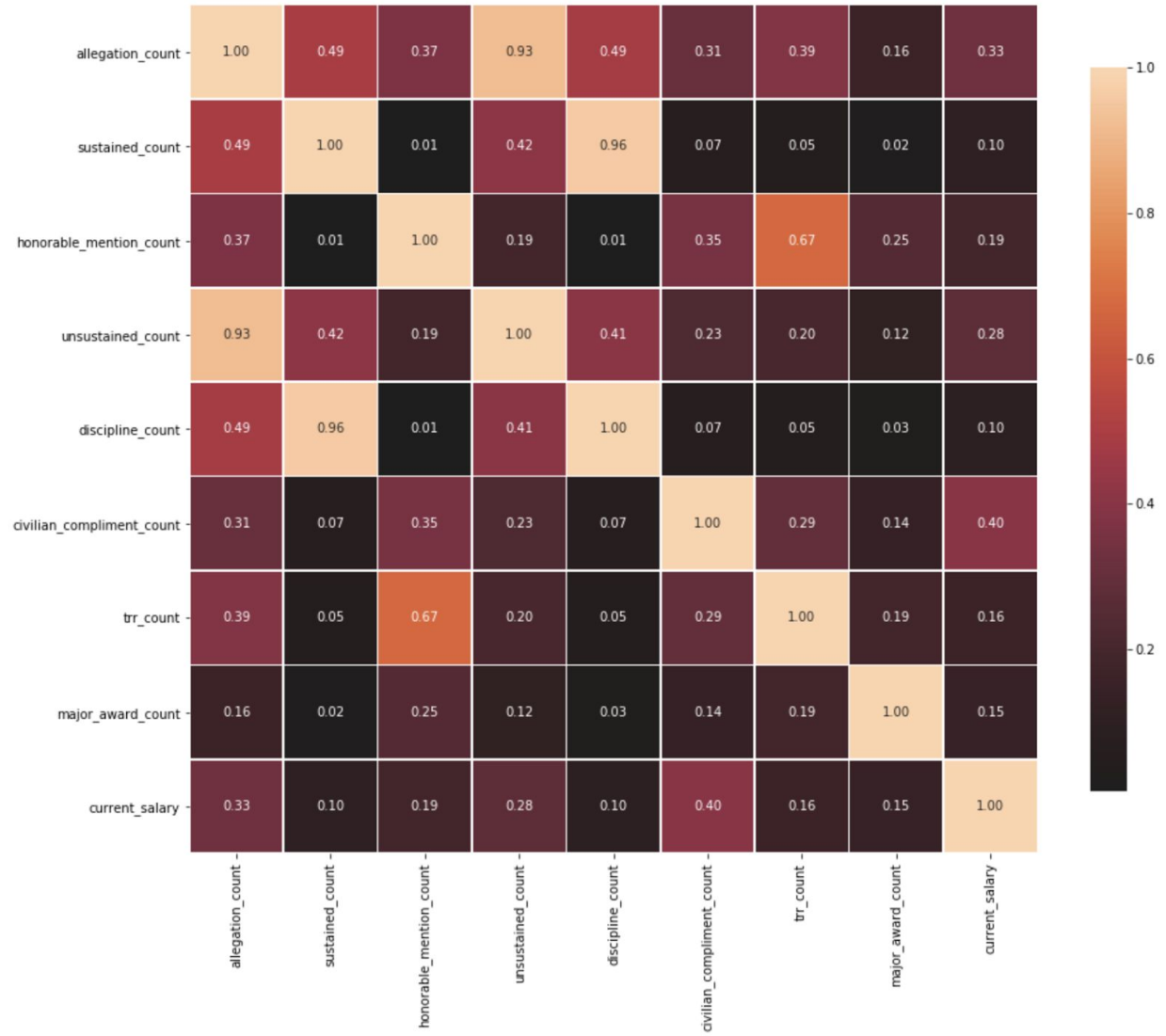
Checkpoint 4

Anam Khan
Madhav Khanna (3184733 mko6761)
Ashish Jeldi (3219879 aej1797)

1. To classify each officer in a group based on the number of allegations that an officer might be involved in.



- **Quantitative Analysis**

  As seen from the correlation matrix derived from the data, the number of allegations for each officer is based on the following fields:
  - Sustained Count
  - Honorable mention count
  - Unsustained Count
  - Civilian Compliment count
  - Discipline Count
  - Current Salary
  - Trr Count

  We see a positive correlation of the number of allegations against the above mentioned features as seen from our Correlation Plot. The data was preprocessed and the null values for the allegation count were dropped. Post this preprocessing we replaced the null values for the current_salary with the mean of the salary. Resulting in a data of about 33000 rows.

  Being a classification problem we decided to use q-cut to come up with 3 classes of varying allegation counts for each officer. The resulting bins had the following number of records:
  - Class 0: 13534
  - Class 1: 11003
  - Class 2: 9302

  The criteria for the each bin were as follows:

  Allegation_count > 1 = class 0
  7 > Allegation_count >= 1 = class 1
  Allegation_count >= 7 = class 2

  The data was randomly split 2:1 as the train-test split.

