IBM HACK CHALLENGE 2023

Identifying Patterns and Trends in Campus Placement Data using Machine Learning

using

Applied Data Science

Team Name: Wingardium Leviosa

S.no	Name	Roll No
1	MADHU SREE K	20BKT0083
2	GOKULAKRISHNAN K	20MIS0256

Submitted to the SmartInternz Programme in fulfilment of the requirements for the Project

(Identifying Patterns and Trends in Campus Placement Data using Machine Learning)

Year of Submission: 2023

SmartBridge Educational Services Pvt. Ltd.

6th Floor, Sundarayya Vignana Kendram, Technical Block, Madhava Reddy Colony,

Gachibowli, Hyderabad, Telangana 500032

DECLARATION:

This is to certify that Report entitled “Identifying Patterns and Trends in Campus Placement Data using Machine Learning” which is submitted by me in fulfilment of the requirement for the award of IBM Hack Challege 2023 using **Applied Data Science** at **SmartBridge Educational Services Pvt. Ltd.** We took the help of other materials in our dissertation which have been properly acknowledged. This report has not been submitted to any other Institute for the award of any other degree.

Date: 30/08/2023

S.no	Name	Roll No	Signature
1	MADHU SREE K	20BKT0083	Madhu Sree K
2	GOKULAKRISHNAN K	20MIS0256	Gokulakrishnan K

CERTIFICATE:

This is to certify that the project report entitled “*Identifying Patterns and Trends in Campus Placement Data using Machine Learning*” submitted to **SmartBridge Educational Services Pvt. Ltd.** in fulfilment of the requirement for the award of the ***IBM Hack Challege 2023 of Applied Data Science*** during the Period (**July – August, 2023**), is a bonafide record of the project work carried out by them under my guidance and supervision.

CONTENT

1. INTRODUCTION

a. Description

Campus placement data comprises information about students, their academic performance, skills, internships, and their eventual placement outcomes. The objective is to extract valuable insights from this data to understand factors influencing placement success and develop strategies for improving the placement process. by identifying patterns and trends in campus placement data using machine learning techniques.

The process of campus placements plays a crucial role in connecting graduating students with potential employers, bridging the gap between academia and industry. In recent years, the use of machine learning techniques has gained prominence in various fields, including human resources and recruitment. By applying machine learning to campus placement data, educational institutions and companies can gain valuable insights into patterns and trends that can aid decision-making and improve the placement process.

Objective:

The primary objective of using machine learning in analyzing campus placement data is to extract meaningful patterns, trends, and insights that can be used to optimize the placement process, enhance career guidance for students, and improve employer-university collaborations.

b. Novelty / Uniqueness:

The novelty and uniqueness of this solution lie in its departure from traditional employment practices and its focus on meritocracy and equal opportunities, while simultaneously acknowledging and addressing existing gender disparities in salary. Here's a breakdown of the different aspects that make this solution stand out:

1. Educational percentages as the primary influencer: Unlike many conventional hiring processes that heavily consider previous employment and work experience, this solution prioritizes the educational percentages of candidates. By doing so, it emphasizes a candidate's academic achievements and potential for success, rather than relying solely on their past professional track record. This approach allows for a more equal footing among candidates, irrespective of their prior work experiences or opportunities.

2. Limited influence of previous employment: While previous work experiences are often valued in recruitment processes, this solution challenges that conventional belief by stating that previous employment has little to no impact on a candidate's placement. This approach helps to level the playing field and prevent candidates with less favorable previous employment experiences from being unfairly disadvantaged when compared to those with more prestigious prior roles.

3. Absence of gender discrimination: The solution affirms that there is no gender discrimination in employment. This aspect highlights the commitment to equal opportunities and ensures that candidates are evaluated solely based on their qualifications, skills, and abilities rather than their gender.

4. Gender-based salary disparity: In a surprising twist, the solution acknowledges that men were offered higher salaries. This admission immediately recognizes an existing problem in gender-based compensation disparities. By acknowledging this disparity, the solution sets the stage for further action to rectify the issue and promote gender equality in terms of compensation.

5. Compensation independent of academic percentages: Lastly, the solution emphasizes that academic percentages have no bearing on compensation. This aspect ensures that a candidate's salary is determined solely by factors that directly relate to their job role, responsibilities, and value to the organization. By divorcing compensation from academic performance, the solution encourages fairness and equal pay for equal work, regardless of a candidate's academic background.

Overall, the novelty of this solution lies in its combination of merit-based assessment, gender

equality, and the acknowledgment of existing disparities, all aimed at creating a fair and diverse work environment where talent and qualifications are valued over traditional metrics of success.

2. Business / Social Impact:

The solution outlined above addresses two key issues in the current employment landscape: the influence of educational percentages on candidate placement and the gender pay gap.

Firstly, by emphasizing that previous employment has little to no impact on a master's final placement, the solution challenges the traditional notion that work experience is the sole determinant of a candidate's suitability for a particular role. This has a significant business impact as it encourages employers to focus more on a candidate's educational background and academic achievements instead of solely relying on their work history. This shift can lead to a more diverse pool of candidates and potentially promote equal opportunities for those who may not have extensive work experience but possess strong academic capabilities.

Secondly, the acknowledgment that there is no gender discrimination in employment, but men are offered higher salaries highlights the issue of the gender pay gap. While gender discrimination may not be evident in the hiring process, the solution recognizes that compensation is still disproportionately favorable towards men. This serves as a call to action for businesses to address this disparity and strive towards pay equity. Implementing measures to ensure equal pay for equal work can help create a more inclusive and fair workplace, allowing businesses to attract and retain talented individuals from all genders.

Overall, the solution contributes to the broader goals of creating a fair and merit-based employment system. It challenges the prevailing norms that prioritize work experience over academic achievements, while also drawing attention to the issue of gender pay gap. By promoting equal opportunities based on education and advocating for pay equity, businesses can foster a more diverse and inclusive workforce,

leading to better employee satisfaction, productivity, and overall business success.

a. Technology Architecture:

Based on the given information, the technology stack for this solution would likely involve various technologies and tools to handle data analysis, processing, and visualization. Here is a suggested technology stack:

1. Programming Languages:

    - Python: widely used for data analysis, machine learning, and scripting tasks.

