**19ZO02 - Social And Economic Network Analysis**

“Student Network Placement Data Analysis”

**Srivathssan VV – 19z247**

**Kousik Nibith Ram V P – 19z253**

**Vignesh M – 19z258**

**Vishwakjith I – 19z260**

**Rajesh G – 20z432**

**BACHELOR OF ENGINEERING**

**Branch: COMPUTER SCIENCE & ENGINEERING**

Of Anna University

**May 2022**

**Department of Computer Science & Engineering**

**PSG College of Technology**

(Autonomous Institution)

**Coimbatore - 641004**

**PROBLEM STATEMENT:**

Unemployment has become a huge threat to the graduates in largely populated nations like India. In order to overcome the risk of getting struck in unemployment, many people choose domains that have lots of opportunities where job vacancies are huge and the scope of getting placed is higher. Thus, the fact that a student will get placed or not depends on his/her specialization of degree as well. Our problem statement is to analyze the placement data of students of a particular institution and draw some valuable insights like the department with highest placement percentage, the department with highest average salary etc. These inferences could guide freshers while they choose their specializations.

**DATASET DESCRIPTION:**

For our analysis, we took the placement data of Kaggle. This dataset consists of the placement data of students in campus recruitment. It includes details of students like secondary and higher secondary school percentage, their degree and specialization. It also includes their work experience and salary offered to them during their recruitment.

It contains about 215 rows with around 15 columns and so the shape of the dataset is 215\*15. And some exploratory analyses were performed on the dataset. On analyzing we found that the dataset contains null values in the salary column, around 67 null values were present. They were replaced with 0 for calculating average salary. The datatypes of all the columns were analyzed in order to verify whether numerical columns are stored as numbers. Around 6 float columns, 8 categorical columns and 1 integer column was found.

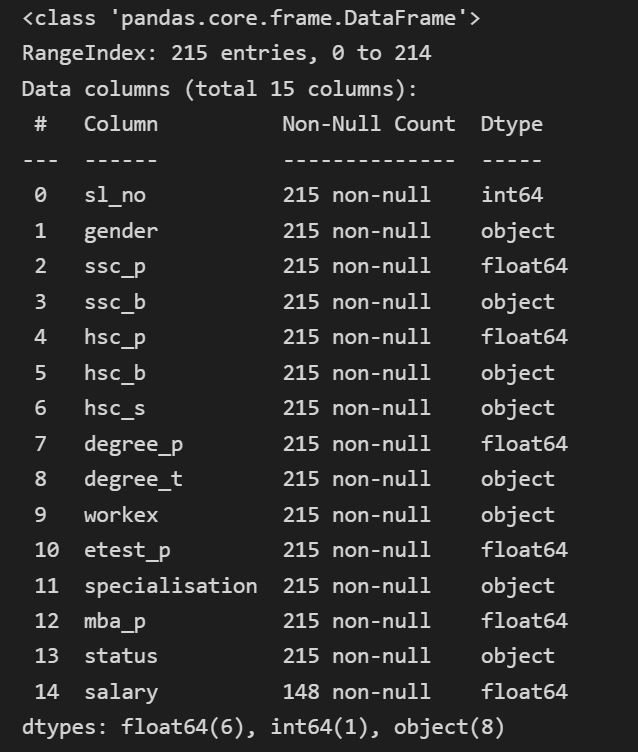


Fig 1. Datatype description of individual columns

The dataset was further analyzed statistically to draw more insights. The mean, median and standard deviation of all the numerical columns was found to draw insights like range in which the salary column varies, the average salary of all the students irrespective of departments and further some more insights regarding the balanced nature of the dataset were drawn as well. It was observed that the dataset had the maximum count of about 86 people from “Commerce and Management” degree under “Market and Finance” specialization. In total there are about 6 different combinations of degree and specialization. The counts of students from other departments and specializations were analyzed as well.

