**PREDICTING MENTAL HEALTH-RELATED DISPOSITIONS AND SENTENCES FROM COOK COUNTY COURT DATA**

**Karmen Hutchinson (kah771), Kelsey Markey (kcm312), Alene Rhea (akr435), Angela Teng (at2507)**Center for Data Science, New York University

***Abstract*** People living with mental illness are especially likely to have encounters with the law; they need dedicated resources and thoughtful treatment as they make their way through the criminal justice system. This project aims to predict mental health-related dispositions and sentences from a set of judicial and case-based features available only at initiation. Early detection of people who are likely to be suffering from mental health illnesses will enable governments and other institutions to provide appropriate support to these people as early as possible.

1. **Business Understanding**

Mental health disorders are three to six times more common among individuals involved in the criminal justice system compared to the general population (Blandford & Osher, 2012). It has also been shown that individuals with mental health disorders spend significantly more time in jail and are nearly twice as likely to be reincarcerated within one year of release, as compared to those without a mental health disorder (Haneberg & Watts 2016; Eno Louden & Skeem 2011). This creates a detrimental environment for individuals with mental health disorders and creates a problematic cycle where they are released into the community only to likely be returned to the justice system in the future.

The goal of this project is to lessen the harmful effects of the movement through the legal system on individuals with mental health disorders, while also minimizing the cost incurred by the county. To do this we aim to predict the likelihood that an individual is suffering from a mental health disorder, as soon as they are initiated into the legal system (without the need for medical records or training). Identifying individuals pre-trial allows for swift and appropriate interventions (i.e. jail-diversion interventions) and resources (i.e. intensive case management programs) so as to avoid a continued involvement with the legal system (Behavioral Health Innovations, 2015).

To maximize impact, this project uses data from Cook County, Illinois, where the number of individuals with mental illness in the Cook County jail has been reported to be as high as 30%, exceeding the national average by nearly 10% (CITE). Cook County is also at the forefront of specialty treatment courts and programs that identify eligible individuals early and link them to community-based services so as to increase successful probation and community reentry, such as the Mental Health Court program (CITE). However, induction into the Mental Health Court Program requires a current case with the health department and happens relatively late in the legal process. To avoid prolonged engagement with the legal system, this project aims to provide earlier detection of mental health disorders so that at initiation, individuals can be provided additional resources and support.

1. **Data**

Our data comes from the Initiation, Dispositions, and Sentencing datasets available on the Cook County Open Data Portal (<https://datacatalog.cookcountyil.gov/>). The researchers have limited the scope of the project to a single large county so that the laws and processes that apply to the area will be uniform. Cook County is also a good choice because it’s very populous, and has a well-kept open data portal that contains detailed metadata.

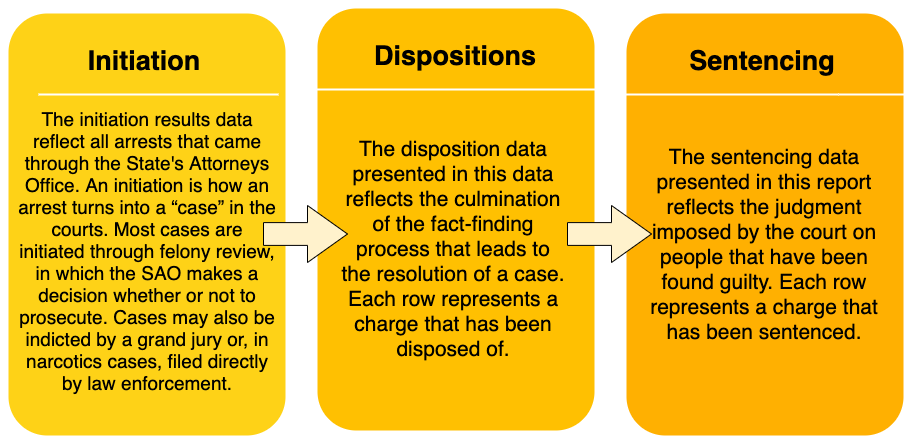


Figure 1: Descriptions of Cook County legal datasets used in this study.

These three datasets contain multiple identification numbers which can be used to link the records between them. The researchers are interested in making predictions at the level of the individual, so we have used case\_participant\_id as our fundamental identifier. Each case\_participant\_id can be linked to multiple charges, with each charge appearing as a separate row in the datasets. In the sentencing dataset, a charge can appear as multiple rows if re-sentencing has occurred.

