AI ASSIGNMENT 4

Steps to run the program:

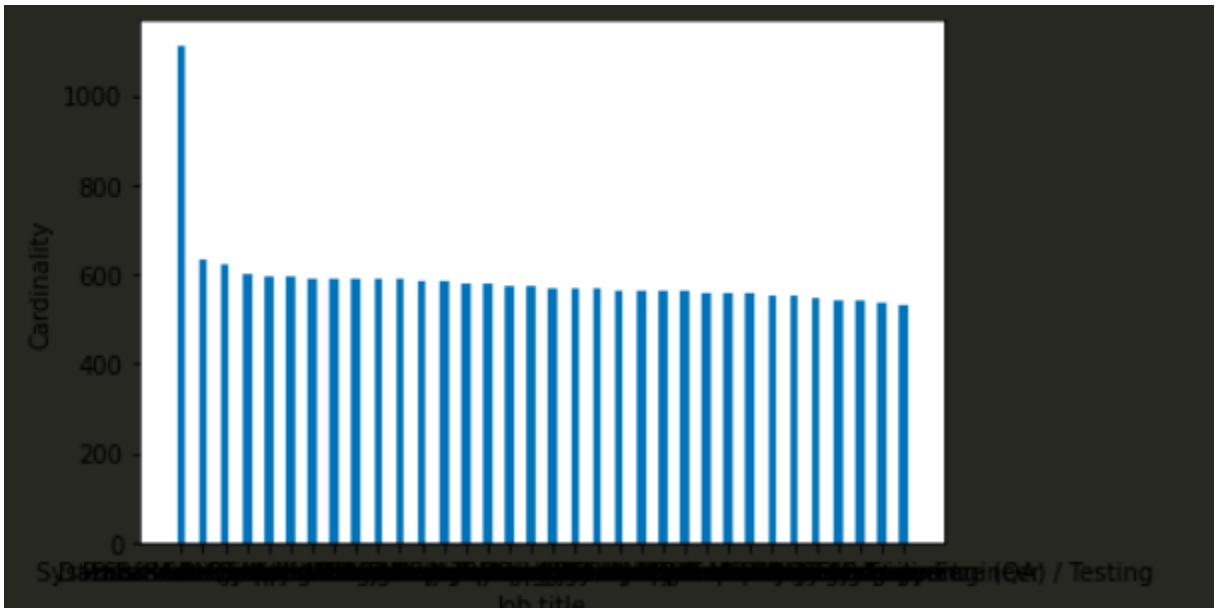
Run the code cells in attached python notebook sequentially starting from the first cell.

Steps for preparation of data and making the model:

- First, the data from the CSV file was read.
- Using LabelEncoder, the columns whose data was in string format were converted to numerical format. This ensures that the ML model can be built upon this data. A sample output for the same is attached:

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?	Introve
21	1	0	1	0	0	0	1	
5	1	1	0	1	0	0	0	
29	0	0	1	0	1	0	0	
23	0	1	0	0	1	1	1	

- For the initial data, the distribution is skewed:

