**AI Assignment – 4:**

**(Udit Kumar – MT21148)**

**Steps to show my analysis :**

**1.**Analysis of the data given roo\_data.csv

This csv file, has first 14 columns as the numeric value and the rest 15 to 39 columns has type of object that means the categorical values.

**2**.The last column is suggested job roles :

It has 34 unique job roles , First of all in our file we separate the object columns in a new data frame called

object\_Columns ,then I checked is there any null values in it.. found none..

**3**.seeing\_Count function can see the count of various object of all columns of object\_Columns.

**4.**Then I pre-process the last column that is our suggested roles as there are 34 unique suggested job roles in 20,000 rows I have written a function from the scratch called “make\_Reduction”

Which will reduced all 34 unique roles broadly in 7 roles,

If any job role has substring Administrator as it will convert into 0…Engineer in 3…and so on..

They are

[Administrator : 0, Ananlyst:1, Developer:2,Engineer: 3, Artitect:4,Manager:5, UX other jobs like testing ,QA :6]

They are label as 0….6 ,total 7 multi classes .

**4**.We use the label encoder( ) on rest object columns to make the values of different categories in number.

**5**.Then we concatenate the numeric and object columns except the suggested role coumn in a data frame called by the name “final\_Data”

**6.**we then train test split the data in the ration 70-30 ,training data 70% and testing data 30%.

// we can use select the best k feature

**7.**Before making ANN model,we perform the feature scaling by using the Standard scaler ()

**8.**First Approach : For making the Ann we have used the **keras library** in which we have used the **Sequential()** model for making ANN.

**9.**We built the first Dense layer of 38 neurons ,then the first hidden layer has 30 neurons, and a last dense layer has only 1 neuron which will predict the lable.

We have use adam as our optimizer and loss function as “categorical\_crossentropy” .

**10.**We fit the model on X train and Y train and get the accuracy of 19.73% .

**11**.Accuracy score with 70-30 training testing split we get an accuracy of 18.53 % on the test set.

**12.**Second approach is with the use of MLP classifier provided by the sklearn. neural network,

We have trained our model with 70-30 training-testing split on **MLP classifier,**

We again get the accuracy of 18.5 % on the test set.

13.I have also try the oversampling technique with **SMOTE()** but the accuracy get dropped by 4% , so I drop the idea..

14.We have also trained our Ann with train test split 80-20, and 90-10,I have found that the accuracy of the model on test set get **increase by 1 percent** with these splits ratio as compared to 70-30 train-test split.