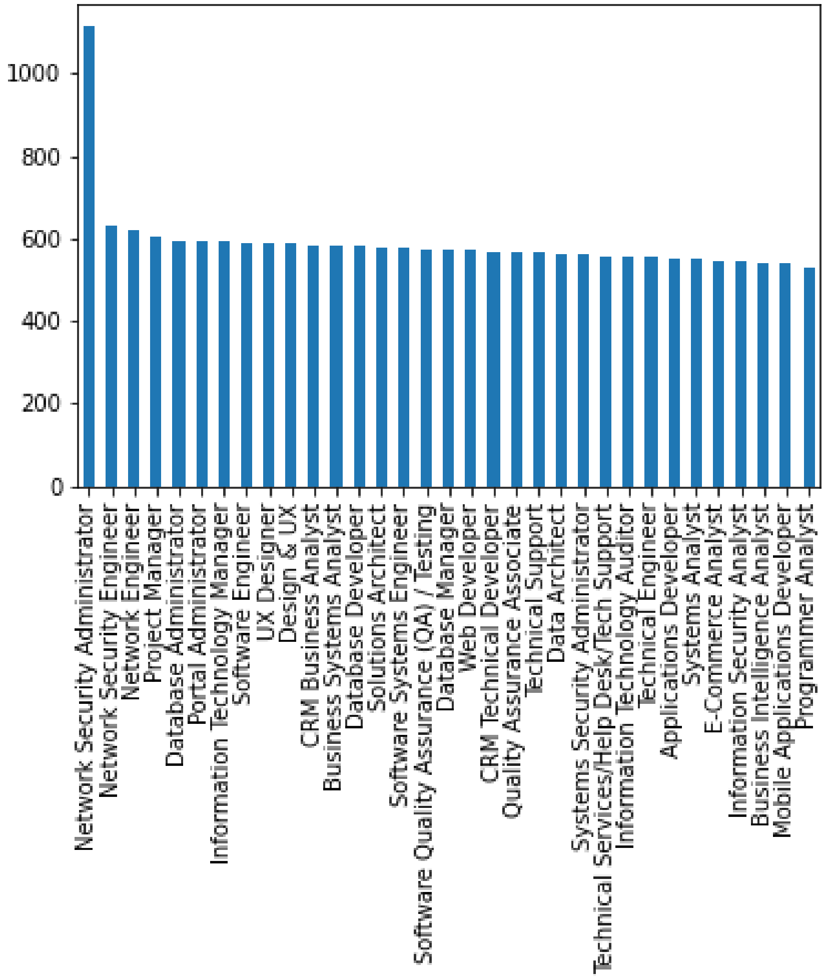
Assignment 4

Manvi Goel

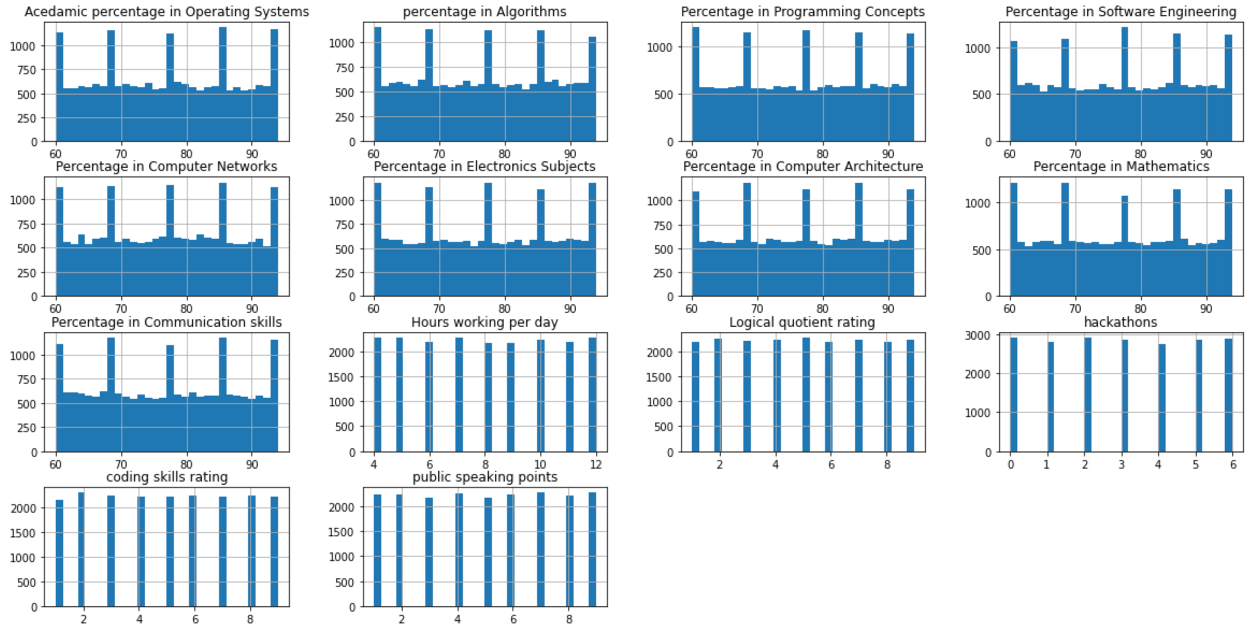
Preprocessing of Data

1. *Class Count*. Getting the class count to see if the data is skewed



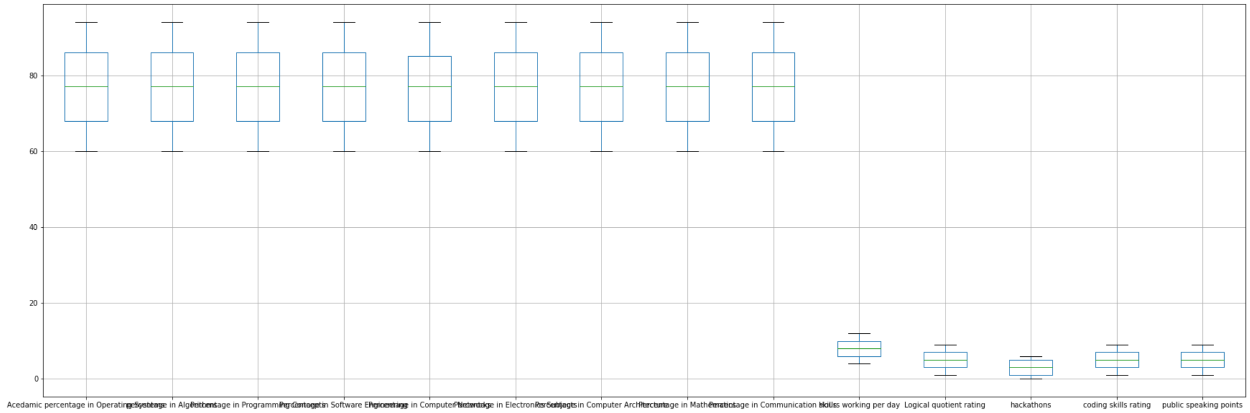
We see the samples for *Network Security Administrator* are almost double the other class samples. But the numbers of samples for the other classes are comparable.

2. *Histogram*. To check if any of the features are skewed.



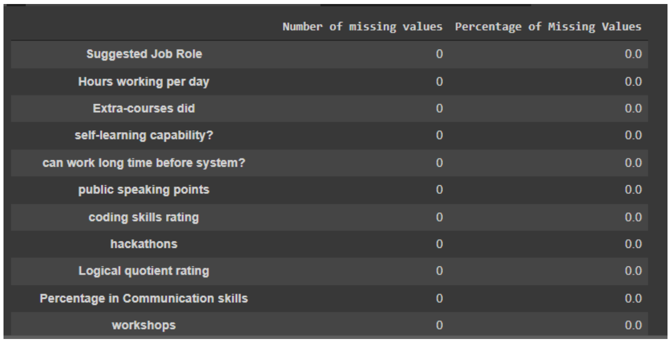
The data is not skewed.

3. *Boxplots*.



4. *Checking Missing Data*

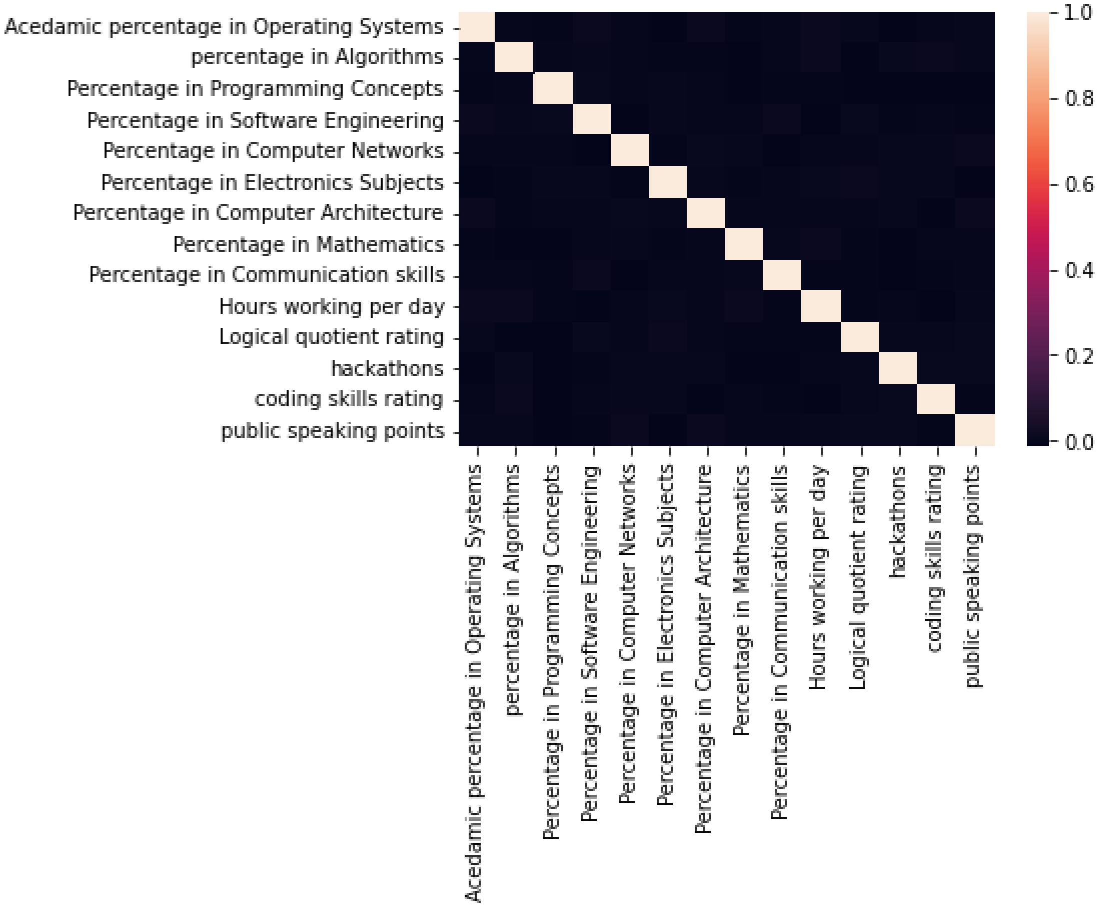
No Missing Data



	Number of missing values	Percentage of Missing Values
Suggested Job Role	0	0.0
Hours working per day	0	0.0
Extra-courses did	0	0.0
self-learning capability?	0	0.0
can work long time before system?	0	0.0
public speaking points	0	0.0
coding skills rating	0	0.0
hackathons	0	0.0
Logical quotient rating	0	0.0
Percentage in Communication skills	0	0.0
workshops	0	0.0

