

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

1. INTRODUCTION

1.1 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.

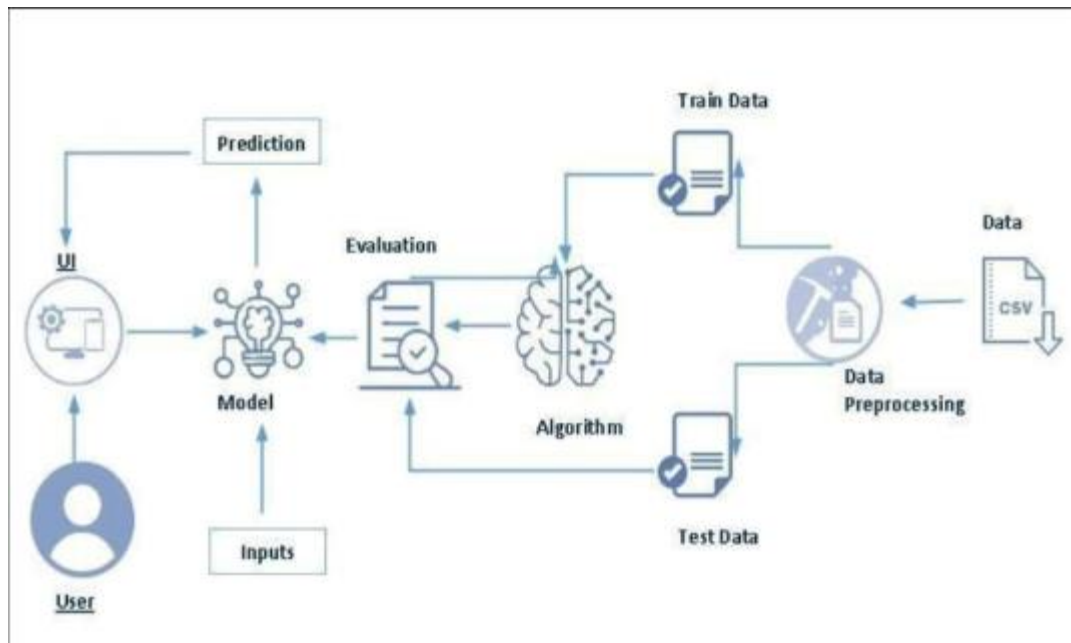
1.2 PURPOSE

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.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 EMPATHY MAP



2.2 IDEATION & BRAINSTORMING MAP



3.RESULT

The dataset for campus placement is collected from the URL.

<https://www.kaggle.com/code/neesham/prediction-of-placements/data>

```
df = pd.read_csv(r"/content/collegePlace.csv")
df.head()
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2966 entries, 0 to 2965  
Data columns (total 8 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   Age                   2966 non-null   int64  
1   Gender                2966 non-null   object  
2   Stream                2966 non-null   object  
3   Internships           2966 non-null   int64  
4   CGPA                  2966 non-null   int64  
5   Hostel                2966 non-null   int64  
6   HistoryOfBacklogs     2966 non-null   int64  
7   PlacedOrNot           2966 non-null   int64  
dtypes: int64(6), object(2)  
memory usage: 185.5+ KB
```

```
df.isnull().sum()
```

```
Age                0  
Gender             0  
Stream            0  
Internships       0  
CGPA              0  
Hostel            0  
HistoryOfBacklogs 0  
PlacedOrNot       0  
dtype: int64
```

```
df = df.drop(['Hostel'], axis=1)
```

df

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot
0	22	0	2	1	8	1	1
1	21	1	0	0	7	1	1
2	22	1	1	1	6	0	1
3	21	0	1	0	8	1	1
4	22	0	3	0	8	0	1
...
2961	23	0	1	0	7	0	0
2962	23	0	3	1	7	0	0
2963	22	0	1	1	7	0	0
2964	22	0	0	1	7	0	0
2965	23	0	5	0	8	0	1

