IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

**TEAM ID**: NM2023TMID25332

**TEAM LEADER:**

**VALLI V :** EE78F53B65A5AC7A5994C176A6E48335

**TEAM MEMBERS:**

**MAHALAKSHMI J**

**SRIDEVI S**

**SANTHINI S**

|  |  |
| --- | --- |
| SI NO | **TITLE** |
| 1. | **INTRODUCTION** |
|  | 1.1 PROJECT OVERVIEW |
|  | 1.2 PURPOSE OF PROJECT |
| 2. | **PROBLEM DEFINITION AND DESIGN THINKING** |
|  | 2.1 EMPATHY MAP |
|  | 2.2 IDEATION AND BRAIN STORMING |
| 3. | **RESULTS** |
| 4. | **ADVANTAGES AND DISADVANTAGES** |
| 5. | **APPLICATIONS** |
| 6. | **CONCLUSION** |
| 7. | **FUTURE SCOPE** |
| 8. | **APPENDIX** |
|  | 1. Source code |



**1.INTRODUCTION:**

In recent years, machine learning has emerged as a powerful tool for analyzing large datasets and making predictions based on patterns identified in that data. One area where machine learning can be particularly useful is in the field of campus placements. Campus placements are an important step in the career development of students, and are also crucial for companies looking to recruit new talent. However, the process of campus placements can be complex, with a wide range of factors that can impact a student's success in getting placed, as well as an employer's success in identifying the best candidates for their open positions.

The objective of this machine learning project is to develop a model that can analyze historical data from campus placements to identify the factors that have the biggest impact on placement success, and to use this information to provide recommendations to both students and employers. The model will be trained on data that includes information on students' academic performance, skill set, internships, and previous work experience, as well as information on the job requirements and employer preferences for specific job openings.

The project will utilize a range of machine learning techniques, including supervised learning (to predict placement success and recommend job openings), unsupervised learning (to identify patterns in the data), and natural language processing (to analyze job descriptions and student resumes).

The model will be evaluated using metrics such as accuracy, precision, recall, and F1 score, and will be optimized through techniques such as hyperparameter tuning and feature selection.

The expected outcomes of this project include a machine learning model that can accurately predict placement success and recommend job openings to students, as well as insights into the factors that impact placement success and the recruiters that have been most successful in hiring students.

These outcomes will help students make more informed decisions about their career paths and job search strategies, and will help employers target their recruitment efforts more effectively. Ultimately, the goal of this project is to use machine learning to improve the efficiency and effectiveness of the campus placement process for both students and employers.

* 1. **OVERVIEW**

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. College recruiting is typically a tactic for medium- to large-sized companies with high-volume recruiting needs, but can range from small efforts (like working with university career centers to source potential candidates) to large-scale operations (like visiting a wide array of colleges and attending recruiting events throughout the spring and fall semester).

Campus recruitment often involves working with university career services centers and attending career fairs to meet in-person with college students and recent graduates. Our solution revolves around the placement season of a Business School in India. Where it has various factors on candidates getting hired such as work experience, exam percentage etc., Finally it contains the status of recruitment and remuneration details.

We will be using algorithms such as KNN, SVM and ANN. We will train and test the data with these algorithms. From this the best model is selected and saved in .pkl format. We will be doing flask integration and IBM deployment.

The project will utilize a range of machine learning techniques, including supervised learning (to predict placement success and recommend job openings), unsupervised learning (to identify patterns in the data), and natural language processing (to analyze job descriptions and student resumes). The model will be evaluated using metrics such as accuracy, precision, recall, and F1 score, and will be optimized through techniques such as hyperparameter tuning and feature selection.

* 1. **PURPOSE**

**Improving placement success for students**: The machine learning model can help students identify job openings that match their skills and preferences, and provide personalized recommendations on how to improve their chances of getting hired. This can lead to higher placement rates and better career opportunities for students.

**Increasing recruiter efficiency**: The model can help employers target their recruitment efforts more effectively by identifying the factors that are most important in predicting successful placements. This can help recruiters save time and resources by focusing on the most promising candidates.

**Providing insights into placement trends and recruiter performance**: The model can provide insights into the factors that impact placement success, as well as the recruiters that have been most successful in hiring students. This information can be used to improve the overall effectiveness of the campus placement process and help both students and employers make more informed decisions.

**Optimizing job search strategies**: The model can provide personalized recommendations to students on how to optimize their job search strategies, including identifying the most promising job openings and optimizing their resumes and cover letters.

**Promoting diversity and inclusion**: The model can be used to identify potential biases in the campus placement process and provide recommendations for promoting diversity and inclusion in the recruitment process.

**Reducing skills gaps**: The machine learning model can identify the skills that are most in demand by employers and provide recommendations to students on how to develop those skills. This can help reduce skills gaps and ensure that students are better prepared for the job market.

**Increasing employer brand awareness**: The model can provide insights into the employers that are most successful in hiring students and promote those employers to students. This can help increase brand awareness and improve the employer's reputation on campus.

