# Assignment - 4

# ML-based prediction System

This Jupyter Notebook helps in predicting the career for a new graduate.

There is one folder named Career Prediction System 2022. For running the Jupyter notebook, the user needs to upload the file on Google colab.

### Exploratory Data Analysis

- The roo Dataset contains 39 features in total, out of which 38 are independent features and the last feature, ‘Suggested Job Role’ is dependent, that we need to predict.
- The dataset has 14 numerical features and the rest are categorical features.

### Data Pre-processing

1. Initially, I have changed the subject names with the ones that I have used in Assignment 1.

```
df.columns
```

```
Index(['Acedamic percentage in Operating Systems', 'percentage in Algorithms',
       'Percentage in Programming Concepts',
       'Percentage in Software Engineering', 'Percentage in Computer Networks',
       'Percentage in Electronics Subjects',
       'Percentage in Computer Architecture', 'Percentage in Mathematics',
       'Percentage in Communication skills', 'Hours working per day',
       'Logical quotient rating', 'hackathons', 'coding skills rating',
       'public speaking points', 'can work long time before system?',
       'self-learning capability?', 'Extra-courses did', 'certifications',
       'workshops', 'talenttests taken?', 'olympiads',
       'reading and writing skills', 'memory capability score',
       'Interested subjects', 'interested career area ', 'Job/Higher Studies?',
       'Type of company want to settle in?',
       'Taken inputs from seniors or elders', 'interested in games',
       'Interested Type of Books', 'Salary Range Expected',
       'In a Realtionship?', 'Gentle or Tuff behaviour?',
       'Management or Technical', 'Salary/work', 'hard/smart worker',
       'worked in teams ever?', 'Introvert', 'Suggested Job Role'],
      dtype='object')
```

```
[6]
df.rename(columns = {'Acedamic percentage in Operating Systems':'Acedamic percentage in ML',
                     'percentage in Algorithms':'Percentage in AI',
                     'Percentage in Programming Concepts':'Percentage in GA',
                     'Percentage in Software Engineering':'Percentage in DMG',
                     'Percentage in Computer Networks':'Percentage in NLP',
                     'Percentage in Computer Architecture':'Percentage in DL',
                     'Percentage in Mathematics':'Percentage in LA',
                     'Percentage in Electronics Subjects':'Percentage in NSC',
                     'Percentage in Communication skills':'Percentage in PSOSM'},inplace = True)
```

2. Secondly, I have converted the total marks in the subjects to corresponding grades to make it discrete. The grading is done from 10 to 4, where 4 is the lowest and 10 is the highest.

```
# Converting Percentage values of the subject to Grades on a scale of 10 to 4
# Where 4 is the lowest grade and 10 is the highest grade

subject_columns = ['Acedamic percentage in ML', 'Percentage in AI', 'Percentage in GA', 'Percentage in DMG', 'Percentage in NLP','Percentage in DL',
'Percentage in LA','Percentage in NSC','Percentage in PSOSM']

for i in subject_columns:
    df.loc[(df[i] < 40), i] = 4
    df.loc[(df[i] >= 40) & (df[i] < 50), i] = 5
    df.loc[(df[i] >= 50) & (df[i] < 60), i] = 6
    df.loc[(df[i] >= 60) & (df[i] < 70), i] = 7
    df.loc[(df[i] >= 70) & (df[i] < 80), i] = 8
    df.loc[(df[i] >= 80) & (df[i] < 90), i] = 9
    df.loc[(df[i] >= 90), i] = 10

```

3. Thirdly, I have changed “`Logical quotient rating`” into three categories of 1,2 and 3.

```
[10]
# Converting Logical quotient rating column to Low, Medium, High

df.loc[df['Logical quotient rating'] < 4, 'Logical quotient rating'] = 1
df.loc[(df['Logical quotient rating'] >= 4) & (df['Logical quotient rating'] < 8), 'Logical quotient rating'] = 2
df.loc[df['Logical quotient rating'] >= 8, 'Logical quotient rating'] = 3
```

4. Fourthly, I have clubbed the values in “`Type of company want to settle in`” into three major categories based on their utility.

```
dict: job_categories
(3 items) {'Product Based': ['product development', 'BPA', '
```

```
services: Service Based, Product based and Sales and finance
job_categories = {'Product Based':['product development', 'BPA','Product based'],
                  'Service Based':['Web Services','SAaS services','Testing and Maintainance Services','Service Based','Cloud Services'],
                  'Sales Based':['Sales and Marketing','Finance']}

for i in job_categories:
    df.loc[df['Type of company want to settle in?'].isin(job_categories[i]),'Type of company want to settle in?'] = i
```