2966 rows × 7 columns

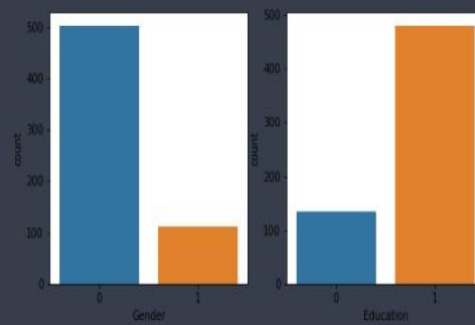
```
#plotting the count plot
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
```

C:\Users\HP\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

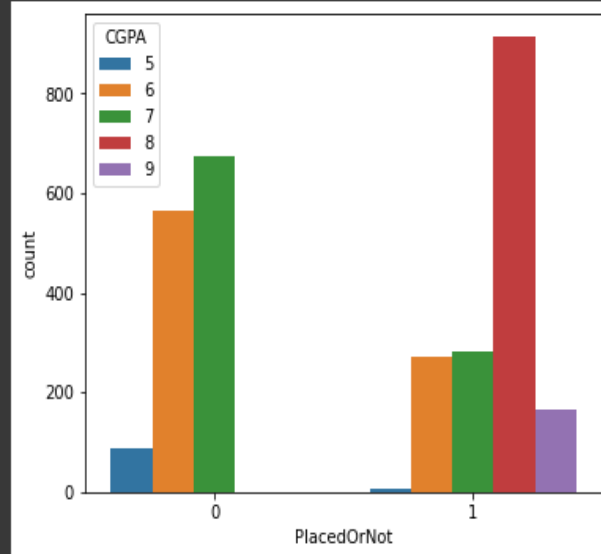
C:\Users\HP\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.

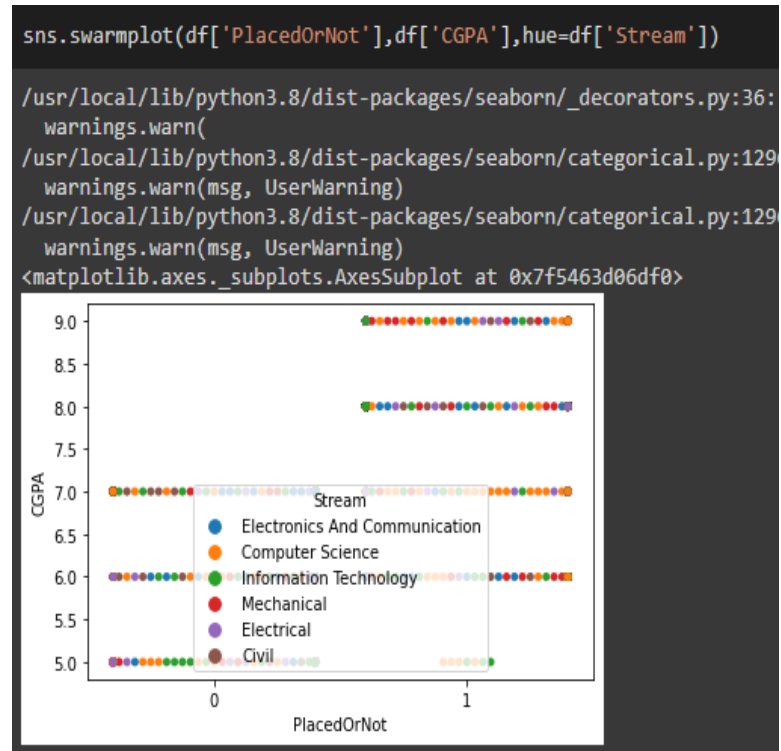
warnings.warn(



```
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(df["PlacedOrNot"],hue=df['CGPA'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f5461cf85b0>
```





3. ADVANTAGES & DISADVANTAGES

ADVANTAGES

Campus recruitment activity offers several advantages to both the companies as well as the job applicants. Following are some of the main advantages of campus recruitment listed below -

The companies will be benefited from getting wide choice of candidates to select for different job posts. Companies can select the right and talented candidate from a vast pool of young applicants within a limited time. On the other hand, students have the advantage of getting a good job according to their qualification level even before the completion of their academic course in college.

Campus recruitment helps in saving time and efforts of the companies. The entire campus recruitment process from a college is not a tedious toil. It prevents the occurrence of unusual expenditures related to recruitment process such as advertisement, initial screening, and final selection procedures etc. This in turn

turns to be useful in reduced manpower effort and time as well.