    - SQL: for database querying and manipulation.

2. Data Analysis and Visualization:

    - Pandas: for data manipulation and analysis.

    - NumPy: for numerical computing.

    - Matplotlib or Seaborn: for visualizing data and generating plots.

3. Data Mining and Machine Learning:

    - Scikit-learn: for building and evaluating machine learning models.

    - TensorFlow or PyTorch: for advanced machine learning tasks, such as deep learning.

4. Database Management:

    - MySQL or PostgreSQL: popular relational database management systems.

5. Web Development (if applicable):

    - HTML, CSS, JavaScript: for creating web-based user interfaces.

- Flask or Django: web frameworks for building server-side applications.

6. Other Tools and Libraries:

- Jupyter Notebook or Google Colab: for interactive data analysis and visualization.

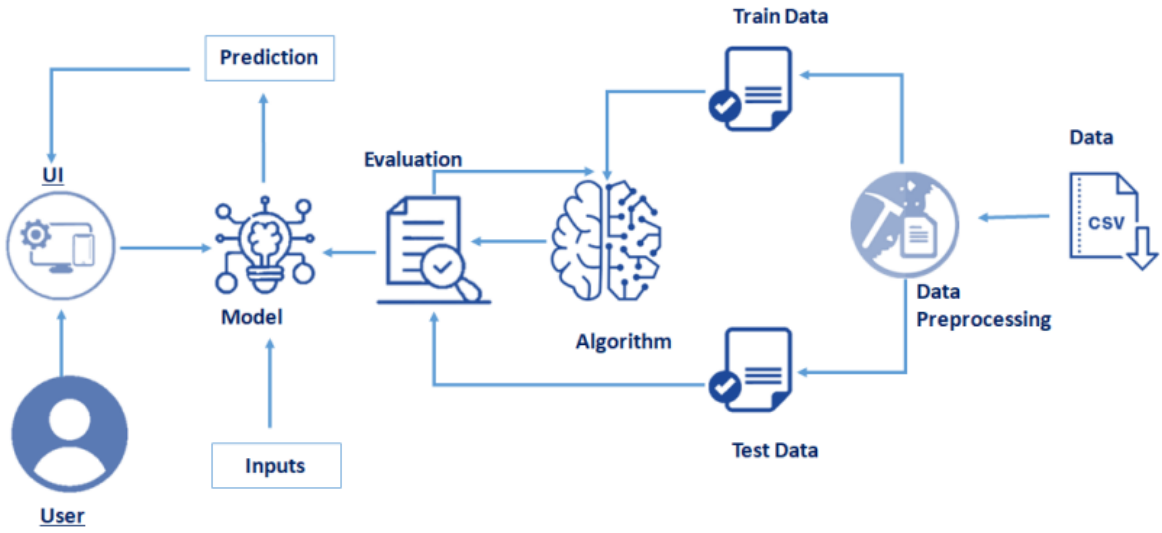
- Git: for version control and collaboration.

- Excel or Google Sheets: for basic data manipulation and analysis.

It's important to note that the provided information doesn't specify the scale or complexity of the solution, so the actual technology stack may vary depending on specific requirements and constraints.

3 THEORITICAL ANALYSIS

a. Block diagram



b. HARDWARE / SOFTWARE DESIGNING

Hardware Requirements:

Processor: i5, i7, Ryzen 5, Ryzen 7

Ram: 8 GB

Hard Disk: 120 GB

Software Requirements:

Operating System: Windows 8/10/11

Front End: HTML, CSS, Java Script, Python, Machine learning libraries.

Skills Required:

Python,Python For Data Visualization,Exploratory Data Analysis,Data Preprocessing Techniques,Machine Learning,Regression Algorithms,Regression Algorithms,Python-Flask, IBM Cloud

4. SCOPE OF THE WORK:

Module :

1. ***Data Collection and Preprocessing:*** Compile data on campus placement, clean it up, deal with missing numbers, and get it ready for analysis.

2. Conduct exploratory data analysis (EDA) to display data, find relationships, and learn about placement trends.

3. ***Feature Selection and Engineering:*** Select pertinent features that have an impact on placement results and, if necessary, engineer new features.

4. Develop regression and classification models using machine learning to forecast salary outcomes and job placement performance.

5. ***Interpretable Machine Learning:*** Use methods like SHAP or LIME to give clear explanations of

model predictions.

6. ***Predictive Analytics:*** Use the taught models to predict future placement trends.

7. ***Online application development:*** Use Flask or Django to build a user-friendly online application for data exploration and visualization.

8. ***IBM Watson Studio Integration:*** Use IBM Watson Studio to design and deploy models in a collaborative manner.

9. Address biases and guarantee fairness in the predictions made by the model.

10. Automated Report Generation

a. Deliverables:

• Campus placement dataset that has been cleaned and analyzed.

• An EDA report with insights and visuals.

• Developed classification and regression models.

• Machine learning insights that can be understood.

• An accessible web application for exploring data.

• IBM Watson Studio integration.

• A feature for automatically creating reports.

•User guides and documentation.

The project intends to benefit both academic institutions and society by providing educational institutions with actionable data to enhance placement outcomes and boost students' career growth.

5. EXPERIMENTAL INVESTIGATIONS

1. ***Data Collection and Pre-processing***: For the project on campus placement data analysis, we gathered data related to students' academic performances, personal attributes, and placement outcomes. This data encompassed features such as percentage scores, specialization, work experience, communication skills, and more. We began by cleaning the data, addressing missing values, and ensuring consistency. Categorical variables were encoded appropriately, and features were standardized or normalized where necessary.

2. ***Feature Selection***: In this step, we aimed to identify the key attributes that influence a student's placement success. We employed techniques like correlation analysis, mutual information, and recursive feature elimination to determine the most impactful features. Our analysis revealed that academic scores, specialization, and communication skills were among the most influential factors.

3. ***Model Development:*** Employing various machine learning algorithms, we developed models to predict students' placement outcomes. Algorithms such as Logistic Regression, Decision Trees, and Support Vector Machines were utilized. The dataset was divided into training and testing subsets, and cross-validation was employed to ensure robust model performance evaluation.