Feature Engineering.

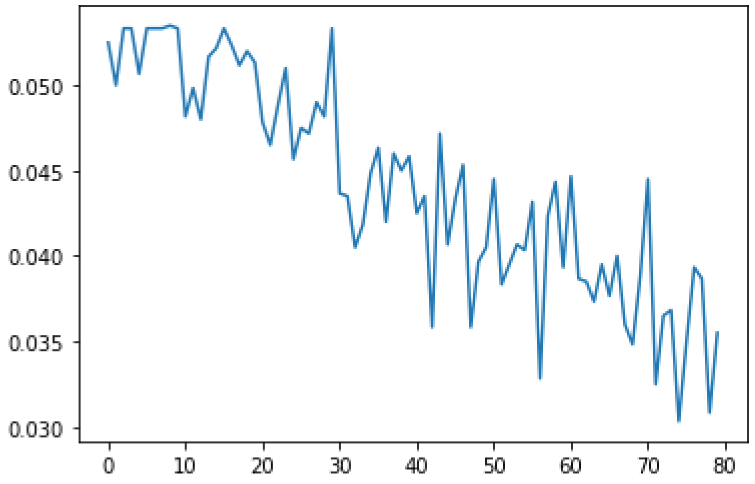
1. *Covariance Heatmap.*



Plot for the covariance between the features. Since all features are nearly independent of each other, we do not need to remove features based on this.

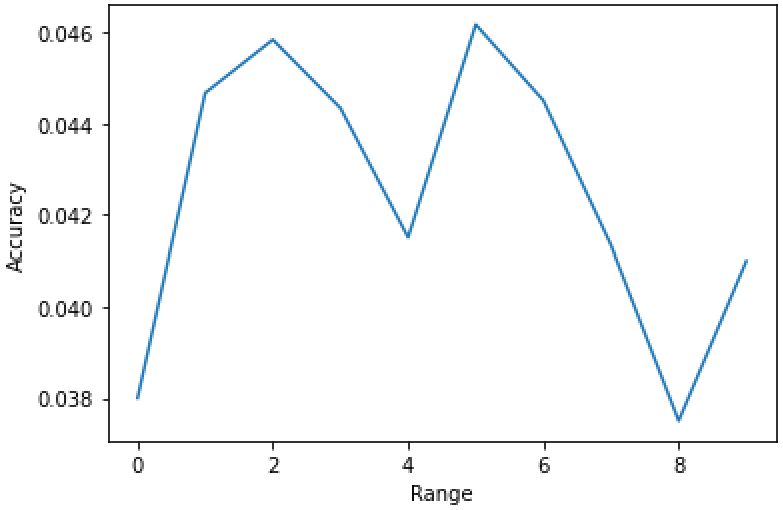
2. *Random Features Selection.*

   To get an idea of features that need to be chosen. I randomly select some features and train a simple Neural Network on the same.



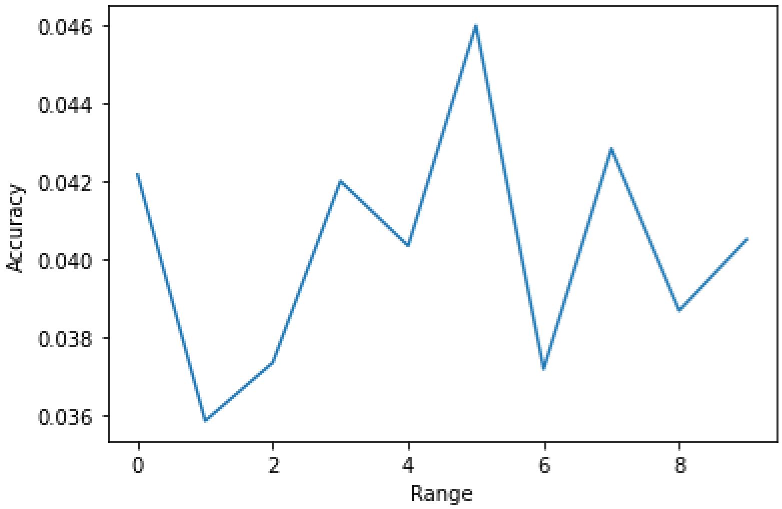
Observation. There is a general trend of accuracy decreasing as the number of features is increased from 2 to 10. But even with limited features, the accuracy is extremely low.

3. Feature Selection using Sklearn: *F_classification*.
   Selecting 15 features.



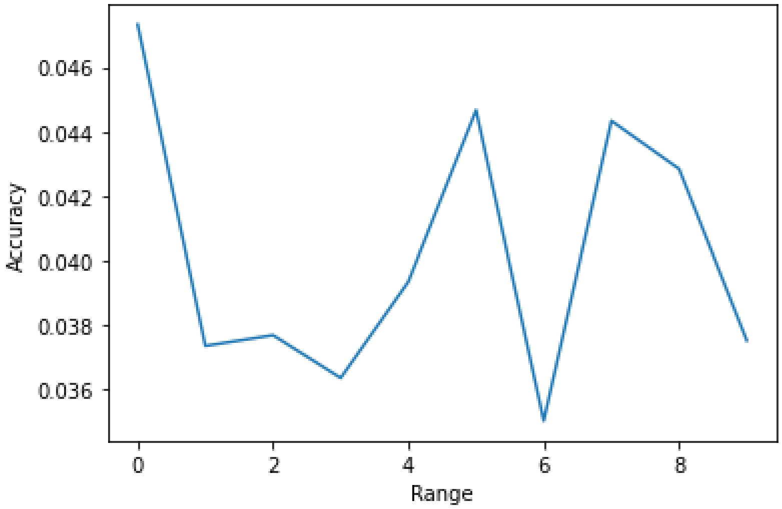
The average accuracy of the model is 0.04248333333333333

Selecting 20 Features.



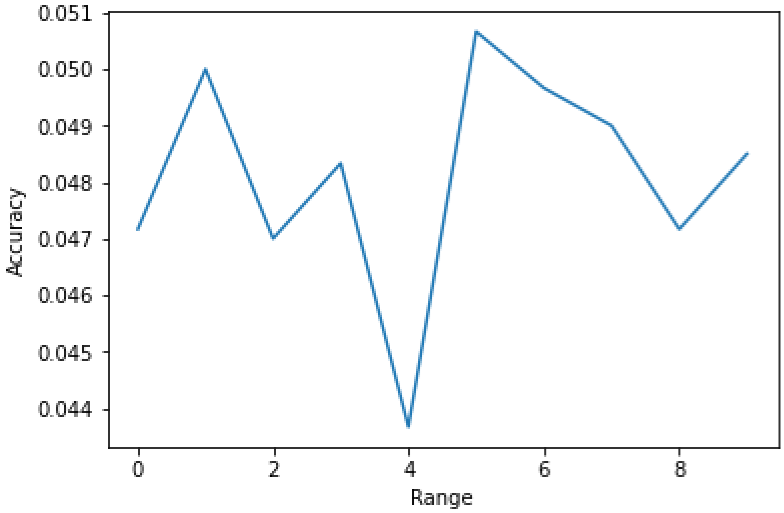
The average accuracy of the model is 0.04028333333333333

Selecting 30 Features.



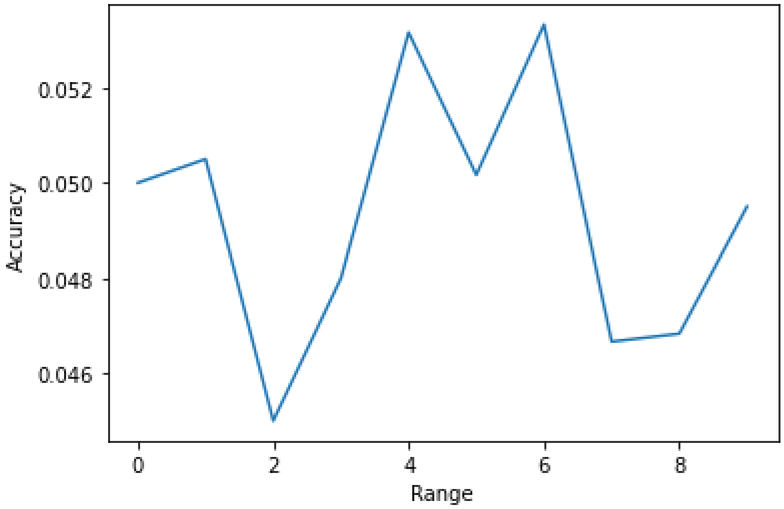
The average accuracy of the model is 0.04023333333333333
We see that 15 features give the best accuracy.

