**Project Title:**

Identifying Patterns and Trends in Campus Placement Data Using Machine Learning With Python

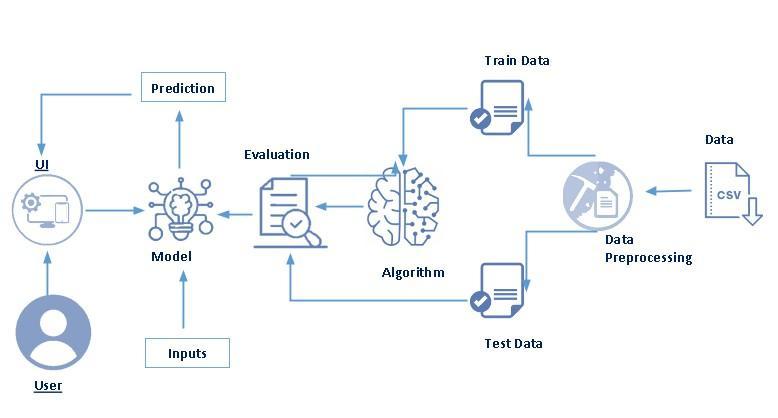
**Team ID:** NM2023TMID34164

**Project Description**

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.

**Technical Architecture**



* Data storage and management: The project would require a robust and secure data storage and management system to store and organize the large amounts of data used in the analysis.
* Infrastructure for model deployment: The project would require infrastructure for deploying the models and algorithms developed, including hardware, software, and cloud-based resources.

**Project Flow**

* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once model analyzes the input the prediction is showcased on the UI
* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting
* rdf.pkl is our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file

To accomplish this, we have to complete all the activities listed below,

* Data collection
  + Collect the dataset or create the dataset
* Visualizing and analyzing data
* Univariate analysis
* Bivariate analysis
* Multivariate analysis
* Descriptive analysis
* Data pre-processing
  + Checking for null values
  + Handling outlier
  + Handling categorical data
  + Splitting data into train and test
* Model building
  + Import the model building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating performance of model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

**Collect the dataset or create the dataset**

**Collect the Database**

There are many popular open sources for collecting the data. Eg:kaggle.com, UCI repository, etc. In this project we have used .csv data. This   data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

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

File Name-collegePlace.csv

**READ THE DATA SET**

df=pd.read\_csv("E:\\NMDS\collegePlace.csv")

df.head()

**IMPORTING THE LIBRARIES**

import numpy as np

import pandas as pd

import os

import plotly.express as px

import plotly.graph\_objects as go

from plotly.subplots import make\_subplots

import plotly.express as px

from sklearn import preprocessing

import seaborn as sns

import matplotlib.pyplot as plt

import math

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

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

from sklearn.model\_selection import cross\_val\_score

from sklearn import preprocessing

from sklearn import svm

from sklearn.svm import SVC

DATA PREPARATION

df['Stream'].value\_counts()

Computer Science 776

Information Technology 691

Electronics And Communication 424

Mechanical 424

Electrical 334

Civil 317

Name: Stream, dtype: int64

#Handling Mising Values

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: 162.3+ KB

#Handling Mising Values

df.isnull().sum()

Age 0

Gender 0

Stream 0

Internships 0

CGPA 0

Hostel 0

HistoryOfBacklogs 0

PlacedOrNot 0

dtype: int64

#Handling Categorical Values

df.describe()

AgeInternshipsCGPAHostelHistoryOfBacklogsPlacedOrNotcount2966.0000002966.0000002966.0000002966.0000002966.0000002966.000000mean21.4858400.7036417.0738370.2690490.1921780.552596std1.3249330.7401970.9677480.4435400.3940790.497310min19.0000000.0000005.0000000.0000000.0000000.00000025%21.0000000.0000006.0000000.0000000.0000000.00000050%21.0000001.0000007.0000000.0000000.0000001.00000075%22.0000001.0000008.0000001.0000000.0000001.000000max30.0000003.0000009.0000001.0000001.0000001.000000

# I tried all the columns and find out that only age column has some outliers.

plt.figure(figsize = (10, 6), dpi = 100)

sns.boxplot(x = "Age", data = df)

