# Assignment 4

## An ML-based prediction system to predict the career of a new graduate


## Name:- Mohammad Atif Quamar
## Roll No:- 2020523


### Contents of each model

- Changes in database
- Building the neural network upon the refined data
- Choosing appropriate (Train : Test) ratio and no of hidden layers.
- Accuracy Score
- Classification Report
- Confusion Matrix
- Classwise Accuracies

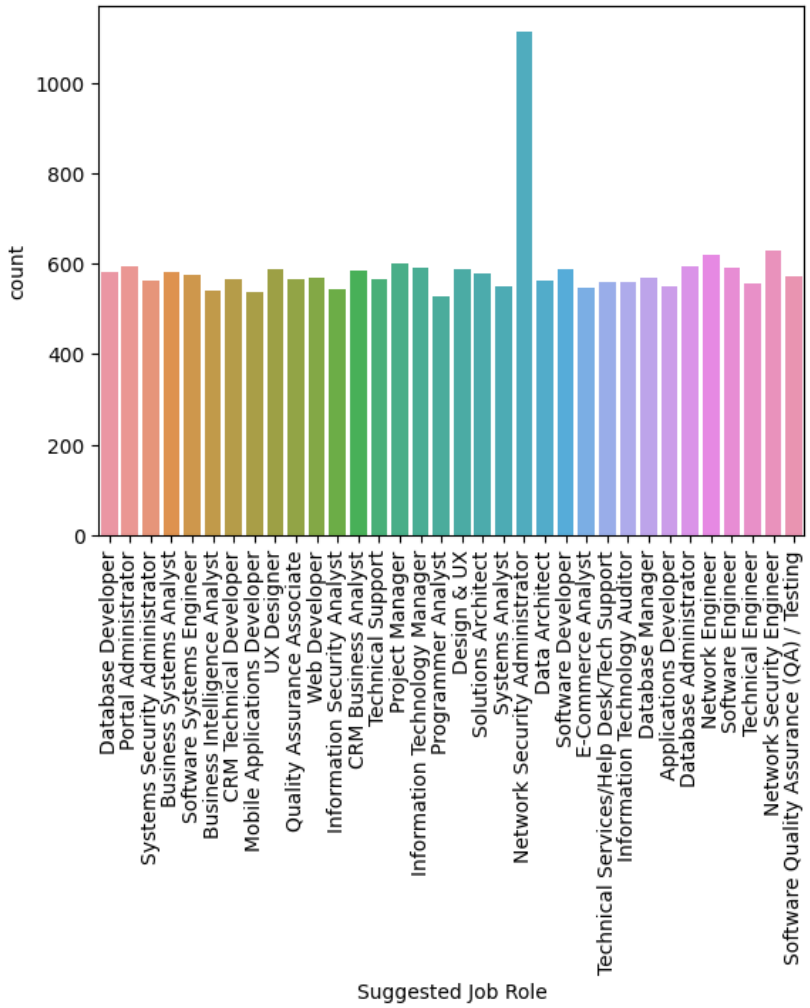
# Step 1

## Preprocessing the data

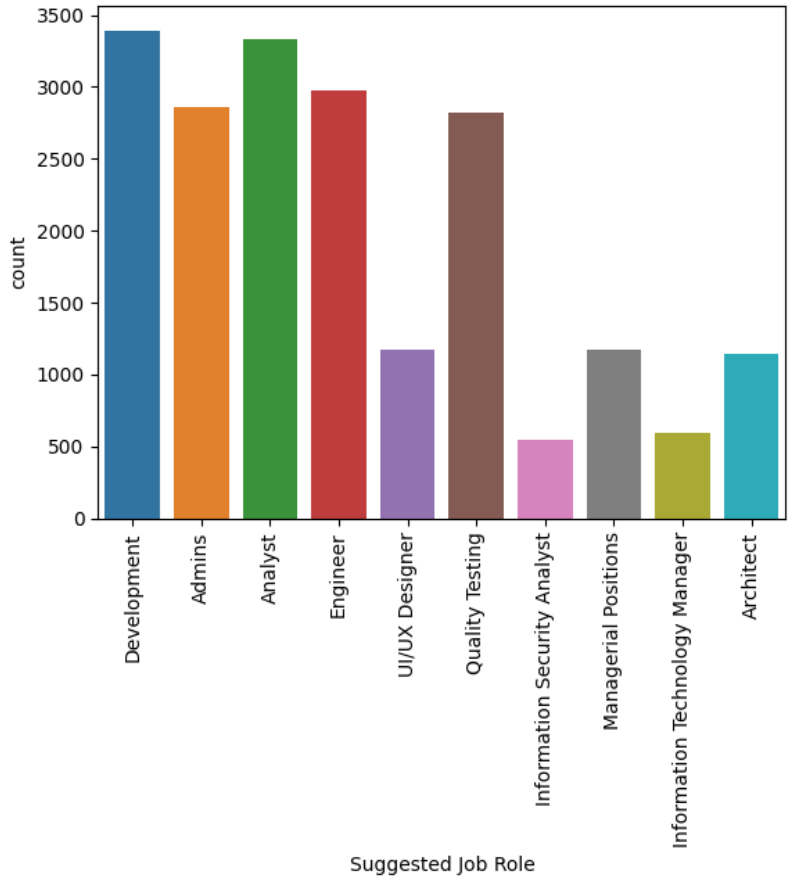
- Steps were taken to preprocess the data beforehand so that our data has some relatability with each other *(amongst their attributes).*

- Some of the steps taken were:-

    - Grouping similar job professions together. The grouping was done based on the type of job it was. Before grouping, the number of jobs for each job looked something like this:-



- After grouping similar job profiles, the data looked like this:-




- Also, changes were made to certain columns to generalize marks. As the marks ranged from 0-100, they had pretty much inconsistency amongst the. So I kept three slabs for the marks. Marks >=70, were classified as good, 70 > Marks >= 40 were classified as average, and 40 > Marks were classified as poor.