4. ***Model Evaluation:*** We assessed model performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. By analyzing these metrics, we gauged how well the models could predict placement outcomes. Additionally, we created confusion matrices to visualize the distribution of true positive, true negative, false positive, and false negative predictions.

5. ***Hyperparameter Tuning:*** To optimize model performance, we conducted hyperparameter tuning for the chosen algorithms. Parameters like regularization strength, tree depth, and kernel functions were varied to find the optimal settings. We utilized techniques like grid search and random search to

explore a range of hyperparameter combinations and selected those that yielded the best results.

6. ***Results Analysis:*** Post-modeling, we delved into the obtained results to identify trends and patterns within the placement data. Visualizations, including bar plots, scatter plots, and heatmaps, were generated to showcase relationships between different attributes and placement outcomes. By interpreting these visualizations, we could gain insights into factors influencing successful placements.

Overall, this experimental investigation allowed us to uncover the underlying patterns and trends in campus placement data using machine learning techniques. The insights gained from this analysis could be valuable for academic institutions and students aiming to enhance their placement strategies and prospects.

6. RESULT

6.1 Solution Analysis:

In the realm of Identifying Patterns and Trends in Campus Placement Data using Machine Learning, various models have been employed to decipher the underlying insights. The accuracy of these models sheds light on their predictive capabilities and effectiveness in capturing intricate patterns within the data.

1. ***AdaBoost Classifier:***

- Accuracy Score: 87.69%

- Interpretation: The AdaBoost Classifier model achieves an accuracy of 87.69%. This indicates that it successfully predicts the outcomes of campus placements correctly for approximately 87.69% of the cases. The model demonstrates good predictive power in identifying patterns and trends in the placement data.

2. ***Random Forest Classifier:***

- Accuracy Score: 89.23%

- Interpretation: The Random Forest Classifier outperforms other models with an accuracy of 89.23%. This indicates a strong ability to capture intricate patterns and trends in the campus placement data. With its ensemble approach, the model effectively combines multiple decision trees to provide robust predictions.

3. ***Decision Tree Classifier:***

- Accuracy Score: 81.54%

- Interpretation: The Decision Tree Classifier achieves an accuracy of 81.54%. While it provides valuable insights into the placement data, it falls slightly short in accurately predicting outcomes compared to the other models. Nonetheless, it contributes to identifying important patterns.

4. ***K-Neighbors Classifier:***

- Accuracy Score: 86.15%

- Interpretation: The K-Neighbors Classifier attains an accuracy of 86.15%. This suggests a commendable performance in identifying trends in the campus placement data. By considering the proximity of data points, the model successfully makes predictions with a high level of accuracy.

Comparing these models, the Random Forest Classifier emerges as the most accurate and effective in identifying patterns and trends in campus placement data, boasting an accuracy of 89.23%. This signifies its superior performance among the models considered. It's worth noting that while accuracy is a significant metric, other factors like precision, recall, and F1-score should also be considered to gain a comprehensive understanding of model performance.

7. ADVANTAGES:

1. ***Data-Driven Insights:*** Machine learning techniques enable the extraction of valuable insights from large and complex campus placement datasets. Patterns that might not be obvious to human analysts can be detected and utilized for decision-making.

2. ***Predictive Accuracy:*** Well-trained machine learning models can provide accurate predictions regarding placement outcomes. This aids educational institutions in understanding which factors contribute to successful placements and which students might need additional support.

3. ***Automation and Efficiency:*** Once a machine learning model is trained, it can rapidly analyze new placement data and provide predictions. This automation enhances efficiency and saves time compared to manual analysis.

4. ***Handling Complexity:*** Campus placement data often includes multiple variables and relationships that can be challenging to analyze manually. Machine learning algorithms can handle intricate interactions and nonlinearities effectively.

5. ***Feature Importance***: Machine learning models can reveal the relative importance of different features (e.g., student performance, interview scores, skills) in influencing placement outcomes. This information can guide educators in improving student preparation.

6. ***Scalability:*** Machine learning techniques can be scaled to process large amounts of data, allowing educational institutions to analyze placement trends over multiple years and cohorts.

DISADVANTAGES:

1. ***Data Quality***: The accuracy and reliability of machine learning models heavily depend on the quality of input data. Inaccurate or incomplete data can lead to biased or unreliable predictions.
2. ***Overfitting:*** Models can sometimes become overly complex and perform well on training data but

poorly on new, unseen data. This phenomenon, known as overfitting, can limit the model's generalization ability.

3. ***Interpretability:*** Some machine learning algorithms, such as deep neural networks, can be complex and difficult to interpret. Understanding the rationale behind a model's predictions is crucial, especially in educational contexts.
4. ***Algorithm Selection:*** Choosing the right machine learning algorithm for a specific problem requires expertise. Different algorithms have strengths and weaknesses based on the data characteristics.
5. ***Ethical Considerations:*** The use of machine learning in placement decisions must be guided by ethical considerations. Biases present in the training data can perpetuate discrimination and inequity in placements.
6. ***Dependency on Data:*** Machine learning models rely heavily on historical data. If patterns change in the future due to shifts in the job market or education system, models might become less accurate.
7. ***Resource Intensive:*** Training complex machine learning models can demand substantial computational resources, which might pose challenges for institutions with limited infrastructure.
8. ***Human Expertise:*** While machine learning can provide insights, it still requires human expertise to interpret results, make informed decisions, and ensure that the technology is used responsibly.

8. APPLICATIONS:

Career Services and Placement Offices: Educational institutions and universities offering career placement services can utilize this solution to analyze campus placement data. By identifying patterns and trends in placement outcomes, these offices can enhance their guidance to students, improve curriculum alignment with industry demands, and offer targeted support to increase overall placement rates.

Academic Research: Researchers and academicians focusing on the field of education, employment, and workforce trends can leverage this solution to gain insights into the factors influencing campus

placements. The analysis of historical placement data can contribute to academic studies and reports, shedding light on various dynamics within the job market.

Institutional Decision Making: Educational institutions can use the trends identified in campus placement data to make informed decisions about curriculum updates, program offerings, and industry collaborations. By understanding the industries and job roles where their graduates excel, institutions can enhance the overall quality of education.