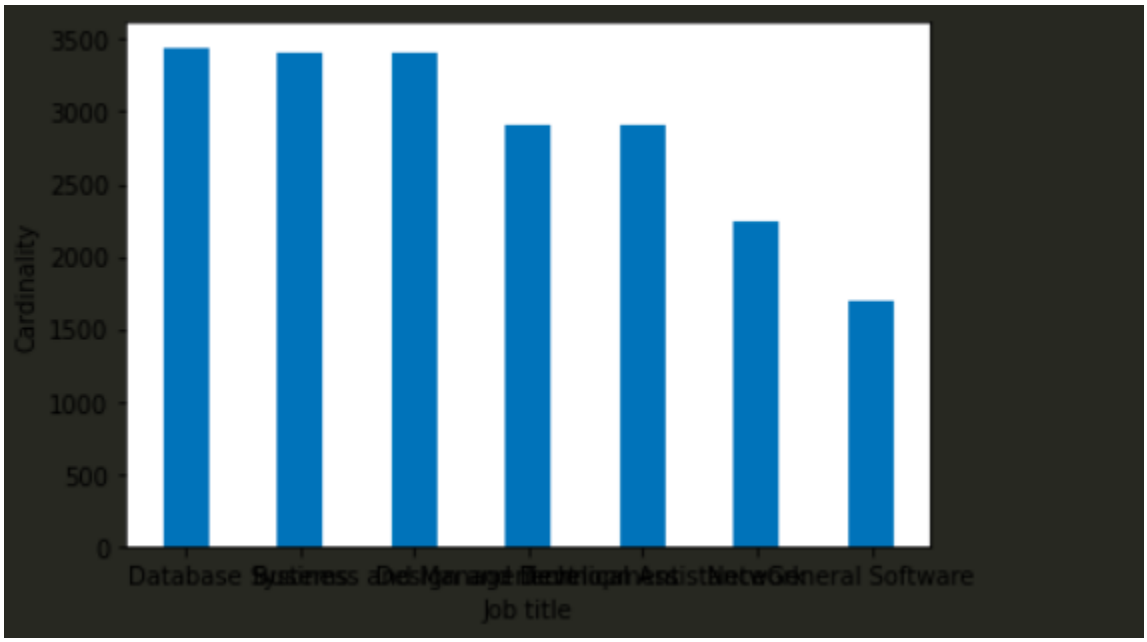


```
1    Network Security Administrator                1112
2    Network Security Engineer                      630
3    Network Engineer                               621
4    Project Manager                                602
5    Database Administrator                         593
6    Portal Administrator                           593
7    Information Technology Manager                 591
8    Software Engineer                              590
9    UX Designer                                    589
10   Design & UX                                    588
11   Software Developer                             587
12   CRM Business Analyst                           584
13   Business Systems Analyst                       582
14   Database Developer                             581
15   Solutions Architect                            578
16   Software Systems Engineer                      575
17   Software Quality Assurance (QA) / Testing      571
18   Database Manager                               570
19   Web Developer                                  570
20   CRM Technical Developer                        567
21   Technical Support                              565
22   Quality Assurance Associate                    565
23   Data Architect                                 564
24   Systems Security Administrator                 562
25   Information Technology Auditor                 558
26   Technical Services/Help Desk/Tech Support      558
27   Technical Engineer                             557
28   Applications Developer                         551
29   Systems Analyst                                550
30   E-Commerce Analyst                             546
31   Information Security Analyst                   543
32   Business Intelligence Analyst                  540
33   Mobile Applications Developer                  538
34   Programmer Analyst                             529
35   Name: Suggested Job Role, dtype: int64
```

- The model for the same does not give even a decent accuracy:

```
The accuracy of the model is: 0.03716666666666667
```

- Because of this, the target columns are modified a bit as in the code. The new distribution is:




```
Business and Management    3445
General Software           3413
Design and Development     3403
Network                    2906
Database                   2901
Systems                    2244
Technical Assistance       1688
Name: Department, dtype: int64
```

- After this modification, the accuracy of the model is significantly improved.

```
The accuracy of the model is: 0.14833333333333334
```

- Now, in order to further improve the accuracy, a few of the columns, which do not seem very important in the career prediction model, are also removed. These columns are:
  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', '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', 'Salary Range Expected',
'Gentle or Tuff behaviour?', 'Management or Technical', 'Salary/work',
'hard/smart worker', 'worked in teams ever?', 'Department'

- After removing these, the accuracy is further improved.

```
The accuracy of the model is: 0.167
The accuracy of the model is: 0.163
The accuracy of the model is: 0.14183333333333334
The accuracy of the model is: 0.16683333333333333
The accuracy of the model is: 0.147
The accuracy of the model is: 0.15566666666666668
The accuracy of the model is: 0.14816666666666667
The accuracy of the model is: 0.17283333333333334
The accuracy of the model is: 0.16883333333333334
The accuracy of the model is: 0.16766666666666666
Average accurace: 0.15988333333333335
```

- Now since our data has been set for our model, we add tinker with certain options of the MLPClassifier so as to get the best accuracy possible.
    - First, we try to change and modify the number of hidden layers, we add 3 hidden layers with 20 neurons each.