# Step 2

## Experimentations performed to increase accuracy

- The experimentation done by us, firstly, was to drop certain columns from the database, which seemed irrelevant as a factor for deciding a Job Role for a college graduate.
    - For example:- Columns such as - ['Hackathons', 'Extra-courses did', 'Certifications', 'Workshops', 'Talent Tests taken?', 'Introvert', 'In a Realtionship?', 'Gentle or Tuff behaviour?', 'Worked in Teams Ever?', 'Olympiads', 'Hours Working Per Day', 'Public Speaking Points', 'Interested Type of Books'] were dropped from the database for several tests.
- Dropping these columns helped us to remove redundancy in the database and make our ML model much better.

## Connecting this data to our Electives Advisory System built in Prolog

- To connect our data to the Electives Advisory System built in Prolog, I modified the data so that the career is predicted by the attributes such as marks in different subjects and interests of students.
- To do this, the only columns kept were:-

  Academic 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 Interested Subjects Interested Career Area Interested in Games

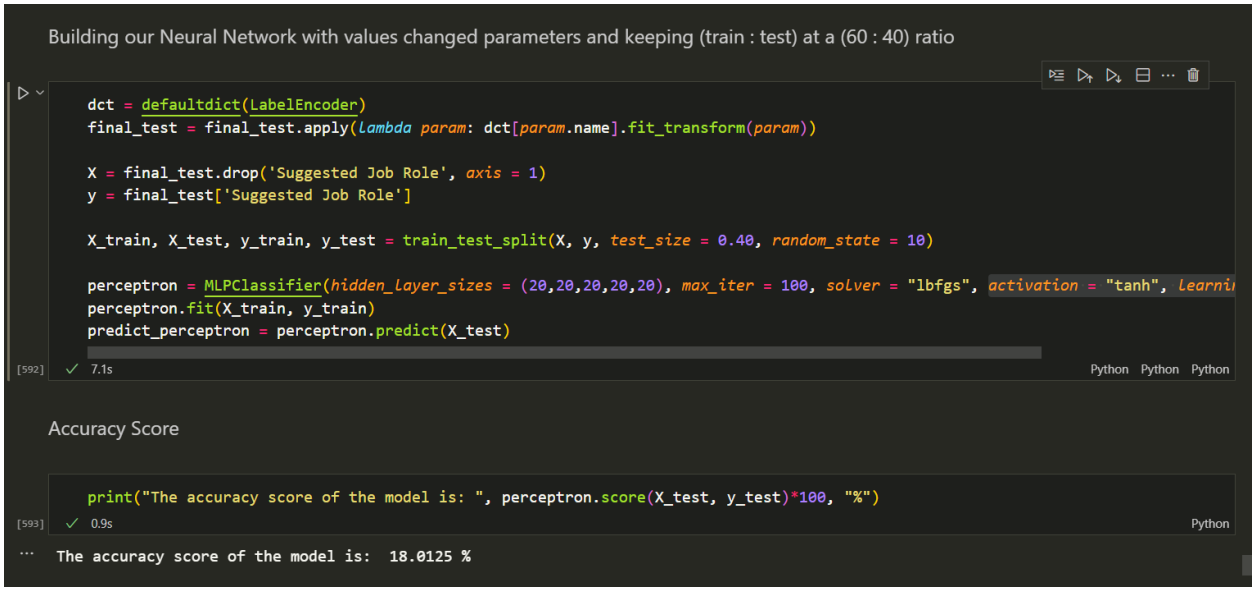
- These columns only signify the grades of the students and his/her interests.

## Results

Note:- Results often vary every time we run the program as the results are much dependent on how many tasks the processor is executing at the time of executing the program. *(tasks apart from the execution of our code)*

- The best average results were found on the 7th attempt when I dropped all more redundant columns and other features kept as in previous iterations. The accuracy ranged between 18-19%.

5 hidden layers of sizes 20 each were kept, and the max iteration was kept at 1000, activation = "tanh", learning_rate='adaptive'. Train : Test ratio was 60 : 40.



## Analysis of the Results

- Before prepping the data, the model accuracy was 4% which was very poor.
- After prepping the data we were able to achieve 18-19% accuracy with the model.
- We observed that as we dropped more columns, the accuracy has been increased. This implies that there is too much data that is not even determining the final suggestion predicted by the ML model. Hence, data has to be refined more, and generalization has to be done logically to achieve more and more accuracy from the model.
- The maximum accuracy is reached in the last stage when more columns are dropped.
- Our model has the highest accuracy percentage of predicting the career of a student related to the development field with an overall accuracy of 74.43559096945552 %

**A listing of the program of all the implementations has been provided alongside the code given in the .ipynb file, and submitted alongside the report.**