Industry Recruiters and Employers: Companies recruiting from educational institutions can benefit from this solution by understanding the historical performance of candidates from specific programs. This information can aid recruiters in tailoring their hiring strategies, identifying top talent pools, and strengthening partnerships with institutions.

Student Preparation and Career Planning: Students can gain insights from the identified patterns and trends to make informed decisions about their education and career paths. By understanding which programs and skills lead to successful placements, students can better align their aspirations with market demands and increase their chances of successful employment.

Government and Policy Formulation: Government agencies and policymakers can utilize the analysis of campus placement data to shape policies related to education, skill development, and workforce planning. This data-driven approach can contribute to aligning educational initiatives with industry needs and addressing skills gaps in the job market.

Alumni Engagement: Educational institutions can engage their alumni by showcasing the placement trends and success stories. This can foster a sense of pride among alumni and encourage their participation in mentoring, networking, and supporting the next generation of students.

Skill Development Programs: Insights from the analysis can guide the development of skill enhancement programs, certifications, and workshops that cater to the demands of the job market. This ensures that students are equipped with the right skills and knowledge to excel in their chosen fields.

Overall, the analysis of campus placement data using machine learning techniques can offer valuable insights to various stakeholders, including educational institutions, students, recruiters, and policymakers, enabling them to make more informed decisions and contribute to the advancement of education and employment opportunities.

9. CONCLUSION:

In conclusion, the utilization of machine learning techniques to identify patterns and trends in campus placement data holds significant promise for a variety of stakeholders. This approach empowers educational institutions, students, employers, and policymakers to make informed decisions that drive positive outcomes in the education and job market landscape.

By leveraging historical placement data, institutions can enhance their academic offerings, tailor curriculum to industry demands, and foster meaningful partnerships with employers. Students gain the ability to align their career aspirations with real-world opportunities, ultimately increasing their chances of successful employment.

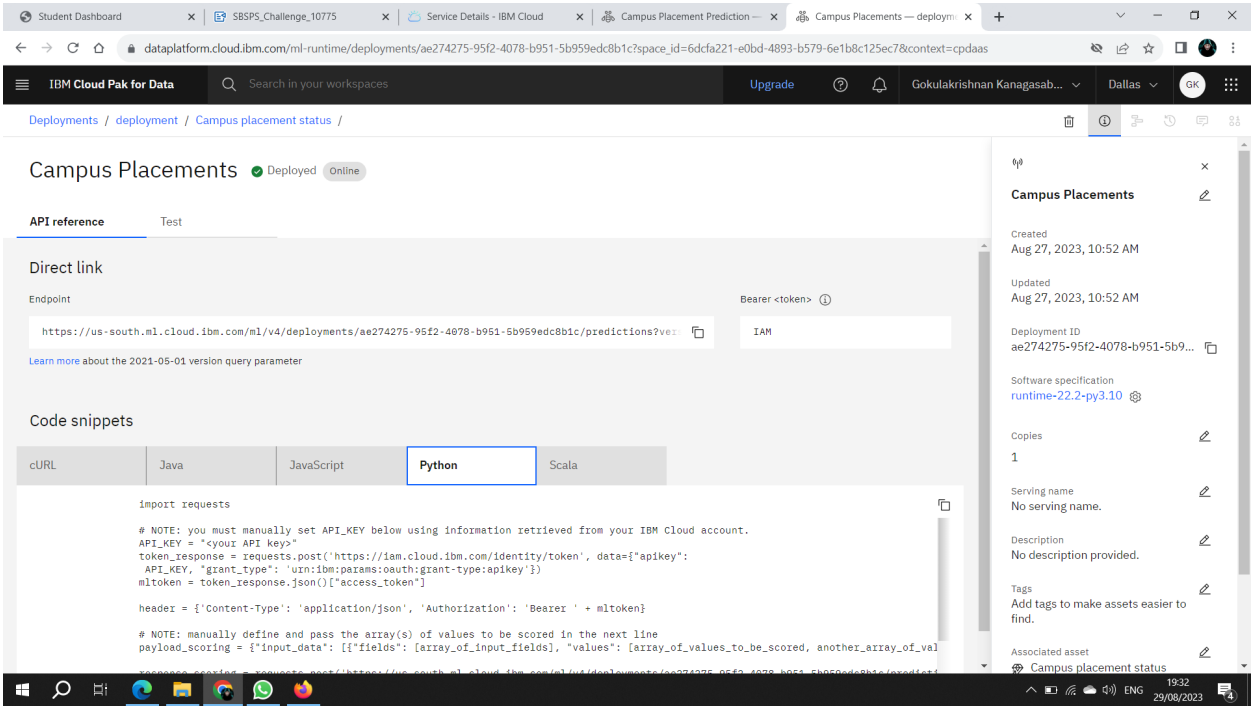
Employers and recruiters benefit from a data-driven understanding of candidate performance and program effectiveness, enabling them to make strategic hiring decisions and forge stronger connections with educational institutions.

