**SOCIALLY RELEVANT PROJECT-2022**

**TEAM - 6**

**Engineering Placement Prediction using Machine Learning**

Team Members:

1)S Meyyappan

2)KN Balasubramanian

3)A S Pruthiev

**PROBLEM STATEMENT :**

* Placements hold great importance for students and educational institutions. It helps the students to build a strong foundation for the professional career as well as a good placement record gives a competitive edge to a college university in the education market.
* This Project focuses on a system that predicts if a student would be placed or not based on the student’s qualifications and historical data
* If a student predicted as not placed, we provide suggestions on the skills to be improved.

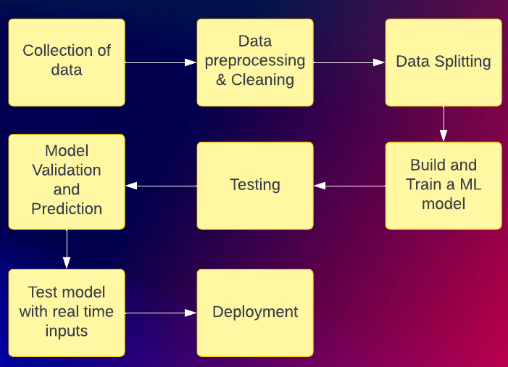
**OBJECTIVE:**

* By using this model we can increase the placement rates in the institutions.
* We will provide suggestions on the areas to be improved for those whom our model predicts as not placed.
* This project provides motivation and boost the self confidence of the students towards placement.

**PROPOSED SOLUTION:**

* The outcome is classified whether the student is placed in product sector, service sector or not placed.
* We collect features such as CGPA, Number of Internships, Projects, Backlogs, Stream, Number of Coding Problems solved.
* The dataset is collected from graduated and final year students from circuit and non-circuit streams.
* A machine learing model is to be developed to predict the student's placement future.

**ARCHITECTURE:**

****

**ENVIRONMENT/ FRAMEWORKS USED:**

* Visual Studio Code
* Java Collection Frame Work
* Intellij IDEA Community version
* Google Forms

**IMPLEMENTATION**:

**Dataset.java**

package NaiveBayesClassifier;

import joinery.DataFrame;

import java.io.IOException;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.Scanner;

public class Dataset {

//placementDataFrame to store the dataset as DataFrame Data Structure

private DataFrame<Object> placementDataFrame;

//Stores the number of rows

private int numberOfRows;

//Stores the number of cols

private int numberOfCols;

//This is used to return an arraylist which is used to set the row indices of the DataFrame instance

public ArrayList<Integer> getRowIndices() {

ArrayList<Integer> rowIndices = new ArrayList<>();

for (int i = 0; i <= 2965; i++) rowIndices.add(i);

return rowIndices;

}

// This is used to return an arraylist which is used to set the column indices of the DataFrame instance

public ArrayList<Integer> getColumnIndices() {

ArrayList<Integer> columnIndices = new ArrayList<>();

for (int i = 0; i <= 7; i++) columnIndices.add(i);

return columnIndices;

}

//This constructor initializes the DataFrame Instance with the Integer row indices and colum indices

public Dataset() {

placementDataFrame = new DataFrame<>(getRowIndices(), getColumnIndices());

numberOfCols = 8;

numberOfRows = 2966;

}

//Used to print the dataset as like a DataFrame

public void printDataset() {

System.out.println(placementDataFrame);

}

//Return the number of rows in the DataFrame Instance

public int getNumberOfRows() {

return numberOfRows;

}

// Return the number of cols in the DataFrame Instance

public int getNumberOfCols() {

return numberOfCols;

}

// This returns the DataFrame which rows with features alone

public DataFrame<Object> getFeatures() {

return placementDataFrame.slice(0, numberOfRows, 0, numberOfCols - 1);

}

//This returns the Dataframe with Outcome Column alone

public DataFrame<Object> getOutcomes() {

return placementDataFrame.slice(0, numberOfRows, 7, numberOfCols);

}

public DataFrame<Object> getDataFrame(){

return placementDataFrame;

}

//Reads the dataset using File Handling from a txt file

public void readDataset() throws IOException {

Path filePath = Paths.get("D:\\SRP\\src\\main\\resources\\PlacementDataset.txt");

Scanner datasetReader = new Scanner(filePath);

int data = -1;

int row = -1;

int col = 0;

while (datasetReader.hasNext()) {

data = datasetReader.nextInt();

if (col == 0) {

row = row + 1;

}

placementDataFrame.set(row, col, data);

col = (col + 1) % 8;

}

}

}

**Main.java**

package NaiveBayesClassifier;

import java.io.IOException;

public class Main {

public static void main(String[] args) throws IOException {

Dataset dataset = new Dataset();

dataset.readDataset();

}

}

**CONCLUSION:**

* Our project provide suggestions to cover up the lagging skills that are required for getting placed in IT sector organisation.
* This project will help in increasing the placement rates among various streams.
* The developed model will be very usefull to self assess one's techinical skills.

**REFERENCES:**

* A. S. Sharma, S. Prince, S. Kapoor and K. Kumar, "PPS — Placement prediction system using logistic regression," 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), 2014, pp. 337-341, doi: 10.1109/MITE.2014.7020299.
* A. Giri, M. V. V. Bhagavath, B. Pruthvi and N. Dubey, "A Placement Prediction System using k-nearest neighbors classifier," 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), 2016, pp. 1-4, doi: 10.1109/CCIP.2016.7802883.
* Harihar, Varsha Kailas, and D. G. Bhalke. "Student Placement Prediction System using Machine Learning." SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology 12, no. SUP 2 (2020): 85-91.