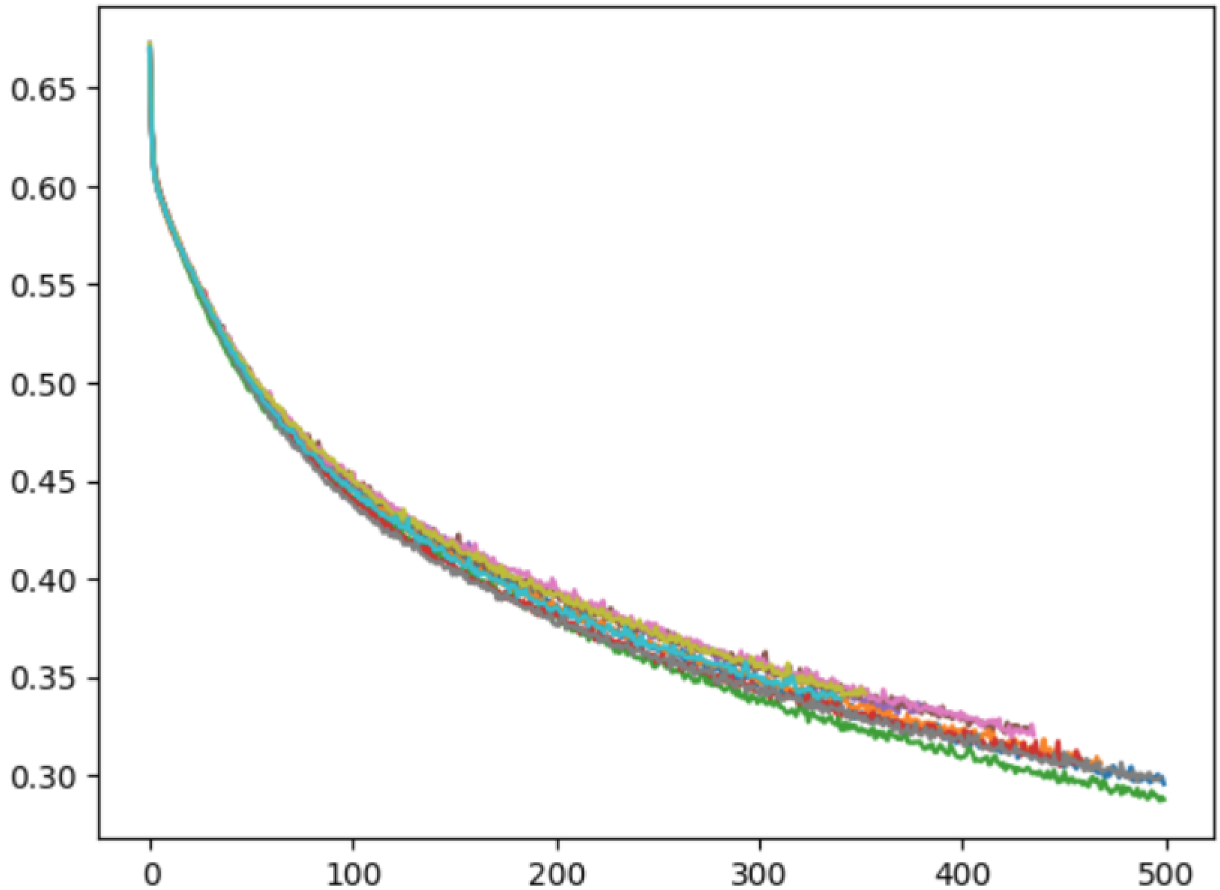
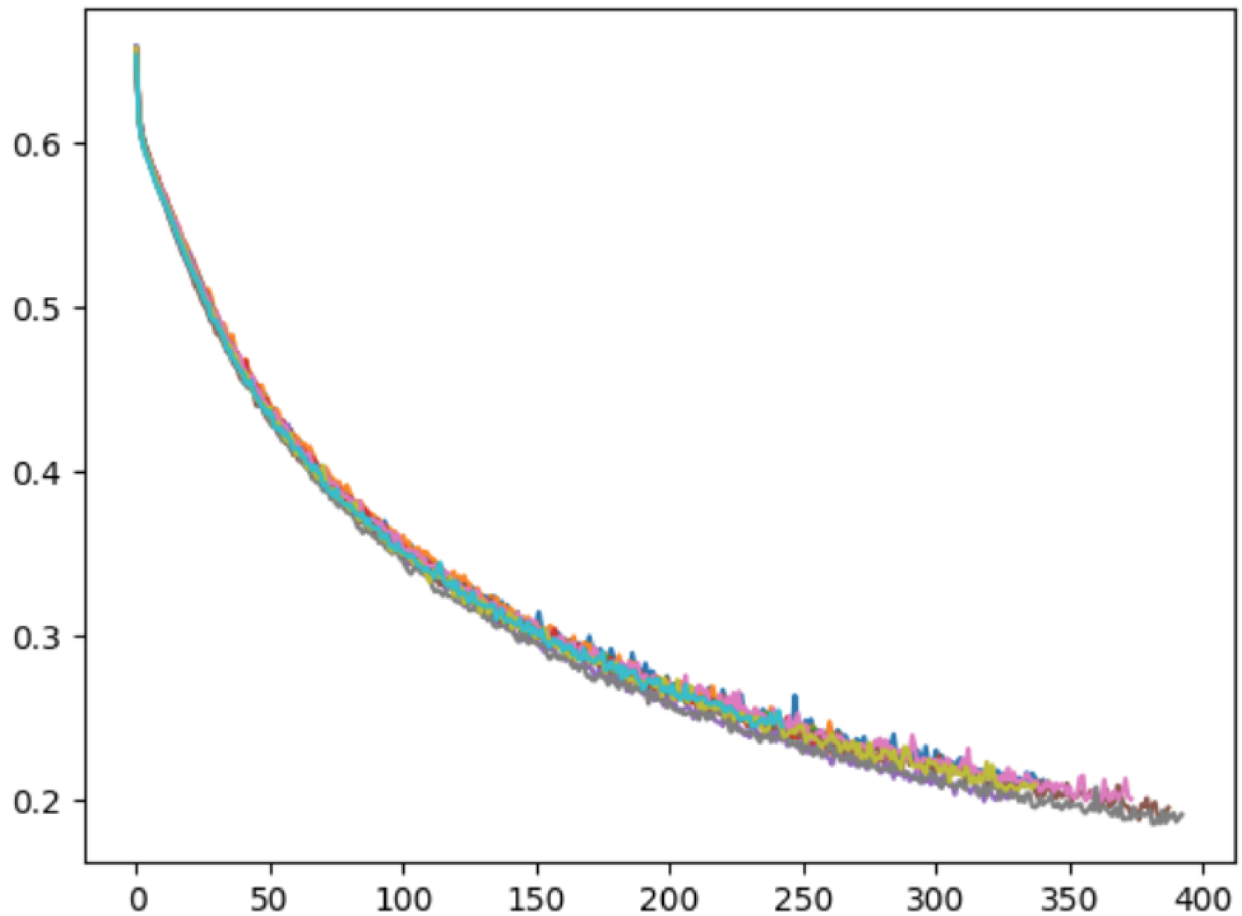
Our overall research question was to determine if patterns could be identified in the execution of search warrants by CPD. We thought it would be interesting to determine if the likelihood of arrest and of property being recovered during the execution of a warrant could be predicted by a machine learning model at a rate higher than chance. This might give some high level insight on the decision process used by the department in determining when, where, and how to serve warrants.

