



---

# PROJECT REPORT

---

## Identifying Patterns and Trends in Campus Placement Data using Machine Learning



**Team Leader** - PRADEEP B

**Team members** - MONIKA M

PRAVEEN R

PRIYADHARSHINI R

# 1. Introduction

## 1.1 Overview

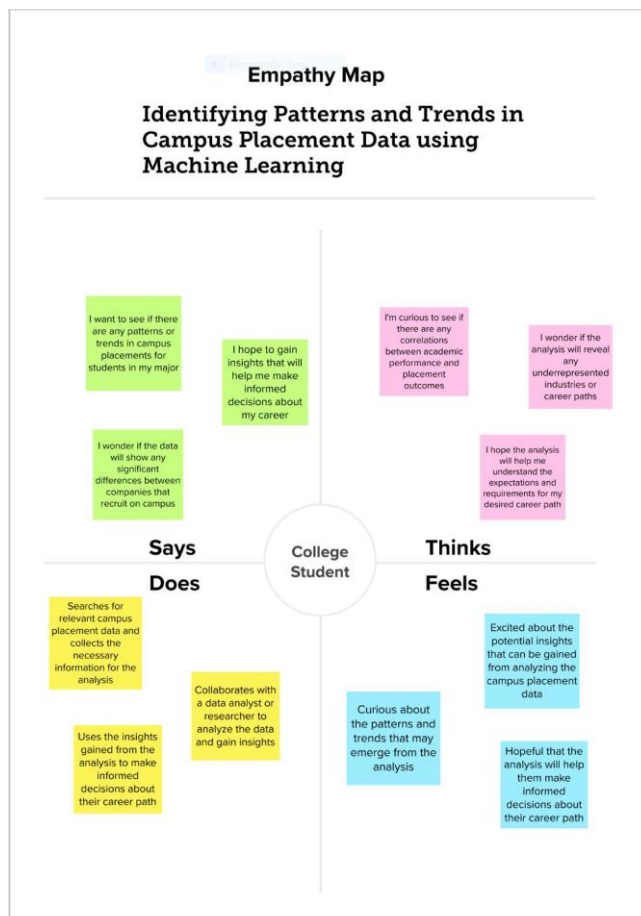
The project aims to analyse and identify patterns and trends in campus placement data using machine learning techniques. The project analyses data collected from campus placement drives conducted by various colleges and universities. The project uses machine learning algorithms to identify patterns and trends in the data that can help in predicting the placement status of future candidates.

## 1.2 Purpose

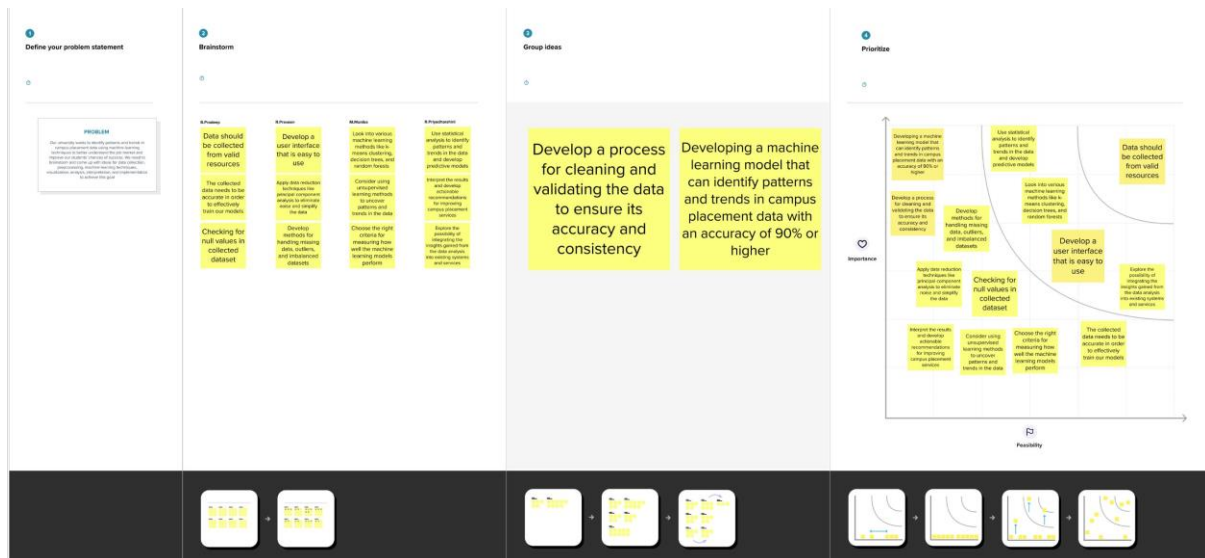
The purpose of the project is to provide a tool for colleges and universities to analyse and understand the placement data of their students. The project helps in identifying the factors that affect the placement status of candidates and provides insights into the placement process. The project also helps in predicting the placement status of future candidates, which can be useful for students, colleges, and recruiters.

