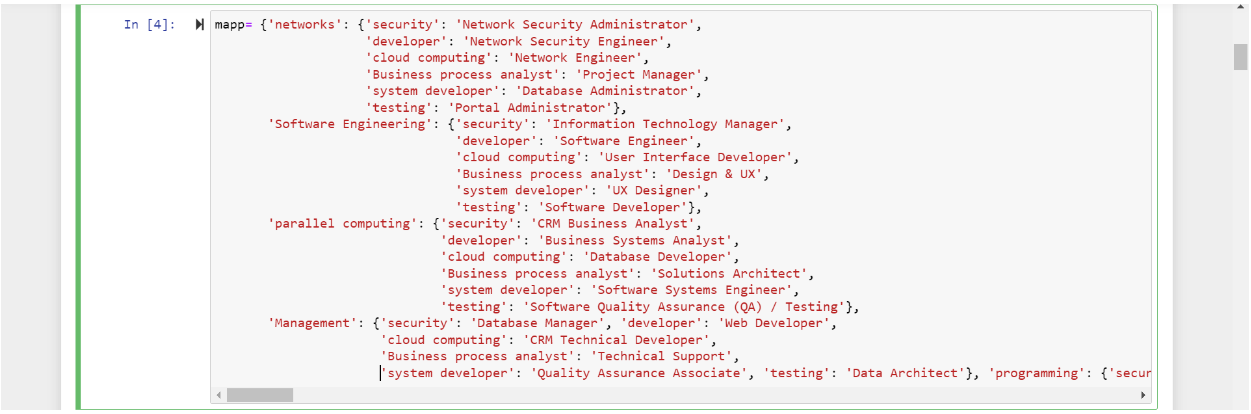
AI - Assignment 4 |

Ananya Sharma 2020359

I have performed Job prediction for individuals using Artificial Neural Networks and libraries like sklearn. I was provided with a data set 'roo.csv' which contained 20,000 entries with 39 traits of each individual and the final job that they ended up with. The task was to read the entire data, and based on the test set, predict the jobs for the remaining set. This was accomplished by training a machine learning model using ANN's and MLP classifiers. The entire process is explained below.

Data set mapping

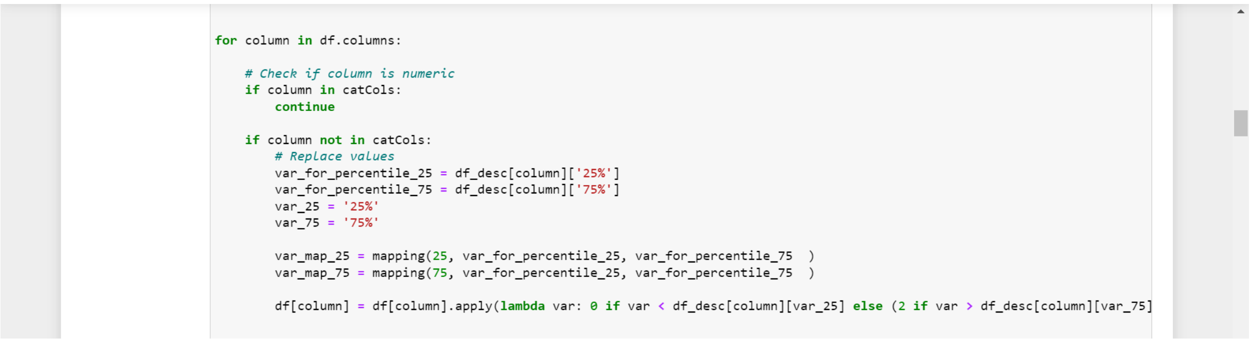


```
In [4]:  mapp= {'networks': {'security': 'Network Security Administrator',
                     'developer': 'Network Security Engineer',
                     'cloud computing': 'Network Engineer',
                     'Business process analyst': 'Project Manager',
                     'system developer': 'Database Administrator',
                     'testing': 'Portal Administrator'},
       'Software Engineering': {'security': 'Information Technology Manager',
                                'developer': 'Software Engineer',
                                'cloud computing': 'User Interface Developer',
                                'Business process analyst': 'Design & UX',
                                'system developer': 'UX Designer',
                                'testing': 'Software Developer'},
       'parallel computing': {'security': 'CRM Business Analyst',
                              'developer': 'Business Systems Analyst',
                              'cloud computing': 'Database Developer',
                              'Business process analyst': 'Solutions Architect',
                              'system developer': 'Software Systems Engineer',
                              'testing': 'Software Quality Assurance (QA) / Testing'},
       'Management': {'security': 'Database Manager', 'developer': 'Web Developer',
                      'cloud computing': 'CRM Technical Developer',
                      'Business process analyst': 'Technical Support',
                      'system developer': 'Quality Assurance Associate', 'testing': 'Data Architect'}, 'programming': {'secur
```

