ARTIFICIAL INTELLIGENCE
ASSIGNMENT-4

AASTHA 2019224

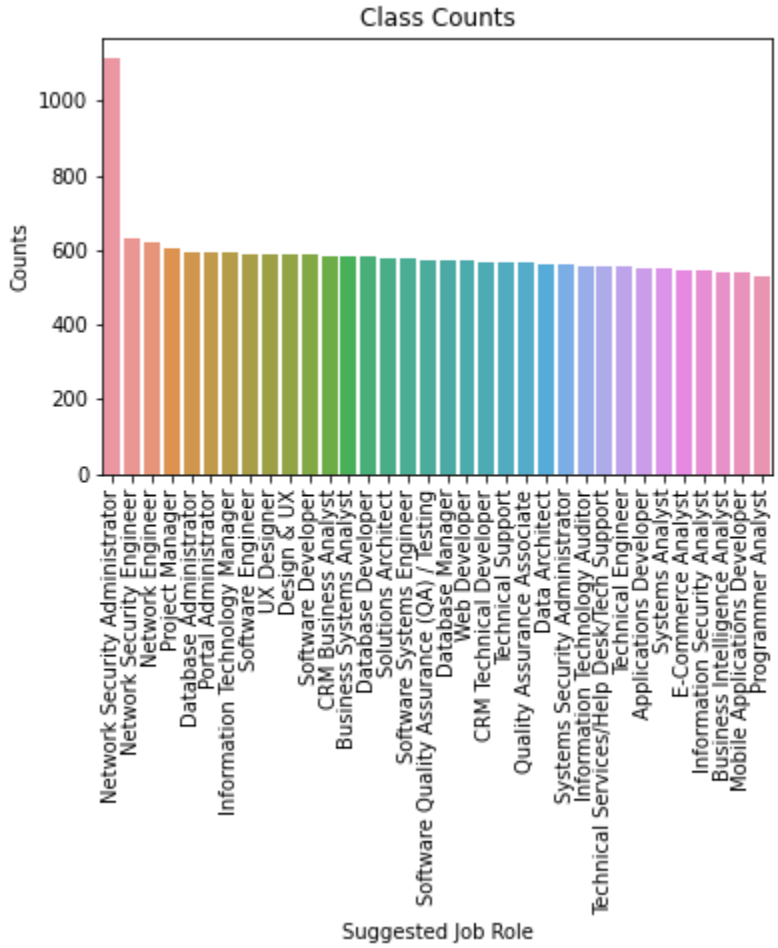
Exploratory Data Analysis

The data contains 20k rows, 38 input features and the output 'Suggested Job Role'.

1) Checking missing values in the dataset

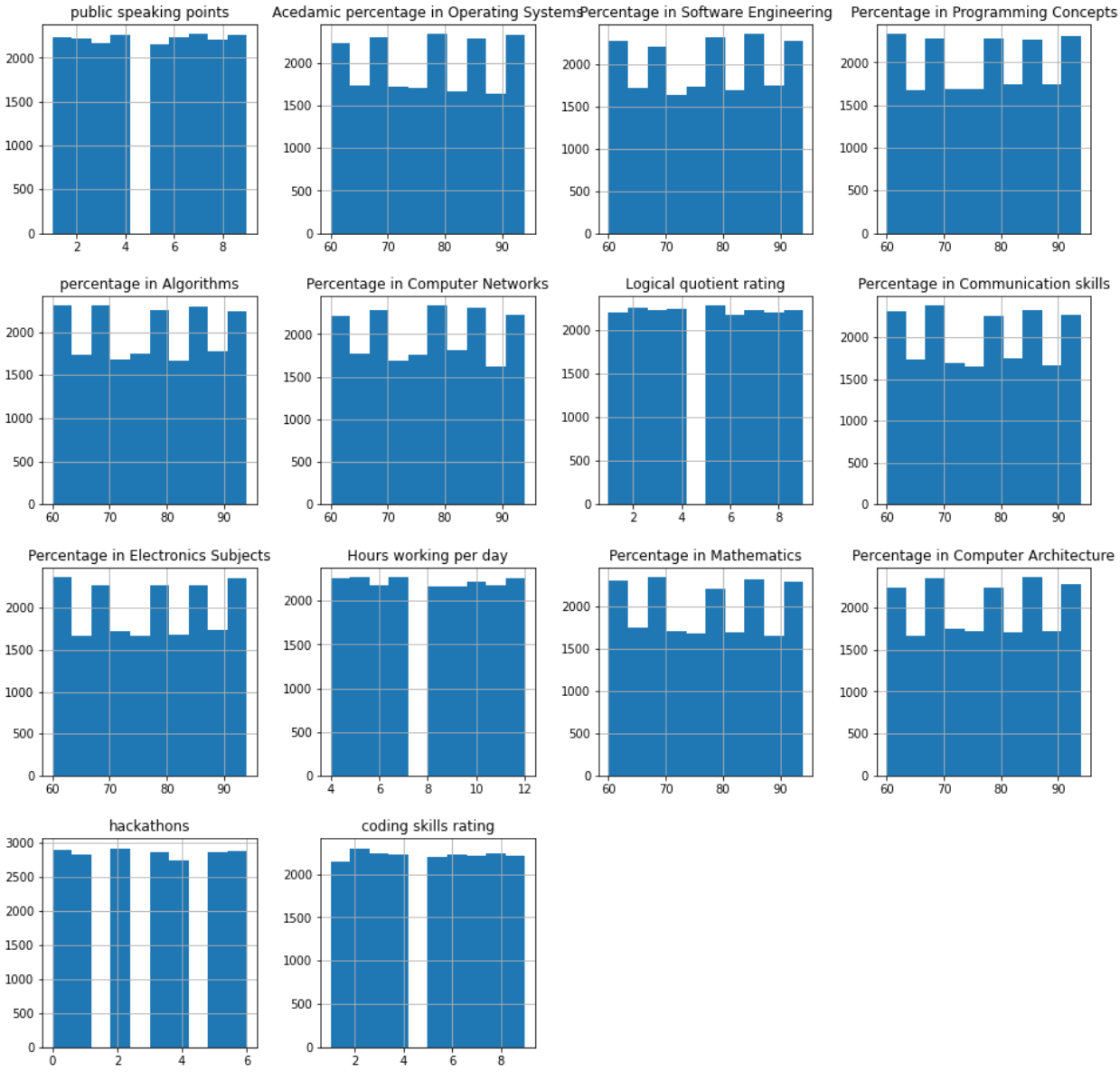
There were no missing values.

2) Plotting Class counts for suggested job role



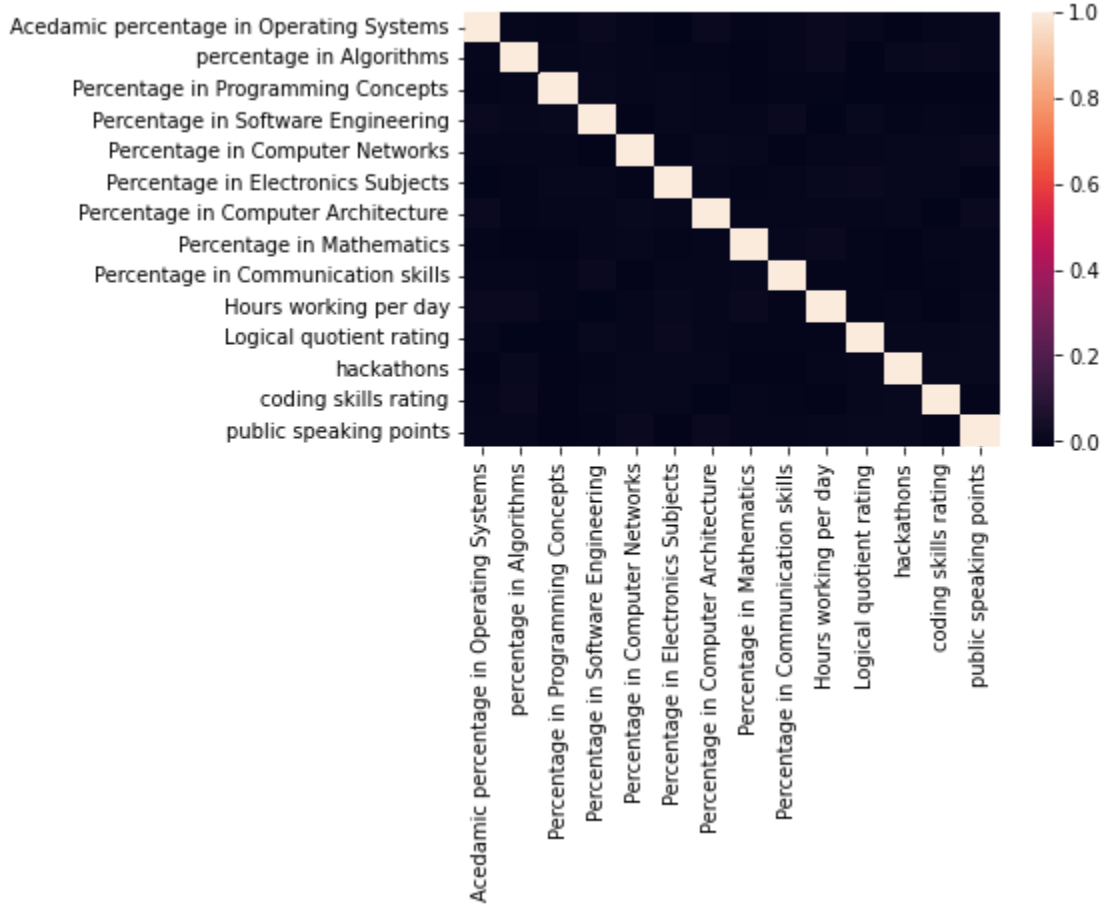
The figure above shows the class counts for the suggested job role. We observe that Network Security Administrator has a higher frequency.