4. Feature Selection using Sklearn: *chi2*.
   Selecting 10 features.



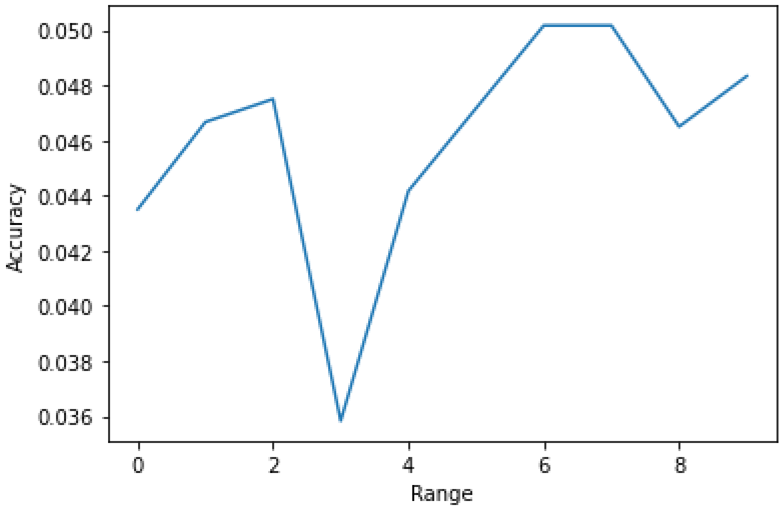
The average accuracy of the model is 0.04811666666666667

Selecting 15 Features.



The average accuracy of the model is 0.04931666666666667

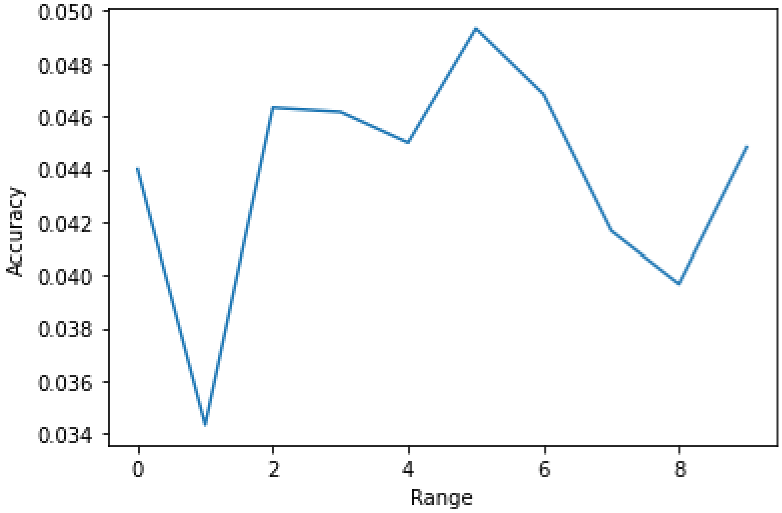
Selecting 20 Features.



The average accuracy of the model is 0.046

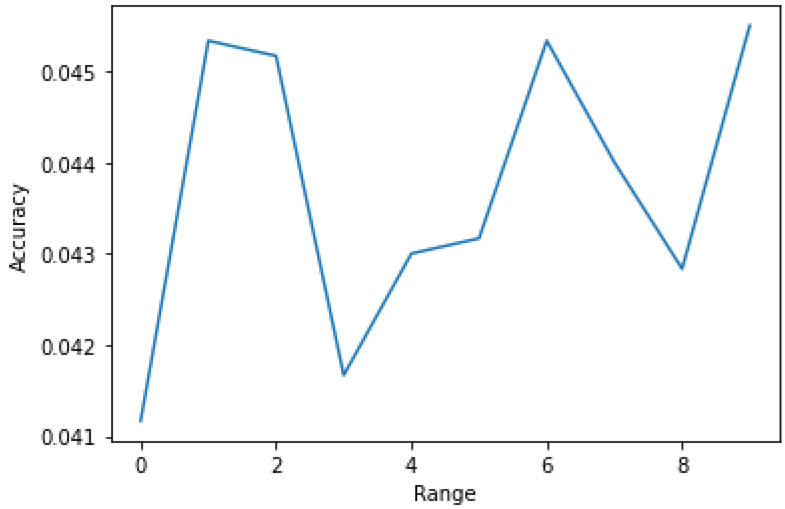
We see that 15 features give the best accuracy.

5. *Mutual Information.*
Selecting 10 Features.



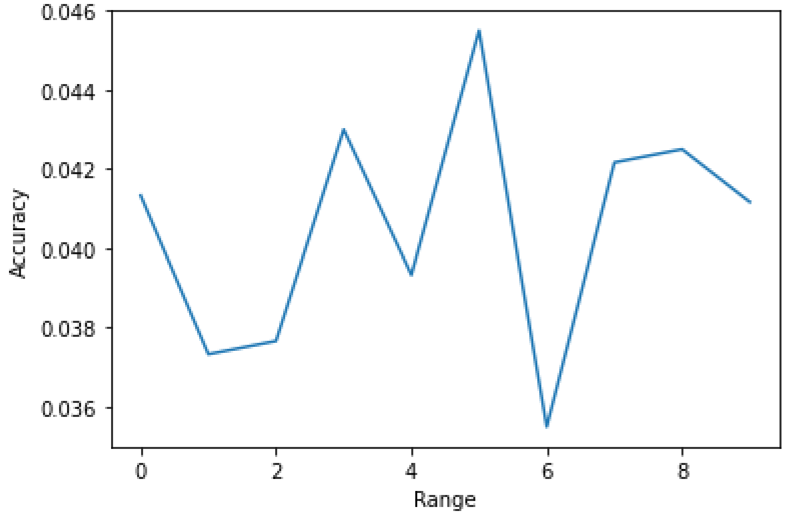
The average accuracy of the model is 0.04381666666666667

Selecting 15 Features.



The average accuracy of the model is 0.04371666666666667

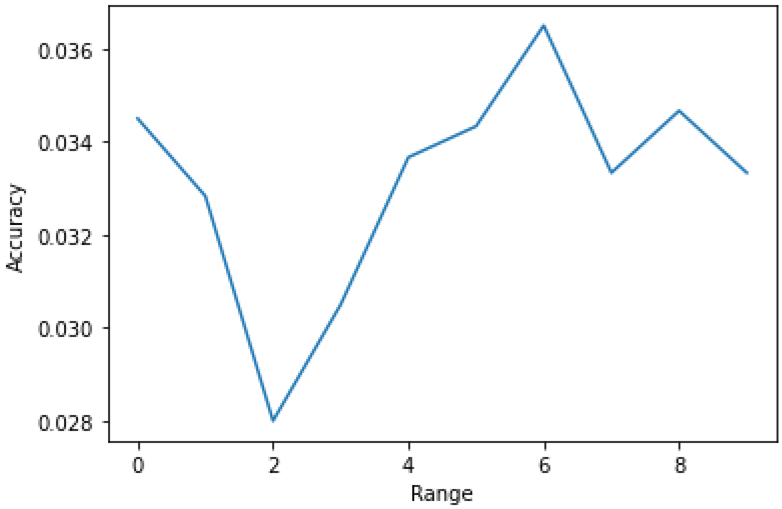
Selecting 20 Features.



The average accuracy of the model is 0.040549999999999996

The best accuracy is given by 15 Features.

6. *Feature Selection using Domain Knowledge and Random Predictions.*
   Columns removed are 'Percentage in Computer Architecture,' 'Interested Type of Books,' 'memory capability score,' 'Percentage in Communication skills,' 'workshops,' 'Percentage in Software Engineering,' 'worked in teams ever?', 'public speaking points' and certifications

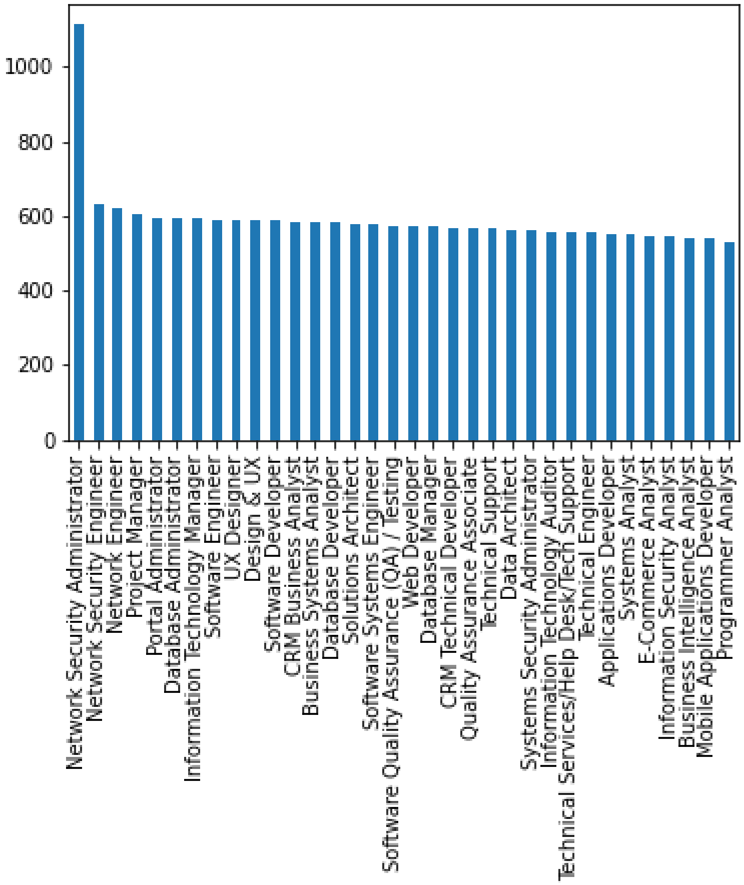


We see that the best accuracy is given by selecting 15 features by the chi2 method. We will use the same in the following sections.

