status using machine learning techniques. Many projects are there related to educational sector, all these mainly concentrate on student performance predictions. All these predictions help the institute to improve the student performance and can come up with 100% results. Many of the previous projects concentrate on a less number of parameters such as CGPA and Arrears for placement status prediction which leads to less accurate results, but proposed work contains many educational parameters to predict placement status which will be more accurate.

FUTURE SCOPE:

- Employing such a system is a proactive way to use data to manage, operate, and evaluate educational institute in a better way. Depending on the quality and implementation of the underlying data, such a system could address a wide range of problems by distilling data from any combination of education records maintenance system. It helps an educational institute improving the quality and placements of students being graduated from their institute. This system can also be implemented in different non educational institutes like business corporates, sports academies and manufacturing companies where the challenge would be taking into consideration the current market scenario as one of the most important factors affecting quality of their products and employee.
- There are many optimization algorithms available which can reduce the number of loops required for higher stability. One of them is known as Stochastic Gradient Decent Algorithm. Newton's gradient method and hessian matrix methods can also give you the good value.

REFERENCES

RESEARCH PAPER REFERENCES:

1. Student Placement Analyzer: A Recommendation System Using Machine Learning - 2017 International Conference on Advanced Computing and Communication Systems (ICACCS - 2017), Jan. 06 – 07, 2017, Coimbatore, INDIA
2. STUDENT PLACEMENT PREDICTION USING MACHINE LEARNING - International Research Journal of Engineering and Technology (IRJET)