3) Plotting histograms of numerical features



The figure above shows histograms of numerical features. It shows that the data is symmetrical.

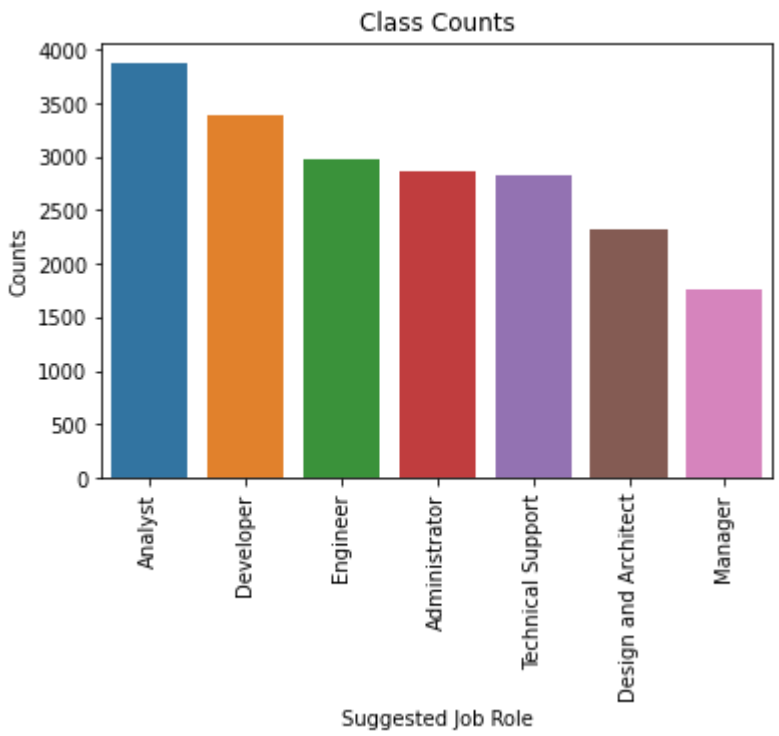
4) Correlation Heatmap



The figure above shows the correlation heatmap for the dataset. From this heatmap, we observe that each feature is dependent only on itself and is independent of the other features. Therefore we cannot drop them.

Preprocessing the data

1) Merging classes in output to reduce the number of classes



The classes were merged to 7 classes. The figure above shows the class counts for the merged classes.

2) Connecting the data to the electives advisory system from Assignment 1

The data was connected to the electives advisory system in Assignment 1. Some columns were renamed in the data, like Interested subjects and certifications, to make predictions based on the electives and interests from Assignment 1.

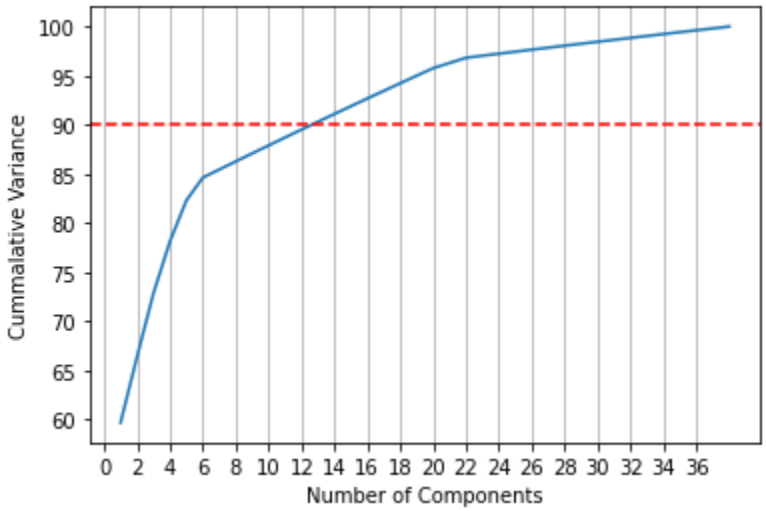
3) Feature scaling using standard scaler

The numerical features were standardized using standard scaler from sklearn.

4) Encoding Categorical Columns

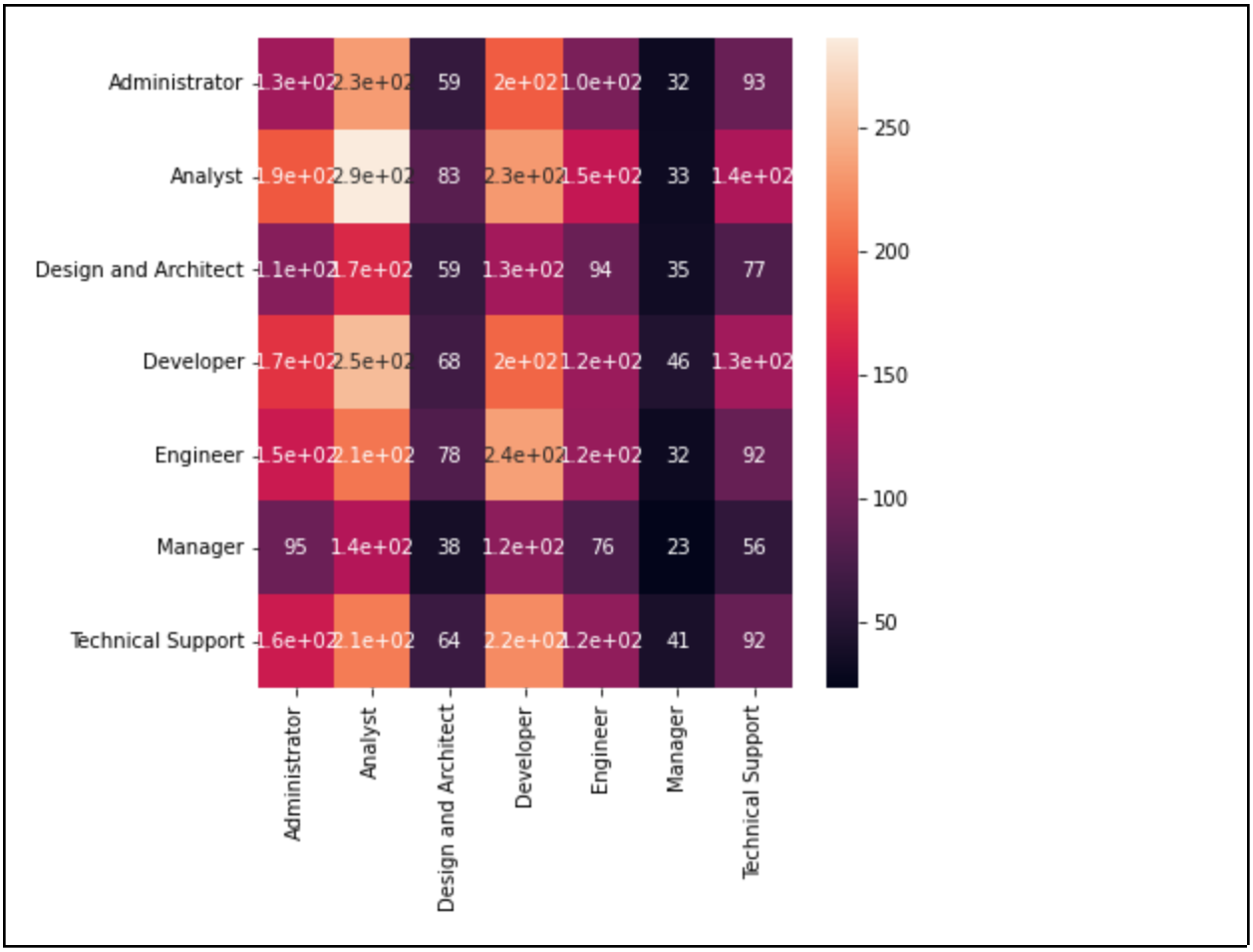
Categorical columns were encoded using label encoding.

5) Principal Component Analysis



PCA was done to reduce the number of components. The graph above shows the cumulative variance vs the number of components. With 13 components, we get a cumulative variance of 90 hence the value 13 is chosen. The metrics were computed on the data with and without pca and it was observed that better results are obtained in dataset without PCA. Hence the dataset without PCA was used for training the model. The following were the results on the dataset with PCA(13 components)-

```
Accuracy Score 0.15183333333333332
Class-wise Accuracies [0.15130024 0.25719424 0.08779762 0.20201005
0.13203463 0.04227941
 0.1014333 ]
Confusion Matrix
```



6) Bucketizing numerical columns into ranges

The numerical columns were bucketized into low, medium and high. However, the accuracy obtained from this was very low hence this was removed. I proceeded with Standard Scaler, feature scaling on the numerical features.