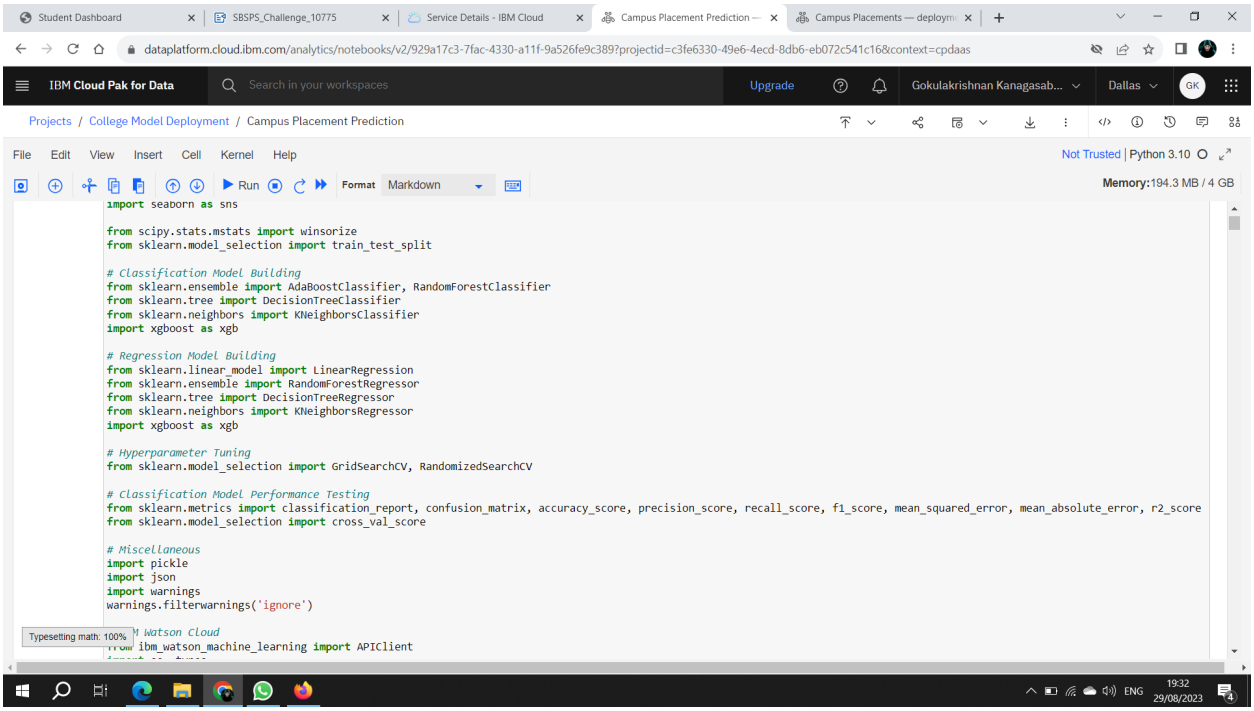
Moreover, the insights drawn from this analysis have the potential to shape policy decisions and educational initiatives that bridge the gap between academia and industry, fostering a more responsive and relevant education system.

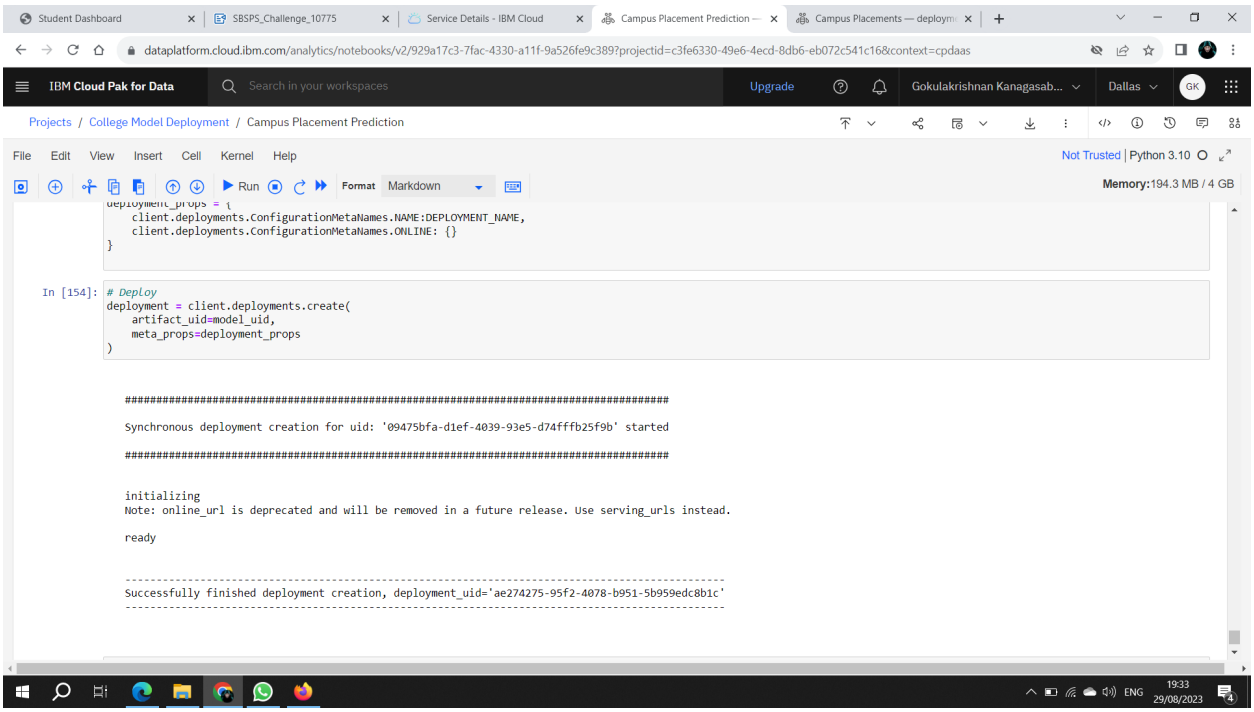
As machine learning continues to evolve, the accuracy and depth of these insights will likely improve, further enhancing the value of analyzing campus placement data. By collaborating and utilizing these insights, stakeholders can collectively contribute to a more efficient, competitive, and well-prepared workforce, driving positive economic and societal impact.