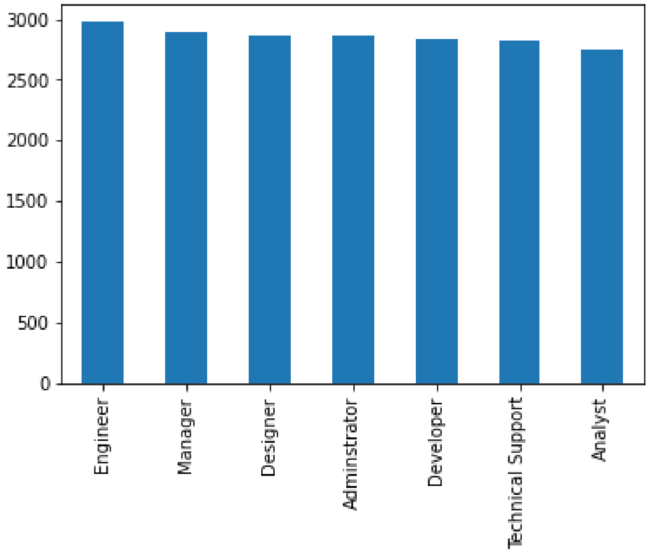
Class Reduction.

We are merging labels by using domain knowledge. We also ensure that the classes are not imbalanced.

Classes before merging.



Classes after merging.

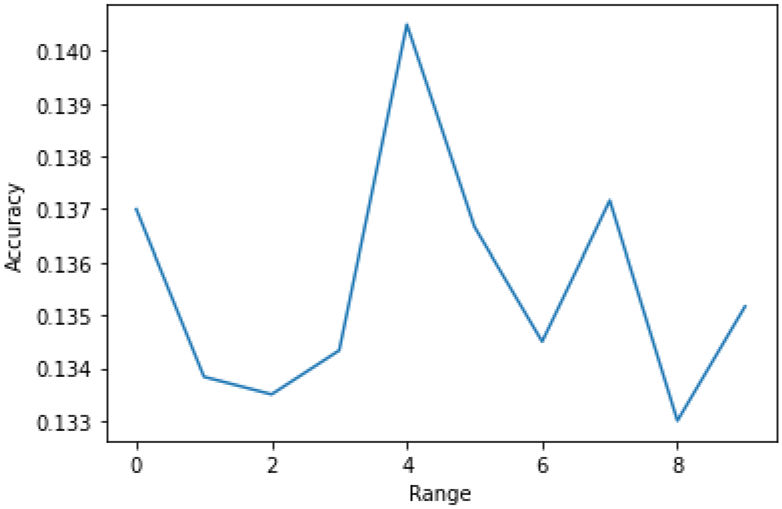


Model

Library Used: Sklearn.

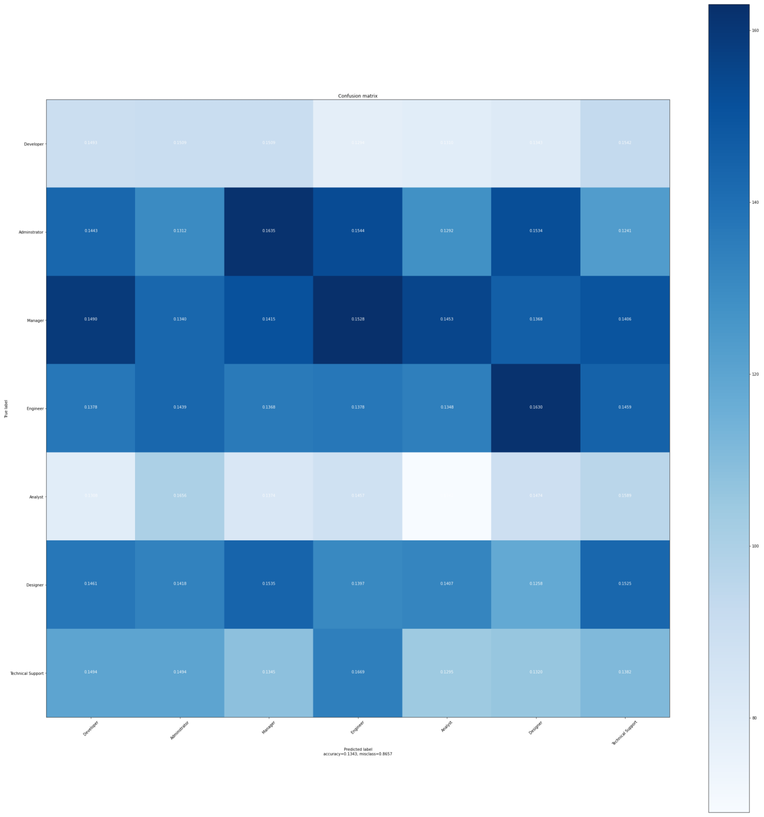
Model Used: Multi-Layer Perceptron or MLPClassifier

1. *Simple ANN Classifier*.

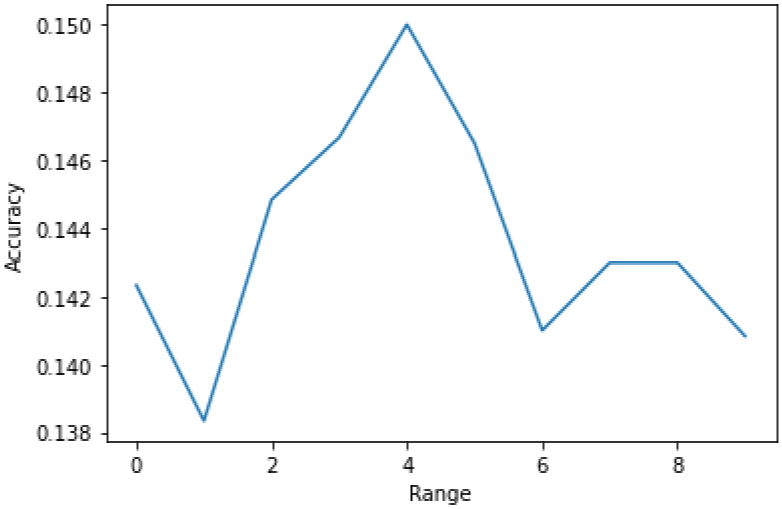


The average accuracy of the model is 0.13556666666666667

Confusion matrix

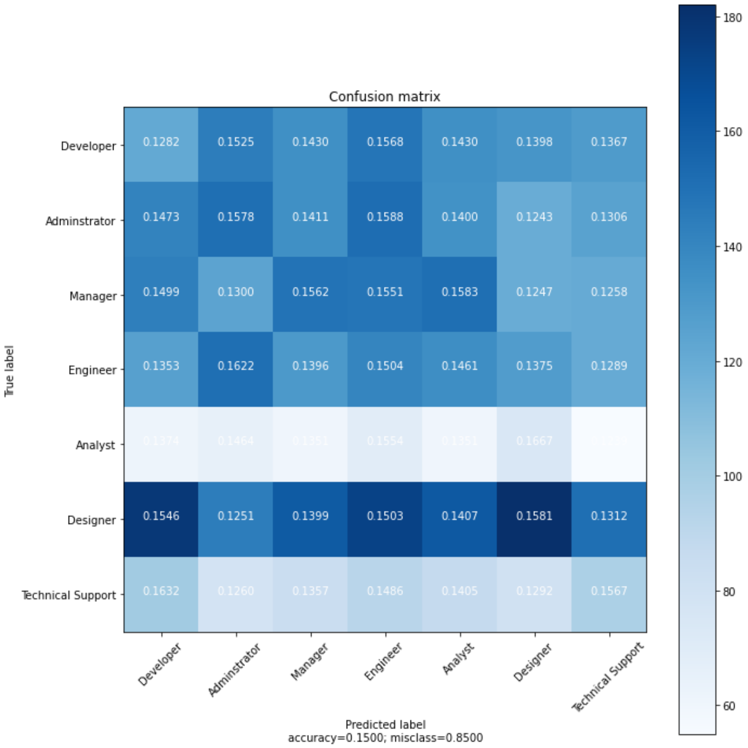


2. *Increasing the number of maximum iterations to remove the convergence warning.*



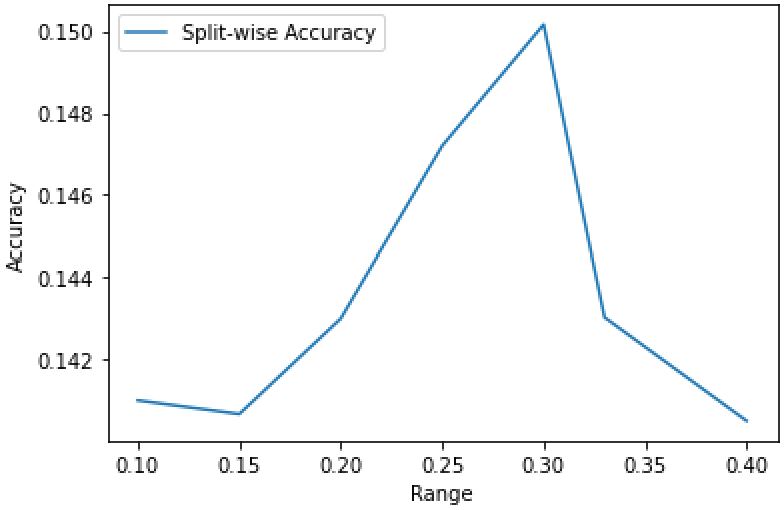
The average accuracy of the model is 0.14365

Classwise Accuracies.
The class wise accuracies are [0.12817797 0.15778474 0.15618449 0.15037594 0.13513514 0.15812337 0.15670436]



3. *Making different train test splits.*

Plot.

