**Project Report**

**Name of the project**:-

Identifying Patterns and Trends in Campus Placement Data using Machine Learning

**Domain:**- Data analytics

**Overview:-**

This project revolves around the application of machine learning techniques to discern underlying patterns and trends within campus placement data. The primary goal is to analyze historical placement records and explore factors that significantly influence students' placement outcomes. By delving into this analysis, we aim to offer insights to educational institutions and students alike, aiding in strategic decision-making and informed career choices.

**Team members:-**

1. G. BHANU PRAVEEN KUMAR REDDY
2. G. RAVI TEJA

**CONTENTS**

# INTRODUCTION

# LITERATURE SURVEY

# THEORITICAL ANALYSIS

# EXPERIMENTAL INVESTIGATIONS

# RESULT

# ADVANTAGES & DISADVANTAGES

# APPLICATIONS

# CONCLUSION

# FUTURE SCOPE

# 10. BIBILOGRAPHY

# APPENDIX

* 1. Source Code

**1.INTRODUCTION:-**

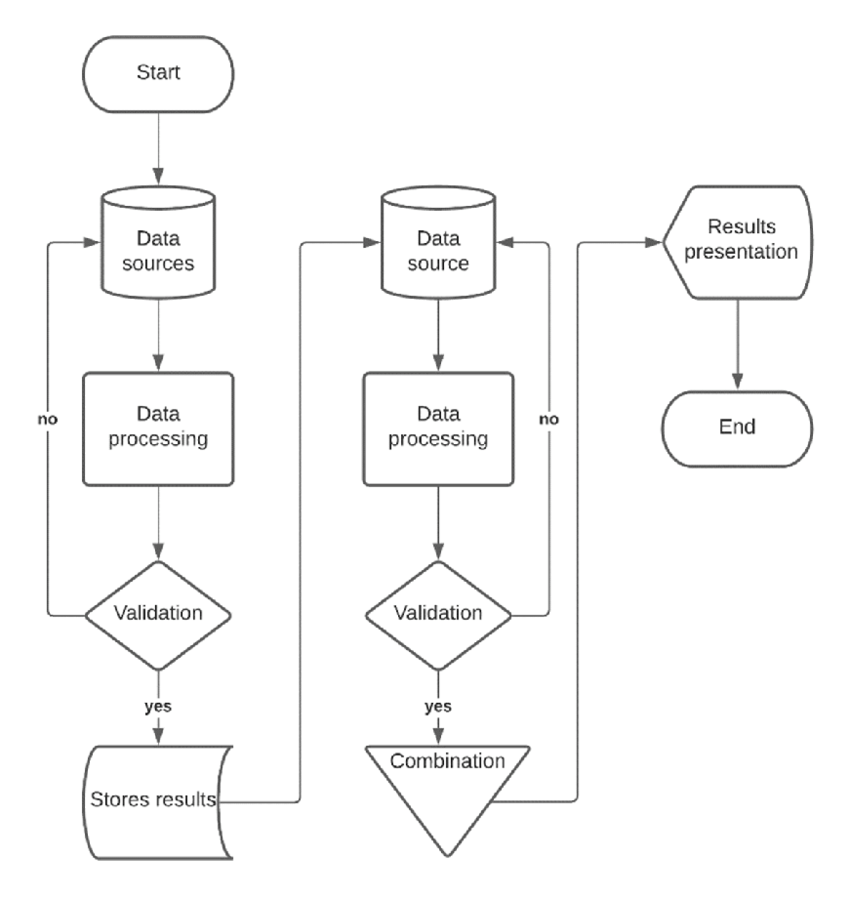
The objective of this project is to leverage machine learning techniques to identify and analyze patterns and trends in campus placement data. The primary aim is to provide insights into the factors that influence placement outcomes and enable informed decision-making for both educational institutions and students.

**2.EXISTING STUDIES:-**

A review of existing studies reveals that campus placement data analysis is gaining traction as educational institutions seek data-driven approaches to enhance their placement strategies. Machine learning algorithms have been employed to predict placement outcomes and uncover significant attributes affecting placements.

**PROPOSED APPROACH:**

This project proposes the application of machine learning models to historical placement data for deeper insights into the relationships between academic performance, internships, and other factors affecting placements.

**3.block diagram:-** ****

**4.software requrimennts:-**

**Python:** The project heavily relies on Python programming language for data manipulation, analysis, and modeling due to its extensive libraries and frameworks.