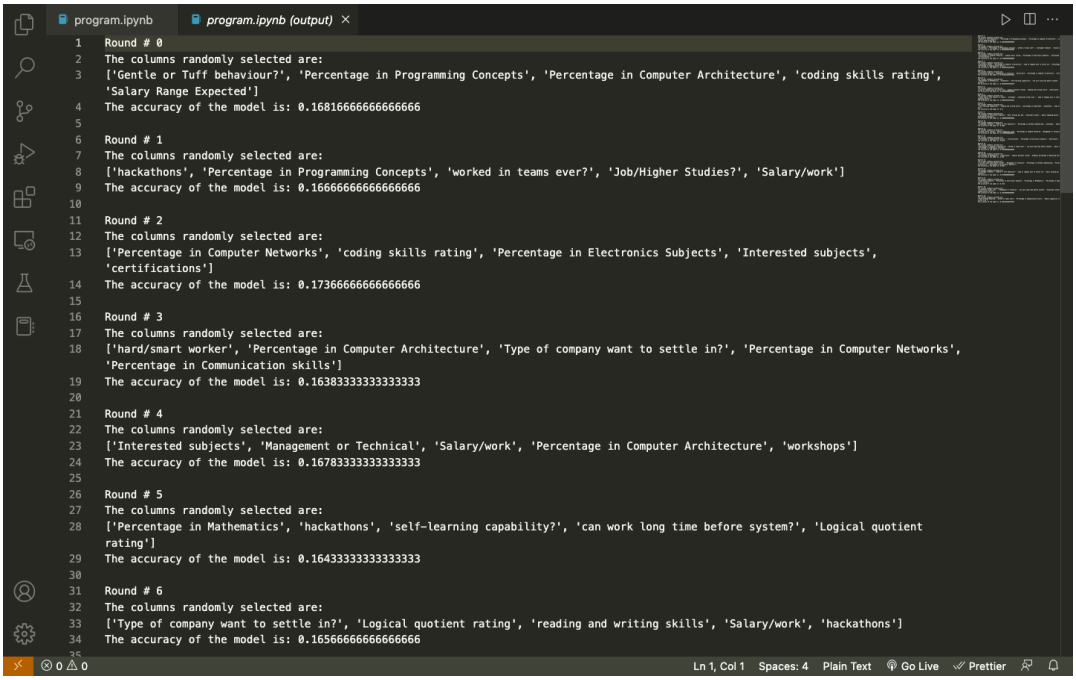
```
The accuracy of the model is: 0.171
The accuracy of the model is: 0.16933333333333334
The accuracy of the model is: 0.16683333333333333
The accuracy of the model is: 0.16066666666666668
The accuracy of the model is: 0.165
The accuracy of the model is: 0.16233333333333333
The accuracy of the model is: 0.16866666666666666
The accuracy of the model is: 0.167
The accuracy of the model is: 0.1665
The accuracy of the model is: 0.16866666666666666
Average Accuracy: 0.16660000000000003
```