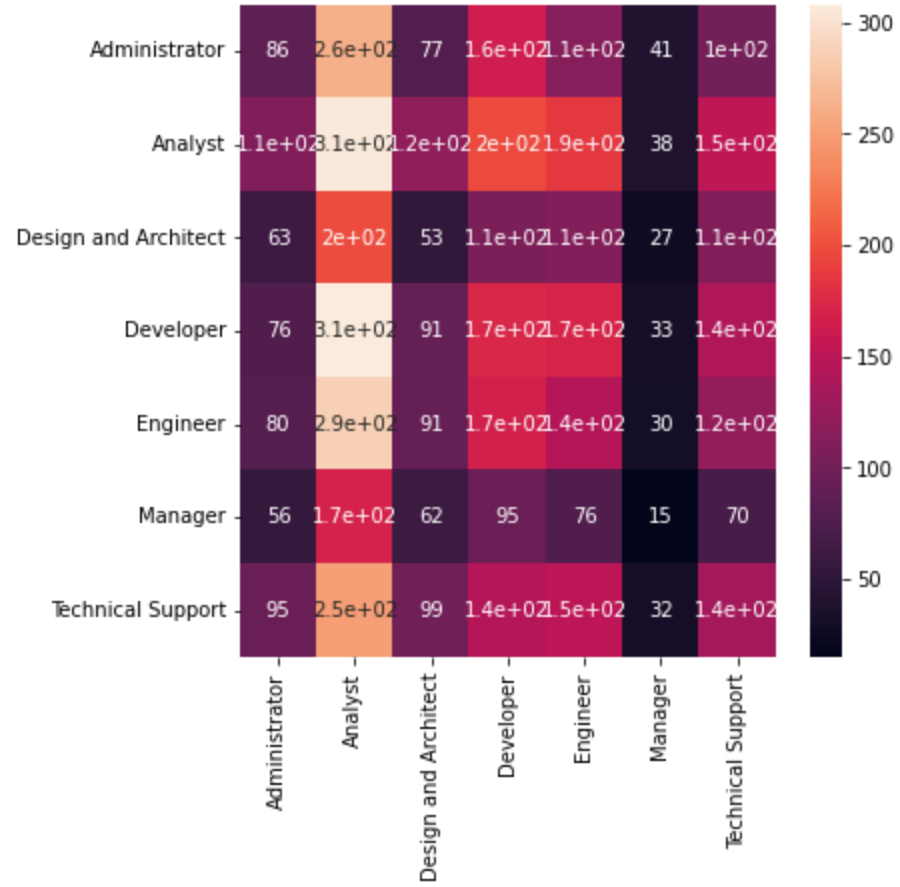
Results With Different Hyperparameters

The following results were obtained on sklearn MLPClassifier model with activation as 'tanh', max_iter=300,learning_rate_init=0.01 and other features as default.

```
Accuracy Score 0.15233333333333332
Class-wise Accuracies [0.10165485 0.27517986 0.07886905 0.17487437
0.15692641 0.02757353
```

```
 0.14884234]

Confusion Matrix
```

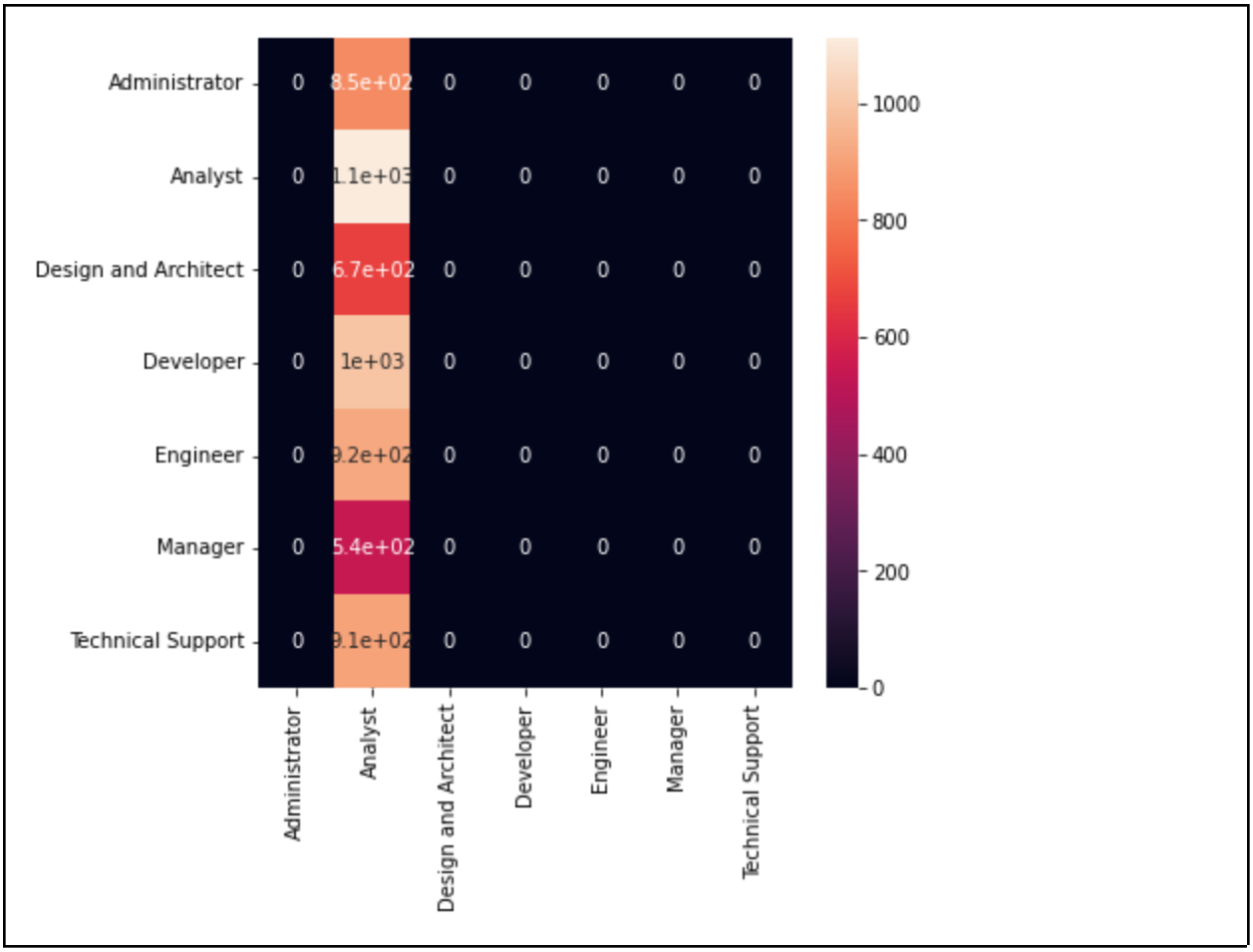


To improve the results I tried out different activation functions, hidden layer sizes and alpha values. The following results were obtained-

1) Trying Different Activation Functions

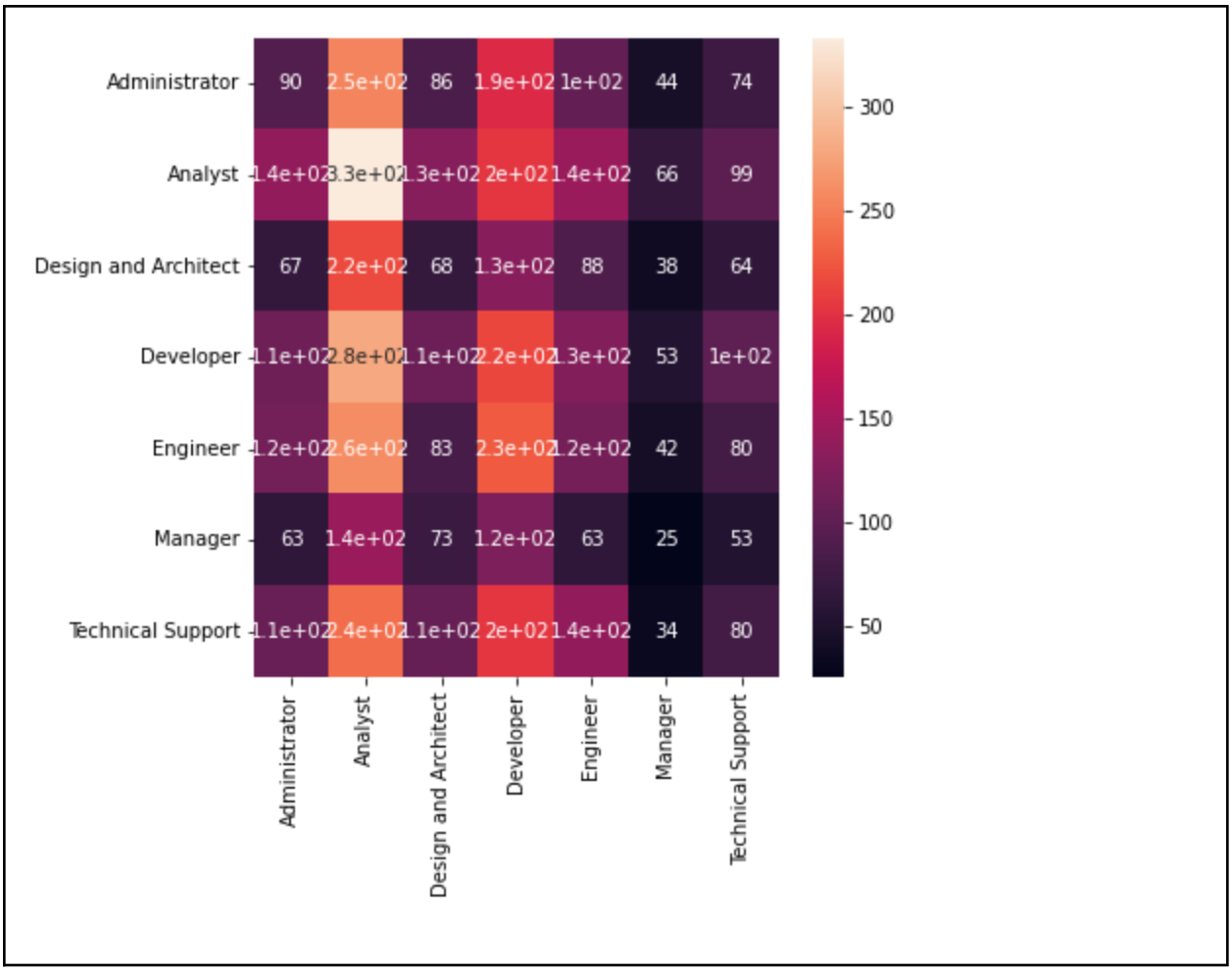
Identity

```
Accuracy Score 0.18533333333333332
Class-wise Accuracies [0. 1. 0. 0. 0. 0. 0.]
Confusion Matrix
```