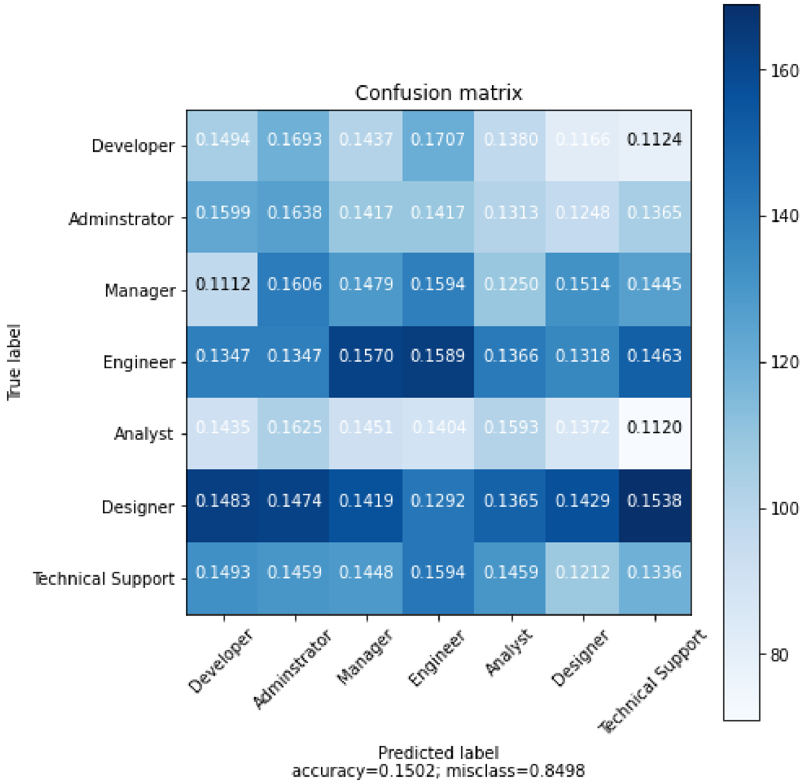


We see the best performance by 70-30 split.

Analysis

We see that the model accuracy decreases when the test size is very small. We can attribute this to the *overfitting of the model*. Due to learning the exact pattern of the train set, the model is not able to give the best accuracy for the test set.

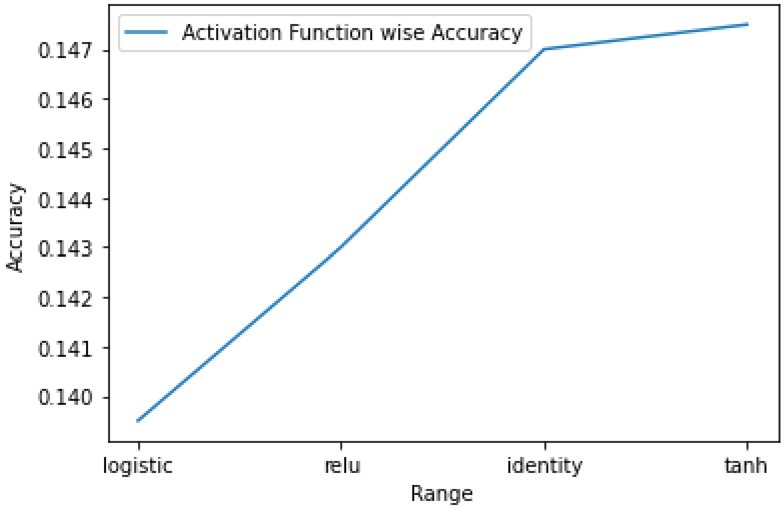
Confusion Matrix for the best prediction.



Class-wise accuracies.

The accuracy of the model is: 0.15016666666666667
The class wise accuracies are [0.14935989 0.16384915 0.14793578 0.15891473 0.15930599 0.14285714 0.1335578]

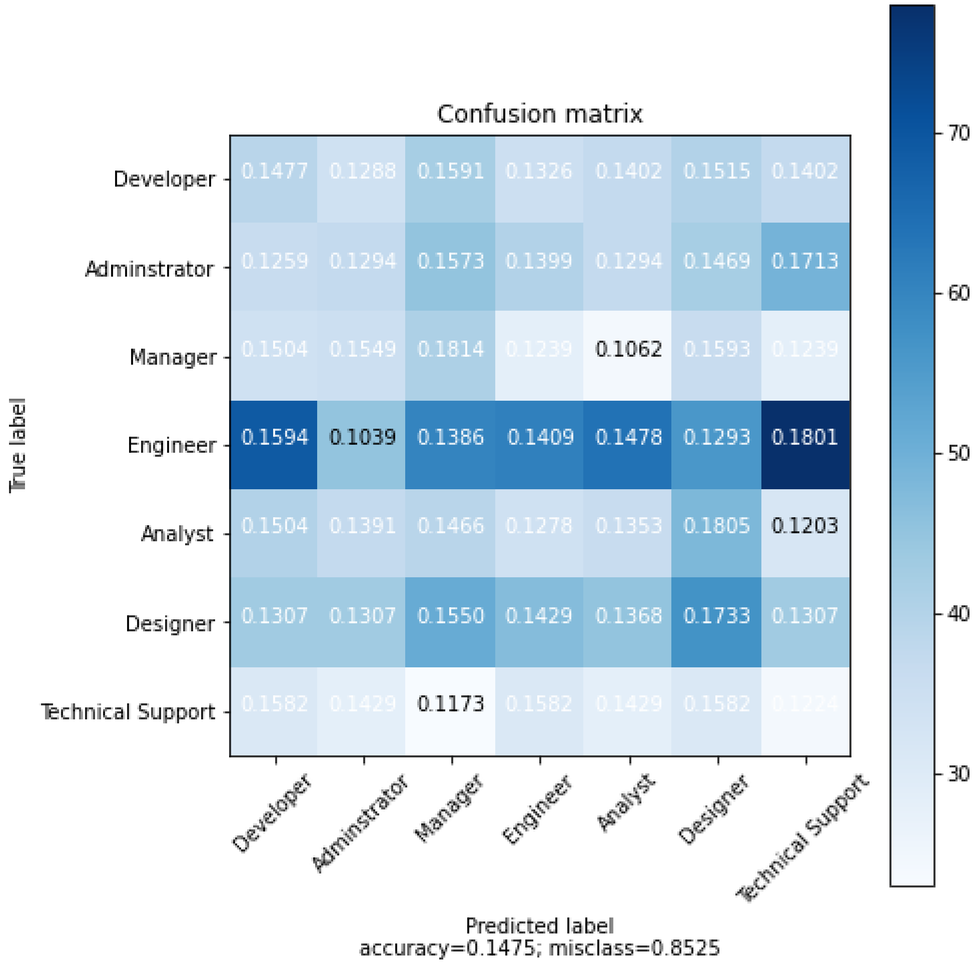
4. *Activation Functions*



We see the best accuracy is given by tanh.

<u>Analysis</u>
Tanh gives a more complicated model than identity and relu. Specific data has different activation functions preference.

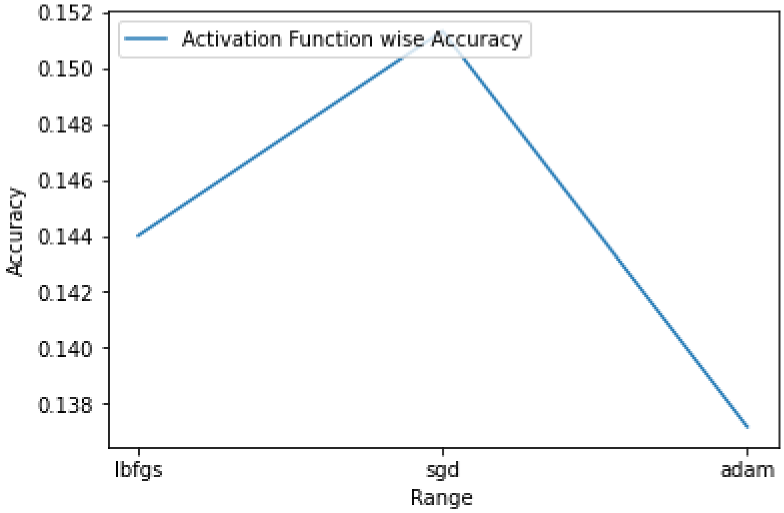
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is: 0.1475
The class wise accuracies are [0.14772727 0.12937063 0.18141593 0.1408776 0.13533835 0.17325228 0.12244898]

5. *Optimizer Functions*

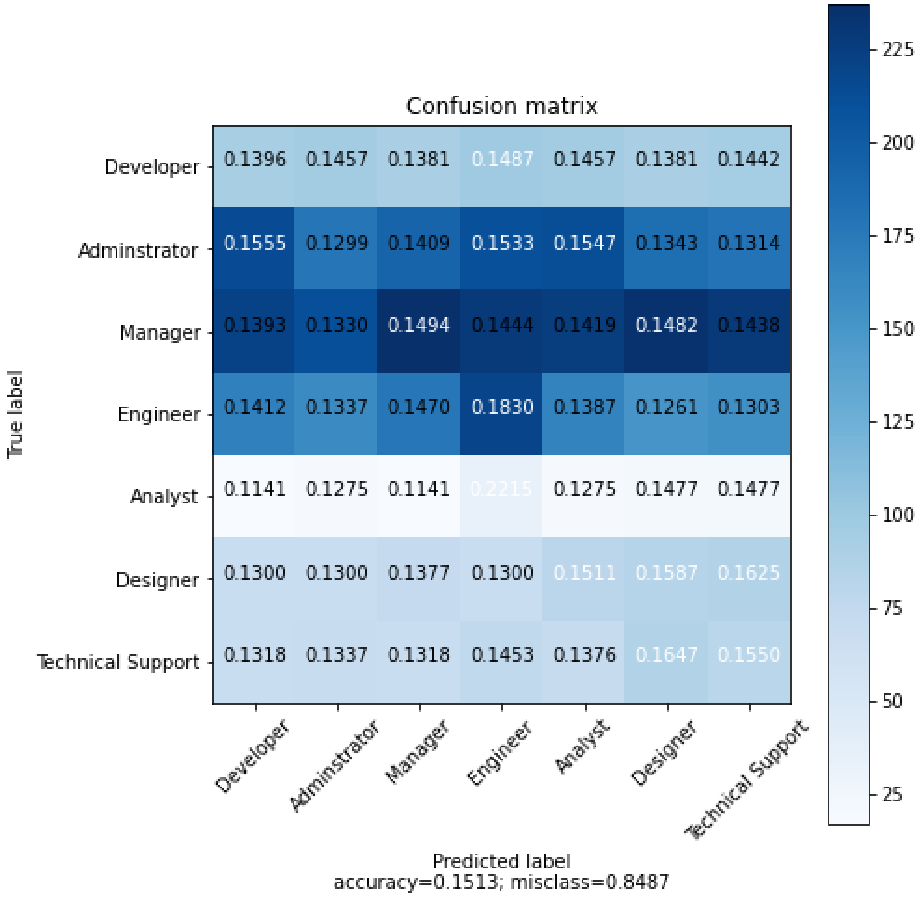


We see the best accuracy is given by the SGD classifier.