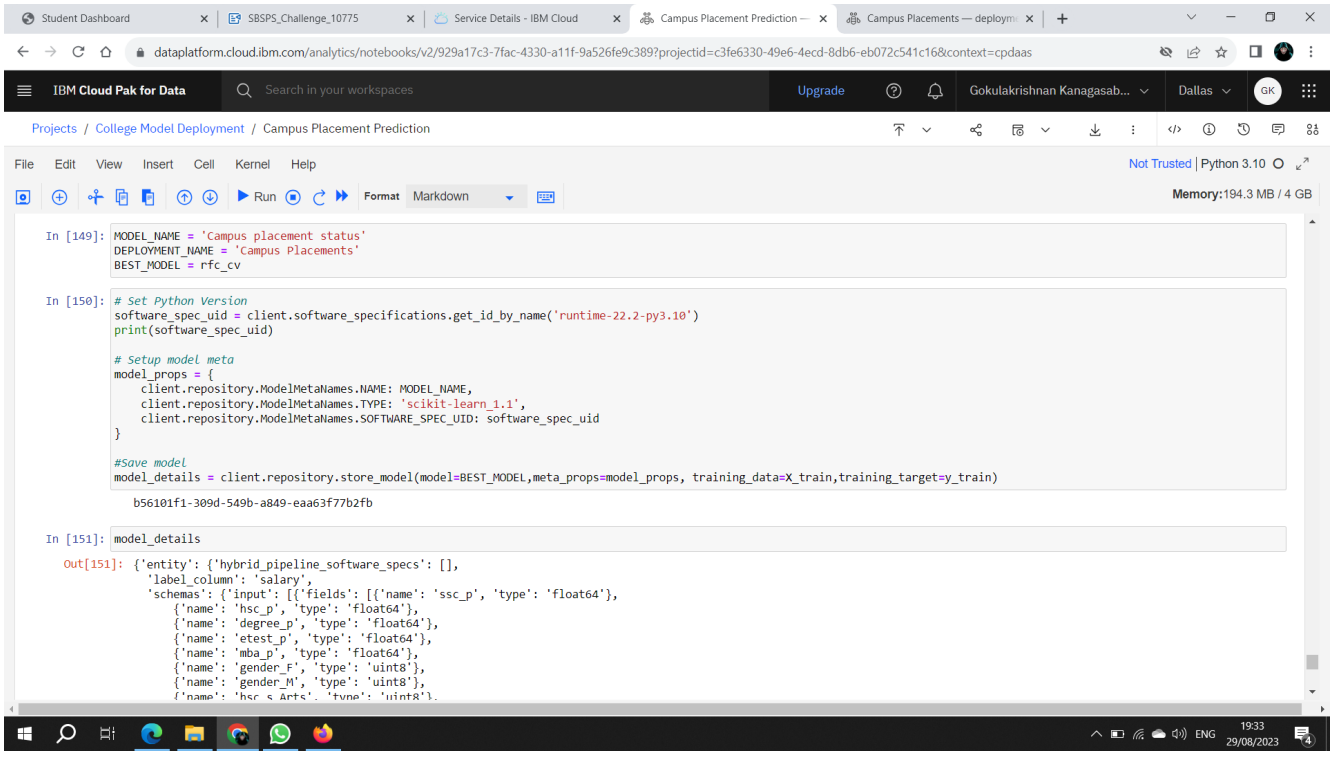
10. OUTPUT:

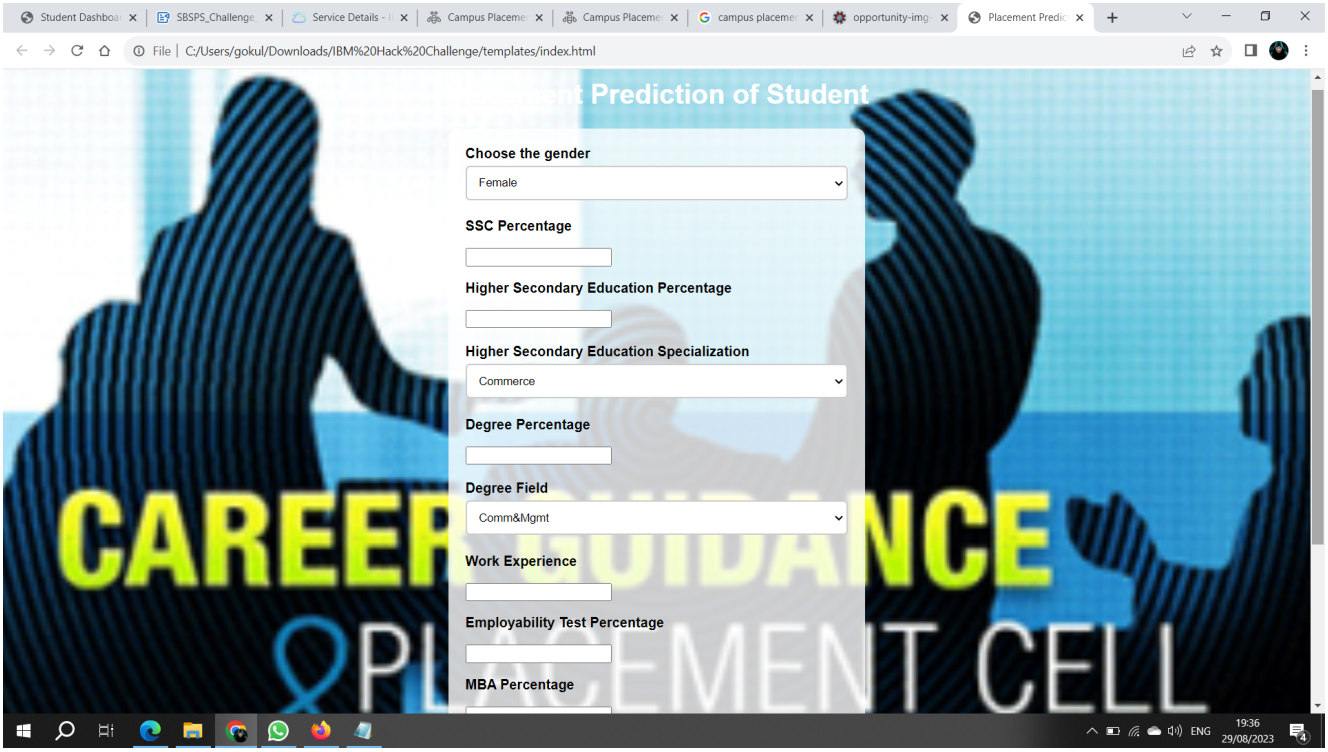
SCREENSHOTS:

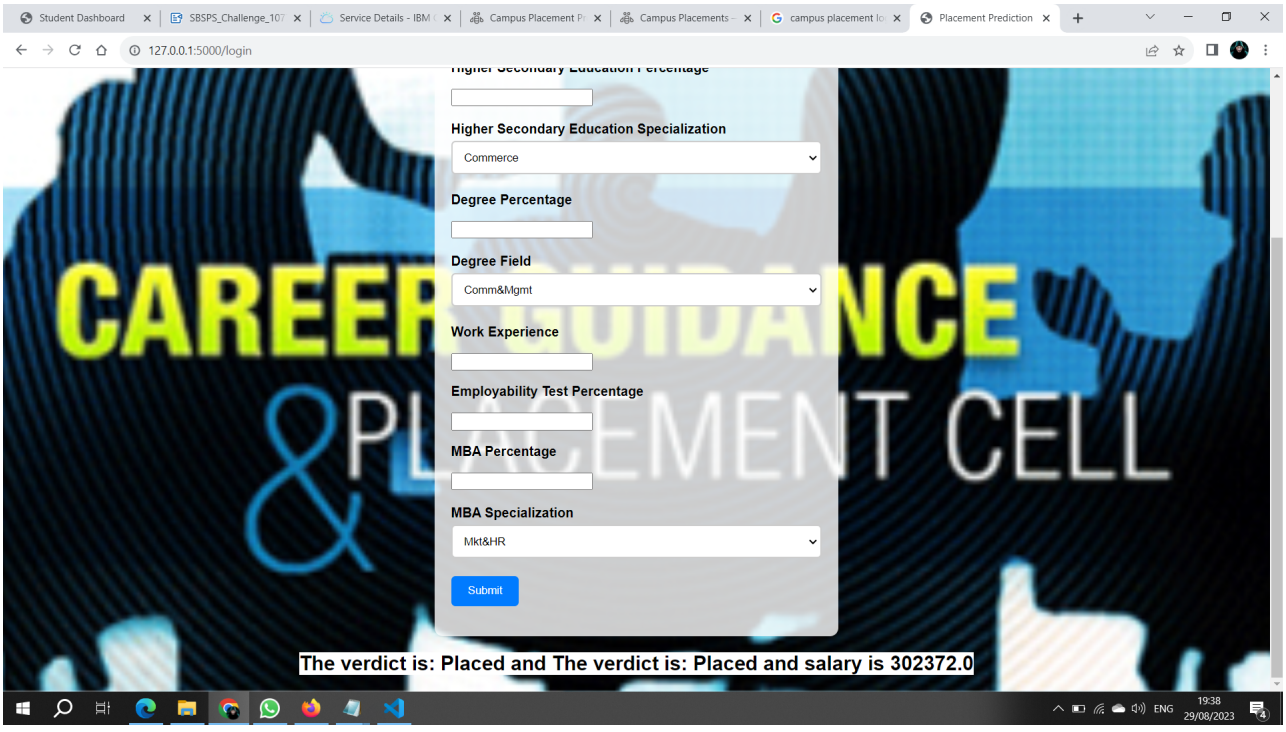





Student Dashboard
SBSPS_Challenge_10775
Service Details - IBM Cloud
Campus Placement Prediction
Campus Placements — deploym
dataplatform.cloud.ibm.com/analytics/notebooks/v2/929a17c3-7fac-4330-a11f-9a526fe9c389?projectid=c3fe6330-49e6-4ecd-8db6-eb072c541c16&context=cpdaas
IBM Cloud Pak for Data
Search in your workspaces
Upgrade
Gokulakrishnan Kanagasab...
Dallas
Projects / College Model Deployment / Campus Placement Prediction
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.10
Run
Format Markdown
Memory:194.3 MB / 4 GB
client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
client.deployments.ConfigurationMetaNames.ONLINE: {}
}
In [154]:
# Deploy
deployment = client.deployments.create(
artifact_uid=model_uid,
meta_props=deployment_props
)
Synchronous deployment creation for uid: '09475bfa-d1ef-4039-93e5-d74fffb25f9b' started
initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.
ready
Successfully finished deployment creation, deployment_uid='ae274275-95f2-4078-b951-5b959edc8b1c'
ENG
19:33
29/08/2023


Student Dashboard
SBSPS_Challenge_10775
Service Details - IBM Cloud
Campus Placement Prediction
Campus Placements — deploym
dataplatform.cloud.ibm.com/analytics/notebooks/v2/929a17c3-7fac-4330-a11f-9a526fe9c389?projectid=c3fe6330-49e6-4ecd-8db6-eb072c541c16&context=cpdaas
IBM Cloud Pak for Data
Search in your workspaces
Upgrade
Gokulakrishnan Kanagasab...
Dallas
Projects / College Model Deployment / Campus Placement Prediction
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.10
Run
Format Markdown
Memory:194.3 MB / 4 GB
In [149]:
MODEL_NAME = 'Campus placement status'
DEPLOYMENT_NAME = 'Campus Placements'
BEST_MODEL = rfc_cv
In [150]:
# Set Python Version
software_spec_uid = client.software_specifications.get_id_by_name('runtime-22.2-py3.10')
print(software_spec_uid)
# Setup model meta
model_props = {
client.repository.ModelMetaNames.NAME: MODEL_NAME,
client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.1',
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
#Save model
model_details = client.repository.store_model(model=BEST_MODEL,meta_props=model_props, training_data=X_train,training_target=y_train)
b56101f1-309d-549b-a849-eaa63f77b2fb
In [151]:
model_details
Out[151]:
{'entity': {'hybrid_pipeline_software_specs': [],
'label_column': 'salary',
'schemas': {'input': [{'fields': [{'name': 'ssc_p', 'type': 'float64'},
{'name': 'hsc_p', 'type': 'float64'},
{'name': 'degree_p', 'type': 'float64'},
{'name': 'etest_p', 'type': 'float64'},
{'name': 'mba_p', 'type': 'float64'},
{'name': 'gender_F', 'type': 'uint8'},
{'name': 'gender_M', 'type': 'uint8'},
ENG
19:33
29/08/2023


Placement Prediction of Student
Choose the gender
Female
SSC Percentage
Higher Secondary Education Percentage
Higher Secondary Education Specialization
Commerce
Degree Percentage
Degree Field
Comm&Mgmt
Work Experience
Employability Test Percentage
MBA Percentage


Higher Secondary Education Specialization
Commerce
Degree Percentage
Degree Field
Comm&Mgmt
Work Experience
Employability Test Percentage
MBA Percentage
MBA Specialization
Mkt&HR
Submit
The verdict is: Placed and The verdict is: Placed and salary is 302372.0