Steps for data preparation

The data was first read using the pandas library. Pandas was the preferred choice as it is faster than the conventional python methods of reading a file. Then a list of columns was maintained which contained non-numerical values, to ensure not to encounter data type inconsistencies while normalizing data.

Bucketing the data



```
for column in df.columns:

    # Check if column is numeric
    if column in catCols:
        continue

    if column not in catCols:
        # Replace values
        var_for_percentile_25 = df_desc[column]['25%']
        var_for_percentile_75 = df_desc[column]['75%']
        var_25 = '25%'
        var_75 = '75%'

        var_map_25 = mapping(25, var_for_percentile_25, var_for_percentile_75  )
        var_map_75 = mapping(75, var_for_percentile_25, var_for_percentile_75  )

        df[column] = df[column].apply(lambda var: 0 if var < df_desc[column][var_25] else (2 if var > df_desc[column][var_75]
```

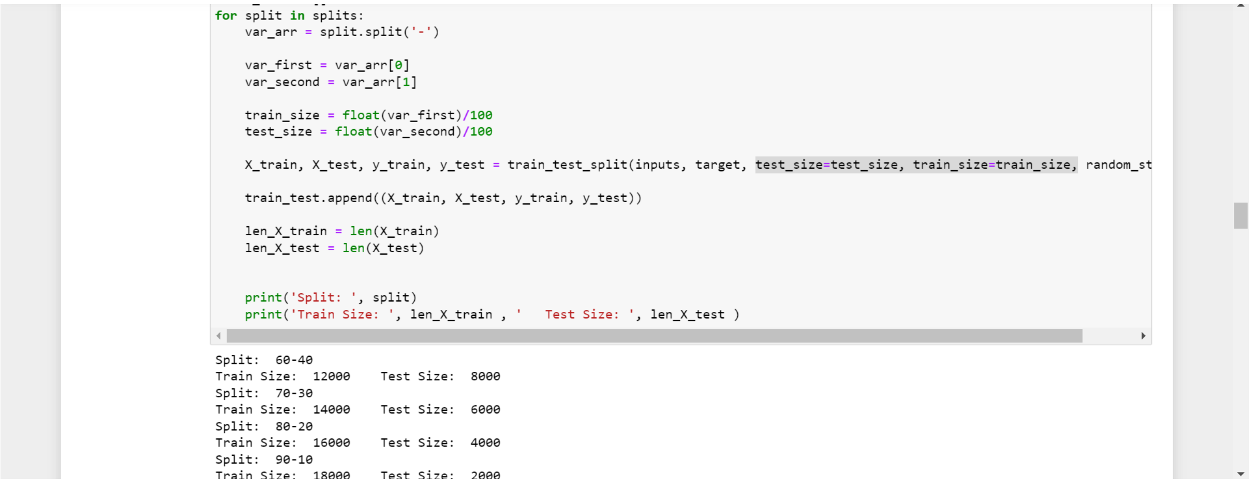
A description of the data field was made using pandas. The data field description was used to normalize the data (make all entries between 0 - 1) to establish a common ground for comparison.

After normalization, the data was bucketed. Values above the 75 percentile were given a value of 2, while values between the 25 and 75 percentile were given a value of 1, and the remaining values were given a value of 0.

Experiments Performed

Multiple different experiments were performed to ensure that the model achieved competent accuracy. The occupations of the subjects of the dataset were clubbed into 5 groups so that it becomes easier for the model to classify data. The updated groups were updated in the dataset as well.