Analysis

Lbfgs is better suited for smaller datasets. While Adam is preferred for large datasets and leads to convergence early, which might deter the model from learning the correct pattern.

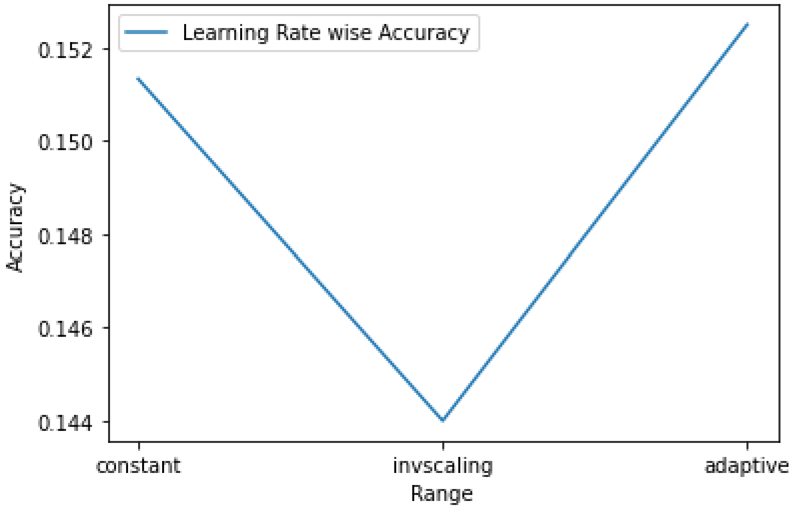
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is: 0.15133333333333332
The class wise accuracies are [0.13960546 0.12992701 0.14943253 0.18295739 0.12751678 0.15869981 0.15503876]

6. *Learning Rates.*

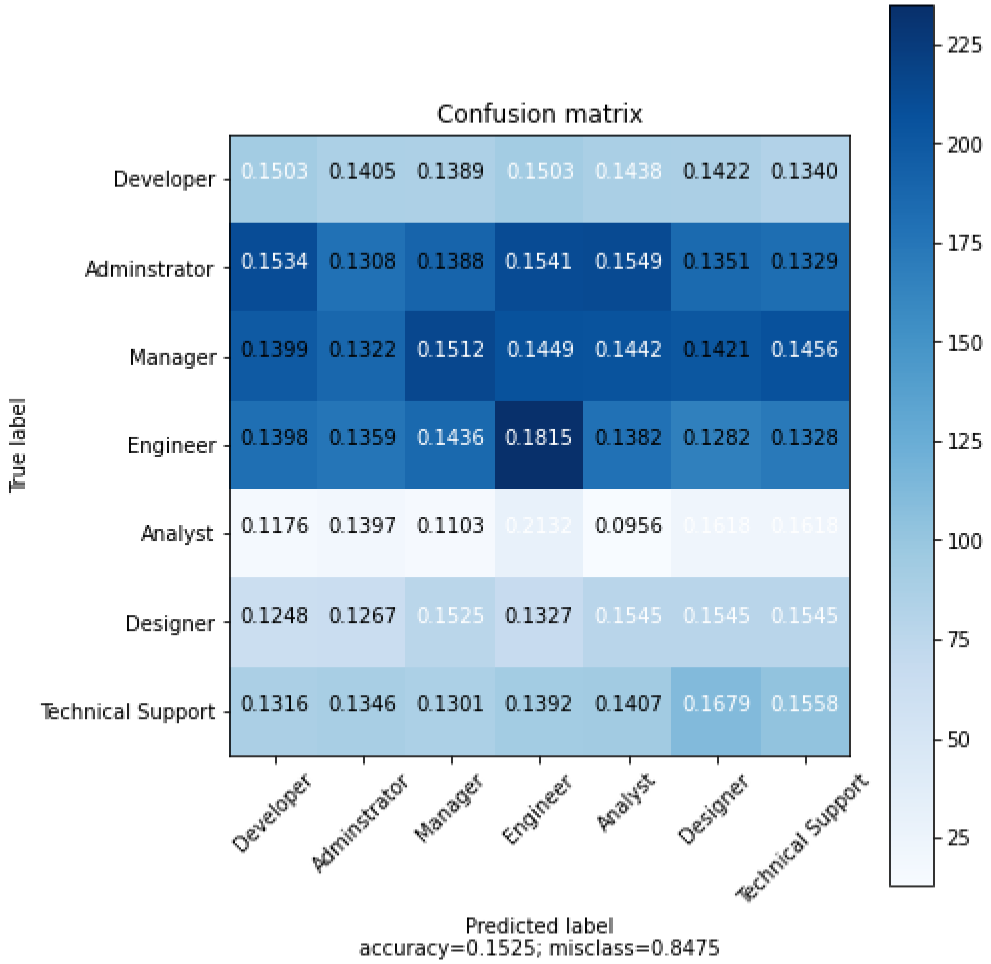


We see the best accuracy is given by adaptive learning rates.

Analysis

Adaptive learning rate implies learning rate changing with the number of epochs to better adapt to the change in costs, which works best in our case with a low accuracy rate.

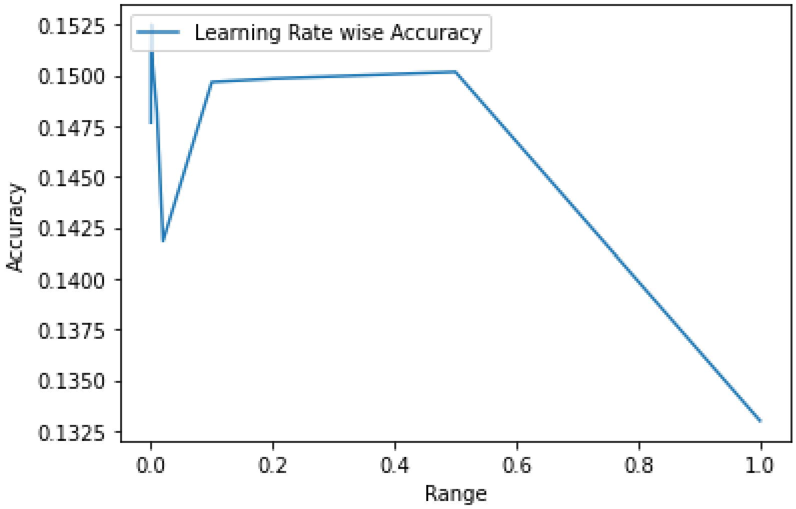
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is 0.1525.
The class wise accuracies are [0.1503268, 0.13075237, 0.1511955, 0.18146718, 0.09558824, 0.15445545, 0.15582451]

7. *Initial Learning Rates.*

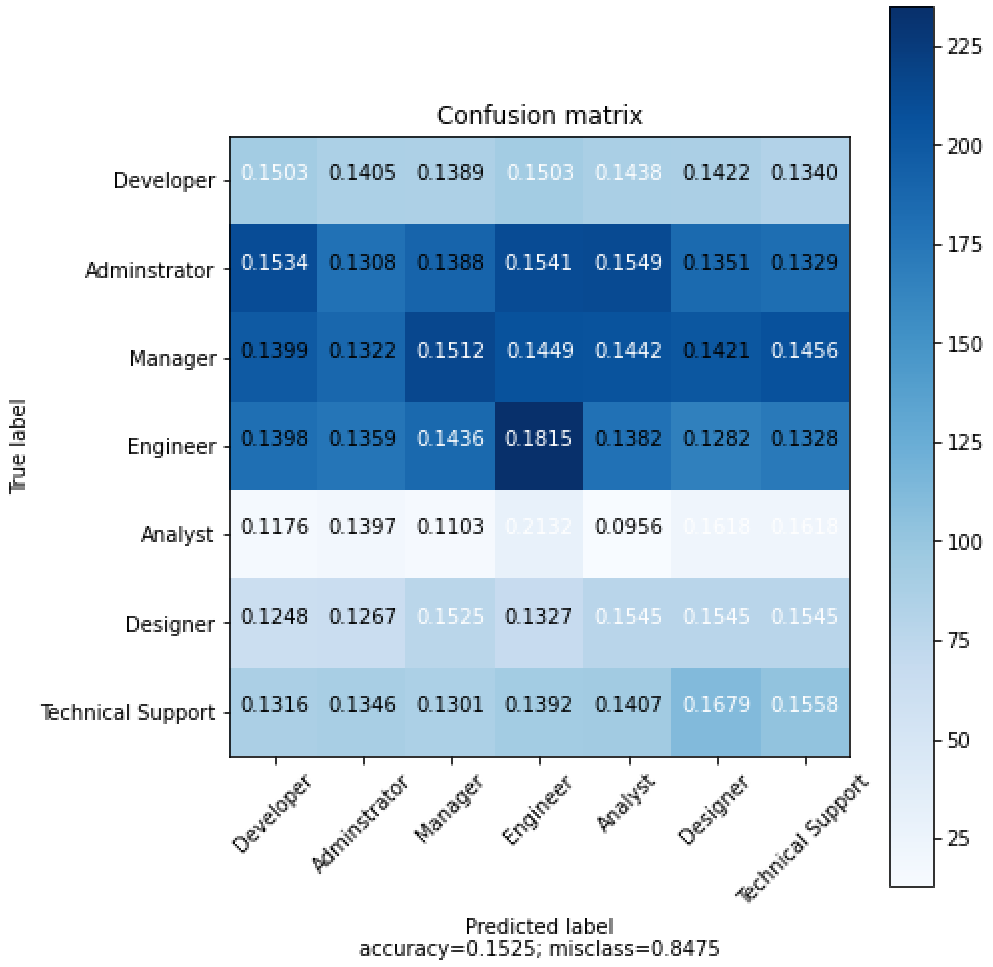


We see the best accuracy is given by 0.001.

Analysis

Smaller learning rates lead to smaller convergence while large learning rates give rise to oscillations. Thus, we need a hand-picked learning rate.

