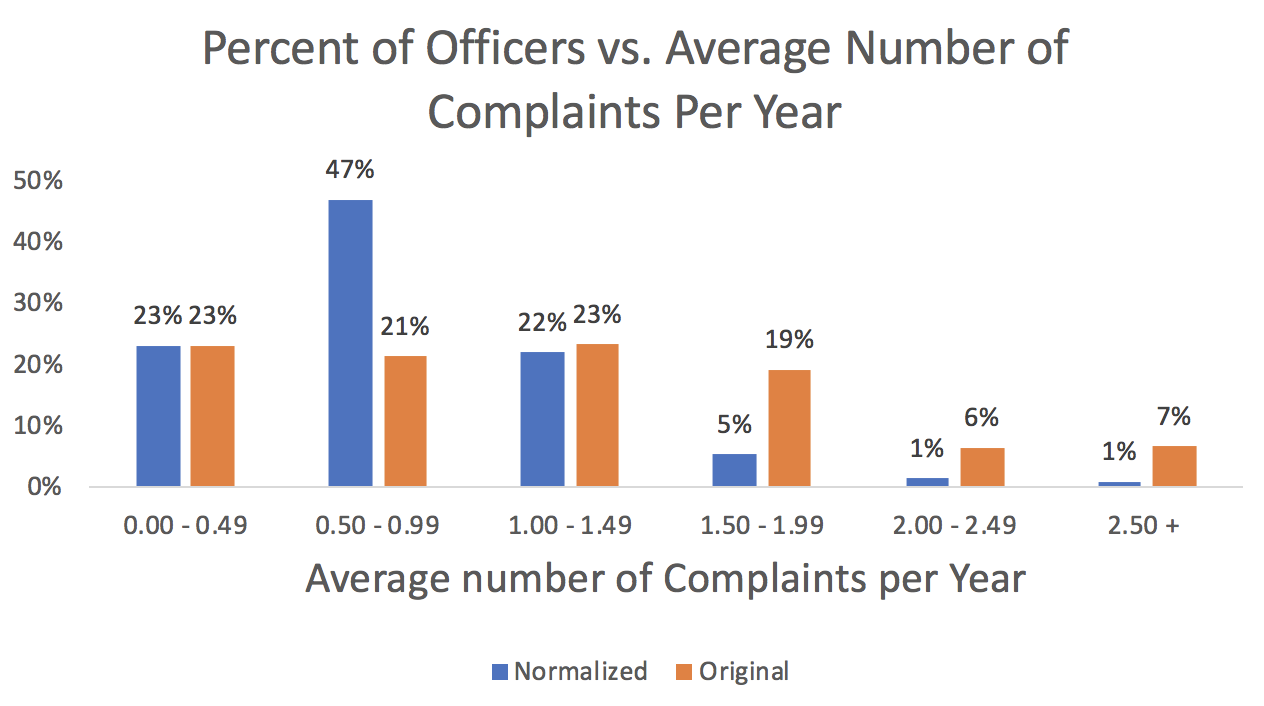
***Chicago Police Department - Checkpoint 4 - Machine Learning***

In terms of applying machine learning to our dataset, our goal was to explore our prior analyses through supervised and unsupervised machine learning methods.

First, we explored utilizing k-means clustering to verify our previous analysis to identify officers who were repeat offenders. To complete this task, we utilized two critical attributes that we previously computed - average number of complaints per year and normalized number of complaints per year at an officer level. In checkpoint 2, we had viewed the figure below and determined that above 1.5 normalized complaints per year looked to be a good boundary between the two groups.