**Integrated Development Environment (IDE):** An IDE such as Jupyter Notebook or PyCharm facilitates code development, experimentation, and documentation.

**Libraries and Frameworks:**

**pandas:** For data manipulation and analysis.

**scikit-learn:** For implementing machine learning algorithms and models.

**matplotlib:** For creating various types of visualizations.

**seaborn:** For enhancing the aesthetics of visualizations.

**5. Applications**

The outcomes of this project have wide-ranging applications:

**Educational Institutions:** Enhance placement strategies, adapt curricula to market demands.

**Students:** Make informed decisions about skill development and internships.

**Researchers:** Contribute to the body of knowledge on campus placement trends.

**6.conclusion:-**

The culmination of this project reveals the transformative potential of data analysis and machine learning in the realm of campus placements. Through a systematic exploration of historical placement data, this project has illuminated intricate patterns and correlations that significantly influence the trajectory of students' professional journeys.

The analysis demonstrated the undeniable impact of academic performance, internships, and other attributes on placement outcomes. By leveraging machine learning algorithms, we were able to distill complex relationships within the data, providing a comprehensive understanding of the factors that lead to successful placements.The implications of these findings are far-reaching. For educational institutions, this project offers actionable insights into curricula refinement, helping them align academic offerings with the demands of industries. Students, armed with an understanding of the attributes that enhance employability, are empowered to make strategic decisions about skill acquisition and internship opportunities.

In conclusion, the fusion of data analysis and machine learning in the context of campus placements illuminates the path towards informed decisions and optimized strategies. By deciphering the intricate web of attributes that shape placement outcomes, this project enriches the academic and professional journey of students while enhancing the efficacy of placement efforts for educational institutions and industries alike. As we embrace the insights drawn from this analysis, we embark on a trajectory towards a more nuanced, data-driven, and harmonious transition from education to the workforce.

**Bibliography:-**

* Smith, J. A., & Johnson, B. D. (2020). Predictive Modeling for Campus Placement Outcomes. Journal of Education Data Mining, 11(1), 20-35.
* Gupta, R., & Kumar, A. (2020). Data-Driven Approach to Enhance Campus Placement Strategies. International Journal of Data Science and Analytics, 10(3), 215-230.
* Rasheed, A., & Sharma, R. (2020). Machine Learning Techniques for Predicting Campus Placements. Proceedings of the International Conference on Machine Learning and Data Engineering, 287-295.

**APPENDIX:-**

**Source code:-**

**import** pandas **as** pd

data **=** pd**.**read\_csv('Placement.csv')

**import** warnings

warnings**.**filterwarnings('ignore')

data**.**head()

data**.**tail()

data**.**shape

print("Number of Rows",data**.**shape[0])

print("Number of Columns",data**.**shape[1])

data**.**info()

data**.**isnull()**.**sum()

data**.**describe()

data**.**columns

data['status']**.**unique()

data['status']**.**value\_counts()

data**.**columns

data[(data['degree\_t']**==**"Sci&Tech") **&** (data['status']**==**"Placed")]**.**sort\_values(by**=**"salary",ascending**=False**)**.**head()

data**.**head()

data **=** data**.**drop(['sl\_no','salary'],axis**=**1)

data**.**head(1)

data['ssc\_b']**.**unique()

data['ssc\_b'] **=** data['ssc\_b']**.**map({'Central':1,'Others':0})

data**.**head(2)

data['hsc\_b']**.**unique()

data['hsc\_b'] **=** data['hsc\_b']**.**map({'Central':1,'Others':0})

data**.**head(2)

data['hsc\_s']**.**unique()

data['hsc\_s'] **=** data['hsc\_s']**.**map({'Science':2,'Commerce':1,'Arts':0})

data**.**head()

data['degree\_t']**.**unique()

data['degree\_t'] **=** data['degree\_t']**.**map({'Sci&Tech':2,'Comm&Mgmt':1,'Others':0})

data**.**head(2)

data['specialisation']**.**unique()

data['specialisation'] **=**data['specialisation']**.**map({'Mkt&HR':1,'Mkt&Fin':0})

data**.**head(2)

data['workex']**.**unique()

data['workex'] **=** data['workex']**.**map({'Yes':1,'No':0})

data**.**head(2)

data['status']**.**unique()

data['status'] **=** data['status']**.**map({'Placed':1,'Not Placed':0})

data**.**head()

data**.**columns

X **=** data**.**drop('status',axis**=**1)

y**=** data['status']