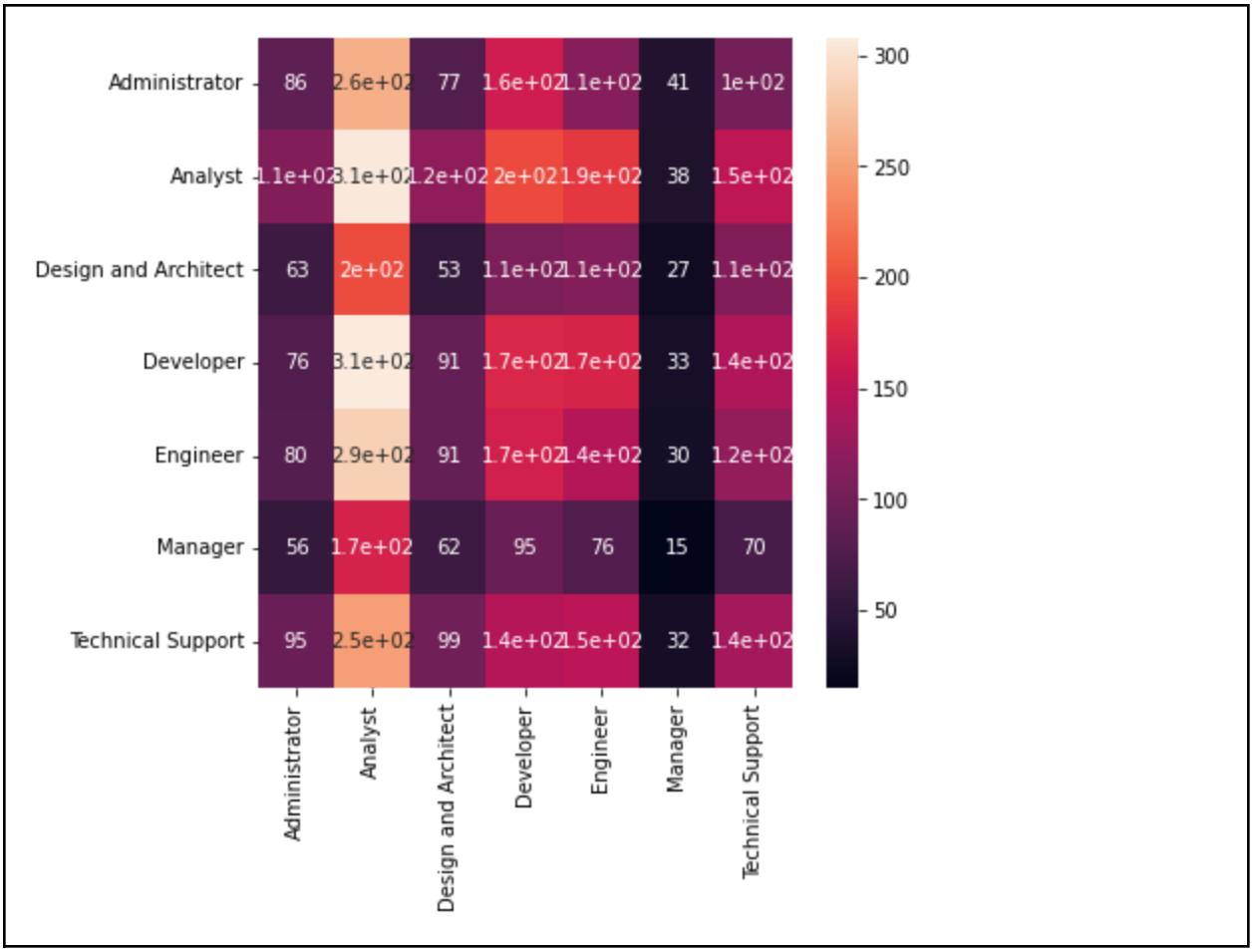
Logistic

```
ConvergenceWarning,
Accuracy Score 0.15466666666666667
Class-wise Accuracies [0.10638298 0.29946043 0.10119048 0.2160804
0.12662338 0.04595588
 0.08820287]
Confusion Matrix
```



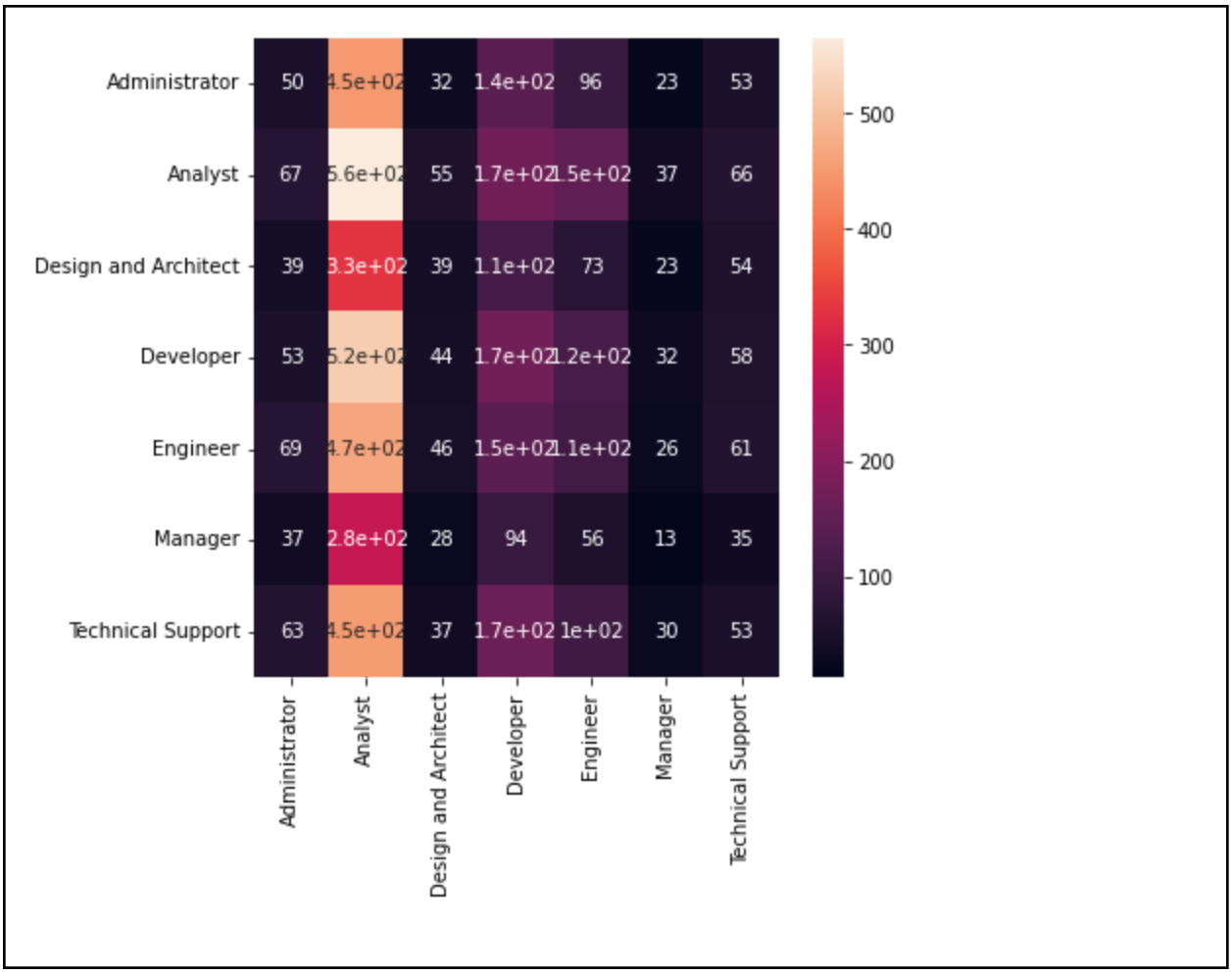
tanh

```
Accuracy Score 0.15233333333333332
Class-wise Accuracies [0.10165485 0.27517986 0.07886905 0.17487437
0.15692641 0.02757353
 0.14884234]
Confusion Matrix
```