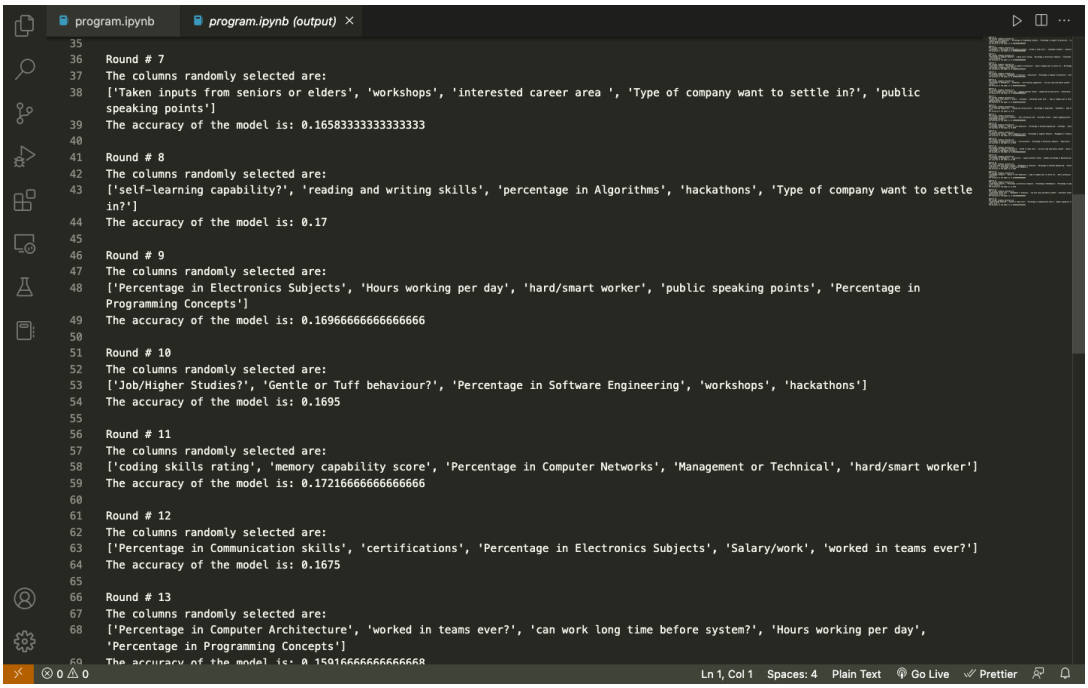
- For 4 hidden layers with 20 neurons each:

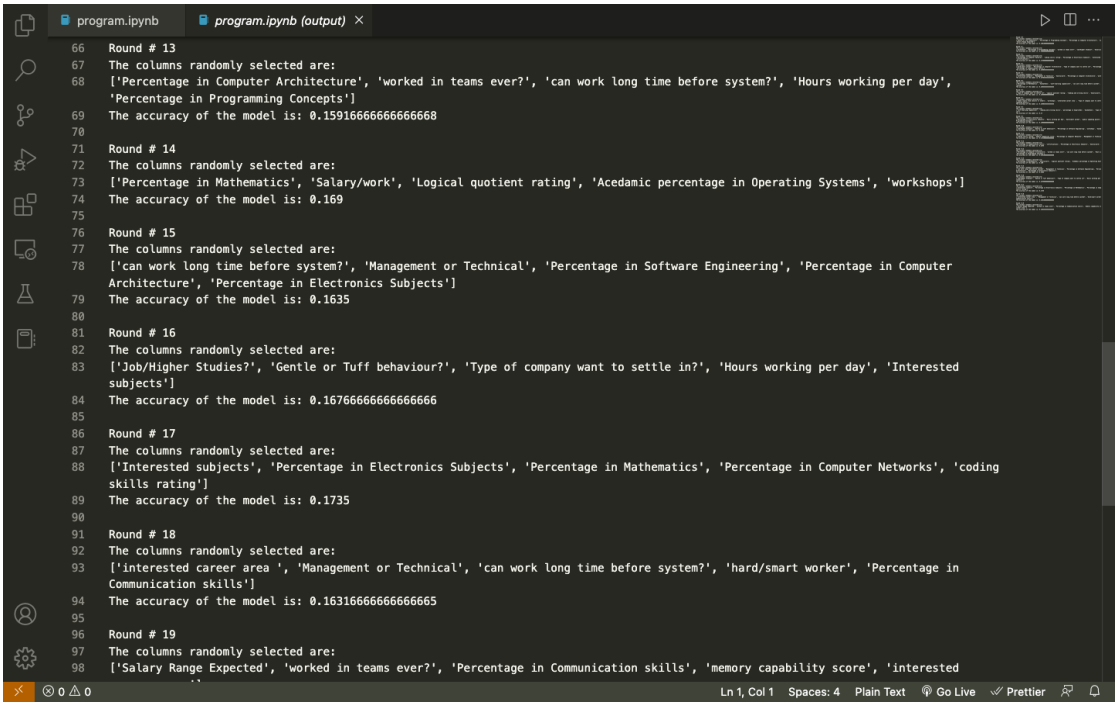
```
The accuracy of the model is: 0.17316666666666666
The accuracy of the model is: 0.16433333333333333
The accuracy of the model is: 0.16583333333333333
The accuracy of the model is: 0.166
The accuracy of the model is: 0.17
The accuracy of the model is: 0.17416666666666666
The accuracy of the model is: 0.1645
The accuracy of the model is: 0.16683333333333333
The accuracy of the model is: 0.1655
The accuracy of the model is: 0.1625
Average Accuracy: 0.16728333333333337
```