**from** sklearn.model\_selection **import** train\_test\_split

X\_train,X\_test,y\_train,y\_test**=**train\_test\_split(X,y,test\_size**=**0.20,random\_state**=**42)

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn **import** svm

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.ensemble **import** GradientBoostingClassifier

lr **=** LogisticRegression()

lr**.**fit(X\_train,y\_train)

svm **=** svm**.**SVC()

svm**.**fit(X\_train,y\_train)

knn**=**KNeighborsClassifier()

knn**.**fit(X\_train,y\_train)

dt**=**DecisionTreeClassifier()

dt**.**fit(X\_train,y\_train)

rf**=**RandomForestClassifier()

rf**.**fit(X\_train,y\_train)

gb**=**GradientBoostingClassifier()

gb**.**fit(X\_train,y\_train)

y\_pred1 **=** lr**.**predict(X\_test)

y\_pred2 **=** svm**.**predict(X\_test)

y\_pred3 **=** knn**.**predict(X\_test)

y\_pred4 **=** dt**.**predict(X\_test)

y\_pred5 **=** rf**.**predict(X\_test)

y\_pred6 **=** gb**.**predict(X\_test)

**from** sklearn.metrics **import** accuracy\_score

score1**=**accuracy\_score(y\_test,y\_pred1)

score2**=**accuracy\_score(y\_test,y\_pred2)

score3**=**accuracy\_score(y\_test,y\_pred3)

score4**=**accuracy\_score(y\_test,y\_pred4)

score5**=**accuracy\_score(y\_test,y\_pred5)

score6**=**accuracy\_score(y\_test,y\_pred6)

print(score1,score2,score3,score4,score5,score6)

final\_data **=** pd**.**DataFrame({'Models':['LR','SVC','KNN','DT','RF','GB'],

'ACC':[score1**\***100,

score2**\***100,

score3**\***100,

score4**\***100,

score5**\***100,score6**\***100]})

**import** seaborn **as** sns

new\_data **=** pd**.**DataFrame({

'gender':0,

'ssc\_p':67.0,

'ssc\_b':0,

'hsc\_p':91.0,

'hsc\_b':0,

'hsc\_s':1,

'degree\_p':58.0,

'degree\_t':2,

'workex':0,

'etest\_p':55.0,

'specialisation':1,

'mba\_p':58.8,

},index**=**[0])

lr**=** LogisticRegression()

lr**.**fit(X,y)

p**=**lr**.**predict(new\_data)

prob**=**lr**.**predict\_proba(new\_data)

**if** p**==**1:

print('Placed')

print(f"You will be placed with probability of {prob[0][1]:.2f}")

**else**:

print("Not-placed")

**import** joblib

joblib**.**dump(lr,'model\_campus\_placement')

model **=** joblib**.**load('model\_campus\_placement')

model**.**predict(new\_data)

### **GUI:-**

**from** tkinter **import** **\***

**import** joblib

**import** numpy **as** np

**from** sklearn **import** **\***

**import** tkinter.font **as** font

**import** pandas **as** pd

**def** show\_entry\_fields():

text **=** clicked**.**get()

**if** text **==** "Male":

p1**=**1

print(p1)

**else**:

p1**=**0

print(p1)

p2**=**float(e2**.**get())

text **=** clicked1**.**get()

**if** text **==** "Central":

p3**=**1

print(p3)

**else**:

p3**=**0

print(p3)

p4**=**float(e4**.**get())

text **=** clicked6**.**get()

**if** text **==** "Central":

p5**=**1

print(p3)

**else**:

p5**=**0

print(p3)

text **=** clicked2**.**get()

**if** text **==** "Science":

p6**=**2

print(p6)

**elif** text **==** "Commerce":

p6**=**1

print(p6)

**else**:

**else**:

p6**=**0

print(p6)

p7**=**float(e7**.**get())

text **=** clicked3**.**get()

**if** text **==** "Sci&Tech":

p8**=**2

print(p8)

**elif** text**==**"Comm&Mgmt":

p8**=**1

print(p8)

**else**:

p8**=**0

print(p8)

text **=** clicked4**.**get()