5. Finally, I have clubbed the target variables into 7 major job types.

- 'Software Developer':['Database Developer', 'CRM Technical Developer', 'Mobile Applications Developer', 'Web Developer', 'Software Developer','Applications Developer']
- 'Engineer':['Network Engineer', 'Software Engineer', 'Technical Engineer', 'Network Security Engineer','Software Systems Engineer','Quality Assurance Associate']
- 'Designer':['UX Designer','Design & UX', 'Solutions Architect','Data Architect']
- 'Manager':['Database Manager','Information Technology Auditor','Information Technology Manager','Project Manager']
- 'Analyst':['Business Systems Analyst', 'Business Intelligence Analyst', 'Information Security Analyst', 'CRM Business Analyst', 'Programmer Analyst', 'Systems Analyst', 'E-Commerce Analyst']
- 'Administrator':[ 'Portal Administrator', 'Systems Security Administrator','Network Security Administrator','Database Administrator']
- 'Site Reliability and Testing':['Software Quality Assurance (QA) / Testing','Technical Support', 'Technical Services/Help Desk/Tech Support']

```
[15]
# Converting Suggested Job Role column into 7 categories

job_role = {'Software Developer':['Database Developer', 'CRM Technical Developer', 'Mobile Applications Developer', 'Web Developer', 'Software Develope
            'Engineer':['Network Engineer', 'Software Engineer', 'Technical Engineer', 'Network Security Engineer','Software Systems Engineer','Quality
            'Designer':['UX Designer','Design & UX', 'Solutions Architect','Data Architect'],
            'Manager':['Database Manager','Information Technology Auditor','Information Technology Manager','Project Manager'],
            'Analyst':['Business Systems Analyst', 'Business Intelligence Analyst', 'Information Security Analyst', 'CRM Business Analyst', 'Programmer
           'Administrator':[ 'Portal Administrator', 'Systems Security Administrator','Network Security Administrator','Database Administrator'],
           'Site Reliability and Testing':['Software Quality Assurance (QA) / Testing','Technical Support', 'Technical Services/Help Desk/Tech Support'

for jobs in job_role:
    df.loc[df['Suggested Job Role'].isin(job_role[jobs]), 'Suggested Job Role'] = jobs
```

```
[16]
df['Suggested Job Role'].unique()
```

```
array(['Software Developer', 'Administrator', 'Analyst', 'Engineer',
       'Designer/Architect', 'Technical Support/ Testing', 'Manager'],
      dtype=object)
```

## Results

ANN results on 90-10 split:

**Classification Report :**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 288 |
| 1 | 0.19 | 0.59 | 0.28 | 382 |
| 2 | 0.00 | 0.00 | 0.00 | 229 |
| 3 | 0.17 | 0.34 | 0.23 | 335 |
| 4 | 0.00 | 0.00 | 0.00 | 235 |
| 5 | 0.15 | 0.05 | 0.07 | 345 |
| 6 | 0.00 | 0.00 | 0.00 | 186 |
| accuracy | | | 0.18 | 2000 |
| macro avg | 0.07 | 0.14 | 0.08 | 2000 |
| weighted avg | 0.09 | 0.18 | 0.11 | 2000 |

**Confusion Matrix :**

```
[[  0 173   0 101   2  12   0]
 [  0 226   0 126   4  26   0]
 [  0 135   0  80   3  11   0]
 [  0 199   0 114   2  20   0]
 [  0 146   0  74   0  15   0]
 [  0 219   0 106   3  17   0]
 [  0 115   0  56   5  10   0]]
```

**Class wise Accuracy: [0. 0.59162304 0. 0.34029851 0. 0.04927536 0.]**

ANN results on 80-20 split:

**Classification Report :**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 846 |
| 1 | 0.19 | 0.51 | 0.28 | 1112 |
| 2 | 0.00 | 0.00 | 0.00 | 672 |
| 3 | 0.21 | 0.28 | 0.24 | 1118 |
| 4 | 0.21 | 0.03 | 0.05 | 717 |
| 5 | 0.18 | 0.27 | 0.22 | 995 |
| 6 | 1.00 | 0.00 | 0.00 | 540 |
| accuracy | | | 0.19 | 6000 |
| macro avg | 0.26 | 0.15 | 0.11 | 6000 |
| weighted avg | 0.22 | 0.19 | 0.14 | 6000 |

**Confusion Matrix :**

```
[[  0 446   0 206   9 185   0]
 [  1 566   0 261  15 269   0]
 [  2 310   0 181  11 168   0]
 [  4 543   0 308  12 251   0]
 [  1 342   0 159  19 196   0]
 [  1 482   0 225  19 268   0]
 [  2 258   0 136   6 137   1]]
```

**Class wise Accuracy: [0. 0.50899281 0. 0.27549195 0.0264993 0.26934673 0.00185185]**

ANN results on 60-40 split:

**Classification Report :**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.06 | 0.00 | 0.00 | 1117 |
| 1 | 0.19 | 0.64 | 0.29 | 1492 |
| 2 | 0.00 | 0.00 | 0.00 | 890 |
| 3 | 0.21 | 0.31 | 0.25 | 1495 |
| 4 | 0.17 | 0.03 | 0.05 | 962 |
| 5 | 0.19 | 0.06 | 0.09 | 1337 |
| 6 | 0.17 | 0.00 | 0.01 | 707 |
| accuracy | | | 0.19 | 8000 |
| macro avg | 0.14 | 0.15 | 0.10 | 8000 |
| weighted avg | 0.15 | 0.19 | 0.12 | 8000 |

**Confusion Matrix :**

```
[[  1 726   0 320  20  49   1]
 [  4 955   2 424  28  77   2]
 [  2 582   0 242  20  43   1]
 [  7 923   1 464  27  72   1]
 [  2 629   1 243  30  56   1]
 [  0 847   1 369  40  76   4]
```

```
 [  1 454   0 198  15  37   2]]
```

**Class wise Accuracy: [0.00089526 0.64008043 0. 0.31036789 0.03118503 0.05684368 0.00282885]**