Relu

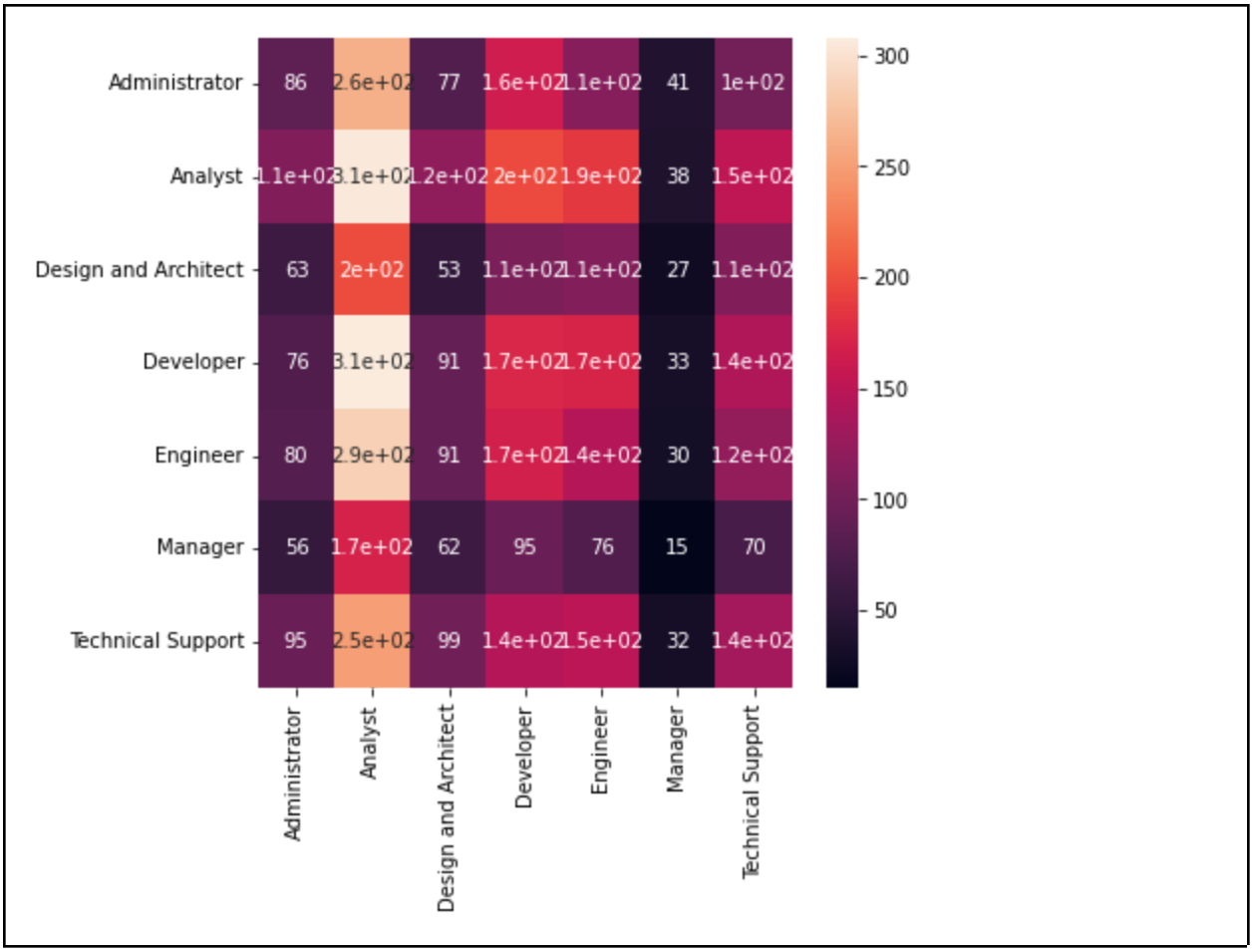
```
Accuracy Score 0.16666666666666666
Class-wise Accuracies [0.05910165 0.50809353 0.05803571 0.1718593
0.11796537 0.02389706
 0.0584344 ]
Confusion Matrix
```



2) Trying Different Hidden Layer Sizes

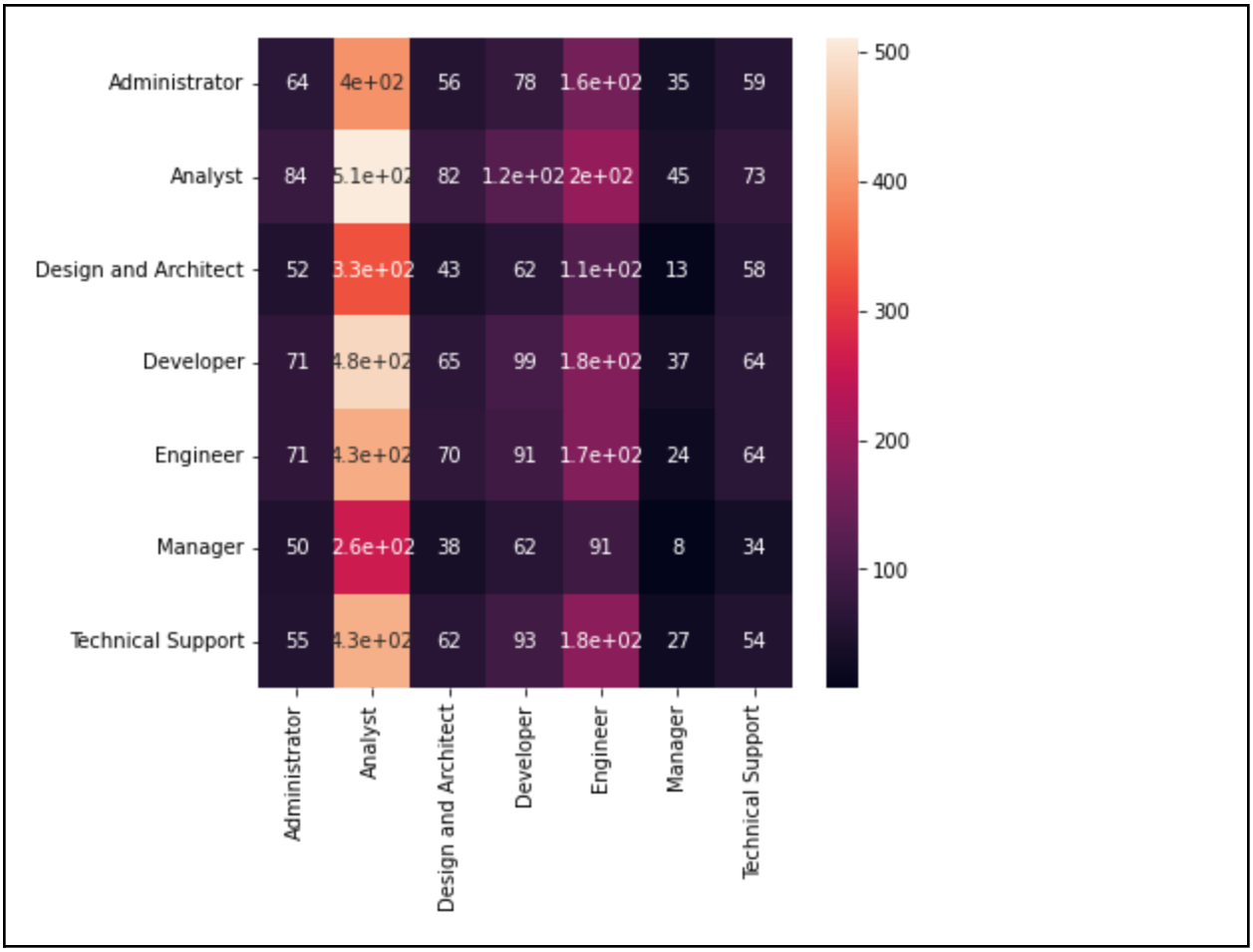
(100,)

```
Accuracy Score 0.15233333333333332
Class-wise Accuracies [0.10165485 0.27517986 0.07886905 0.17487437
0.15692641 0.02757353
 0.14884234]
Confusion Matrix
```