DATASET URL: <https://www.kaggle.com/code/secunsexto/placement-data/data>

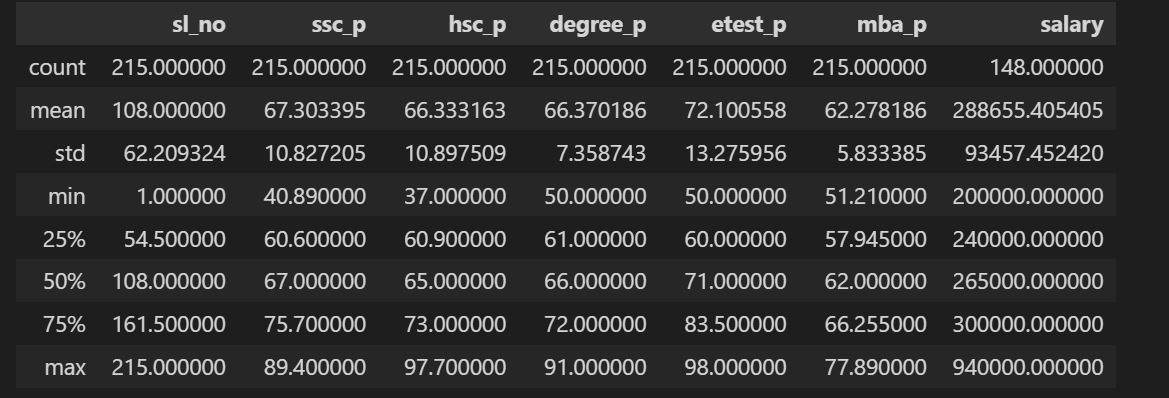


Fig 2. Statistics of the Dataset

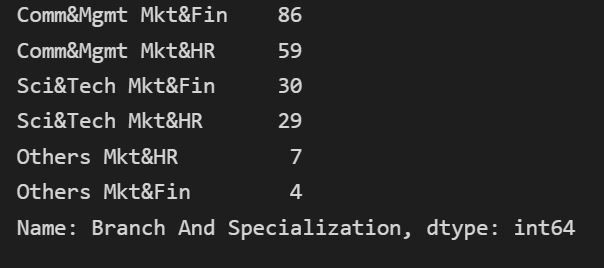


Fig 3. Number of records in each department - specializations

**TOOLS USED:**

In this section, we briefly discuss about the various tools and packages that we have used in order to accomplish our project

1. **NETWORKX**
   1. This python package helps in creating graphs out of edges and nodes given in a csv file.
   2. In order to visualize the various clusters in our dataset, we used this package to draw a simple graph that depicts the 6 different clusters that we mentioned earlier in fig 3
   3. This package helped us to analyze the graph as well and some features like the average cluster co-efficient, diameter and the degree distribution were obtained. It was observed that the graph turned out to have infinite diameter since the graph is not fully connected
2. **MATPLOTLIB.PYPLOT**
   1. This package is most widely used to plot bar graphs and other visualizations that help us plot clearly what we want.
   2. We used this to plot the various departments versus the average salary and departments versus their placement percentage
   3. This helped us to view visually the highest placement percentage and maximum average salary
3. **PANDAS**
   1. This python package is used for processing csv (Comma Separated Values) files. The dataset is mostly available as .csv files
   2. We used this to remove null values and clean our dataset. Also, we used pandas to create our own custom dataset from the original one for our convenience
   3. We used pandas to perform exploratory analysis on the dataset for drawing more insights regarding the dataset.

**CHALLENGES FACED:**

1. Since the graph was not a fully connected one and edges were defined based on the department and specialization, link prediction became infeasible as links were static. So, we took cluster-based analyses.
2. To do cluster-based analyses, we had to brainstorm to decide on the valuable features that could give us valuable insights and whose analysis would be fruitful as well.
3. The raw dataset available was not suitable enough to consume directly. So, we had to clean and prepare the data before making our analysis

**CONTRIBUTION OF TEAM MEMBERS:**

|  |  |  |
| --- | --- | --- |
| Roll Number | Name | Contribution |
| 19z247 | Srivathssan VV | Cluster Analysis – Average Salary |
| 19z253 | Kousik Nibith Ram V P | Cluster Analysis – Maximum placement percentage |
| 19z258 | Vignesh M | Dataset Analysis |
| 19z260 | Vishwakjith I | Visualization graphs |
| 20z432 | Rajesh G | Graph Analysis |

**ANNEXURE 1: CODE:**

**#import statements**

import networkx as nx

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

**#Read csv files**

df = pd.read\_csv('Edges.csv')

df.head()

df.columns

**#Create Graph From edge list**

G = nx.from\_pandas\_edgelist(df,source='Source',target='Target')

plt.figure(figsize=(20,20))

nx.draw(G)