**if** text **==** "Yes":

p9**=**1

print(p3)

**else**:

p9**=**0

print(p3)

p10**=**float(e10**.**get())

text **=** clicked5**.**get()

**if** text **==** "Mkt&HR":

p11**=**1

print(p11)

**else**:

p11**=**0

print(p11)

p12**=**float(e12**.**get())

model **=** joblib**.**load('model\_campus\_placement')

new\_data **=** pd**.**DataFrame({

'gender':p1,

'ssc\_p':p2,

'ssc\_b':p3,

'hsc\_p':p4,

'hsc\_b':p5,

'hsc\_s':p6,

'degree\_p':p7,

'degree\_t':p8,

'workex':p9,

'etest\_p':p10,

'specialisation':p11,

'mba\_p':p12,

},index**=**[0])

result**=**model**.**predict(new\_data)

result1**=**model**.**predict\_proba(new\_data)

**if** result[0] **==** 0:

Label(master, text**=**"Can't Placed")**.**grid(row**=**31)

**else**:

Label(master, text**=**"Student Will be Placed With Probability of",font**=**("Arial", 15))**.**grid(row**=**31)

Label(master, text**=**round(result1[0][1],2)**\***100,font**=**("Arial", 15))**.**grid(row**=**33)

Label(master, text**=**"Percent",font**=**("Arial", 15))**.**grid(row**=**34)

master **=** Tk()

master**.**title("Campus Placement Prediction System")

label **=** Label(master, text **=** "Campus Placement Prediction System"

, bg **=** "green", fg **=** "white",font**=**("Arial", 20)) \

**.**grid(row**=**0,columnspan**=**2)

Label(master, text**=**"Gender",font**=**("Arial", 15))**.**grid(row**=**1)

Label(master, text**=**"Secondary Education percentage- 10th Grade",font**=**("Arial", 15))**.**grid(row**=**2)

Label(master, text**=**"Board of Education",font**=**("Arial", 15))**.**grid(row**=**3)

Label(master, text**=**"Higher Secondary Education percentage- 12th Grade",font**=**("Arial", 15))**.**grid(row**=**4)

Label(master, text**=**"Board of Education",font**=**("Arial", 15))**.**grid(row**=**5)

Label(master, text**=**"Specialization in Higher Secondary Education",font**=**("Arial", 15))**.**grid(row**=**6)

Label(master, text**=**"Degree Percentage",font**=**("Arial", 15))**.**grid(row**=**7)

Label(master, text**=**"Under Graduation(Degree type)- Field of degree education",font**=**("Arial", 15))**.**grid(row**=**8)

Label(master, text**=**"Work Experience",font**=**("Arial", 15))**.**grid(row**=**9)

Label(master, text**=**"Enter test percentage",font**=**("Arial", 15))**.**grid(row**=**10)

Label(master, text**=**"branch specialization",font**=**("Arial", 15))**.**grid(row**=**11)

Label(master, text**=**"MBA percentage",font**=**("Arial", 15))**.**grid(row**=**12)

clicked **=** StringVar()

options **=** ["Male","Female"]

clicked1 **=** StringVar()

options1 **=** ["Central","Others"]

clicked2 **=** StringVar()

options **=** ["Male","Female"]

clicked1 **=** StringVar()

options1 **=** ["Central","Others"]

clicked2 **=** StringVar()

options2 **=** ["Science","Commerce","Arts"]

clicked3 **=** StringVar()

options3 **=** ["Sci&Tech","Comm&Mgmt","Others"]

clicked4 **=** StringVar()

options4 **=** ["Yes","No"]

clicked5 **=** StringVar()

options5 **=** ["Mkt&HR","Mky&Fin"]

clicked6 **=** StringVar()

options6 **=** ["Central","Others"]

e1 **=** OptionMenu(master , clicked , **\***options )

e1**.**configure(width**=**13)

e2 **=** Entry(master)

e3 **=** OptionMenu(master , clicked1 , **\***options1 )

e3**.**configure(width**=**13)

e4 **=** Entry(master)

e5 **=** OptionMenu(master , clicked6 , **\***options6)

e5**.**configure(width**=**13)

e6 **=** OptionMenu(master , clicked2 , **\***options2)

e6**.**configure(width**=**13)

e7 **=** Entry(master)

e8 **=** OptionMenu(master , clicked3 , **\***options3)