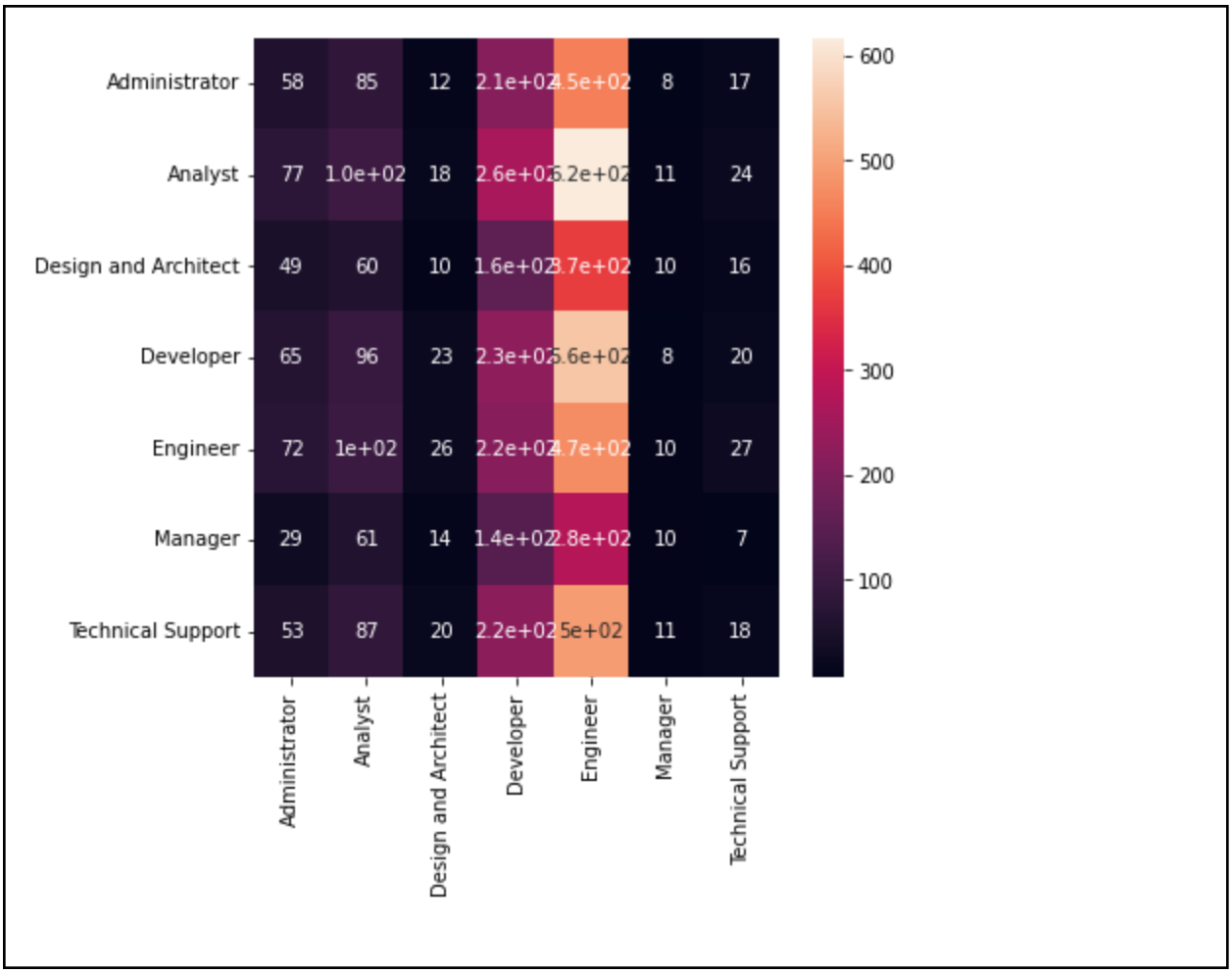
(100,50)

```
Accuracy Score 0.15883333333333333
Class-wise Accuracies [0.07565012 0.45953237 0.0639881  0.09949749
0.18831169 0.01470588
 0.05953693]
Confusion Matrix
```



(100,50,20)

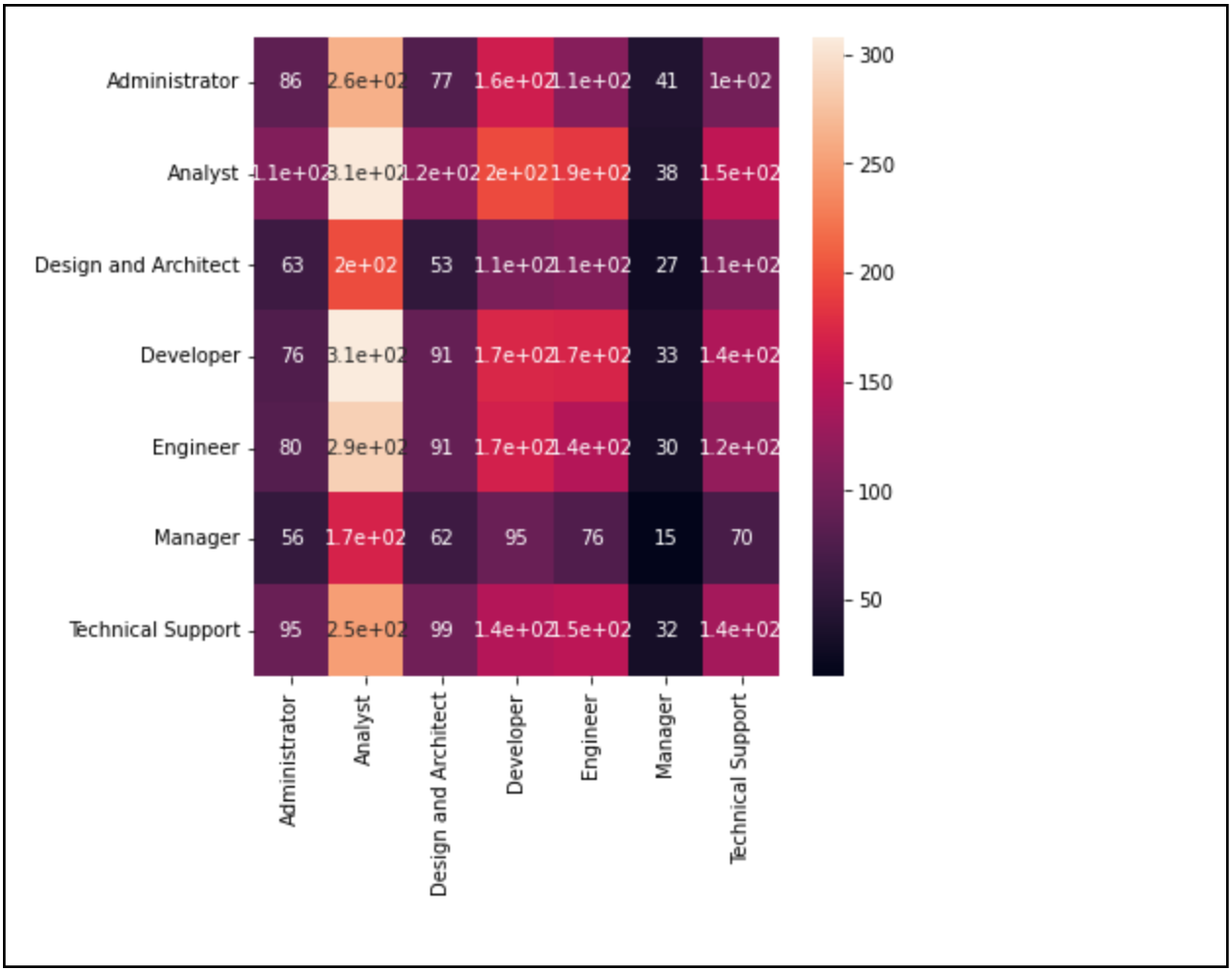
```
Accuracy Score 0.14983333333333335
Class-wise Accuracies [0.06855792 0.09442446 0.01488095 0.22713568
0.51082251 0.01838235
 0.01984564]
Confusion Matrix
```



3) Trying Different Alpha Values

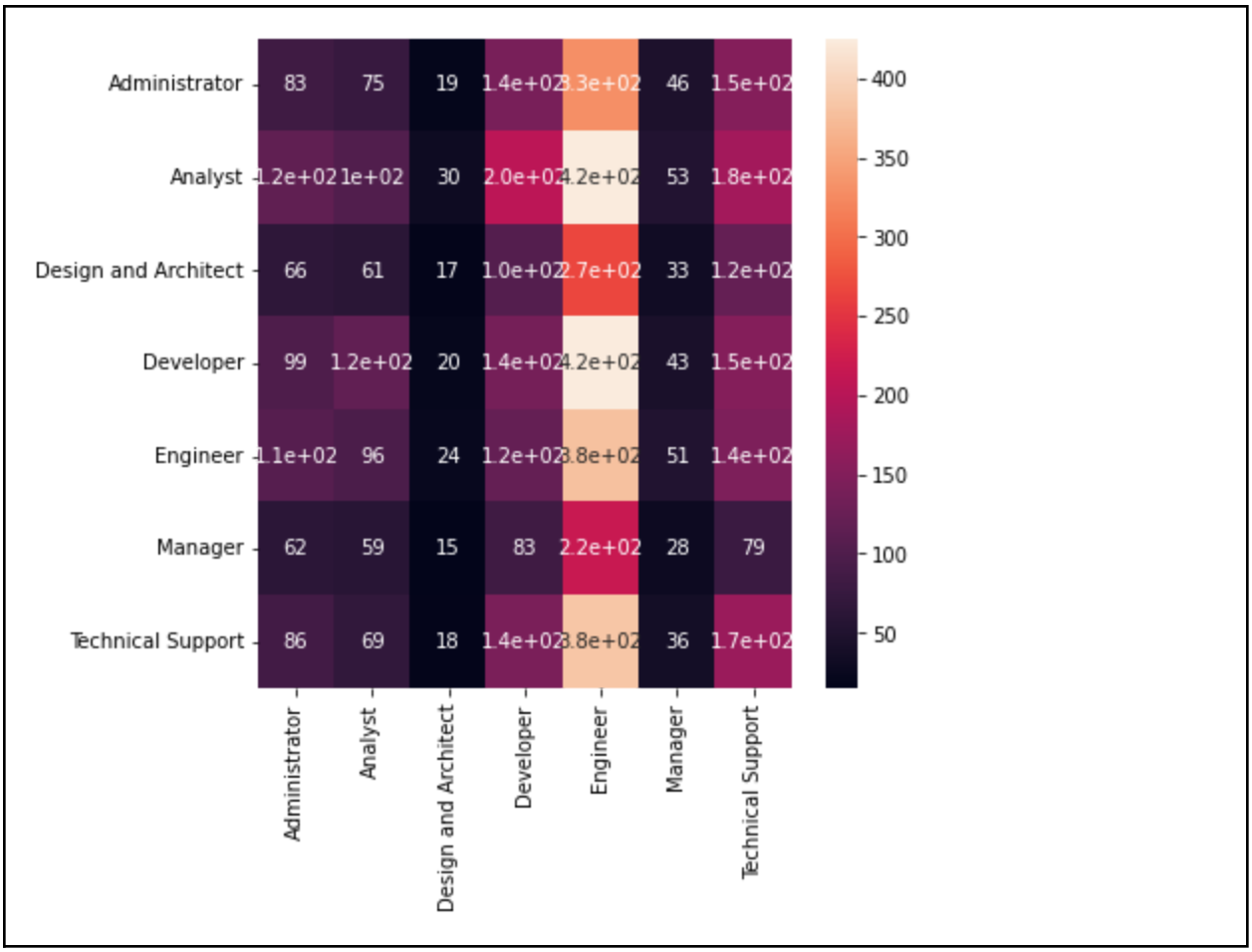
0.001

```
Accuracy Score 0.15233333333333332
Class-wise Accuracies [0.10165485 0.27517986 0.07886905 0.17487437
0.15692641 0.02757353
 0.14884234]
Confusion Matrix
```