**#Plot histogram**

plt.hist([v for k,v in nx.degree(G)])

**#print average clustering co efficient**

nx.cluster.average\_clustering(G)

**#print diameter - infinite diameter**

try:

nx.diameter(G)

except Exception as e:

print('Infinite Diameter')

**#Conversion of graph to edge matrix for clustering**

def graph\_to\_edge\_matrix(G):

edge\_mat = np.zeros((len(G),len(G)),dtype=int)

for node in G:

for neighbor in G.neighbors(node):

edge\_mat[node-1][neighbor-1]=1

edge\_mat[node-1][node-1]=1

return edge\_mat

**#Read csv for cluster analyses**

df = pd.read\_csv('Placement\_Data\_Full\_Class.csv')

df.head()

df.columns

**#Create custom dataset**

dataset = {

'Sl\_no':[],

'Branch And Specialization':[],

'Placement Status':[],

'Salary':[]

}

for index,row in df.iterrows():

dataset['Sl\_no'].append(row['sl\_no'])

dataset['Branch And Specialization'].append(row['degree\_t']+' '+row['specialisation'])

dataset['Placement Status'].append(row['status'])

dataset['Salary'].append(row['salary'])

len(df)

len(dataset['Salary'])

**#Dataset analysis**

df.info()

**#Statistical analysis**

df.describe()

**#null values checking**

df.isna().sum()

**#Creation of custom dataset**

Dataframe = pd.DataFrame(dataset)

Dataframe.head()

Dataframe.isnull().sum()

**#Filling of null values**

Dataframe.fillna(value=0.0,inplace=True)

Dataframe.head()

**#Printing count of students in each brnch and specialisation**

Dataframe['Branch And Specialization'].value\_counts()

**#Count of students who are placed in each department**

lst={

'Sci&Tech Mkt&HR':Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&HR']['Placement Status'].value\_counts()[‘Placed’],

'Sci&Tech Mkt&Fin':Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&Fin']['Placement Status'].value\_counts()),

'Comm&Mgmt Mkt&Fin':list(Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&Fin']['Placement Status'].value\_counts()),

'Comm&Mgmt Mkt&HR':list(Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&HR']['Placement Status'].value\_counts()),

'Others Mkt&HR':list(Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&HR']['Placement Status'].value\_counts()),

'Others Mkt&Fin':list(Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&Fin']['Placement Status'].value\_counts()),}

avg\_salary = []

**# Average Salary of Sci&Tech Mkt&HR**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&HR']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&HR'])).round(2))

**# Average Salary of Comm&Mgmt Mkt&Fin**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&Fin']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&Fin'])).round(2))

**# Average Salary of Comm&Mgmt Mkt&HR**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&HR']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Comm&Mgmt Mkt&HR'])).round(2))

**# Average Salary of Sci&Tech Mkt&Fin**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&Fin']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Sci&Tech Mkt&Fin'])).round(2))

**# Average Salary of Others Mkt&HR.**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&HR']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&HR'])).round(2))

**# Average Salary of Others Mkt&Fin**

avg\_salary.append((Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&Fin']['Salary'].sum()/len(Dataframe[Dataframe['Branch And Specialization']=='Others Mkt&Fin'])).round(2))

**#plotting dept vs placed students count**

bxaxis = ['Sci&Tech Mkt&HR','Sci&Tech Mkt&Fin','Comm&Mgmt Mkt&Fin','Comm&Mgmt Mkt&HR','Others Mkt&HR','Others Mkt&Fin']

byaxis = [lst[bxaxis[\_]][1] for \_ in range(len(lst))]

plt.figure(figsize=(14,5))

plt.bar(bxaxis, byaxis, color='g')

plt.title("Number of Placed Students")

plt.xlabel("Department")

plt.ylabel("Number of Students")

plt.show()

**#plotting dept vs average salary**

gxaxis = ['Sci&Tech Mkt&HR','Comm&Mgmt Mkt&Fin','Comm&Mgmt Mkt&HR','Sci&Tech Mkt&Fin','Others Mkt&HR','Others Mkt&Fin']

plt.figure(figsize=(14,5))

plt.bar(gxaxis, avg\_salary, color='b')

plt.title("Student's Average Salary")

plt.xlabel("Department")

plt.ylabel("Average Salary")

plt.show()