The researchers have chosen to limit our training data to the 27 columns present in Initiation (link appendix), in order to simulate the use case. Excluding age\_at\_initiation and the \_#\_ of datetime columns, all \_#\_ of these features were categorical text variables

While there are a few columns that aren’t interpretable, all attempts to contact representatives from Cook County for clarification failed.

**Target Variable: MHI**

Since we are interested in classifying individuals based on mental health, we need a binary target variable that indicates whether or not an individual was identified to have a mental health disability. To construct our target variable, which we’ve named Mental Health Indicator (MHI),

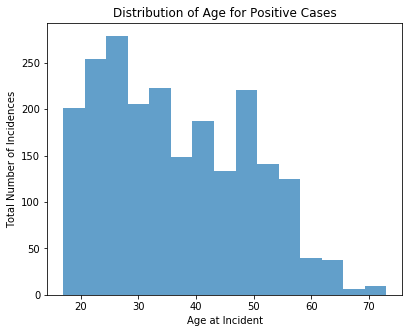
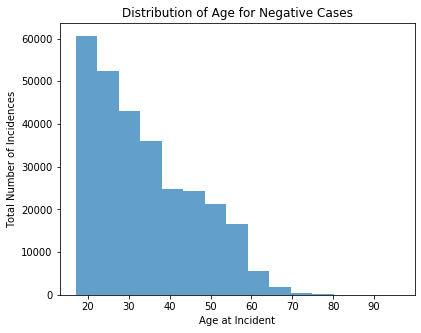
The researchers performed an exhaustive analysis of all the values in Sentencing and Disposition which indicate a mental health-related outcome. We identified \_\_#\_ values in \_#\_ columns which reveal an individual is mentally ill (link appendix here).

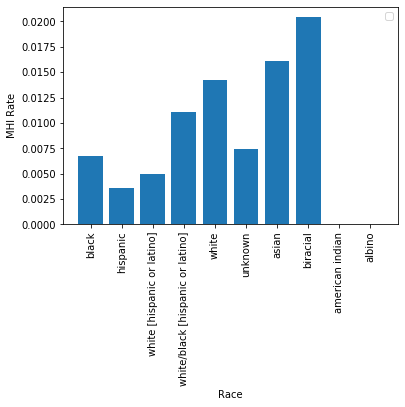
First we marked each of the rows in Sentencing and each of the rows in Disposition

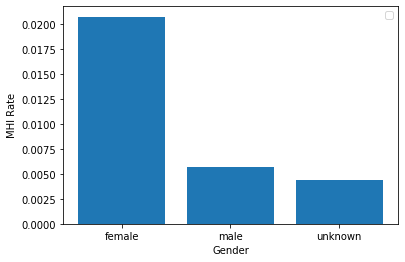
Since there may be multiple rows pertaining to a case\_participand\_id, we created a separate dataset that contains one row for each unique case\_participant\_ID, along with the corresponding MHI. Each unique ID was assigned a 1 if any rows corresponding to that ID had an MHI of 1.

If such an instance was found, for example the individual was sentenced to a mental health facility, the observation was assigned an MHI of 1. If no such instance occurred, the observation was assigned an MHI of 0.

**MHI Distribution**

****

****

****

**Filtering**

Beginning with the initiation and disposition data, we removed all IDs from initiation that didn’t appear in both sets. This ensures that we have a full set of information pertaining to each unique ID and that our model isn’t trained on partial processes. Our dataset is fairly large, so incomplete observations don’t contribute to the model.

**Cleaning**

Cleaning the filtered dataset consisted of changing data types, filling in missing information, and verifying that inputs within a column are uniform. Given that all information is recorded by many different arresting officers, there exist uniformity issues within the data. To remedy this problem, all string variables were converted to lowercase, all missing or outlying numeric values were replaced with medians, and all missing non-numeric inputs were replaced with ‘unknown’, which is the commonly used filler already found within the data. (Describe filling all nan values… distribution charts)

We paid close attention to how we dealt with both age and date/time features…(1900 & other specific stuff) . After our pseudo-baseline random forest indicated that age was an extremely important factor (see below), we considered building a separate model to predict and impute the missing ages. This was deemed infeasible given the time constraints of the project.