  For further analysis, this 4 hidden layer model will only be used

- Now in order to get a better model based upon the inputs, I try to select 5 random features for the model using the sample() function.


program.ipynb
program.ipynb (output)
Round # 0
The columns randomly selected are:
['Gentle or Tuff behaviour?', 'Percentage in Programming Concepts', 'Percentage in Computer Architecture', 'coding skills rating', 'Salary Range Expected']
The accuracy of the model is: 0.16816666666666666
Round # 1
The columns randomly selected are:
['hackathons', 'Percentage in Programming Concepts', 'worked in teams ever?', 'Job/Higher Studies?', 'Salary/work']
The accuracy of the model is: 0.16666666666666666
Round # 2
The columns randomly selected are:
['Percentage in Computer Networks', 'coding skills rating', 'Percentage in Electronics Subjects', 'Interested subjects', 'certifications']
The accuracy of the model is: 0.17366666666666666
Round # 3
The columns randomly selected are:
['hard/smart worker', 'Percentage in Computer Architecture', 'Type of company want to settle in?', 'Percentage in Computer Networks', 'Percentage in Communication skills']
The accuracy of the model is: 0.16383333333333333
Round # 4
The columns randomly selected are:
['Interested subjects', 'Management or Technical', 'Salary/work', 'Percentage in Computer Architecture', 'workshops']
The accuracy of the model is: 0.16783333333333333
Round # 5
The columns randomly selected are:
['Percentage in Mathematics', 'hackathons', 'self-learning capability?', 'can work long time before system?', 'Logical quotient rating']
The accuracy of the model is: 0.16433333333333333
Round # 6
The columns randomly selected are:
['Type of company want to settle in?', 'Logical quotient rating', 'reading and writing skills', 'Salary/work', 'hackathons']
The accuracy of the model is: 0.16566666666666666
Ln 1, Col 1
Spaces: 4
Plain Text
Go Live
Prettier


program.ipynb
program.ipynb (output)
Round # 7
The columns randomly selected are:
['Taken inputs from seniors or elders', 'workshops', 'interested career area ', 'Type of company want to settle in?', 'public speaking points']
The accuracy of the model is: 0.16583333333333333
Round # 8
The columns randomly selected are:
['self-learning capability?', 'reading and writing skills', 'percentage in Algorithms', 'hackathons', 'Type of company want to settle in?']
The accuracy of the model is: 0.17
Round # 9
The columns randomly selected are:
['Percentage in Electronics Subjects', 'Hours working per day', 'hard/smart worker', 'public speaking points', 'Percentage in Programming Concepts']
The accuracy of the model is: 0.16966666666666666
Round # 10
The columns randomly selected are:
['Job/Higher Studies?', 'Gentle or Tuff behaviour?', 'Percentage in Software Engineering', 'workshops', 'hackathons']
The accuracy of the model is: 0.1695
Round # 11
The columns randomly selected are:
['coding skills rating', 'memory capability score', 'Percentage in Computer Networks', 'Management or Technical', 'hard/smart worker']
The accuracy of the model is: 0.17216666666666666
Round # 12
The columns randomly selected are:
['Percentage in Communication skills', 'certifications', 'Percentage in Electronics Subjects', 'Salary/work', 'worked in teams ever?']
The accuracy of the model is: 0.1675
Round # 13
The columns randomly selected are:
['Percentage in Computer Architecture', 'worked in teams ever?', 'can work long time before system?', 'Hours working per day', 'Percentage in Programming Concepts']
Ln 1, Col 1
Spaces: 4
Plain Text
Go Live
Prettier




```
Round # 18
The columns randomly selected are:
['interested career area ', 'Management or Technical', 'can work long time before system?', 'hard/smart worker', 'Percentage in
Communication skills']
The accuracy of the model is: 0.16316666666666665

Round # 19
The columns randomly selected are:
['Salary Range Expected', 'worked in teams ever?', 'Percentage in Communication skills', 'memory capability score', 'interested
career area ']
The accuracy of the model is: 0.16433333333333333
```