**Promoting employer-student matching**: The model can help match students with employers that are the best fit for their skills and preferences. This can lead to better job satisfaction for students and more successful placements for employers.

**Improving recruitment marketing**: The model can provide insights into the job search behavior and preferences of students, which can be used to improve recruitment marketing efforts and better target potential candidates.

**Providing insights into industry trends**: The model can provide insights into the trends and challenges facing different industries, which can be used by both students and employers to make more informed decisions about their career paths and recruitment strategies.

**2 PROBLEM DEFINITION AND DESIGN THINKING:**

**Problem Definition:**

The campus placement process can be challenging for both students and employers. Students often struggle to find job openings that match their skills and preferences, while employers face difficulties in identifying the most promising candidates from a large pool of applicants. Additionally, the campus placement process can be time-consuming and resource-intensive for both parties. As a result, there is a need for a more efficient and effective way to match students with potential employers.

**Design Thinking:**

Design thinking is a problem-solving approach that focuses on understanding the needs and preferences of users and creating solutions that meet those needs. To apply design thinking to the campus placement problem, we can follow the following steps:

**Empathize**:

We need to understand the needs and pain points of both students and employers when it comes to the campus placement process. This can involve conducting interviews, surveys, and focus groups to gather insights.

**Define**:

Based on our research, we can define the problem statement and identify the key challenges facing students and employers in the campus placement process.

**Ideate**:

We can brainstorm potential solutions to the problem statement, considering both traditional and innovative approaches. This can involve creating a brainstorm map to organize our ideas and identify the most promising solutions.

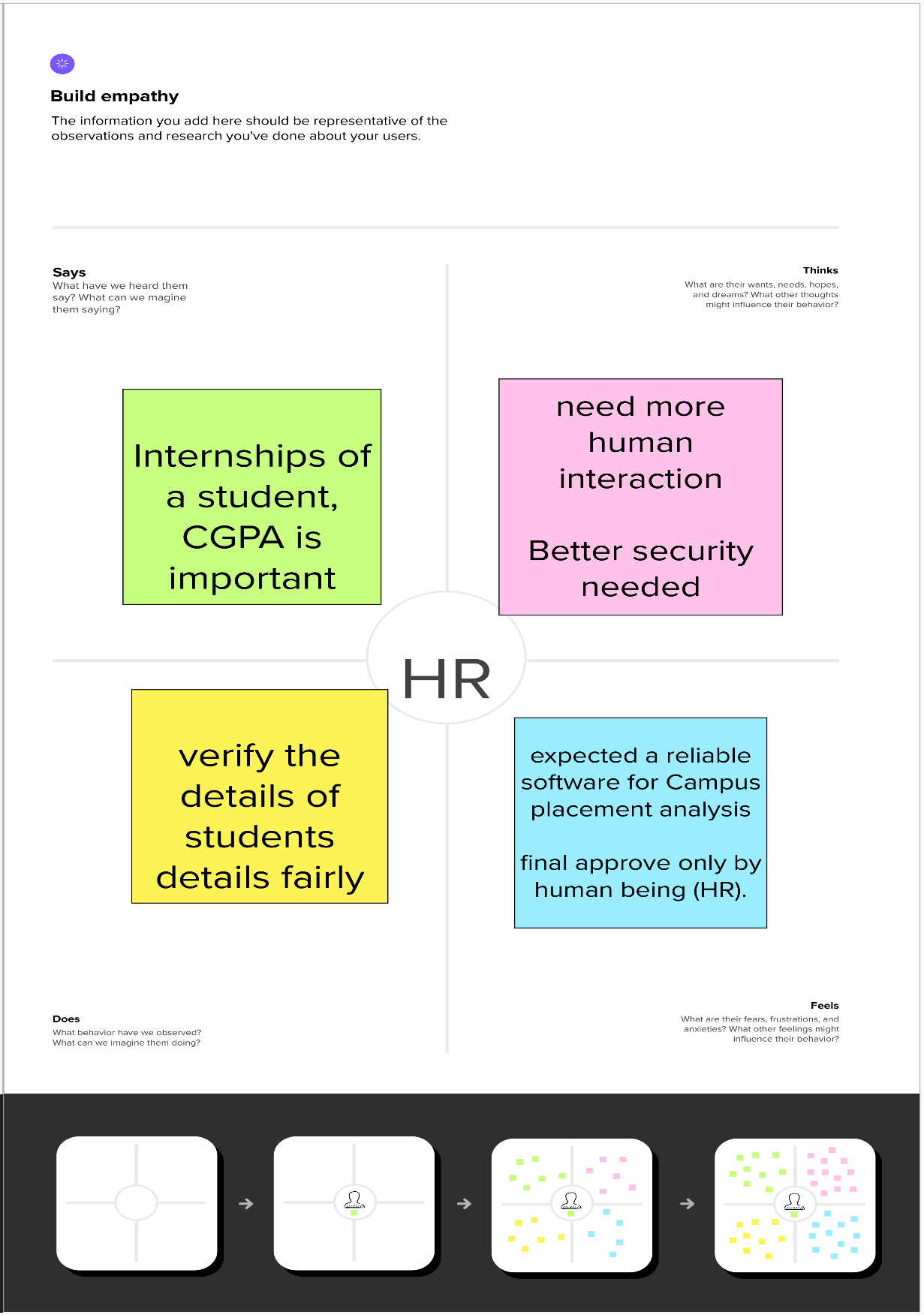
**Prototype**:

We can create a prototype of our solution, such as a machine learning model that uses student data to match them with potential employers.