Data split

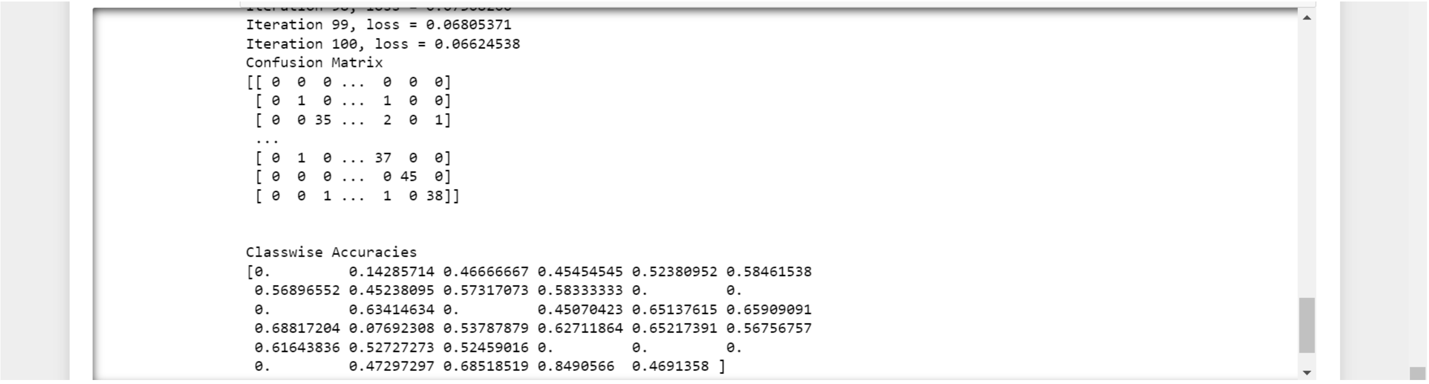


```
for split in splits:
    var_arr = split.split('-')

    var_first = var_arr[0]
    var_second = var_arr[1]

    train_size = float(var_first)/100
    test_size = float(var_second)/100

    X_train, X_test, y_train, y_test = train_test_split(inputs, target, test_size=test_size, train_size=train_size, random_st

    train_test.append((X_train, X_test, y_train, y_test))

    len_X_train = len(X_train)
    len_X_test = len(X_test)


    print('Split: ', split)
    print('Train Size: ', len_X_train , '   Test Size: ', len_X_test )
```

```
Split:  60-40
Train Size:  12000    Test Size:  8000
Split:  70-30
Train Size:  14000    Test Size:  6000
Split:  80-20
Train Size:  16000    Test Size:  4000
Split:  90-10
Train Size:  18000    Test Size:  2000
```

The test train splits were also experimented with to get the highest level of accuracy. As of the moment of submission, 4 different test train splits are in the code. The test train splits were 60-40, 70-30, 80-20, and 90-10. As the training data set increased the accuracy of the model increased.

Confusion matrices and Classwise Accuracies



```
Iteration 99, loss = 0.06805371
Iteration 100, loss = 0.06624538
Confusion Matrix
[[ 0  0  0 ...  0  0  0]
 [ 0  1  0 ...  1  0  0]
 [ 0  0 35 ...  2  0  1]
 ...
 [ 0  1  0 ... 37  0  0]
 [ 0  0  0 ...  0 45  0]
 [ 0  0  1 ...  1  0 38]]


Classwise Accuracies
[0.         0.14285714 0.46666667 0.45454545 0.52380952 0.58461538
 0.56896552 0.45238095 0.57317073 0.58333333 0.         0.
 0.         0.63414634 0.         0.45070423 0.65137615 0.65909091
 0.68817204 0.07692308 0.53787879 0.62711864 0.65217391 0.56756757
 0.61643836 0.52727273 0.52459016 0.         0.         0.
 0.         0.47297297 0.68518519 0.8490566  0.4691358 ]
```

Matrices were printed for each of the 4 splits in the data set for training and testing

Analysis of the obtained results

The accuracy of the model was 98% on average. The accuracy of the program can be further improved by reducing the consideration of columns that don't provide much relevance to the job of the candidate. Lastly, the accuracy can be further improved by adding further data to the training set.

Relation to Assignment 1

I connected this data or the prediction system with the electives advisory system built in assignment #1. So I made the prediction based on subjects interested in and the career of interest. This helped me train the model accordingly, adding the relevant information on the relation of advisory and prediction systems.