- Now we use the Feature Engineering feature of the sklearn library, again we select 5 features using feature engineering options.
- First using chi2 as the scoring function:

```
The columns selected using SelectKBest are:
Index(['Logical quotient rating', 'hackathons', 'public speaking points',
       'workshops', 'Type of company want to settle in?'],
      dtype='object')
The accuracy of the model is: 0.17266666666666666

The accuracy of the model is: 0.16683333333333333

The accuracy of the model is: 0.16533333333333333

Average Accuracy: 0.168277777777777778
```

- Using f_classif as scoring function

```
The columns selected using SelectKBest are:
Index(['hackathons', 'public speaking points', 'self-learning capability?',
       'Salary/work', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.166

The accuracy of the model is: 0.16616666666666666

The accuracy of the model is: 0.16916666666666666

The accuracy of the model is: 0.16916666666666666

The accuracy of the model is: 0.16966666666666666

Average Accuracy: 0.1680333333333333
```

- From the above, it is clear that chi2 performs better than f_classif, so I will be using that scoring function for remaining trials.
- Now exploring different combinations of solver and activation function.
- Using ‘lbfgs’ solver:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.17233333333333334
```

- Using ‘sgd’ solver:

```
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.16566666666666666
```

- Since ‘lbfgs’ solver gave better accuracy than ‘sgd’ and ‘adam’ (default), I will use ‘lbfgs’ solver itself for further analysis
- Using ‘identity’ activation function:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.17066666666666666
```

- Using ‘logistic’ activation function:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.16433333333333333
```

- Using ‘tanh’ activation function:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.175
```

- ‘tanh’ activation function gives the best result as compared to ‘relu’(default), ‘logistic’ and ‘identity’ so it will be used for further analysis.

- The above analysis was done for a 70-30 train-test split. It was found that the best accuracy is shown by ‘lbfgs’ solver, ‘tanh’ activation function and ‘chi2’ algo for feature engineering. Now I will use these parameters to show confusion matrix and other parameters by also changing the train-test split:
  - 70-30 train-test split:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.17133333333333334

[[ 28  20  32  33  29  21  36]
 [  0   0   0   0   0   0   0]
 [282 237 344 290 158 263 336]
 [230 181 268 271 120 280 273]
 [  0   0   0   0   0   0   0]
 [ 36  16  33  37  11  33  26]
 [298 207 369 355 184 311 352]]