**Figure 1**: Percent of Officers vs. Average Number of Complaints Per Year

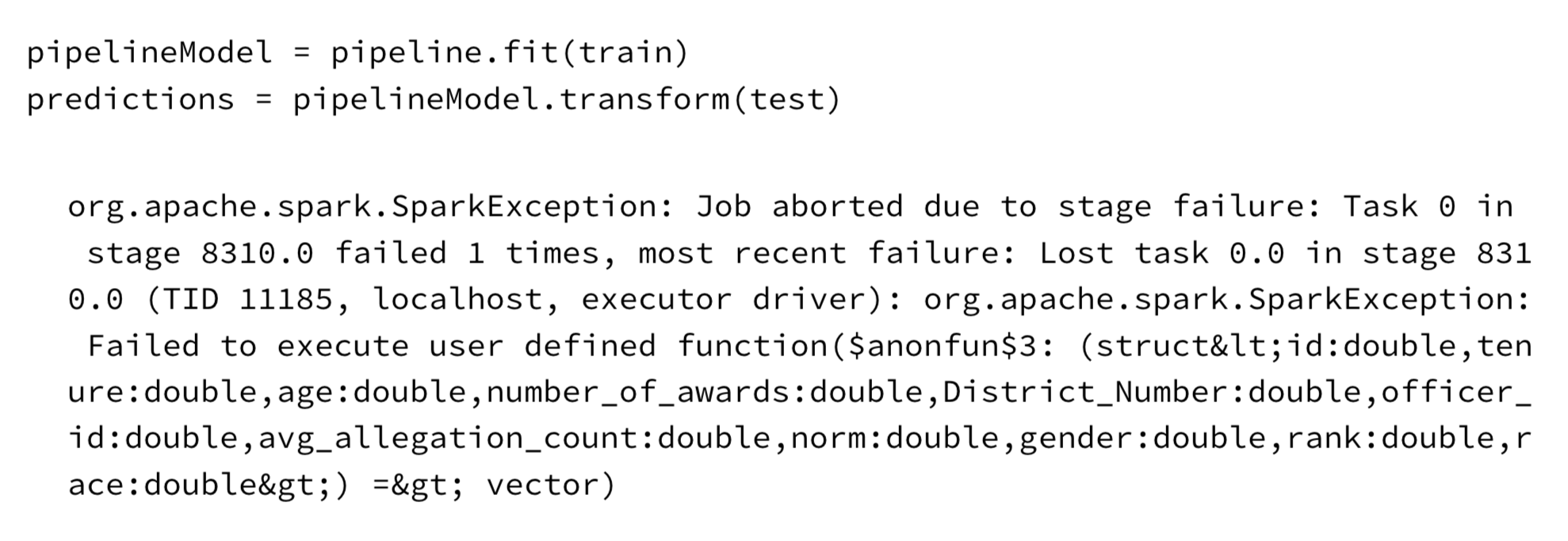


By running our first set of spark code as shown in the queries document, we were able to cluster the officers into two distinct groups. After examining the two groups, we were very happy to see that we had only misclassified 887 total officers out of about 14,000 officers. Our output has been provided in a supplemental CSV file named k\_means\_evaluation. As our project moves forward, following our next two checkpoints we would like to dig into performing this analysis with additional dimensions, and also clustering officers move tightly using smaller timeframes.

For our supervised method, we desired to build a decision tree model to predict officers that will become repeat offenders using the following features:

* Demographics
* Tenure
* Rank
* # of awards
* District/Beat
* Normalized Complaints per Year
* Average Complaints per year

To do this, we performed joins on the overall table from checkpoint 2 with the normalized allegation counts off of crime and arrest data, with the k-means clustering classifications. We also had to do an additional join from an aggregation of awards for each officer from the DATA\_AWARD table. With all of the features we want to use for the prediction in one table, we then had to vectorize the columns, which required string indexing on the attributes: Rank, Tenure and Race. By using the clustering classification as the label, we attempted to create a decision tree model in Spark, with a 70/30 cross-validation split on our data. Previously, we ran into errors when executing the display of the prediction model. Below is an image of the previous error message our spark notebook was giving:



To alleviate our issue, we tried testing our model with different classifiers, but similar errors kept happening. However, we restarted our cluster completed (not a clone) and ran the code again and it was successful. One slight modification that we added to our original code was to cache our training dataset and the pipeline - which helped with runtime speed as well.

The specific decision tree model that we ran was a Gradient Boosted Forest (Trees in Spark) with Regression. There were 100 different trees created in our specific ensemble method, but after using the feature Importances attribute, we obtained the following values of significance for each attribute:

Tenure: 0.0582

Age: 0.0393

Number\_of\_awards: .0988

District\_Number: 0.0073

Average\_Allegation\_Count: 0.4296

Normalized\_Allegation\_Count: 0.2971

Gender: 0.0008

Rank: 0.026

Looking at the above significance, we were interested to see that the non-normalized allegations resulted in a much higher significance as a feature than the normalized. Additionally, we were surprised to see that the number of awards an officer had was the largest impact outside of our core indicators. This is something we want to look into further as the project closes out. We also did not receive results from Race - indicating that the attribute was thrown out due to insignificance, something we were very surprised at.

With all three of our models trained (K-means clustering, normalized allegation count, GBT regression), we wished to see how they compared to each other. We took a random sample of 1000 from the Decision Tree output and matched our results to both previous models - our own evaluation cutoff and the K-means clustering. We were very pleased to see that our GBT regression matched to our K-means results with an accuracy of about 99%, and about 93% to our original cutoff. This further validated that our original normalized model with a top 7% threshold cutoff was significant, and that our K-means results were an even better nuanced indicator. We plan to use the K-means clustering evaluation of other officers moving forward. The results for the ~1,000 officers is included in a attached CSV file. In this file a value of 0 indicates a repeat offending officer.

**Conclusion**

From our k-means clustering analysis, we confirmed that our method of using a threshold on normalized allegation count was a legitimate classification on repeat offenders. With the decision tree, we also validated both models further. Additionally, although we were hoping to find more features that are indicative of repeat offending officers, one we believe to be significantly valuable at this time is number of awards.