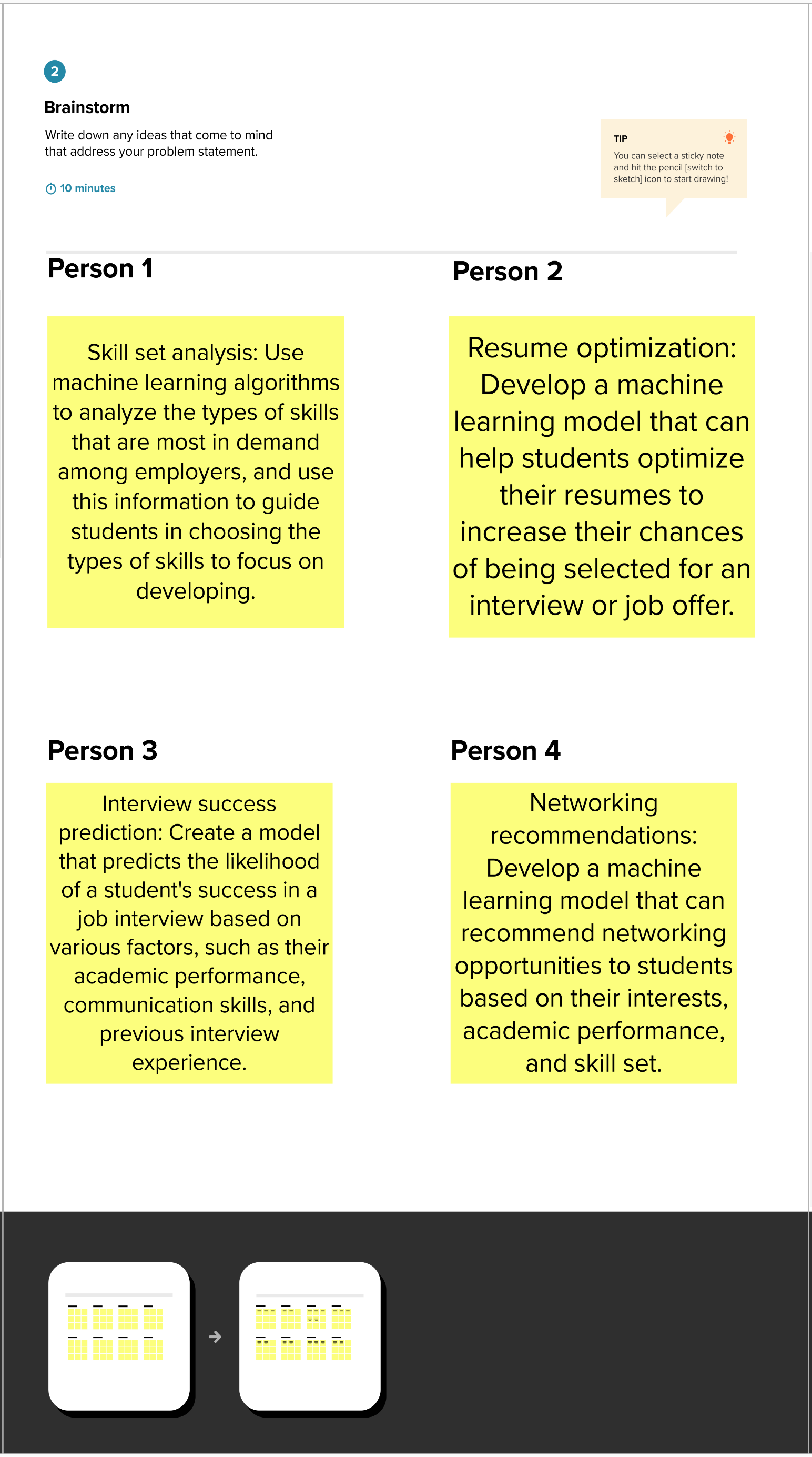
**Test**:

We can test our prototype with a small group of users and gather feedback on its effectiveness and usability. Based on the feedback, we can refine our solution and iterate on the design.