Classwise Accuracies:
Database :    0.1407035175879397
Systems :    nan
Business and Management :    0.18010471204188483
Design and Development :    0.16697473813924832
Technical Assistance :    nan
Network :    0.171875
General Software :    0.16955684007707128
```

- 60-40 train-test split:

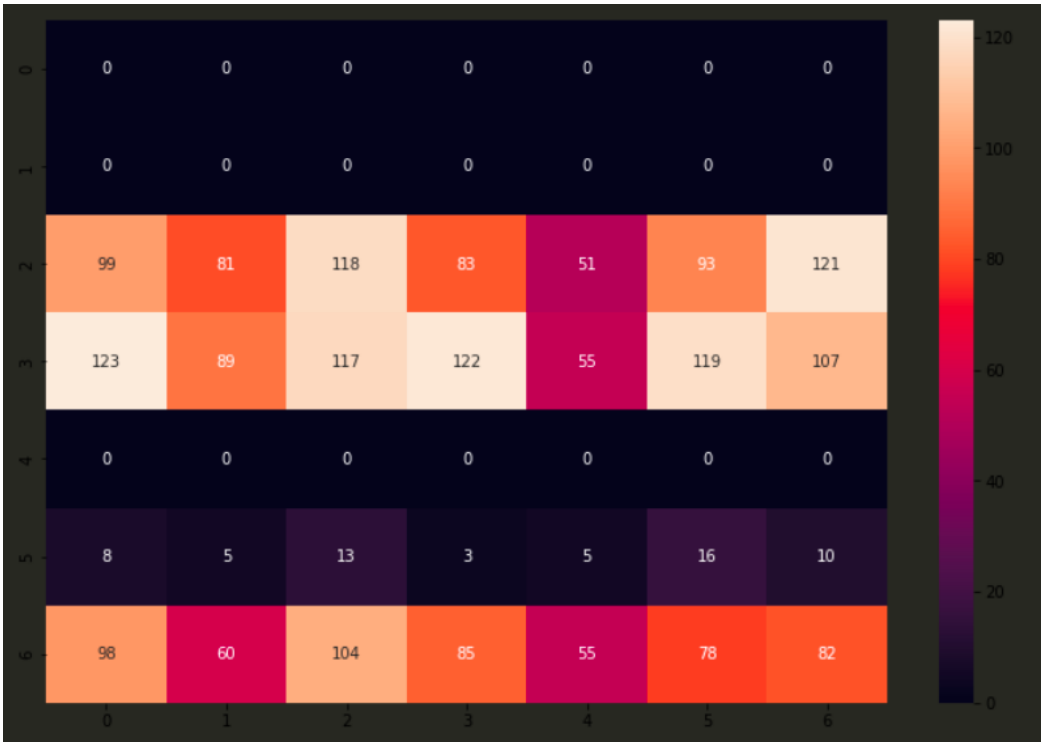
```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.172875

[[  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [426 336 489 460 250 377 475]
 [387 296 444 465 215 405 456]
 [  0   0   0   0   0   0   0]
 [ 70  46  94  84  41  83  80]
 [304 202 351 346 166 306 346]]

Classwise Accuracies:
Database :    nan
Systems :    nan
Business and Management :    0.17383576253110558
Design and Development :    0.174287856071964
Technical Assistance :    nan
Network :    0.16666666666666666
General Software :    0.17120237506185057
```

- 80-20 train-test split:

```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.17325

[[ 15  10  18  22  21  16  23]
 [  0   0   0   0   0   0   0]
 [184 166 236 189 106 175 221]
 [179 135 195 206  97 200 189]
 [  0   0   0   0   0   0   0]
 [ 26  13  25  29   5  21  19]
 [185 121 227 207 110 194 215]]