An organization through effective campus recruitment finds an opportunity to establish a link with the next batch of students. This in turn paves way to serve the future and long term recruitment needs of the company. Students participating in internships and summer training programs may have direct recruitment to different job positions offered by the company.

Campus recruitment helps in increased selection ratio. More number of quality candidates can be selected through this recruitment process.

The organizations can built up more company loyalty through campus selection process. Fresh and talented graduates will work more closely with their first company. Hence, this in a way will increase the brand loyalty among different applicants.

DISADVANTAGES

Campus recruitment is an expensive affair for majority of the companies as it adds up costs to the bottom line. Companies incur different expenses related to travel, boarding, training etc while conducting campus selection process. The experienced and skilled candidates having practical job exposures cannot be recruited through campus placements. Fresh candidates selected through campus placements require adequate training for work. This is an additional expense for the company. Also, students can't work with their dream company and will have to remain satisfied with the

company that recruits them during campus selection.

4. APPLICATIONS

The main objective of the Placement Management System is to manage the details of the Student and Placement Cell, to reduce manual work and time.

5. CONCLUSION

Maximum work goes manually in the present placement system which makes it take time to avail changes. This includes main problems like searching for the data of students and sorting them along with it. Also, updating student data is a cumbersome job and does not have a method to notify the student in time which makes the management of the placements very difficult. In the proposed system, all of these problem become automated. The registration of the student for an upcoming placement, the addition of a new user, notifying students, sharing information, the privacy of the student, etc is all met. The admin validates the information and gives the student list based on the criteria required which otherwise would have been very difficult to manage.

6. FUTURE SCOPE

The project has a wide scope. Our project mainly helps in improving productivity and makes use of utilization of resources. There is no duplication of work as this was not the case when done manually. Thus it reduces labor and increases morale. The system intends user friendly operations which may resolve ambiguity. The project is a total management and informative system, which provides the up- to-date information of all the students in the college. Our system also help the college to overcome the difficulty in keeping records of hundreds of students and searching for a student eligible for recruitment criteria from the whole thing. It helps in effective and timely utilization of resources. The project facilitates user friendly, reliable and fast management system. The placement officer itself can carry out operations in a smooth and effective manner. They need not concentrate on recordkeeping. The college can maintain

computerized records thus reducing paper work, time and money.

7. APPENDIX

```
1 Import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from tensorflow.keras import layers

[ ] classifier = Sequential()

#add input layer and first hidden layer
classifier.add(keras.layers.Dense(6,activation = 'relu', input_dim = 6))
classifier.add(keras.layers.Dropout(0.50))
#add 2nd hidden layer
classifier.add(keras.layers.Dense(6,activation = 'relu'))
classifier.add(keras.layers.Dropout(0.50))

#final or output layer
classifier.add(keras.layers.Dense(1, activation = 'sigmoid'))

[ ] #Compiling the model

loss_1 = tf.keras.losses.BinaryCrossentropy()

classifier.compile(optimizer = 'Adam', loss = loss_1 , metrics = ['accuracy'])

[ ] #fitting the model
classifier.fit(X_train, Y_train, batch_size = 20, epochs = 100)
```

```
[ ] import pickle

pickle.dump(knn,open("placement.pkl",'wb'))
model = pickle.load(open('placement.pkl', 'rb'))
```

```
from flask import Flask, render_template , request
app=Flask(__name__)
import pickle
import joblib
model=pickle.load(open("placement123.pkl",'rb'))
ct=joblib.load('placement')
```

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

```

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

    sen1=request.form["sen1"]
    sen2=request.form["sen2"]
    sen3=request.form["sen3"]
    sen4=request.form["sen4"]
    sen5=request.form["sen5"]
    sen6=request.form["sen6"]

@app.route('/y_predict', methods = ["POST"])
def y_predict():
    x_test = [[(yo) for yo in request.form.values()]]

    prediction =model.predict(x_test)

    prediction = prediction[0]

    return render_template("secondpage.html",y=prediction)

```

```

app.run(debug=True)

```

```

* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a p
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 146-359-021
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```