**2.1. Empathy Map**

****

**2.2 IDEATION AND BRAINSTROMING MAP:**

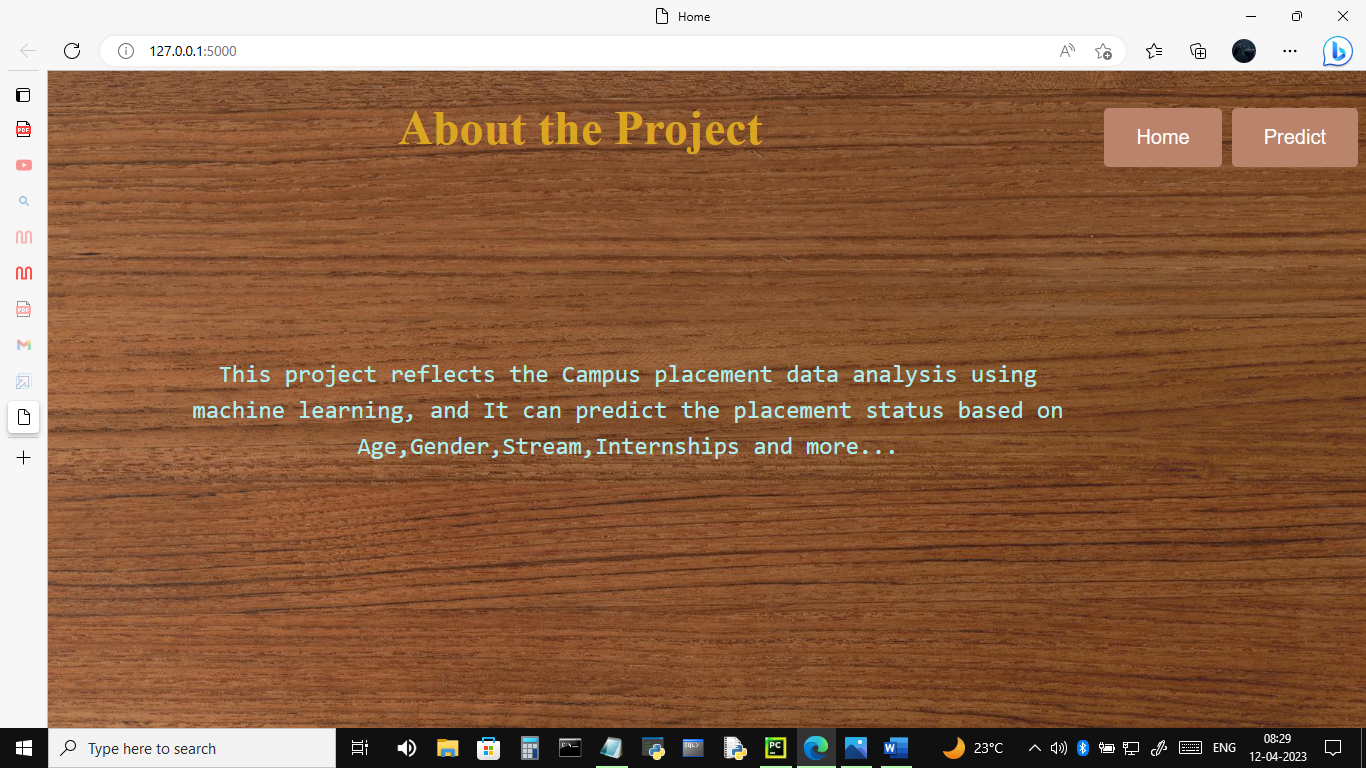
****

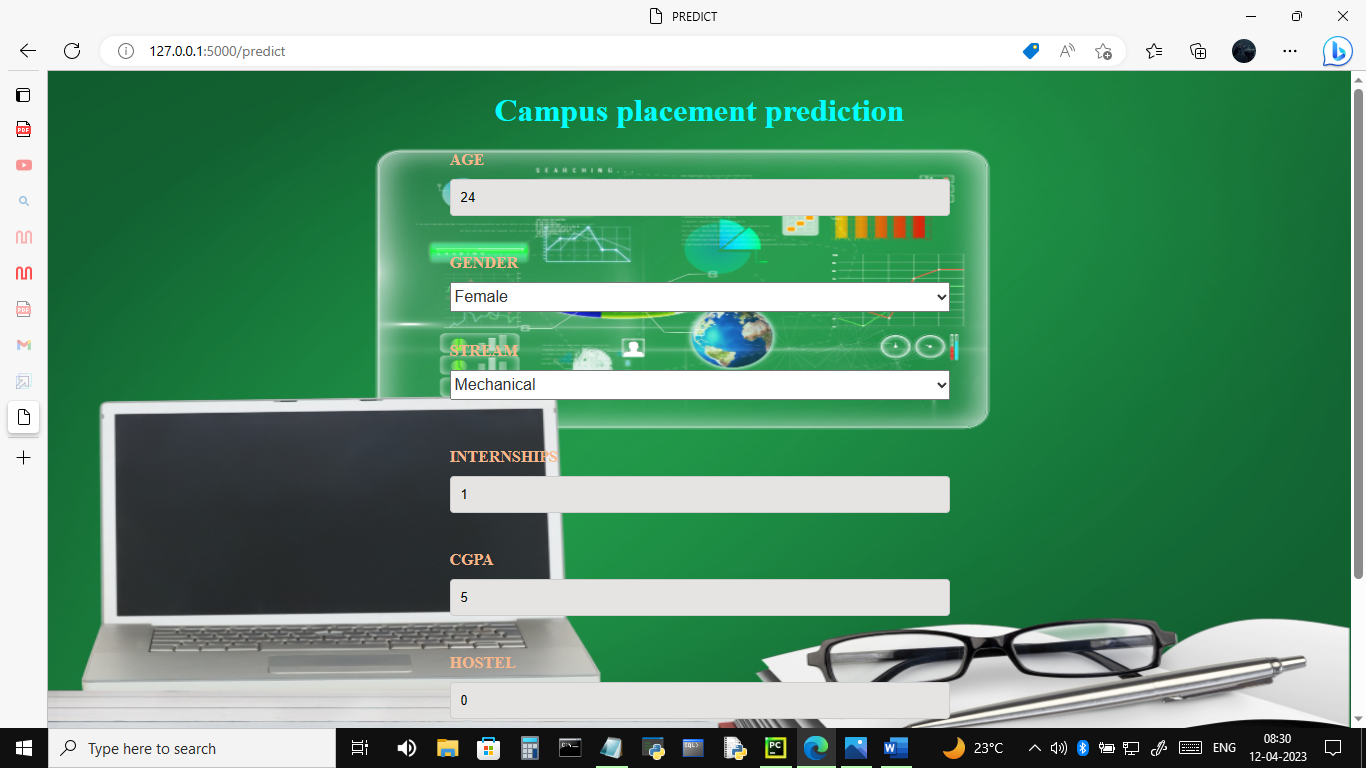