**# Final Analysis**

print("Hence from this analysis we can know that " + gxaxis[avg\_salary.index(max(avg\_salary))] + " has high Average Salary")

print("Hence from this analysis we can know that " + bxaxis[byaxis.index(max(byaxis))] + " has Highest Placement")

**ANNEXURE 2: SNAPSHOTS:**

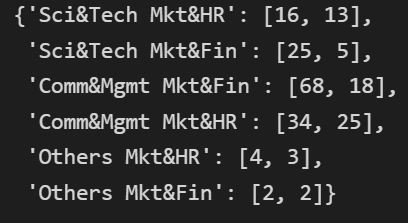
****

Fig 4. Placement Count in each Department - Specialization

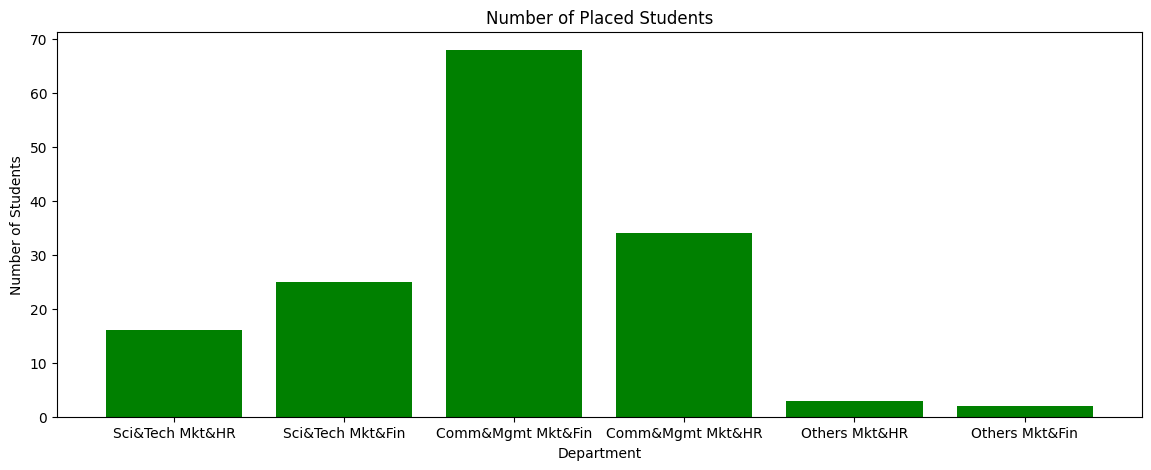
****

Fig 5. Department Vs Number of Placed Students

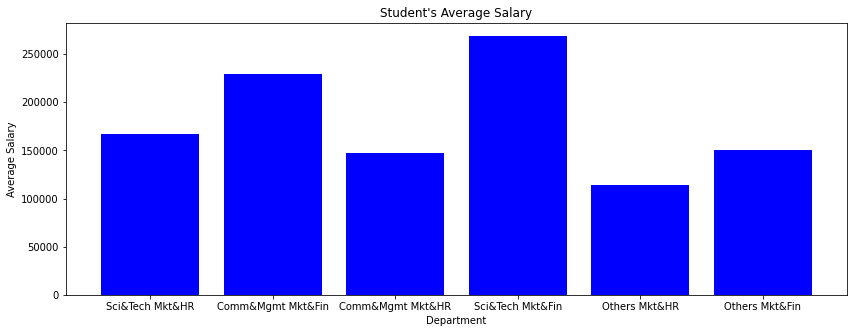
****

Fig 6. Department Vs Average Salary

****

Fig 7. Final Inference

**REFERENCES:**

1. <https://www.kaggle.com/code/secunsexto/placement-data/notebook>
2. <https://www.learndatasci.com/tutorials/k-means-clustering-algorithms-python-intro/>
3. <https://www.geeksforgeeks.org/networkx-python-software-package-study-complex-networks/>
4. <https://www.geeksforgeeks.org/different-types-clustering-algorithm/>
5. <https://www.geeksforgeeks.org/top-7-clustering-algorithms-data-scientists-should-know/>
6. <https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68>
7. <https://byjus.com/maths/cluster-analysis/>
8. <https://www.geeksforgeeks.org/introduction-to-social-networks-using-networkx-in-python/>
9. <https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/>
10. <https://www.qualtrics.com/experience-management/research/cluster-analysis/>