Since standardization is a common requirement for many models, the researchers used StandardScaler to transform our data such that its distribution will have a mean value 0 and standard deviation of 1. All numerical columns (age, charge count, incident length) were scaled as well as the categorical features that varied within unique case\_participant\_id groups and were summed (chapter, act, class, section, charge, etc.).

**Dummy Variables**

Followed standard practice of getting dummy variables for our categorical variables.

**Aggregation**

In order to turn the multiple rows per case\_participant\_id from the initiation dataset into one row per case\_participant\_id, we needed to grouped by case\_participant\_id and then aggregate those rows.

Aggregation can only be performed on numeric variables, so we removed all features that were still not numeric after getting dummy variables (i.e., datetime features).

To determine which functions to apply during aggregation, the researchers divided the columns into those which were always consistent within case\_partipant\_id groups, and those which sometimes had different values within a group. During aggregation, we took the median of the consistent categorical columns (which was functionally equivalent to taking the min, max, or mode), and we took the sum of the inconsistent categorical columns. We made a special case for charge\_count, where we took the max.

same: ['case\_id', 'case\_participant\_id', 'offense\_category', 'event', 'event\_date', 'age\_at\_incident', 'gender', 'race', 'incident\_begin\_date', 'arrest\_date', 'law\_enforcement\_agency', 'received\_date', 'arraignment\_date', 'updated\_offense\_category', 'incident\_city', 'unit', 'incident\_end\_date', 'age\_over\_100', 'age\_unknown']

different: ['primary\_charge', 'charge\_id', 'charge\_version\_id', 'charge\_offense\_title', 'chapter', 'act', 'section', 'class', 'aoic', 'charge\_count', '402']

**Finalizing column list**

Added datetime columns back into the aggregated dataset. (This was possible because they were always consistent within CP ID groups.) Removed all ID numbers -- removed case\_id, charge\_id, and charge\_version\_id, and set CP ID to index.

**Downsampling**

The dataset is extremely class imbalanced. (See above.) In order to help our models learn to identify positive classes our dataset, we downsampled the negative cases in our training set using random stratified sampling. (Given access to greater computing resources, the team would have liked to also experiment with upsampling.) The researchers used 100% of the positive instances, and sampled without replacement from the negative instances. Initially, we downsampled the negative instances such that the positive instances comprised 50% of the training set population. In tuning our final model, we tested this ratio and confirmed that it was ideal (see below). The validation and test sets were not downsampled, to replicate deployment.

**Training, Validation, and Test Sets**

The researchers chose to split the dataset into a training set with 85% of the data, a static validation set with 15% of the data, and a test set with 15% of the data.

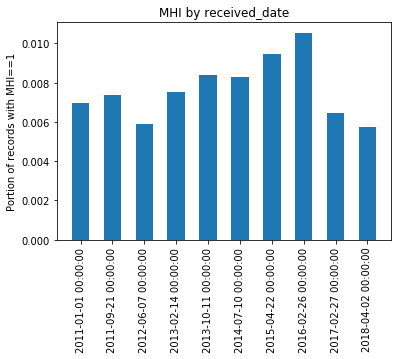
To simulate the deployment environment, in which our model will be used to predict forward in time, we partitioned our training, validation, and test sets based on received\_date. (The cases with the earliest received\_date became our training set, and the lastest cases become our test set.) Received\_date was chosen over other datetime columns, because it was the only one without missing values and because it replicates the use case in which individuals are evaluated when their cases are received by the SAO.

Partitioning based on time also helps prevent data leakage. This is especially important because the researchers have no way of linking the records of individuals who have multiple cases -- a person is assigned a new case\_participant\_id every time they re-enter the system with a new case. Ideally we would not have one individual appear in both the training and test sets, but by dividing the dataset based on time, we can at least ensure that we won’t predict an individual’s past based on data from their future.

Because we wanted to downsample the training set, but not the validation set (see above), we were unable to use scikit-learn’s built-in TimeSeriesSplit method. With more time, the team would have liked to implement our own walk-forward cross-validation method, downsampling the training set for each fold.