**3. RESULT:**

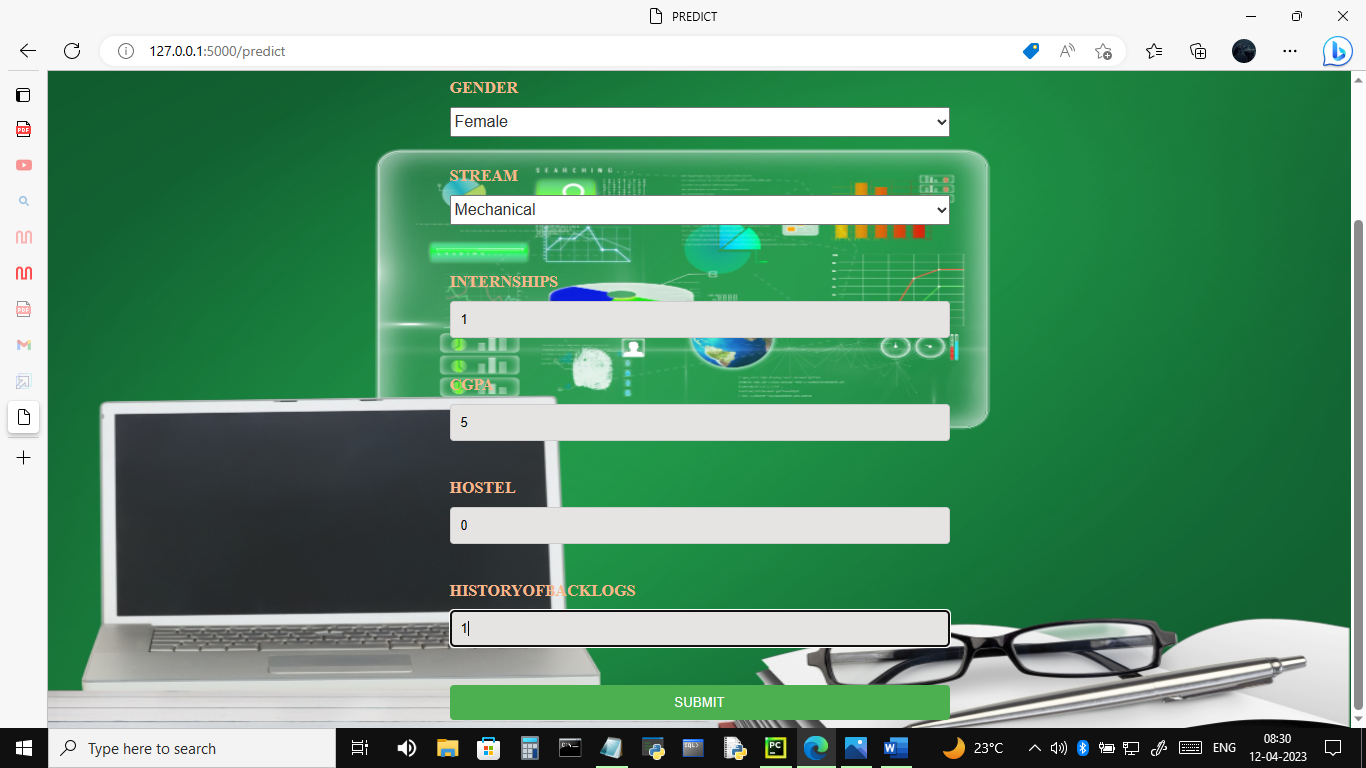
As a result of our project we use a K-Neighbors Classifier algorithm to predict the values based on details. It predict the accuracy of 83%.