We began by using a random forest, as this model is applicable with minimal feature engineering. While it was able to perform at level slightly better than chance at predicting property_recovered, we found that it did poorly at predicting arrests. Afterwards, we experimented with a MLP classifier. This required some additional feature engineering. Categorical columns (district, policeunit_id) were encoded as one-hot vectors. The timestamp column was decomposed into hour, day of year, month, and year. We suspected there might be some cyclical dependence upon both time of day and day of year, so we encoded the hour, month, and day of year columns as sinusoids. This encoding was chosen to respect the fact that e.g. January 1st 2013 is close to December 31st 2012. Encoding month or day of year categorically would fail to capture this relationship. We also made the problem a bit easier by dropping rows which did not have a police unit ID listed.

The MLP model was chosen with some reasonable default parameters. We ran two networks, both with 2 hidden layers. One network had a width of 64, while the second had a width of 128. All trials were performed with 10 random resets. Both models used the same learning rate and regularization, and both used a max iteration count of 500. The 64 wide model had the following training loss curves



Note the non-convergence. Compare the training loss curves of the 128 wide network:



Note that the 128 wide network reaches a training accuracy of above 80 percent. However, the validation accuracy for the 128 wide network is marginally lower (61.67% compared to 61.96% on average for the 64 wide network). This suggests that the 128 wide network is quickly overfitting. This indicates that early stopping should be considered for the 128 wide network.

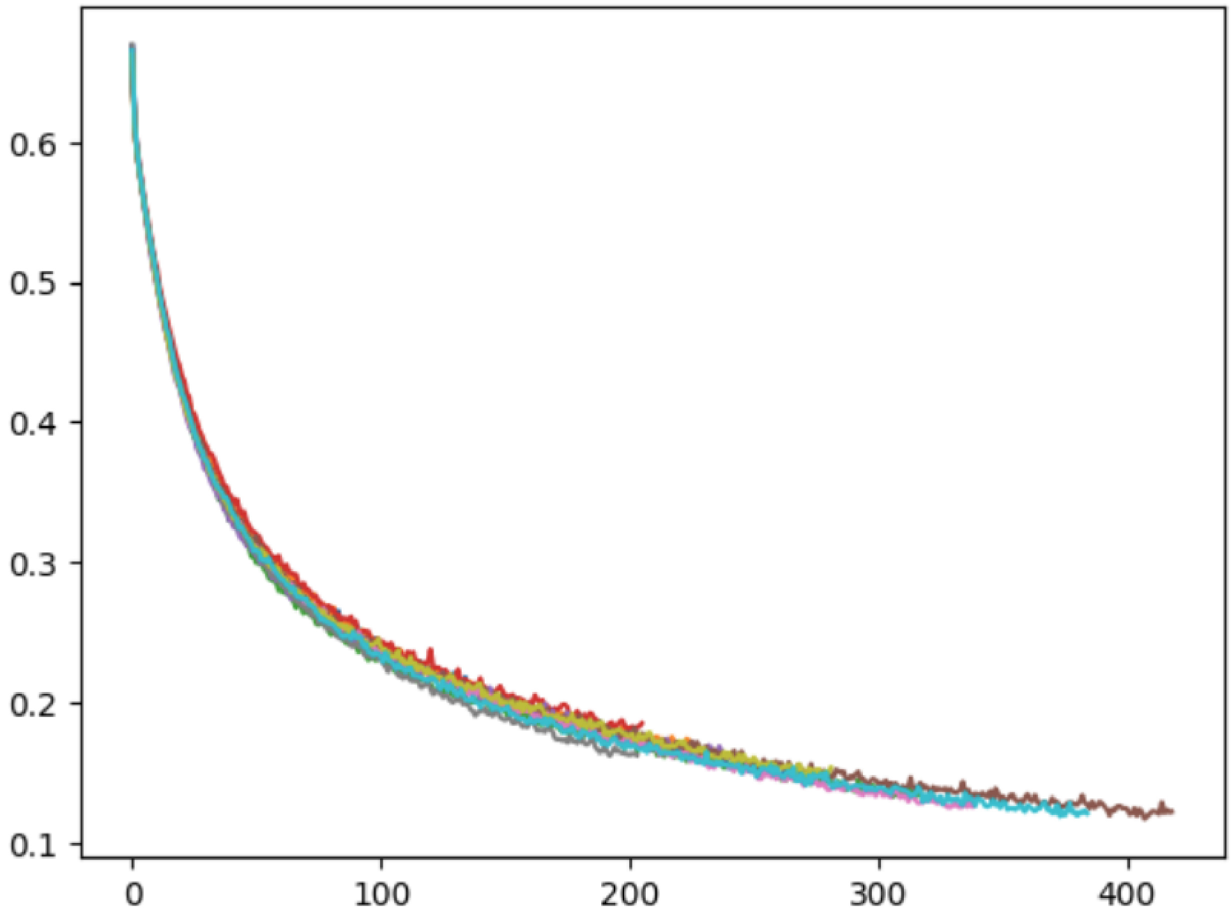
Afterwards, we used permutation importance to interrogate which features were most pertinent for the network's classifications. Permutation importance measures the decrease in a model's score when one column is randomly shuffled. Columns with mean score drop greater than or equal to twice the standard deviation of the score decrease are listed in the following table.

```
policeunit_149 0.017 +/- 0.004
hour_sin 0.014 +/- 0.006
district_11 0.014 +/- 0.003
day_of_year_cos 0.012 +/- 0.005
month_cos 0.011 +/- 0.004
policeunit_215 0.010 +/- 0.002
policeunit_206 0.010 +/- 0.002
year     0.009 +/- 0.005
policeunit_174 0.007 +/- 0.002
policeunit_8 0.006 +/- 0.001
district_16 0.005 +/- 0.001
policeunit_155 0.005 +/- 0.001
policeunit_6 0.004 +/- 0.001
policeunit_173 0.004 +/- 0.002
policeunit_175 0.004 +/- 0.001
policeunit_16 0.004 +/- 0.001
policeunit_11 0.003 +/- 0.001
policeunit_55 0.003 +/- 0.001
policeunit_26 0.003 +/- 0.001
policeunit_12 0.002 +/- 0.001
policeunit_17 0.001 +/- 0.000
policeunit_218 0.001 +/- 0.000
policeunit_161 0.001 +/- 0.000
```