# 2. Problem Definition & Design Thinking

## 2.1 Empathy Map



## 2.2 Ideation & Brainstorming Map



### 3. Result

The final output of the project includes a machine learning model that can predict the placement status of future candidates based on the analysis of the campus placement data. The project also provides various visualizations and insights into the placement data that can help in understanding the placement process better.

127.0.0.1:5000

127.0.0.1:5000

Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Go to previous page

Fill the details

Age

Gender M(0),F(1)

Stream CS(0),IT(1),ECE(2),Mech(3),EEE(4),Civil(5)

Internships

CGPA

Number of backlogs

submit



Fill the details

22

0

2

1

8

1

submit



## **4. Advantages & Disadvantages**

### **4.1 Advantages**

- Provides insights into the placement process
- Helps in identifying factors that affect the placement status of candidates
- Predicts the placement status of future candidates
- Helps in making data-driven decisions

### **4.2 Disadvantages**

- The accuracy of the machine learning model depends on the quality of the data
- The project may not be able to capture all the factors that affect the placement status of candidates

## **5. APPLICATIONS**

The project can be applied in various domains, including:

- Educational institutions: To analyse the placement data of their students and improve the placement process.
- Recruiters: To predict the placement status of future candidates and make data-driven hiring decisions.

## **6. CONCLUSION**

The project provides a tool for analysing and understanding the campus placement data using machine learning techniques. The project helps in identifying patterns and trends in the data that can provide valuable insights into the placement process. The project also provides a machine learning model that can predict the placement status of future candidates.

## **7. FUTURE SCOPE**

The project can be further improved by:

- Adding more data sources to improve the accuracy of the machine learning model
- Using advanced machine learning algorithms to improve the accuracy of the predictions

## 8. APPENDIX

### A. Source Code

```
from flask import Flask, render_template, request
import pickle
import numpy as np

app = Flask(__name__)

model = pickle.load(open("rdf.pkl", 'rb'))

@app.route('/')
def hello():
    return render_template("index.html")

@app.route('/guest', methods = ["POST"])
def Guest():
    return render_template("secondpage.html")

@app.route('/y_predict', methods=["POST"])
def y_predict():
    if request.method == "POST":
        sen1 = request.form["sen1"]
        sen2 = request.form["sen2"]
        sen3 = request.form["sen3"]
        sen4 = request.form["sen4"]
        sen5 = request.form["sen5"]
        sen6 = request.form["sen6"]
        X_test = np.array([[sen1, sen2, sen3, sen4, sen5, sen6]],
dtype=float)
        prediction = model.predict(X_test)
        prediction = prediction[0]
        return render_template("secondpage.html", y=prediction)
    else:
        return "Invalid request method"

if __name__ == '__main__':
    app.run(debug=True)
```