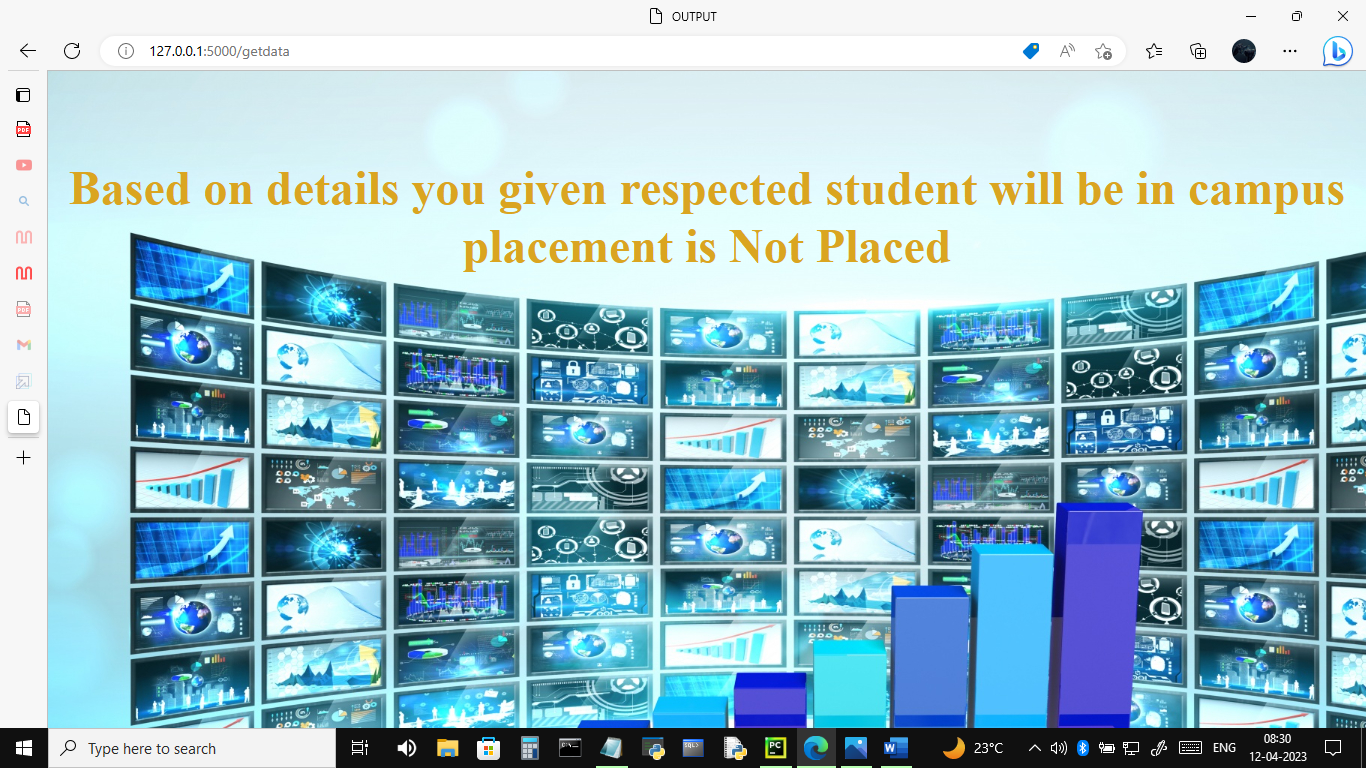
The output of the KNN algorithm is a classification label for a new input based on its k-nearest neighbors in the training data.

The performance of the KNN algorithm would depend on various factors, such as the choice of distance metric, the value of k, and the quality and quantity of the training data. The output of the KNN algorithm can be evaluated using various performance metrics, such as accuracy

****

****

****

****

**4. ADVANTAGES AND DISADVANTAGES**

**Advantages**:

* Increased efficiency: Machine learning can help automate the campus placement process by analyzing large amounts of student data and identifying patterns that can help optimize the process.
* Personalized learning paths: Machine learning algorithms can analyze the skills and performance of individual students and recommend personalized learning paths to help them increase their employability and enhance their chances of getting placed.
* Accurate prediction: Machine learning models can be trained on historical placement data to predict the likelihood of a student getting placed in a particular company or industry based on their profile and skills.
* Improved candidate matching: Machine learning algorithms can match student profiles with job requirements and recommend the best candidates for a particular job based on their skills, experience, and other factors.