Classwise Accuracies:
Database :    0.12
Systems :    nan
Business and Management :    0.18480814408770557
Design and Development :    0.17152373022481265
Technical Assistance :    nan
Network :    0.15217391304347827
General Software :    0.17077045274027006
```

- 90-10 train-test split:

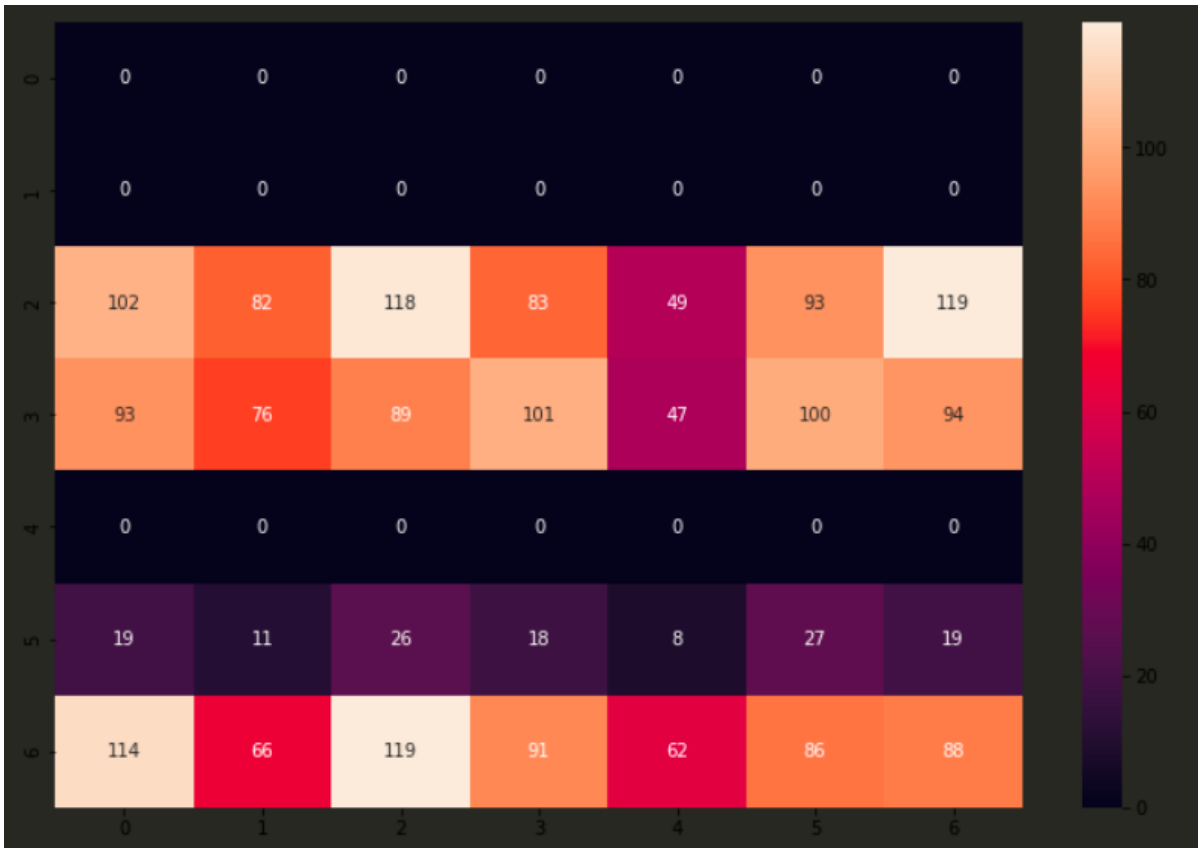
```
The columns selected using SelectKBest are:
Index(['self-learning capability?', 'Job/Higher Studies?', 'Salary/work',
       'hard/smart worker', 'worked in teams ever?'],
      dtype='object')
The accuracy of the model is: 0.169

[[  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [ 99  81 118  83  51  93 121]
 [123  89 117 122  55 119 107]
 [  0   0   0   0   0   0   0]
 [  8   5  13   3   5  16  10]
 [ 98  60 104  85  55  78  82]]

Classwise Accuracies:
Database :    nan
Systems :    nan
Business and Management :    0.1826625386996904
Design and Development :    0.16666666666666666
Technical Assistance :    nan
Network :    0.26666666666666666
General Software :    0.14590747330960854
```



For comparison with Assignment 1:

In assignment 1, I had considered various courses done to suggest what courses the person should take. Taking that forward, this time, I considered the grades in the courses to train the model using the above best parameters only.

```
The accuracy of the model is: 0.167

[[  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [102  82 118  83  49  93 119]
 [ 93  76  89 101  47 100  94]
 [  0   0   0   0   0   0   0]
 [ 19  11  26  18   8  27  19]
 [114  66 119  91  62  86  88]]

Classwise Accuracies:
Database :    nan
Systems :    nan
Business and Management :    0.1826625386996904
Design and Development :    0.16833333333333333
Technical Assistance :    nan
Network :    0.2109375
General Software :    0.14057507987220447
```



References:
https://www.geeksforgeeks.org/how-to-convert-categorical-string-data-into-numeric-in-python/
https://www.pluralsight.com/guides/machine-learning-neural-networks-scikit-learn
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
https://www.youtube.com/watch?v=-AOQieESISw
https://www.quora.com/How-do-you-measure-the-accuracy-score-for-each-class-when-testing-classifier-in-sklearn