  *(You can find the decision tree output file **.dot** files in the Results folder)*

- **Qualitative Analysis**

  We used a couple of different models on our data, and got the following results:

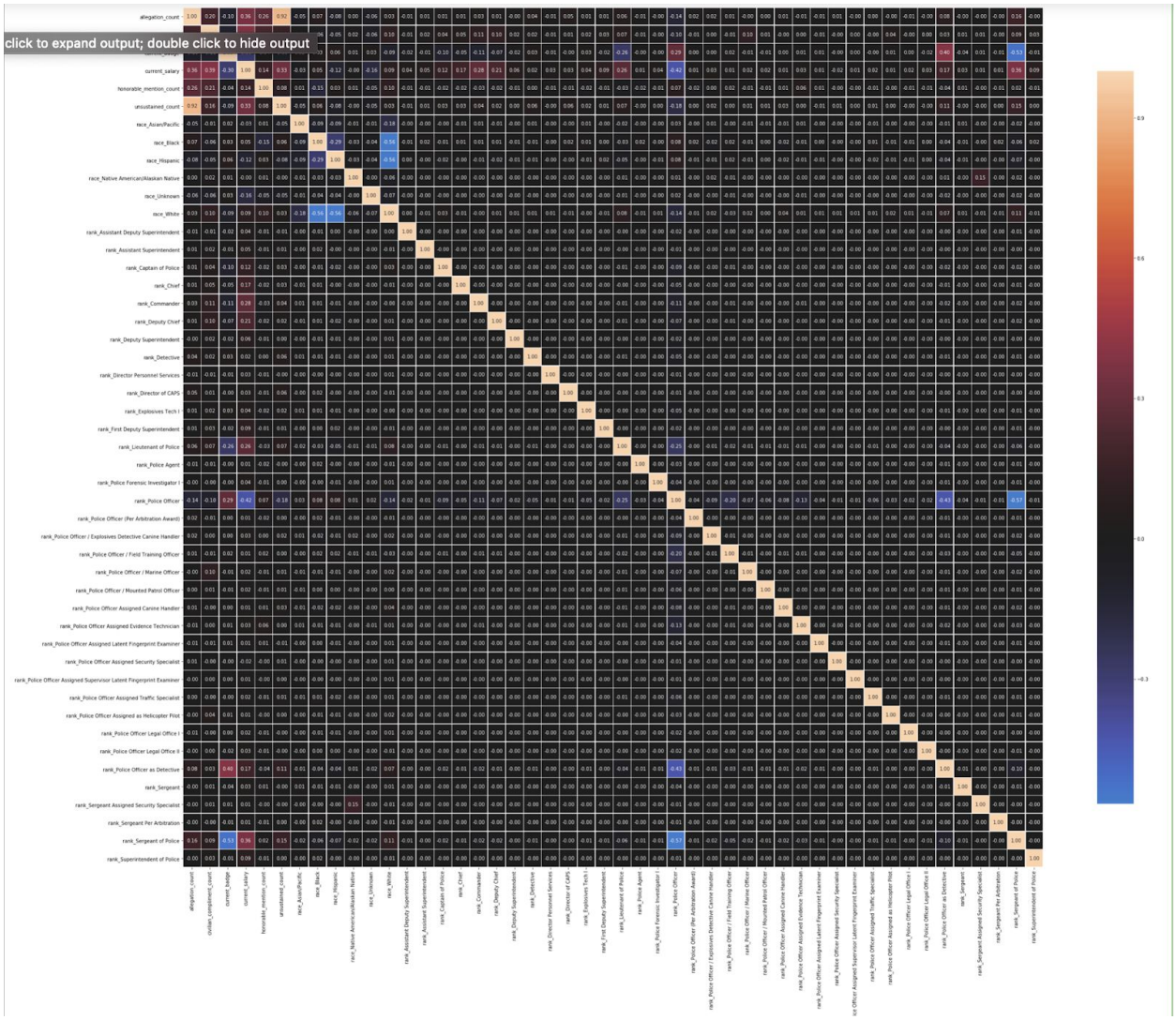
  Decision tree with a depth of 4 gave us an accuracy of 0.8482.

  We used GridCV Search on our tree to find the best possible parameters and it gave us an accuracy of 0.8642.

A random forest with 100 trees and a depth of 6 gave us an accuracy of 0.8639.

Gradient Boosting Classifier gave us a mean accuracy of 0.8754

2. To detect ambiguity in the salaries of officers



- **Quantitative Analysis:**

Our goal is to train a model to learn how much each officer is paid based on certain features and then have it estimate how much an officer should be paid. This would help us identify if

an officer is being paid significantly higher or lower than they should be, thus detecting ambiguity in officer payscale.

The correlation matrix shows that the salary of officers is dependent majorly on features like rank, current badge, unsustained count, civilian compliment count, allegation count. We have built Linear Regression, Polynomial Regression and Multi Layer Perceptron Models to predict the salary which officers should be given and we used the above features as inputs for these models.

The records where the currect_salary were NULL were removed as they are useless to the dataset since salaries is our target variable. Also, there was a minor fraction of records where the currect_badge values were NULL, these were discarded too since we had a big enough dataset to work with.

The values with strings like ‘race’ and ‘rank’ were one hot encoded to be able to be fed into our predictive models. We had made a split of 4:1 on the data for validation purposes. The linear regression model gave us an RMSE value of 0.36, the polynomial regression model with a degree of 2 gave us an RMSE value of 0.52, the polynomial regression model with a degree of 3 gave us an RMSE value of 0.63, the polynomial regression model with a degree of 4 gave us an RMSE value of 0.24.

Also, we observed that office pay isn’t dependent very much on gender and race.

- **Qualitative Analysis:**

It is expected that the linear regression model would not yield good results as the data is not linear and thus the model would not fit well.
We can see that as we increase the degree of the polynomial regression model from 2 to 3, it gives us better accuracy, meaning it fits the data better. However, when we use a degree of 4, the accuracy values drop significantly. This is a sign that our model is overfitting on the training data and hence isn’t generalizing well on the test data.
R2 compares the fit of the chosen model with that of a horizontal straight line (the null hypothesis). If the chosen model fits lower than a horizontal line, then $R2$
R2 is negative. This is the case in our Neural Networks, as it is trying to fit the data with a straight line and hence yields poor results.