In forgoing random sampling, we may be exposing our models to bias induced by type-1 censoring. We hypothesize that older cases may have a higher base rate because they have had more time to be assigned an MHI of 1. Individuals can be re-sentenced multiple times, and there is no way for the researchers to mark a case as complete or incomplete. The distribution of MHI across time deciles bears out this hypothesis. (See plot.) The implications of censoring bias and concept drift will be discussed in the deployment section.

|  |  |
| --- | --- |
| **Training, Validation, and Test Sets** | |
| Training base rate | 0.0077 |
| Validation base rate | 0.0092 |
| Testing base rate | 0.0060 |



**PCA**

In general, it is not advisable to feed a large number of features directly into a machine learning model, since some features “may be irrelevant or the ‘intrinsic’ dimensionality may be smaller than the number of features.” (SAS BLOG) Thus, a couple of dimensionality reduction methods, such as principal component analysis (PCA), singular value decomposition (SVD), and latent Dirichlet allocation (LDA), can be used to avoid the curse of dimensionality, and other issues such as overfitting in high dimensional space and running out of computational power (Nagpal, 2017).

The researchers decided to focus on PCA, because of its simplicity, efficiency and non-parametric applications for extracting relevant information from perplexing datasets (Shlens, 2014). In particular, PCA “reduces the dimension of [a] data[set] with the aim of retaining as much information as possible… this method combines highly correlated variables together to form a smaller number of an artificial set of variables which is called “principal components” that account for most variance in the data.” (Nagpal, 2017). In particular, for this research project, the researchers used sparse PCA, a specialized technique that extends the classic method of PCA by introducing sparse inputs (in this case, matrices that contain one hot encoded columns that are comprised of many 0 values) (Lu, 2019).

**V. Modeling & Evaluation**

**Evaluation Metrics**

We focused on two performance metrics in the evaluation of our models: (1) area under the receiver operating curve (AUC), and (2) sensitivity. The researchers believe the cost of a false negative in our use case to be significantly higher than the cost of a false positive; to miss an instance of mental illness could be detrimental to that individual, but to offer support and services to an individual without a particular need is likely to incur only a marginal cost. However, without specific knowledge of Cook County’s available resources and budgetary constraints surrounding mental health, we sought to provide a model which could perform well at various thresholds. We therefore optimized our models with respect to AUC, paying close attention to the effects on sensitivity at each iteration. This choice is bolstered by the knowledge that due to our low base rate (0.77%), accuracy would be a non-ideal metric since it could be very high even if the minority class was not well predicted. AUC on the other hand is more appropriate for our business goal since it is sensitive to class imbalance in the sense that it treats the minority class with as much weight as the majority class.

**Baseline Metrics**

Optimizing for AUC provides a natural baseline, as an AUC of 0.5 represents a model which assigns class probabilities randomly (Brownlee, 2019). The expected sensitivity of such a model, using a 50% probability cutoff, would also be 0.5. Bringing that threshold down to 0% (i.e., assigning every case to the positive class) would be the easiest way to maximize sensitivity; indeed the sensitivity of such a model would be 1. This further illustrates the reason it is better in this case to optimize for AUC rather than for sensitivity: our goal is not simply to identify positive instances, but to do so with minimal type I errors.

**Pseudo-Baseline Model**

The first model the researchers ran was a random forest model with out-of-the-box parameters, fit on our cleaned and downsampled training set. (We call this a “pseudo-baseline,” because at this stage we had already invested significant time into data munging.) An ensemble tree-based method was chosen as a baseline because they are known to perform well on categorical variables (Tutuz, G and Berger, M, 2017). The team treated this model as a baseline from which to start, before feature engineering and hyperparameter tuning. This unrefined model yielded an AUC of 0.78 and a sensitivity of 0.78.