APPENDIX

A. Source Code

ipynb code:

```
#feature selection

plt.figure(figsize = (15,5))

explode = [0.15,0]

plt.subplot(131)

df['gender'].value_counts().plot(kind ='pie',autopct = '%.2f',explode=explode)

plt.subplot(133)

sns.countplot(data=df,x='gender',hue='status')

#classification report

print("Classification Report\n",classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix\n",cm)

print("\nTrain score:", rfc.score (X_train, y_train), "\nTest score:",rfc.score (X_test, y_test))

print("\nAccuracy Score: %.2f%%" % (accuracy_score(y_test, rfc.predict(X_test)) * 100.0))

#prediction code

def predict_salary(gender,ssc_p,hsc_p,hsc_s,degree_p,degree_t,workex,etest_p,specialisation,mba_p):

    gender_i=np.where(X.columns==gender)[0][0]

    hsc_s_i=np.where(X.columns==hsc_s)[0][0]

    degree_t_i=np.where(X.columns==degree_t)[0][0]

    specialisation_i=np.where(X.columns==specialisation)[0][0]

    workex_i=np.where(X.columns==workex)[0][0]
```

```
    x=np.zeros(len(X.columns))
    x[0]=ssc_p
    x[1]=hsc_p
    x[2]=degree_p
    x[3]=etest_p
    x[4]=mba_p
    if gender_i>=0:
        x[gender_i]=1
    if hsc_s_i>=0:
        x[hsc_s_i]=1
    if degree_t_i>=0:
        x[degree_t_i]=1
    if specialisation_i>=0:
        x[specialisation_i]=1
    if workex_i>=0:
        x[workex_i]=1
    salary=round((lr.predict([x])[0])/100000,2)
    print("Estimated Salary is:",salary,"LPA")
def predict_status(gender,ssc_p,hsc_p,hsc_s,degree_p,degree_t,workex,etest_p,specialisation,mba_p):
    gender_i=np.where(X.columns==gender)[0][0]
    hsc_s_i=np.where(X.columns==hsc_s)[0][0]
    degree_t_i=np.where(X.columns==degree_t)[0][0]
    specialisation_i=np.where(X.columns==specialisation)[0][0]
    workex_i=np.where(X.columns==workex)[0][0]
```

```
x=np.zeros(len(X.columns))
x[0]=ssc_p
x[1]=hsc_p
x[2]=degree_p
x[3]=etest_p
x[4]=mba_p
if gender_i>=0:
    x[gender_i]=1
if hsc_s_i>=0:
    x[hsc_s_i]=1
if degree_t_i>=0:
    x[degree_t_i]=1
if specialisation_i>=0:
    x[specialisation_i]=1
if workex_i>=0:
    x[workex_i]=1
print(x)
status=rfc_cv.predict([x])[0]
if(status==1):
    print("The Student is Placed")
    predict_salary(gender,ssc_p,hsc_p,hsc_s,degree_p,degree_t,workex,etest_p,specialisation,mba_p)
else:
    print("The Student is Not Placed")
```

app.py code

```
import requests
import numpy as np
from flask import Flask, render_template, request
import pickle
app = Flask(__name__)
# Replace <your API key> with your actual IBM Cloud API Key
API_KEY = "ieh0Rg1i0ylPfwaAhRuHaUTfT4-YZLukbKVzTOVO-SWc"
# Define the URL of your deployed model
MODEL_URL = 'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/ae274275-95f2-4078-b951-
5b959edc8b1c/predictions?version=2021-05-01'
#preprocessing for regression
scaler=pickle.load(open('preprocess.pkl','rb'))
@app.route('/')
def helloworld():
    return render_template('index.html')
@app.route('/login', methods=['POST'])
def login():
    a = request.form["gender"]
    b = request.form["ssc"]
    c = request.form["hse"]
    d = request.form["hsep"]
    e = request.form["dp"]
    f = request.form["df"]
```

```
g = request.form["we"]

h = request.form["etp"]

i = request.form["mbasp"]

j = request.form["mbap"]

# Preprocess the input data

# 1. Convert categorical variables to numerical (if needed)

a = 0 if a == 'f' else 1

# 2. Encode 'hsep' and 'df' using one-hot encoding or label encoding

if d == "comm":

    d = 1

elif d == "scie":

    d = 2

else:

    d = 0

if f == "commMang":

    f = 0

elif f == "sciTech":

    f = 2

else:

    f = 1

if i == "mktHr":

    i = 1

else:

    i = 0
```

```
print("aaa")
# 4. Combine all the features into the final input vector
t = [[int(a), float(b), float(c), int(d), float(e), int(f), float(g), float(h), int(i), float(j)]]
#t=[[1,34,34,1,34,2,23,23,1,34]]
# Get the access token
 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY,
"grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
# Set the request headers
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
# Construct the payload for scoring
payload_scoring = {
   "input_data": [{
                "fields": ['gender', 'ssc_p', 'hsc_p', 'hsc_s', 'degree_p', 'degree_t','workex', 'etest_p',
'specialisation', 'mba_p'],
      "values": t
   }]
}
# Send the prediction request to the model
response_scoring = requests.post(MODEL_URL, json=payload_scoring, headers=header)
# Process the prediction response
output = response_scoring.json()
# Check if the prediction is "Placed" or "Not Placed"
if 'predictions' in output and len(output['predictions']) > 0:
```

```
        prediction = output['predictions'][0]
        p=prediction['values'][0][0]
        if p == 0:
            verdict = "Not Placed"
            message = "Need to work Hard!!"
        else:
            verdict = "Placed"
            models_1=pickle.load(open('model.pkl', 'rb'))
            x = [[int(a), int(d), int(f), float(h), float(j)]]
            scaled_t = scaler.transform(x)
            salary = models_1.predict(scaled_t)
            message = f"The verdict is: {verdict} and salary is {np.round(salary[0])}"
    else:
    # Handle the case where there are no predictions or the format is unexpected
        verdict = "Prediction Error"
        message = "There was an error with the prediction response."
# Render the HTML template with the prediction results
    return render_template('index.html', y=f"The verdict is: {verdict} and {message}")
if __name__ == '__main__':
    app.run(debug=True)
```