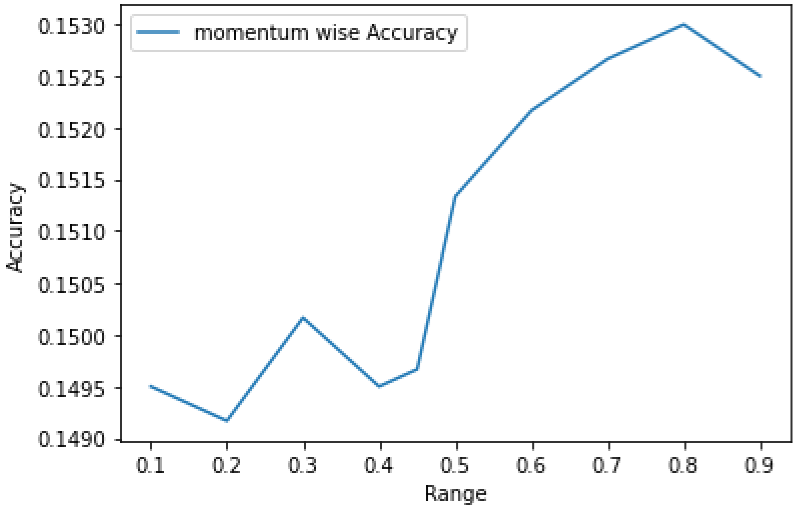
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is: 0.1525
The class wise accuracies are [0.1503268 0.13075237 0.1511955 0.18146718
0.09558824 0.15445545
 0.15582451]

8. *Momentum for merging SGD*

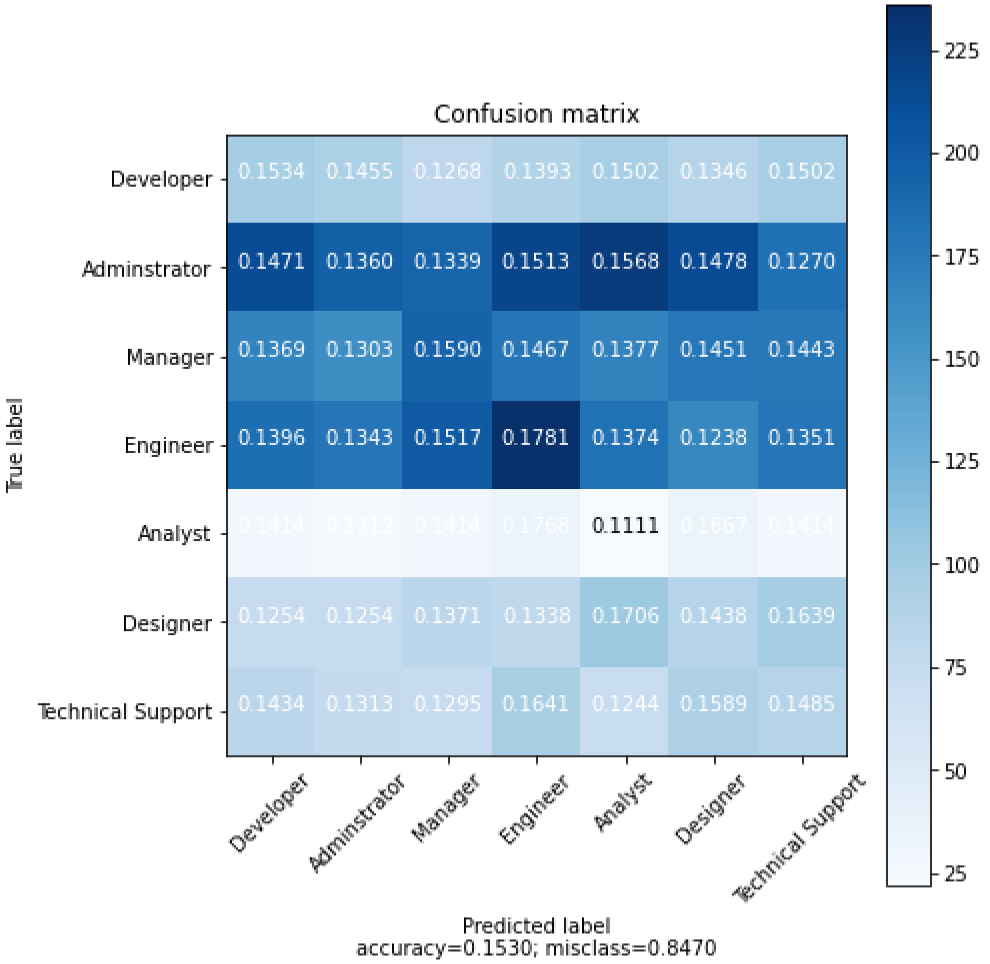


We see the best accuracy is given by 0.8.

Analysis

Momentum leads to faster convergence by pushing in the correct direction. Thus we need to fix the correct value of momentum.

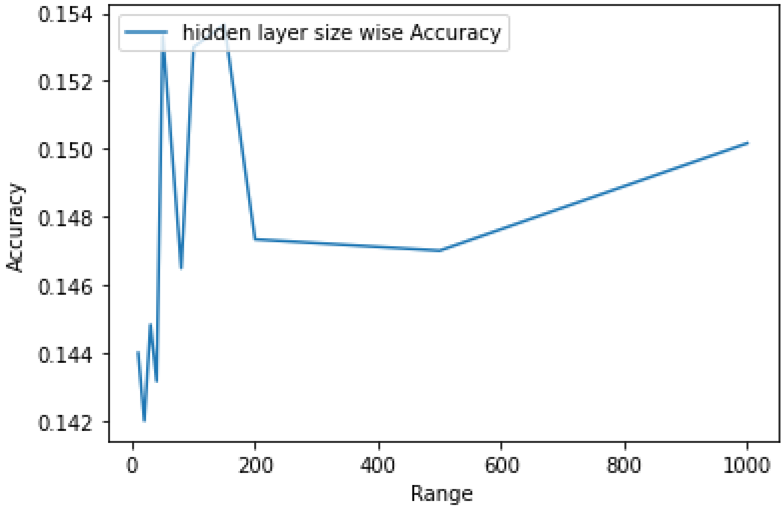
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is: 0.153
The class wise accuracies are [0.15336463 0.13601666 0.15901639 0.17811321 0.11111111 0.14381271 0.14853195]

9. *Size of first hidden layer*

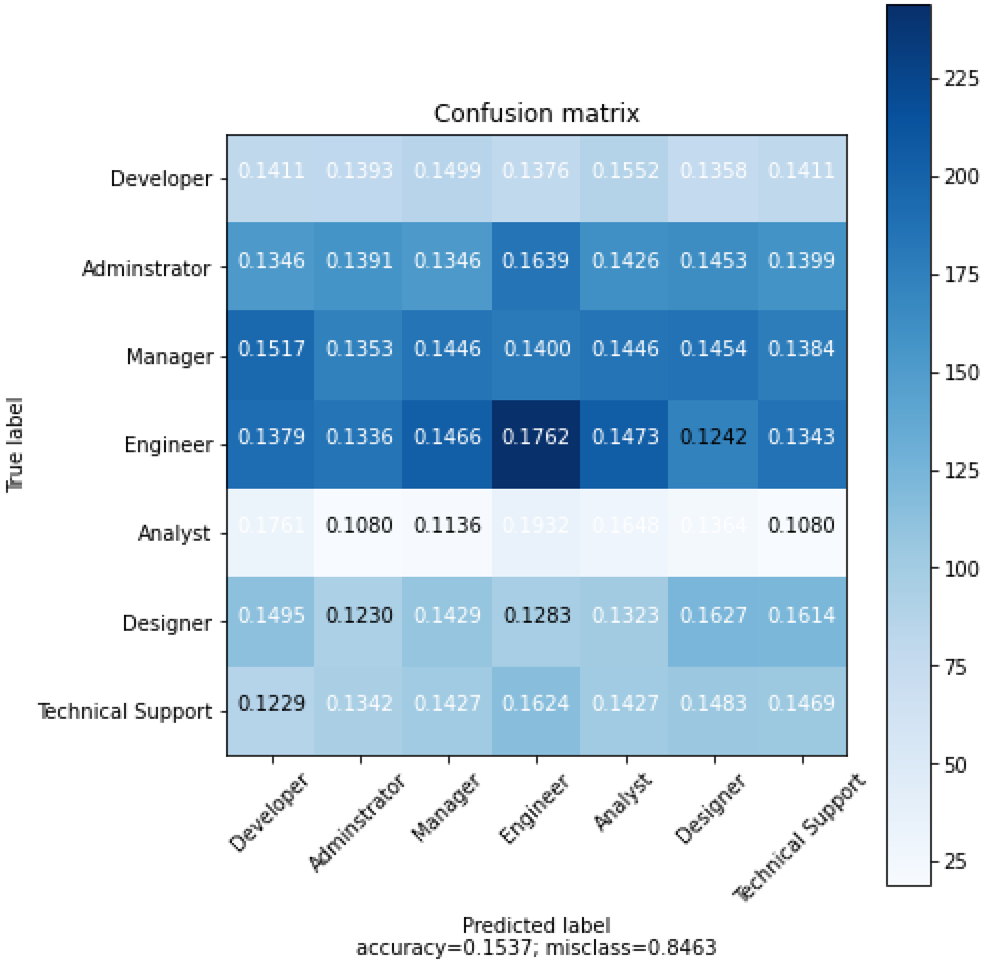


We see the best accuracy is given by 150.

Analysis

Different sizes of hidden layers allow ANN to learn different models. The higher the number of layers or neurons the more complicated the model, ANN will learn. Here we see increasing the size from 100 to 150 lets the model learn better.