**Disadvantages**:

* Dependence on data quality: The quality of the data used to train machine learning models can greatly impact the accuracy of the results. If the data is incomplete or biased, it can lead to incorrect predictions and recommendations.
* Lack of transparency: Some machine learning models can be difficult to interpret and may lack transparency, making it challenging to understand how the model arrived at its predictions.
* Limited scope: Machine learning models are only as effective as the data they are trained on. If the training data is limited or does not represent the full range of job opportunities, the model may not be able to make accurate predictions.
* Cost: Developing and implementing machine learning algorithms can be expensive, requiring significant investment in hardware, software, and personnel.

**5. APPLICATIONS:**

**HR Departments**: HR departments can use a campus placement analysis project to analyze data from past placement programs and identify factors that contribute to successful placements. This information can be used to improve recruitment strategies, identify skill gaps in candidates, and make better hiring decisions.

**Recruitment Agencies**: Recruitment agencies can use machine learning algorithms to analyze job requirements and candidate profiles to recommend suitable job roles. This can help recruitment agencies save time and resources by automating the candidate selection process and providing clients with a more targeted and efficient recruitment service.

**Education Institutions**: Educational institutions can use a campus placement analysis project to track the performance of their students in placement programs and identify areas where students need additional support. This can help educational institutions develop more effective training programs and improve the overall quality of their graduates.

**Staffing Firms**: Staffing firms can use machine learning models to analyze job requirements and match candidates with suitable job roles. This can help staffing firms improve the quality of their candidate selection process and increase the success rate of their placements.

**Job Boards**: Job boards can use machine learning algorithms to analyze job postings and recommend suitable job roles to candidates based on their profiles and interests. This can help job boards improve the quality of their recommendations and increase engagement among their users.

**7. CONCLUSION:**

a campus placement analysis project using machine learning can provide valuable insights into the recruitment process, improving the efficiency and effectiveness of the placement program. By analyzing factors such as academic performance, work experience, and skill set, machine learning models can predict the likelihood of a student getting placed, recommend suitable job roles, identify skill gaps, and track the progress of placements in real-time. This information can be used by recruiters, HR departments, recruitment agencies, education institutions, staffing firms, and job boards to improve their recruitment strategies, make better hiring decisions, and provide more targeted and efficient recruitment services. Overall, a campus placement analysis project using machine learning has the potential to benefit companies across a variety of industries by improving the quality of their placements and increasing the success rate of their recruitment programs.

**7. FUTURE SCOPE:**

The future scope of campus placement analysis using machine learning is vast, with the potential to revolutionize the recruitment process. Here are some potential future developments:

**Personalized Recommendations**: As machine learning algorithms become more sophisticated, they will be able to provide more personalized recommendations to candidates based on their unique profile, interests, and skills.

**Predictive Analytics**: Machine learning models will be able to use data from a variety of sources, such as social media and online platforms, to predict which students are most likely to succeed in a particular job role or industry.

**Improved Candidate Screening**: Machine learning algorithms can be used to screen candidates more efficiently and accurately, reducing the time and resources required to select suitable candidates for a job.

**Automated Interviews**: With the development of natural language processing (NLP) technology, machine learning algorithms can be used to conduct automated interviews with candidates, providing a more objective and standardized assessment of their skills and suitability for a job.

**Increased Transparency**: Machine learning models can be used to provide more transparent and unbiased hiring decisions, reducing the impact of human bias and discrimination in the recruitment process.

**9. APPENDIX: A)Source code:**

**app.py**

import pickle

from flask import Flask, render\_template, request

app = Flask(\_\_name\_\_)

with open('Placement\_knn.pkl', 'rb') as f:

data = pickle.load(f)

@app.route('/')

def entry():

return render\_template('home.html')

@app.route('/home')

def home():

return render\_template('home.html')

@app.route('/predict')

def predict():

return render\_template('predict.html')

@app.route('/getdata', methods=['post'])

def submit():

age = request.form['age']

gender = request.form['gender']

stream = request.form['stream']

internships = request.form['internships']

cgpa = request.form['cgpa']

hostel = request.form['hostel']

historyofbacklogs = request.form['historyofbacklogs']

variables = [[int(age), int(gender), int(stream), int(internships), int(cgpa), int(hostel), int(historyofbacklogs)]]

result = data.predict(variables)

result.astype(int)

if result == 0:

output = 'Not Placed'

else:

output = 'Placed'

return render\_template('submit.html', output=output)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

**home.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Home</title>

<style>

body{

background-image: url('static/images/home.jpg');

}

.wrapper {

display: flex;

justify-content: space-between;

align-items: center;

text-align:center;

padding-top:20px;

}

.button-wrapper{

display: flex;

gap: 10px;