0.05

```
Accuracy Score 0.15333333333333332
Class-wise Accuracies [0.09810875 0.0926259  0.02529762 0.13768844
0.41125541 0.05147059
 0.18963616]
Confusion Matrix
```



Best Hyperparameters

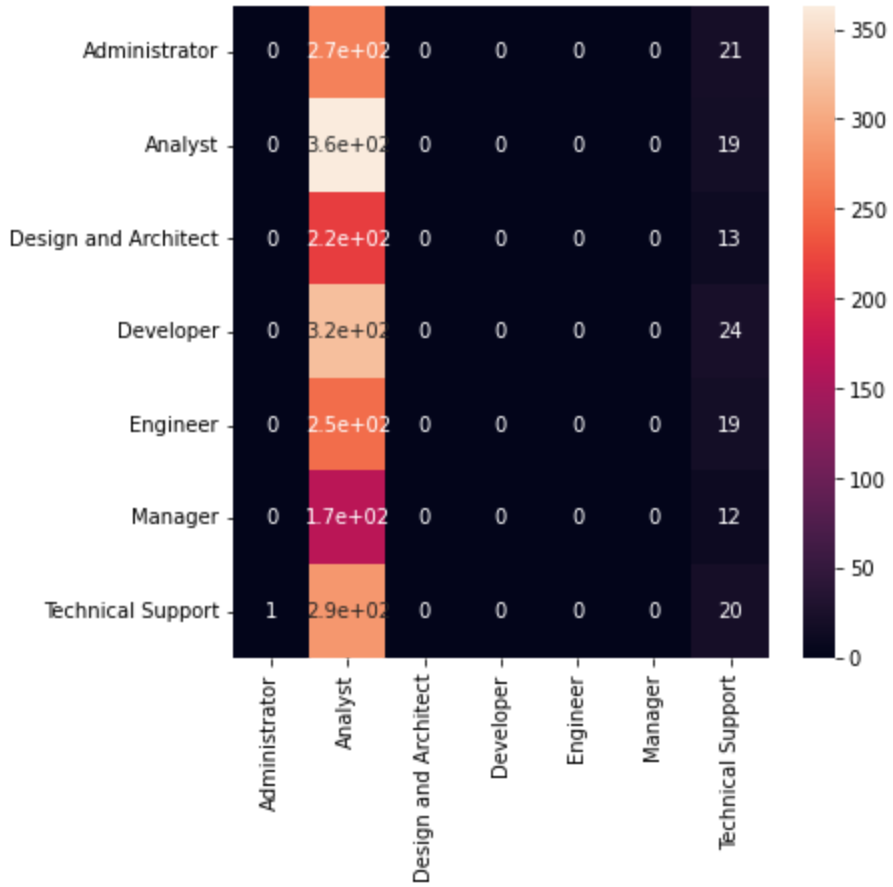
With alpha=0.05, hidden_layer_sizes = (100,50), activation='tanh' , max_iter = 300, learning rate as 0.01.

Data Modifications

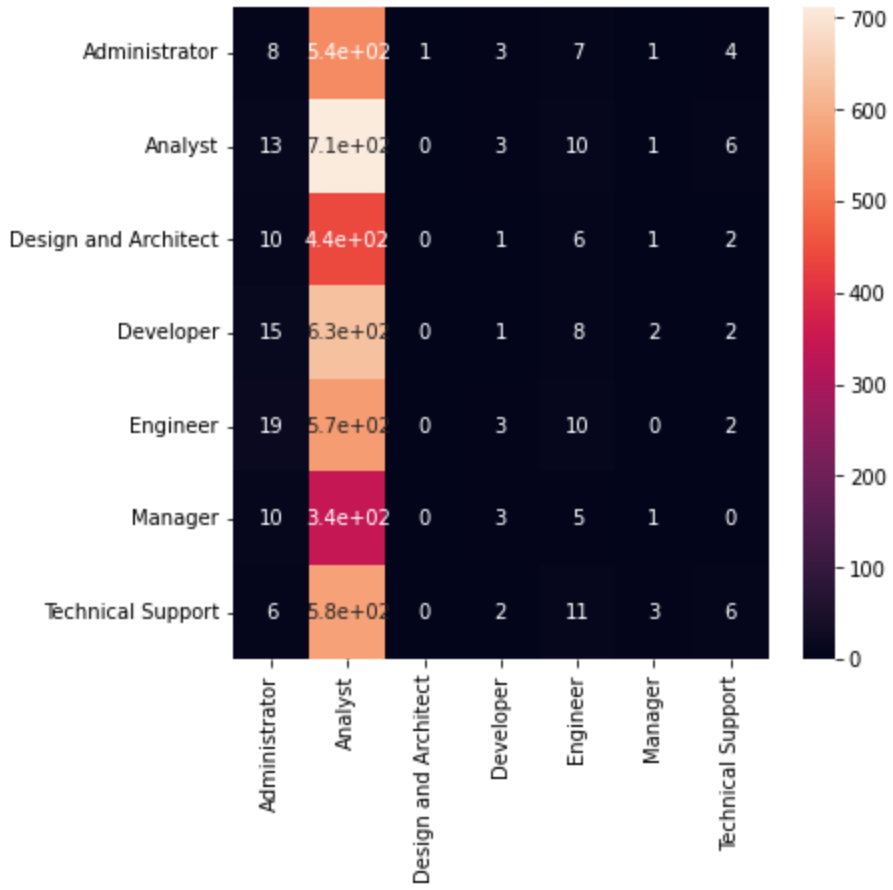
1) Trying Different Ratios of Train-test split

Different train-test split ratios were tried. The below table shows the results on the ratios 10-90, 20-80, 30-70 and 40-60. Looking at the classwise accuracies the best results were obtained in the 30-70 split ratio though the highest accuracy is obtained in 10-90 split ratio.

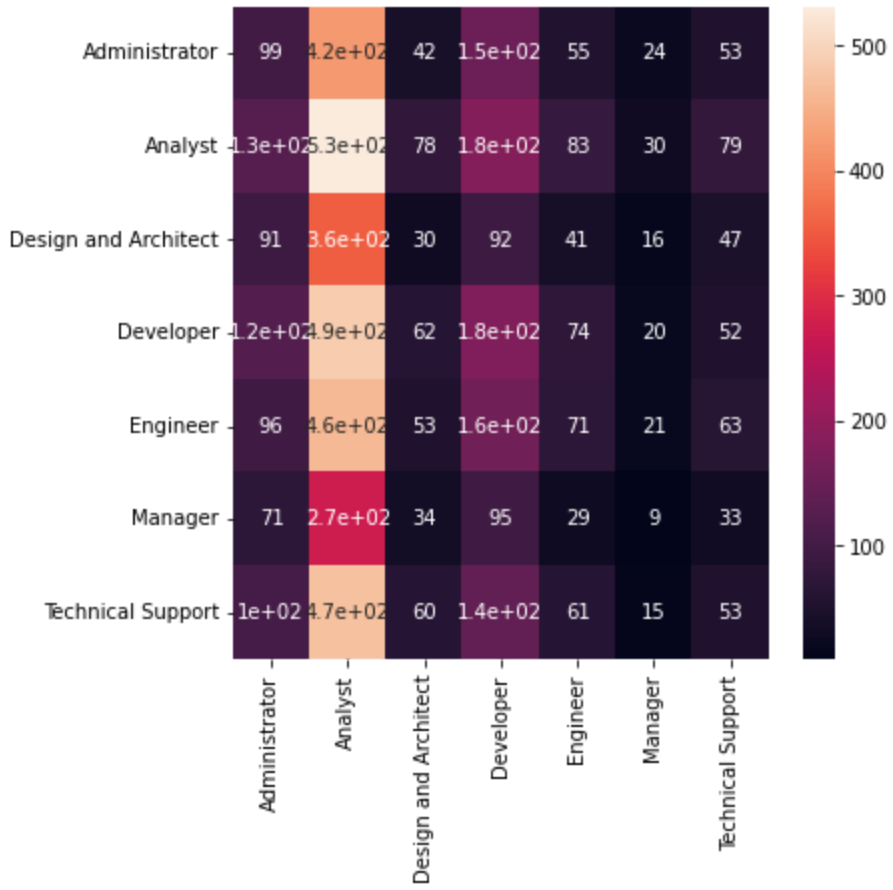
```
Accuracy Score 0.1915
Class-wise Accuracies [0.         0.95026178 0.         0.         0.
0.
 0.06514658]
Confusion Matrix
```