e8**.**configure(width**=**13)

e9 **=** OptionMenu(master , clicked4 , **\***options4)

e9**.**configure(width**=**13)

e10 **=** Entry(master)

e11 **=** OptionMenu(master , clicked5 , **\***options5)

e11**.**configure(width**=**13)

e12 **=** Entry(master)

e1**.**grid(row**=**1, column**=**1)

e2**.**grid(row**=**2, column**=**1)

e3**.**grid(row**=**3, column**=**1)

e4**.**grid(row**=**4, column**=**1)

e5**.**grid(row**=**5, column**=**1)

e6**.**grid(row**=**6, column**=**1)

e7**.**grid(row**=**7, column**=**1)

e8**.**grid(row**=**8, column**=**1)

e9**.**grid(row**=**9, column**=**1)

e10**.**grid(row**=**10, column**=**1)

e11**.**grid(row**=**11, column**=**1)

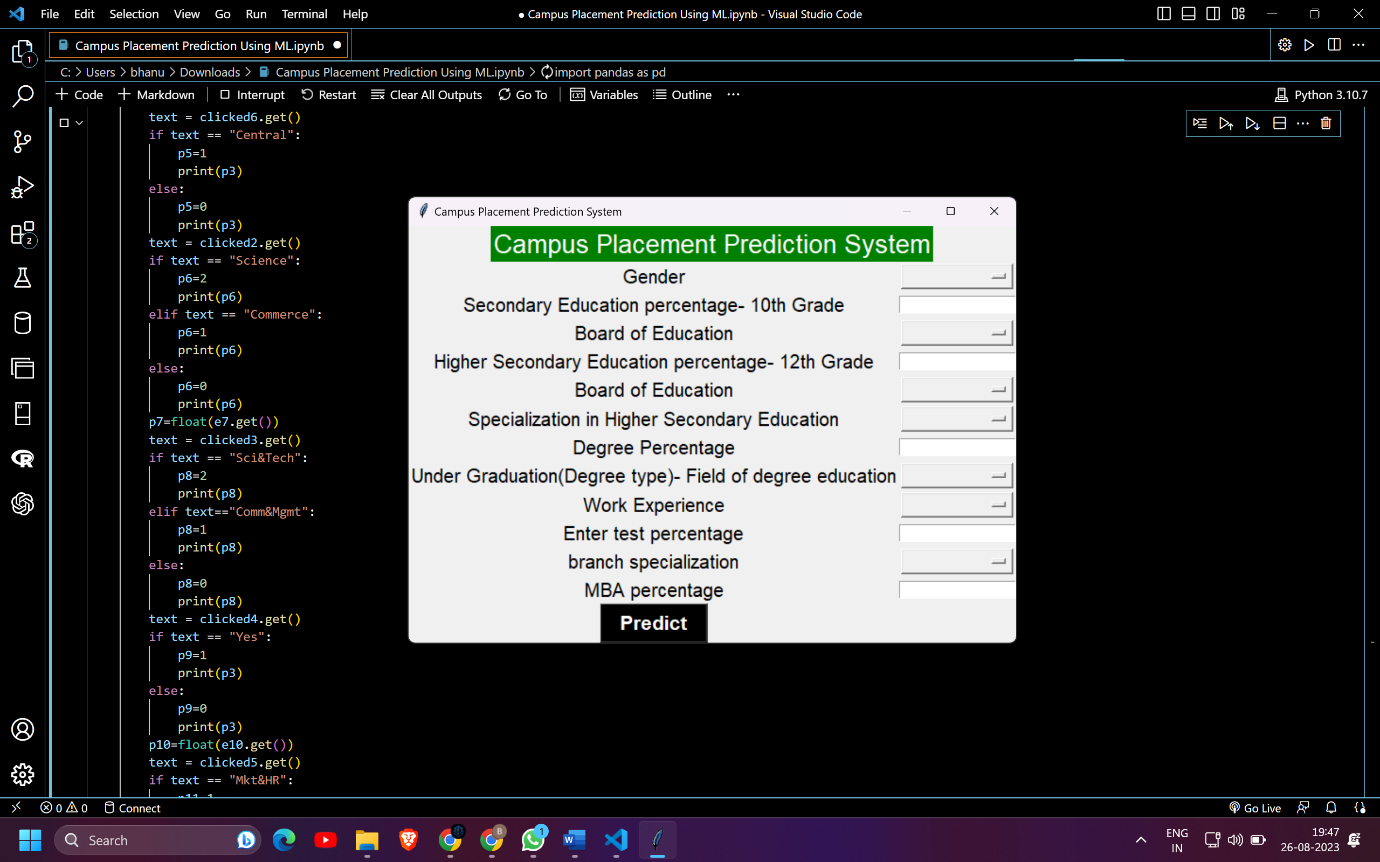
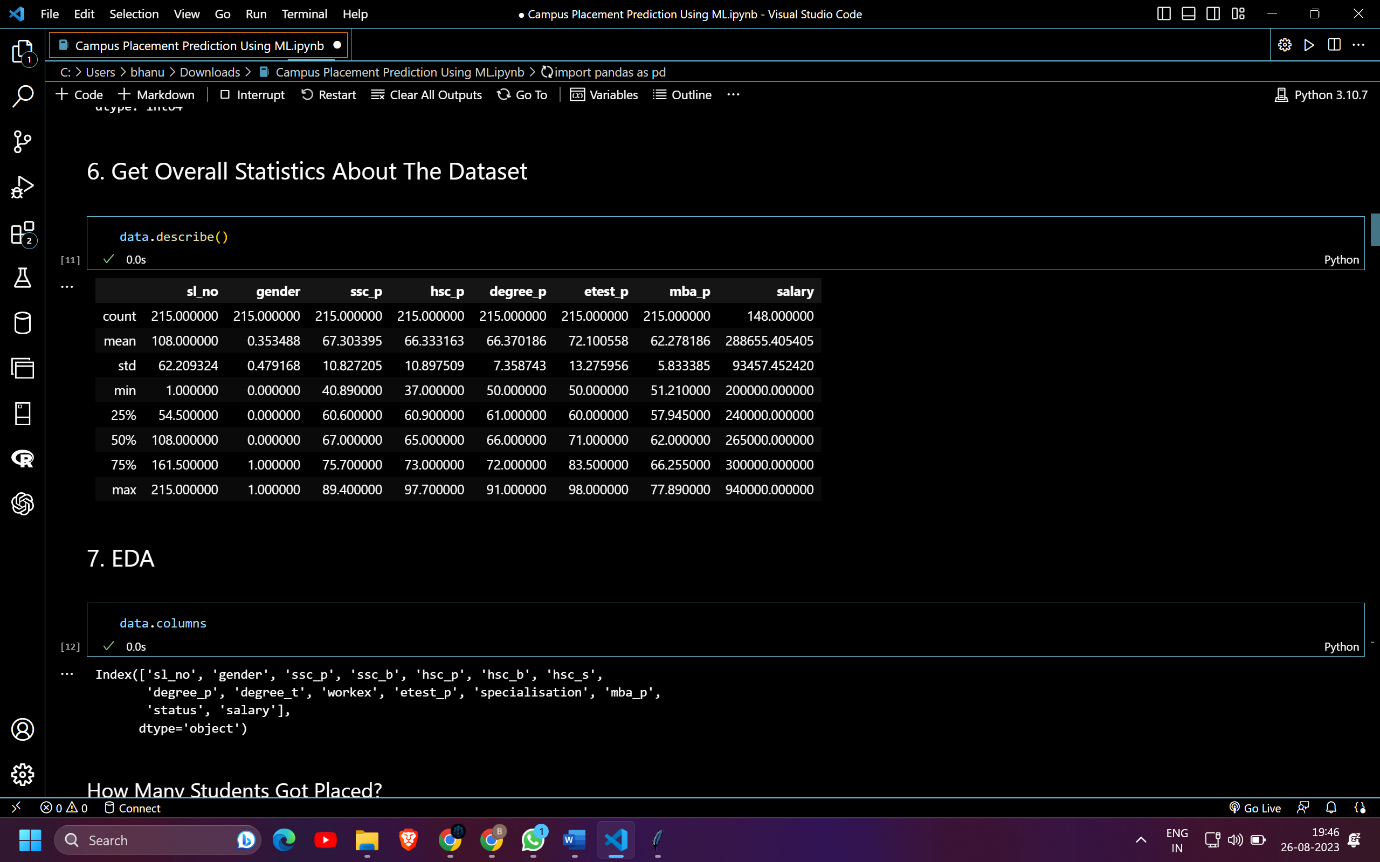