**Feature Engineering**

The feature importances demonstrated in our pseudo-baseline random forest (see Appendix B, Table 1) guided much of our initial feature engineering. For example, ‘Section 402(c)’ was within the top fifteen feature importances and further research indicated that this corresponds to legal sections related to narcotics or possession of narcotics (Illinois Secretary of State, Accessed 2019). As such, the researchers engineered a new indicator variable to encode if the section column contains other ‘402’ sections outside of just 402(c). In order to look more granularly at the location in which the arrest took place, we geoencoded incident city to latitude and longitude in order to determine which regions of Chicago may be prone to mental health issues. Geocoding also ensures that our proxies for location are uniform throughout the dataset. There exists a specific code for values/places that are unknown, so both unknown and missing values are handled accordingly. A number of datetime features were also engineered in an attempt to better represent what we postulated might be relevant relationships between mental health incidents and time. We first created a binary feature to encode whether the arrest date fell on a weekday (positive label) or a weekend. We also encoded the arrest date into season, and one hot encoded these to be binary columns for each season. We also created a feature for incident length, by calculating the distance between the incident begin and end dates. Finally, since age at incident had nearly 4% missing values and approximately 40 (unrealistic) outlier ages that were over 100 we created a binary feature for whether age at incident was null and another for whether age was over 100.

**Algorithm Selection**

The team identified five algorithms to explore: logistic regression, decision trees, random forest, gradient boosting, and support vector machine (Table 1). Initial performance for each of these models was established using out-of-the-box parameters on cleaned, scaled, and downsampled data, after all feature engineering was complete. The researchers chose not to experiment with a k-nearest neighbors model (kNN) because of the high dimensionality of the dataset. In such cases, instances which may in fact be similar can have very large distances, and so kNN would perform poorly (Brownlee, 2016). The researchers also decided not to implement a Naive Bayes Classifier, because the use of dummy variables to encode categorical data explicitly violates the algorithm’s assumption of conditional independence.

|  |  |  |
| --- | --- | --- |
| **Out-of-the-box Model Performance** | **AUC** | **Sensitivity** |
| **Random Forest without Feature Selection** | 0.78 | 0.78 |
| **Logistic Regression** | 0.72 | -- |
| **Random Forest with engineered features** | 0.79 | 0.75 |
| **Support Vector Machine** | 0.76 | 0.46 |
| **Gradient Boosting** | 0.82 | 0.79 |

Table 1: Performance metrics on validation set using out-of-the-box parameters

**Support Vector Machine**

Often viewed as the general purpose algorithm for machine learning, we choose an SVM as one of our exploratory models because of its ability to capture complex relationships through linear or non-linear kernels. However, the SVM took significantly longer to train than any of our other models, likely because of our high number of features and the constrained optimization problem that backs SVM (Ragnar, 2016). Additionally, it did not yield results that justified the long training time. The researchers determined that the run-time and the extremely low sensitivity (0.46) of the out-of-the-box SVM model meant that it would not be a candidate for hyperparameter tuning.

**Logistic Regression**

Logistic regression was chosen for its robustness, reliability, and intuitive interpretation. Moreover, logistic models are relatively easy to update with new data, using the method of stochastic gradient descent, and can easily be regularized to avoid overfitting (Li, 2017). After making appropriate transformations and prior to tuning, the model failed to converge when using all ~4800 features. Increasing max iterations, testing different solvers, and testing different non-linear transformations all failed to get the model to converge. Assuming that multicollinearity may be an issue, we reduced the number of columns to the top ten feature importances from our random forest and found that the model (with solver = ‘liblinear’ and C = 1e30) was able to fit the data with an AUC of 0.72. However, we were unable to estimate recall on this model because…...(add piece/link about recall not working).

Karmen add info about justification for not tuning/using logreg: i.e. recall doesn’t work (bc AUC of 0.72 is totally respectable for 10 features, so that can’t be the reason that we didn’t pursue it)

**Interpretable Decision Tree**

Decision trees are easily scalable and are able to model non-linear and categorical variables relatively well. Although ensemble methods usually outperform decision trees on key metrics, singular decision trees can provide valuable transparency. The researchers decided to experiment with creating interpretable decision trees because transparency is especially important in the context of the problem at hand. Models employed by government to aid decision making are subject to scrutiny by the public, so the ability to extract an intuitive set of rules to explain their decisions may be worth a decrease in performance metrics.

Trees were trained on unscaled data so that numerical values would be interpretable. The researchers iterated values of max\_depth (2, 3, 4), min\_samples\_leaf (1, 10, 100, 500), and max\_features (10, 5, 3, None). The best combination of hyperparameters turned out to be max\_depth=4, min\_samples\_leaf=10, and max\_features=None, with an AUC of 0.75 and a sensitivity of 0.86. The resultant tree, which is redundant and needs pruning, can be found in Appendix C.