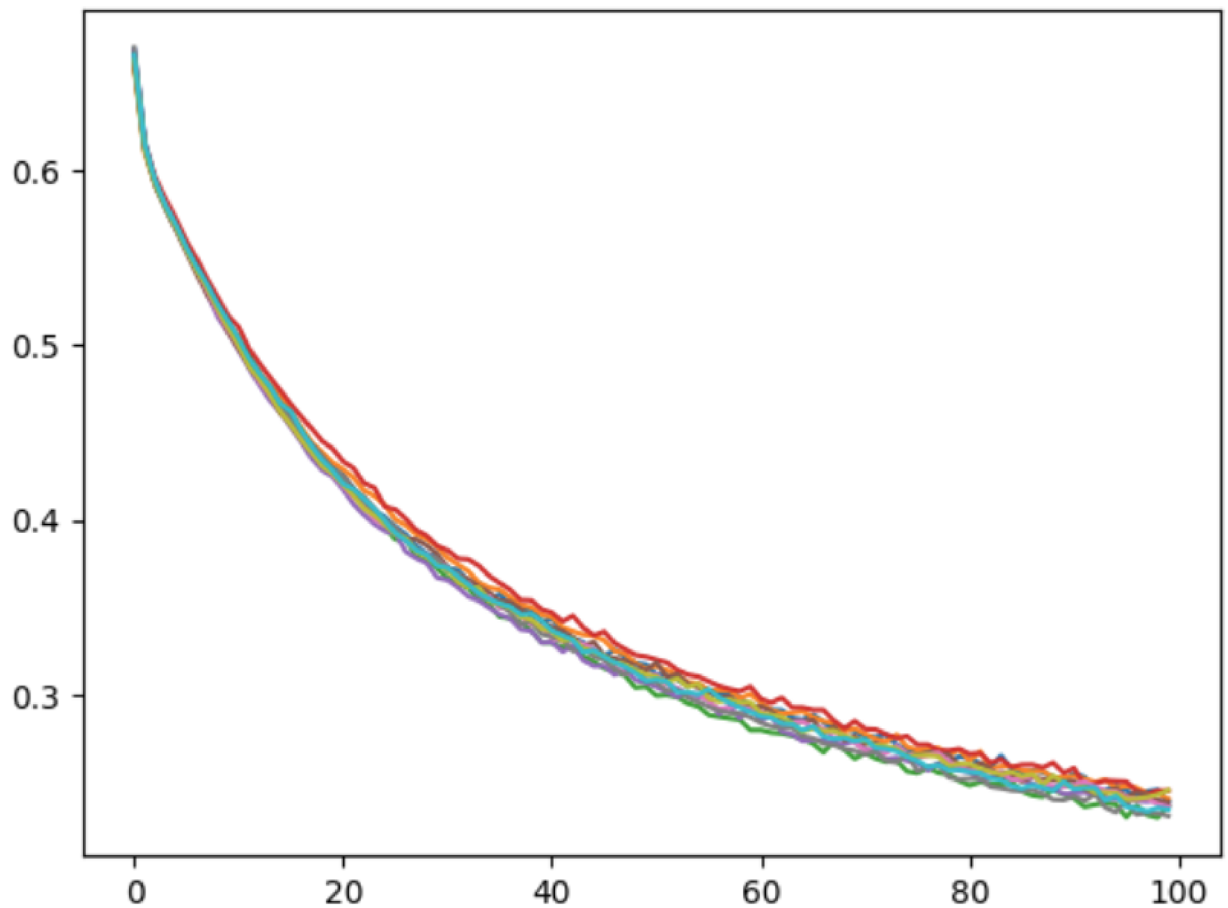
There are a few things to note here. First, policeunit_id is one of the best predictors of arrest outcomes. Whether the warrant was served by policeunit_id 149 (narcotics unit) was the single most pertinent input feature for the model. Hour, day of year, and month were also highly pertinent, suggesting the existence of some cyclical dependencies in arrest frequency. Finally, district ID was a poor predictor of arrest outcomes. Latitude and longitude might have been more useful. These findings are robust to random restarts.