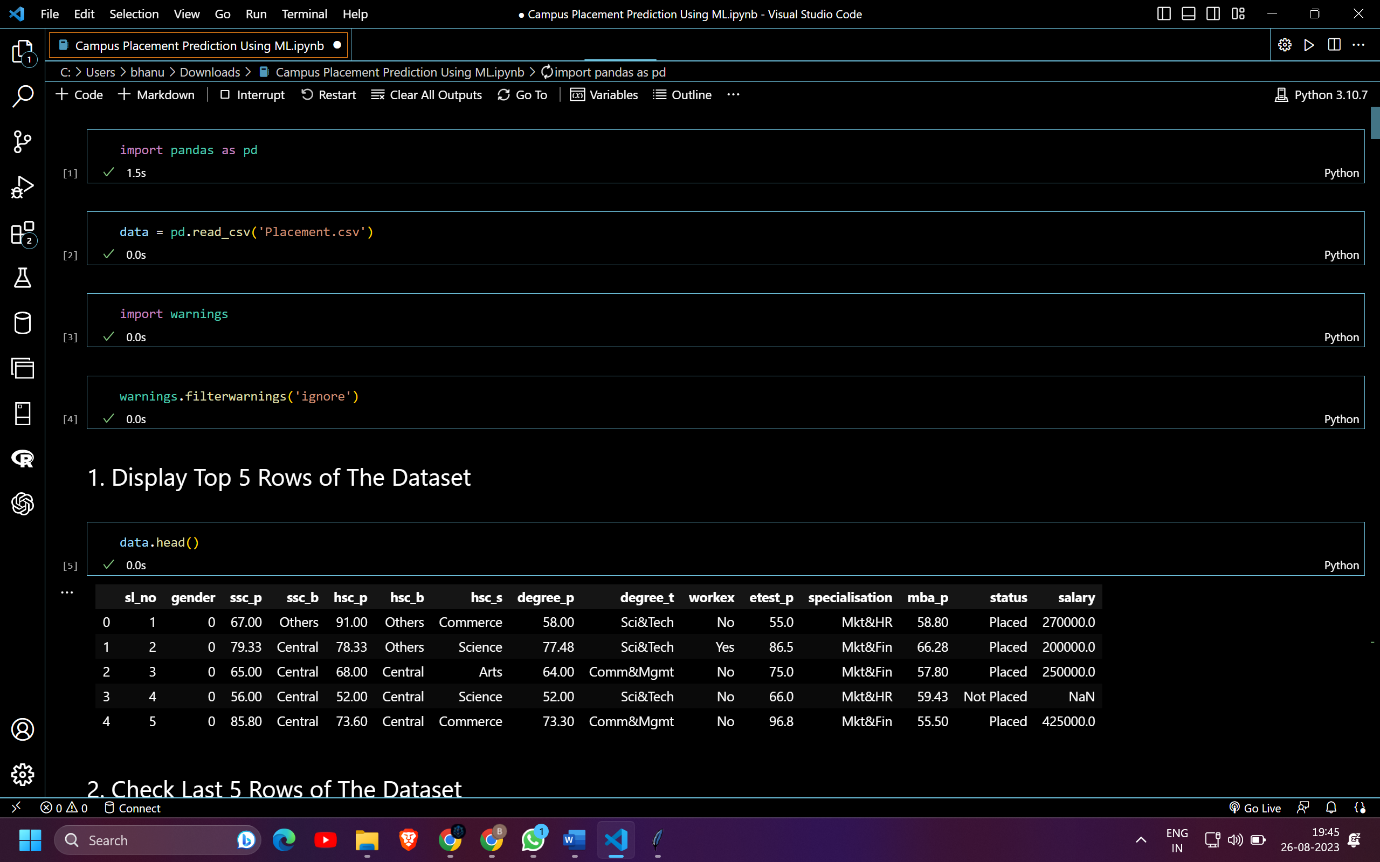
e12**.**grid(row**=**12, column**=**1)

buttonFont **=** font**.**Font(family**=**'Helvetica', size**=**16, weight**=**'bold')

Button(master, text**=**'Predict',height**=** 1, width**=**8,activebackground**=**'#00ff00',font**=**buttonFont,bg**=**'black', fg**=**'white',command**=**show\_entry\_fields)**.**grid()

mainloop()

**output:-**

****