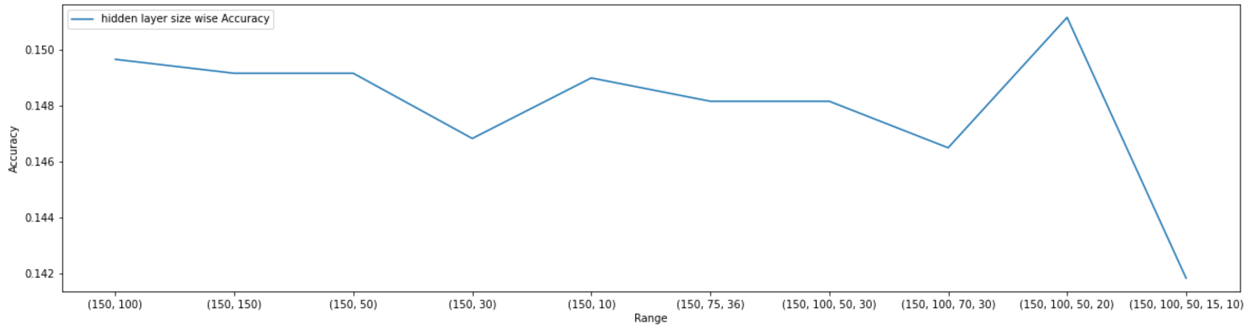
Confusion matrix for the best performance.



Class-wise accuracy for the best performance.

The accuracy of the model is: 0.15366666666666667
The class wise accuracies are [0.14109347 0.13906112 0.14464425 0.17617329 0.16477273 0.16269841 0.14689266]

10. Multiple Hidden Layers.

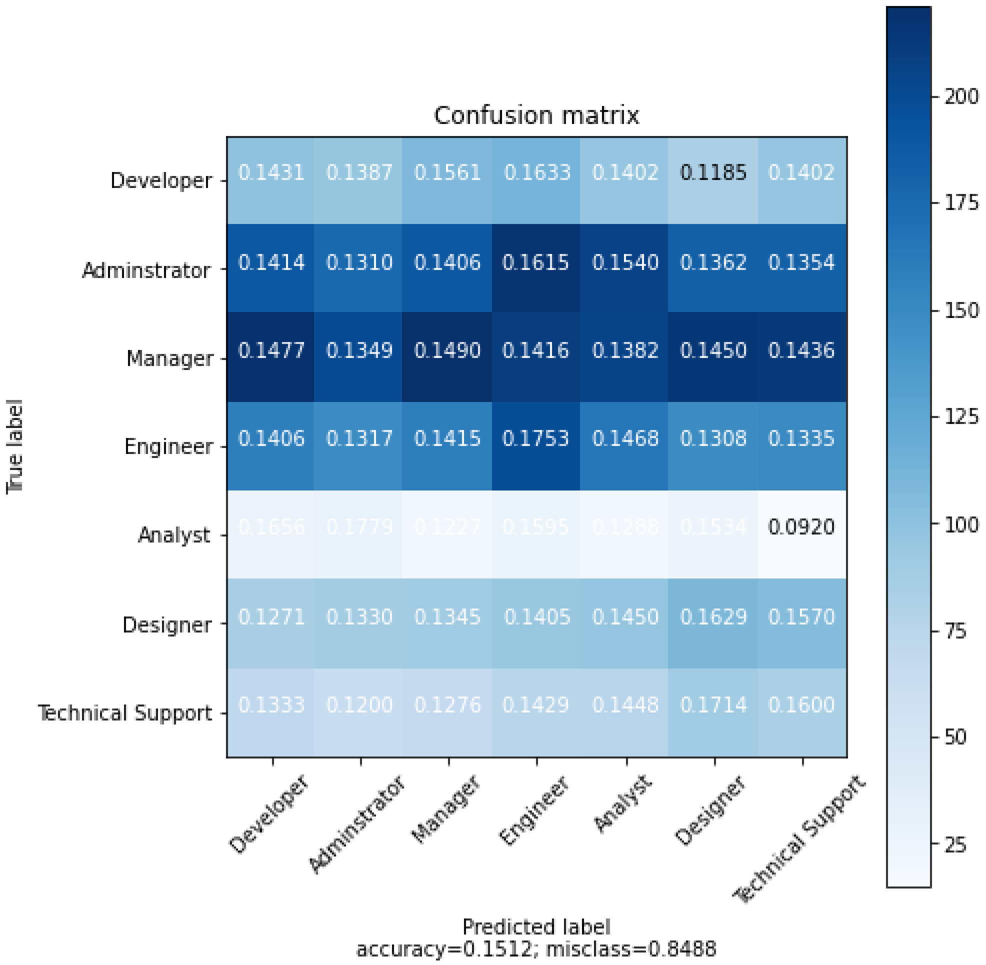


We see the best accuracy is given by (150, 100, 50, 20).

Analysis

Different sizes of hidden layers allow ANN to learn different models. The higher the number of layers or neurons the more complicated the model, ANN will learn. Here we see increasing the number of hidden layers lets the model learn better.

Confusion matrix for the best performance.



	Developer	Adminstrator	Manager	Engineer	Analyst	Designer	Technical Support
Developer	0.1431	0.1387	0.1561	0.1633	0.1402	0.1185	0.1402
Adminstrator	0.1414	0.1310	0.1406	0.1615	0.1540	0.1362	0.1354
Manager	0.1477	0.1349	0.1490	0.1416	0.1382	0.1450	0.1436
Engineer	0.1406	0.1317	0.1415	0.1753	0.1468	0.1308	0.1335
Analyst	0.1656	0.1779	0.1227	0.1595	0.1288	0.1534	0.0920
Designer	0.1271	0.1330	0.1345	0.1405	0.1450	0.1629	0.1570
Technical Support	0.1333	0.1200	0.1276	0.1429	0.1448	0.1714	0.1600

Class-wise accuracy for the best performance.
The accuracy of the model is: 0.15116666666666667
The class wise accuracies are [0.14306358 0.13095238 0.14902225 0.1752669 0.12883436 0.16292975 0.16   ]

Final Model

Using the various techniques, we finally arrive at the best model.

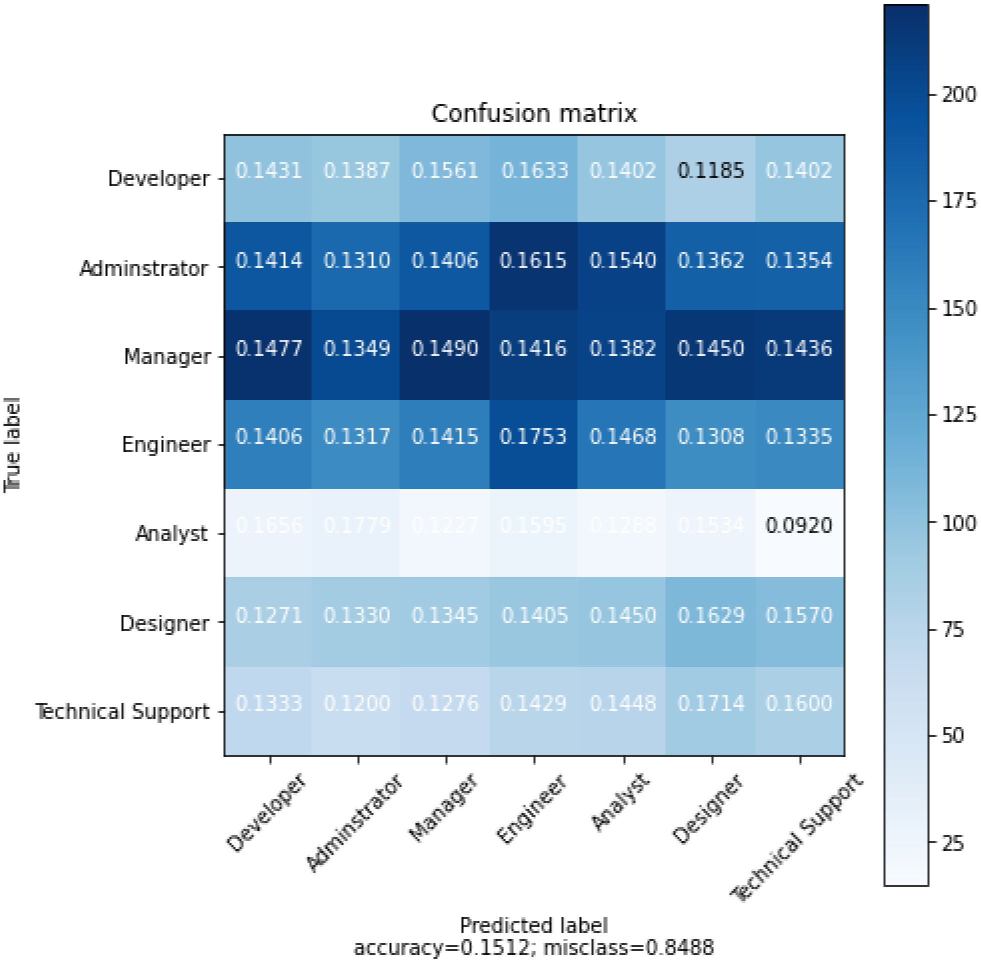
Model.

MLPClassifier

- Max iterations = 1000
- Hidden Layers = (150, 100, 50, 20)
- Activation Function = *tanh*
- Optimizer Function = *SGD or Stochastic Gradient Descent*
- Learning Rate = *Adaptive*
- Initial Learning Rate = 0.001
- Momentum = 0.8
- Train-test Split = 70% Train and 30% Test.

Evaluation Metrics.

- Confusion Matrix



	Developer	Adminstrator	Manager	Engineer	Analyst	Designer	Technical Support
Developer	0.1431	0.1387	0.1561	0.1633	0.1402	0.1185	0.1402
Adminstrator	0.1414	0.1310	0.1406	0.1615	0.1540	0.1362	0.1354
Manager	0.1477	0.1349	0.1490	0.1416	0.1382	0.1450	0.1436
Engineer	0.1406	0.1317	0.1415	0.1753	0.1468	0.1308	0.1335
Analyst	0.1656	0.1779	0.1227	0.1595	0.1288	0.1534	0.0920
Designer	0.1271	0.1330	0.1345	0.1405	0.1450	0.1629	0.1570
Technical Support	0.1333	0.1200	0.1276	0.1429	0.1448	0.1714	0.1600

- Accuracy

  The accuracy of the model is: 0.15116666666666667

- Class-Wise Accuracies
  The class wise accuracies are [0.14306358 0.13095238 0.14902225 0.1752669 0.12883436 0.16292975 0.16   ]

Thank You