**Random Forest**

After feature engineering the researchers again evaluated a random forest model with default parameters and found that the AUC rose to 0.79 and sensitivity lowered to 0.75. The top 11 feature importances for the models with and without engineered features are plotted in Table 1 in Appendix B.

Considering the high performance of the model, the team decided to continue tuning the random forest model, both with hyperparameter tuning and feature extraction (since the team’s original model was extremely wide with more than 4800 features). The researchers began with tuning of hyperparameters and tested a range of max\_depths (None, 3, 10, 30), min\_samples\_leaf (2, 50, 100, 200), and n\_estimators (10, 100, 500, 1000). We found that under these conditions AUC was optimized at 0.816 with max\_depth = None, min\_samples\_leafs = 2, and n\_estimators = 1000. We then decided to investigate with various PCA components and tested each of the same tree parameters with PCA components of 1, 3, 100, and 1000, as well as without PCA. Results of this are shown in table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **AUC** | **Recall** | **PCA components** | **Tree Parameters** |
| **Highest AUC** | 0.801 | 0.698 | 3 | Max\_depth=10, min\_samples\_leafs=2, n\_estimators= 100 |
| **Highest sensitivity** | 0.538 | 0.836 | 1 | Max\_depth=3, min\_samples\_leafs=100, n\_estimators=1000 |
| **Highest combined AUC & sensitivity** | 0.794 | 0.736 | 100 | Max\_depth=None, min\_samples\_leafs=2, n\_estimators=1000 |

Table 2: Random forest tuning and performance on validation set

**Gradient Boosting Model:**

LINE ABOUT WHY GRADIENT BOOSTING. The out-of-the-box gradient boosting model had the highest AUC of 0.82, with a sensitivity of 0.76 and thus was a logical candidate for further hyperparameter tuning.

Optimizing for sensitivity, because of the business needs defined above, the researchers found that a gradient boosting model with parameters: {max\_depth: 3, min\_samples\_leaf: 10, learning\_rate: 0.01, n\_estimators: 10} achieved the highest sensitivity, of 0.87, and a relatively average AUC of 0.72, in comparison to the other tree-based models tested.

**Tuning the winning GB model**

Tested winning GB models for diff levels of downsampling; 20% had terrible recall; 10% had fully abysmal recall; didn’t bother testing without any downsampling; confirmed that 50% was best.

Left all variables unscaled bc time. Scaling shouldn’t matter for tree models, but just to make sure i tested 50% as one of the downsampled levels -- this basically duplicated the winning GB model, confirmed that scaling doesn’t matter

**Final model:**

Took all test & val, downsampled to 50%, tested on test. No scaling. AUC was v good

* **Discuss why and how this model should “solve” the business problem (i.e., improve along some dimension of interest to the firm).**

**VI. Deployment**

* **Discuss how the result of the data mining will be deployed.**
* **Discuss how it should be monitored and evaluated in an actual production system.**
  + **Discuss threshold… FN worse than FP?**

**This is about mental health court, make that more clear. In order to get into mental health court you need to have a pre-existing case with the health department**

It’s important for firms to keep in mind that the model isn’t fool proof. First, we can only detect cases of mental illnesses that have been identified by the courts, giving the court full control over what’s considered a mental health disability. This also means that pre-existing cases with the health department must have existed prior to an arrest in order for it to be documented at the time of the arrest. This may create a disadvantage for demographics that either cannot afford mental healthcare in general or are systematically misdiagnosed by healthcare providers. Prior research shows that certain demographics are less likely to be taken seriously by healthcare professionals (link article), and thus some individuals may not be linked to mental health issues they may indeed have. One way we hoped to capture this effect was by altering the entries for race as little as possible. We did not combine groups or filter particular entries, as how an individual’s race is perceived by the arresting officer could potentially play a part in how that individual is able to navigate the judicial system, especially in terms of their mental health.

In a more general sense, the model doesn’t take into account an individuals history within the Cook County judicial system. Thus it is advised that firms pay close attention to individuals belonging to historically underrepresented and disadvantaged groups, as well as the nature of the crime committed. This will ensure that firms are able to capture at least some individuals that may be overlooked by the model.