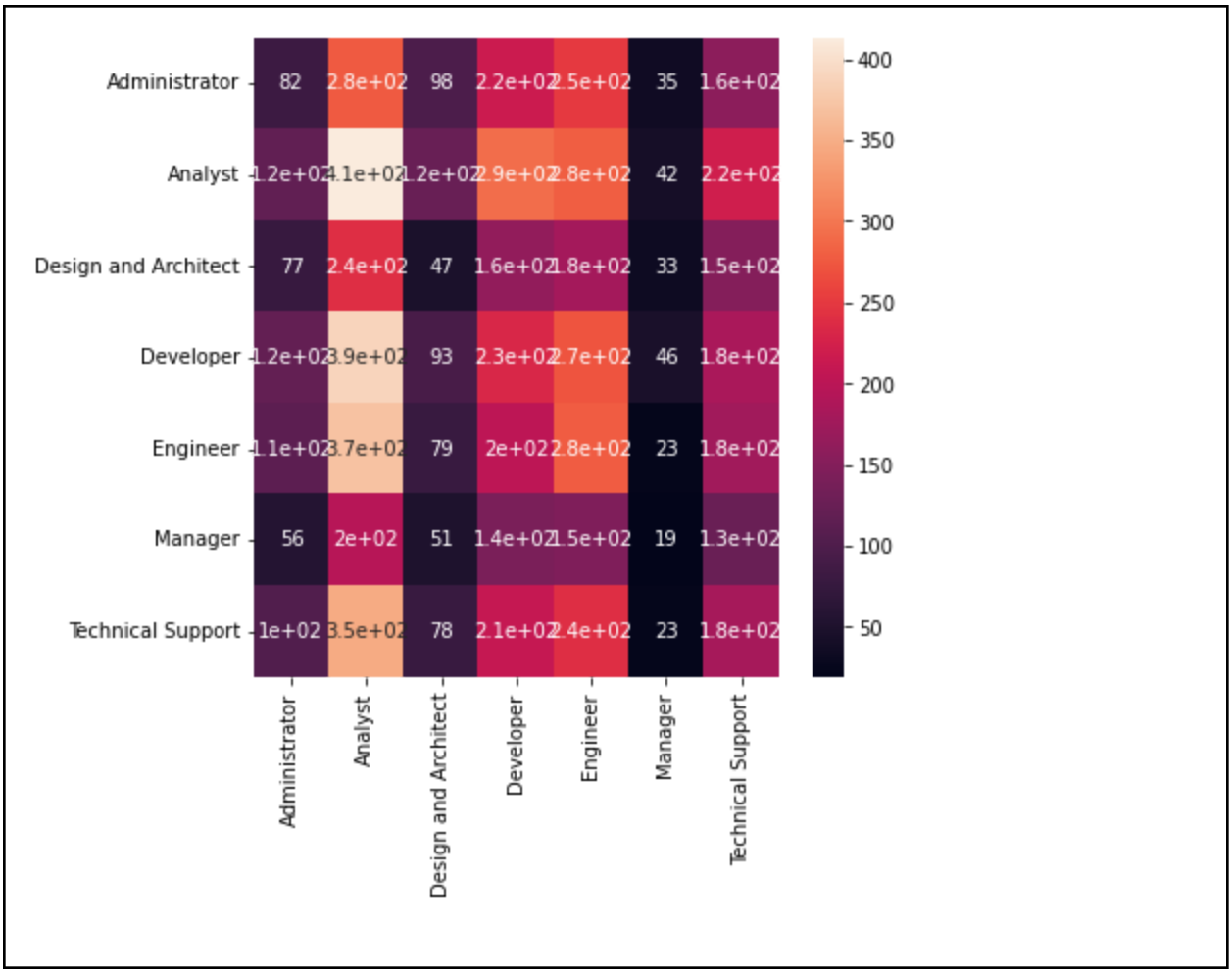

```
Accuracy Score 0.1845
Class-wise Accuracies [0.0141844  0.9557047  0.         0.00151286
0.01658375 0.00276243
 0.00991736]
Confusion Matrix
```




```
Accuracy Score 0.16183333333333333
Class-wise Accuracies [0.11702128 0.47751799 0.04464286 0.17889447
0.07683983 0.01654412
 0.0584344 ]
Confusion Matrix
```



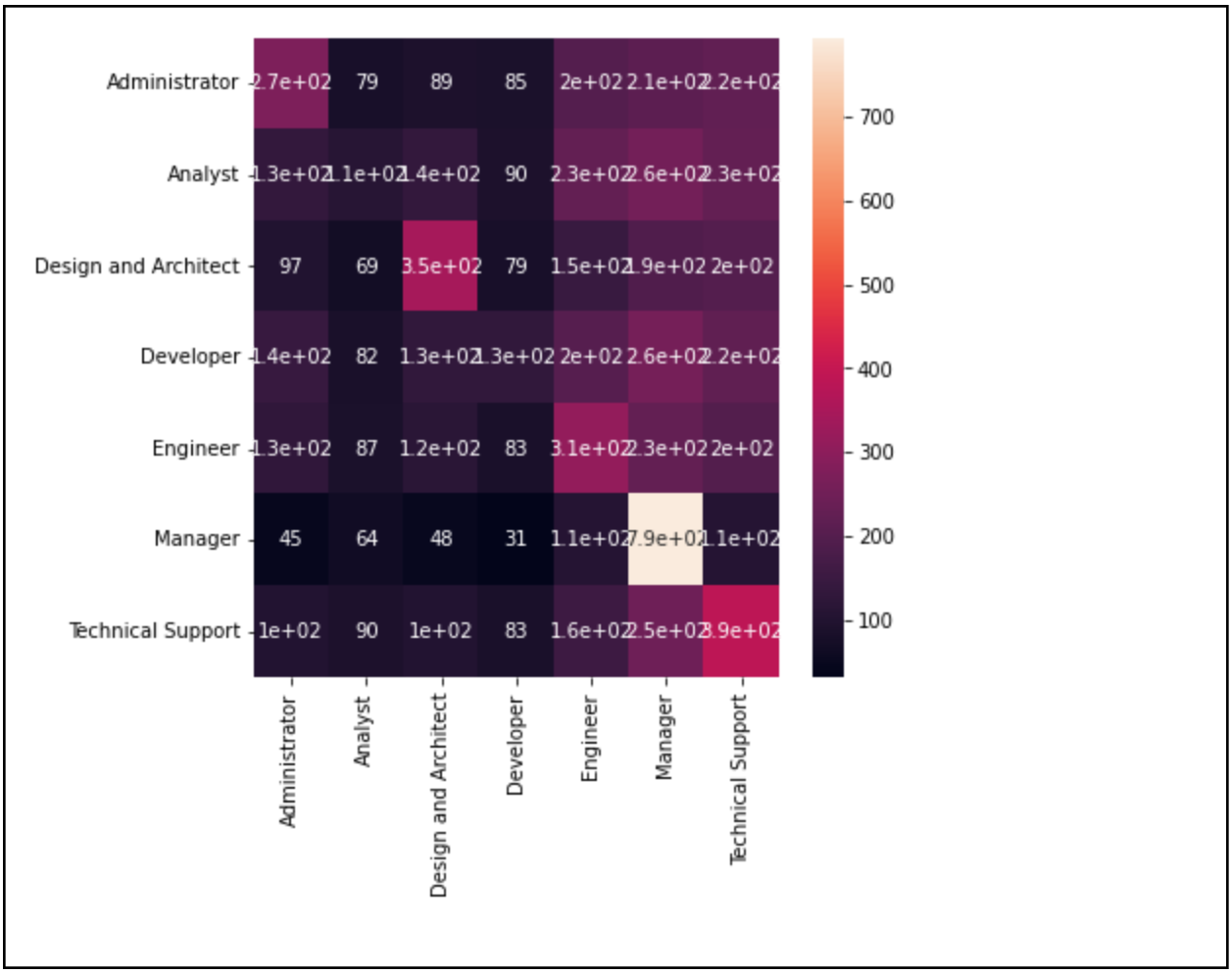

```
Accuracy Score 0.15675
Class-wise Accuracies [0.07341092 0.27680965 0.05280899 0.17352281
0.22409639 0.02585034
 0.15371622]
Confusion Matrix
```



2) Oversampling

The following results were obtained on data with oversampling and the best accuracy of **28.9 %**. So this was chosen as the final model with hyperparameters the same as best hyperparameters and trained on oversampled data.

```
Accuracy Score 0.2892084562438545
Class-wise Accuracies [0.23514212 0.09268707 0.30755556 0.11320755
0.27217391 0.66499162
 0.33161512]
Confusion Matrix
```


Administrator
Analyst
Design and Architect
Developer
Engineer
Manager
Technical Support
2.7e+02 79 89 85 2e+02 2.1e+02 2.2e+02
1.3e+02 1.1e+02 1.4e+02 90 2.3e+02 2.6e+02 2.3e+02
97 69 3.5e+02 79 1.5e+02 1.9e+02 2e+02
1.4e+02 82 1.3e+02 1.3e+02 2e+02 2.6e+02 2.2e+02
1.3e+02 87 1.2e+02 83 3.1e+02 2.3e+02 2e+02
45 64 48 31 1.1e+02 7.9e+02 1.1e+02
1e+02 90 1e+02 83 1.6e+02 2.5e+02 3.9e+02
Administrator
Analyst
Design and Architect
Developer
Engineer
Manager
Technical Support
700
600
500
400
300
200
100