}

h1{

margin:auto;

display: inline-block; /\* Set display to inline-block \*/

vertical-align: middle;

text-align:center;

color:#daa520;

font-size: 48px; /\* Set font size to 32 pixels \*/

font-weight: bold; /\* Set font weight to bold \*/

line-height: 1.2; /\* Set line height to 1.2 \*/

margin-bottom: 20px; /\* Set margin bottom to 20 pixels \*/

}

p {

font-size: 18px; /\* Set font size to 16 pixels \*/

line-height: 1.5; /\* Set line height to 1.5 \*/

text-align:center;

font-size:24px;

color:#afeeee;

text-size:28px;

font-family:monospace,san-serif;

margin: 10px 0; /\* Set margin to 10 pixels top and bottom, 0 pixels left and right \*/

position: absolute; /\* Set position to absolute \*/

top: 50%; /\* Set top to 50% \*/

left: 30%; /\* Set left to 50% \*/

transform: translate(-30%, -50%);

}

button {

background-color: #BA836C; /\* Set the background color \*/

color: white; /\* Set the text color \*/

padding: 18px 32px; /\* Set the padding \*/

border: none; /\* Remove the border \*/

border-radius: 5px; /\* Add rounded corners \*/

cursor: pointer; /\* Change cursor on hover \*/

font-size:20px;

text-align:right;

}

button:hover {

background-color: #3e8e41; /\* Change background color on hover \*/

}

.container {

position: relative; /\* Set position to relative \*/

text-align:right;

}

.button {

position: absolute; /\* Set position to absolute \*/

top: 0; /\* Align to top of container \*/

right: 0; /\* Align to right of container \*/

}

</style>

</head>

<body>

<div class="wrapper">

<h1>About the Project</h1>

<div class="button-wrapper">

<a href="/home"><button>Home</button></a>

<a href="/predict"><button>Predict</button></a>

</div>

</div>

<p>This project reflects the Campus placement data analysis using machine learning, and It can predict the placement status based on Age,Gender,Stream,Internships and more... </p>

<h2></h2>

</body>

</html>

**Predict.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>PREDICT</title>

<style>

body {

background-image: url('static/images/predict.jpg');

background-repeat:no-repeat;

background-attachment:fixed;

background-size:100% 100%;

}

form {

display: flex;

flex-direction: column;

max-width: 500px;

margin: 0 auto;

color:#F8B88B;

}

select{

margin-bottom:30px;

height:30px;

font-size: 16px;

color: #333;

}

select:hover {

background-color: #eee;

border: 1px solid #aaa;

}

select:focus {

background-color: #fff;

border: 2px solid #007bff;

outline: none;

}

label {

font-weight: bold;

margin-bottom: 10px;

}

input, textarea {

padding: 10px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

background-color: #4CAF50;

color: white;

border: none;

padding: 10px 20px;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #3e8e41;

}

h1{

text-align:center;

color:#00ffff;

}

input[type="text"] {

background-color: #E5E4E2;

}

</style>

</head>

<body>

<h1>Campus placement prediction</h1>

<form action="/getdata" method="post" autocomplete="off">

<label for='age'>AGE</label>

<input type="text" id="age" name="age" required><br>

<label for='gender'>GENDER</label>

<select id="gender" name="gender" required>

<option value="" disabled selected hidden></option>

<option value=1>Male</option>

<option value=0>Female</option>

</select>

<label for='stream'>STREAM</label>

<select id="stream" name="stream" required>

<option value="" disabled selected hidden></option>

<option value=0>Electronics and Communication</option>

<option value=1>Computer Science</option>

<option value=2>Information Technology</option>

<option value=3>Mechanical</option>

<option value=4>Electrical</option>

<option value=5>Civil</option>

</select><br>

<label for='internships'>INTERNSHIPS</label>

<input type="text" id="internships" name="internships" required><br>

<label for='cgpa'>CGPA</label>

<input type="text" id="cgpa" name="cgpa"required><br>

<label for='hostel'>HOSTEL</label>

<input type="text" id="hostel" name="hostel"required><br>

<label for='historyofbacklogs'>HISTORYOFBACKLOGS</label>

<input type="text" id="historyofbacklogs" name="historyofbacklogs"><br>

<button type="submit">SUBMIT</button>

</form>

</body>

</html>

**Submit.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>OUTPUT</title>

<style>

body{

background-image:url('/static/images/submit.jpg');

}

h1{

margin:auto;

display: inline-block; /\* Set display to inline-block \*/

vertical-align: middle;

text-align:center;

color:#daa520;

font-size: 48px; /\* Set font size to 32 pixels \*/

font-weight: bold; /\* Set font weight to bold \*/

line-height: 1.2; /\* Set line height to 1.2 \*/

margin-bottom: 20px; /\* Set margin bottom to 20 pixels \*/

padding-top:80px;

</style>

</head>

<body>

<h1>Based on details you given respected student will be in campus placement is {{output}}</h1>

</body></html>