The MLP model was poor at recovering the data_recovered attribute. This column is highly skewed towards false, which the decision forest is more robust to. The confusion matrix for this task reveals that out of 521 true values in the validation set, only 121 were correctly predicted as true by the neural network, meaning the network has essentially learned to always predict false. This issue could potentially be mitigated by oversampling true values.

We then continued by performing some additional feature engineering to attempt to extract more useful features that the network could be trained on. For these trials, we did not perform an assessment of permutation importance as the computational cost increased considerably after concatenating more columns. We did an alternative trial using the cleaned judge name column. With judge names concatenated, we see the following training loss curve for the 64 wide network.



This indicated overfitting due to the validation accuracy being only 61.74% after 500 iterations. After this, we reduced the number of training epochs from 500 to 100. Even with this manual early stopping, adding judge names did not improve validation accuracy. We show the training loss curves for these trials in the following diagram:



Note that even with the low number of training epochs, the network is still training to the point of overfitting. After 100 trials, the small network achieves a validation accuracy of 62.08%. The large network performs comparably (62.23%). Full results can be viewed in the .txt log files in the src directory of the project.

Our final trial was one in which pre-execution approver names were used instead of judge names. This column lists the name of the sergeant or higher ranking officer who oversaw the preparations for executing the warrant. For these trials, we also used 100 training epochs. This addition improved validation set accuracy by only about 1% over the baseline (63.08% validation accuracy for the small network, 62.86% for the large).

We conclude some more powerful machinery seems necessary to properly utilize the information content of the judge and approver names. Both of these are sparse distributions, with most individuals participating in few warrants. The overfitting in both augmented data sets seems to support this analysis. Some natural language processing techniques could be of use here. This naive fully connected classifier is not very effective at resolving the complex interdependencies of the name columns in this data set. The approver name column lends itself to further data engineering, as these names can be looked up in data_officer. This is a reasonable option if a more robust network architecture can be found for this task. Nevertheless, the network does seem to be able to identify some patterns both in differential arrest distributions for specific police units and cyclic temporal dependencies at multiple levels of granularity.