* **Identify the risks associated with your proposed plan and how you would mitigate them.**
  + **Concept Drift -- drug court expanding capacity over time (citations for this in the research document) → increase in MHI rate over time**
  + **Censoring bias (the “rolling in” effect) -- could be solved by plotting MHI rates of fairly small bins of days\_since\_received, identifying a point at which the MHI levels off, and then using that as a “cutoff” heuristic. Ie, not training on data that is at least X old.**
  + **“Irreducible error”? Brian talked about this a lot last class, maybe just a line about this?**
  + **MHI 1 very low for updated offense category narcotics, and yet…. Mental health court contains lots of drug support. This seems inconsistent**

**CONCLUSION and Future Work**

1. **What it means that the data is bias (from Julia slides?).**
2. **This is not currently an implementable model because of the protected class (race, age, gender)**
   1. **We still have things in that could reconstruct the sensitive factors, so removing them wouldn’t solve the problem (and is why we left them in).**
3. **How to mitigate**

# 

# 

# Bibliography

Poularikas, Alexander D. (September 1998). *Handbook of Formulas and Tables for Signal Processing* (1 ed.). CRC Press. p. 42-8 <https://en.wikipedia.org/wiki/Upsampling#cite_note-1>

Shlens, Jonathon. “A Tutorial on Principal Component Analysis.” *ArXiv:1404.1100v1 [Cs.LG] 3 Apr 2014*, https://arxiv.org/pdf/1404.1100.pdf.

Johnstone, Iain M., and Arthur Yu Lu. “On Consistency and Sparsity for Principal Components Analysis in High Dimensions.” *Journal of the American Statistical Association*, vol. 104, no. 486, 2019, pp. 682–693., doi:10.1198/jasa.2009.0121.

Brownlee, Jason. “K-Nearest Neighbors for Machine Learning.” *Machine Learning Mastery*, 12 Aug. 2019, <https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/>.

Abhishek Sharma, and Abhishek Sharma. “Decision Tree Introduction with Example.” *GeeksforGeeks*, 25 Nov. 2019, <https://www.geeksforgeeks.org/decision-tree-introduction-example/>.

Li, Hui, and SAS Data Science Blog. “Which Machine Learning Algorithm Should I Use?” *The SAS Data Science Blog*, 12 Apr. 2017, <https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/#prettyPhoto>.

Nagpal, Anuja. “Principal Component Analysis- Intro.” *Medium*, Towards Data Science, 22 Nov. 2017, <https://towardsdatascience.com/principal-component-analysis-intro-61f236064b38>.

Botev, Zdravko, and Ad Ridder. “Variance Reduction.” *Wiley StatsRef: Statistics Reference Online*, 2017, pp. 1–6., doi:10.1002/9781118445112.stat07975.

Parikh, Rajul, et al. “Understanding and Using Sensitivity, Specificity and Predictive Values.” *Indian Journal of Ophthalmology*, Medknow Publications, 2008, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636062/.

Narkhede, Sarang. “Understanding AUC - ROC Curve.” *Medium*, Towards Data Science, 26 May 2019, <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

“Sklearn.metrics.recall\_score¶.” *Scikit*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\_score.html.

Brownlee, Jason. “Metrics To Evaluate Machine Learning Algorithms in Python.” *Machine Learning Mastery*, 21 Nov. 2019, https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/.

Yiu, Tony. “Understanding Random Forest.” *Medium*, Towards Data Science, 14 Aug. 2019, <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>.

“Modern Machine Learning Algorithms: Strengths and Weaknesses.” *EliteDataScience*, 25 Jan. 2019, <https://elitedatascience.com/machine-learning-algorithms>.

State, Illinois Secretary of. “Online Services.” *The Official Website for the Illinois Secretary of State*, <https://www.cyberdriveillinois.com/>.

Ragnar. “What Kinds of Learning Problems Are Suitable for Support Vector Machines?” *Data Science Stack Exchange*, 1 Feb. 2016, https://datascience.stackexchange.com/questions/9736/what-kinds-of-learning-problems-are-suitable-for-support-vector-machines.

Kelsey citations

~~Baillargeon, J., Binswanger, I. A., Penn, J. V., Williams, B. A., & Murray, O. J. (2009). Psychiatric disorders and repeat incarcerations: The revolving prison door.~~ **~~American Journal of Psychiatry, 166(1), 103-109.~~**

**Behavioral Health Innovations. Mental Health and Justice in Cook County Bond Courts An Examination of the Management of Persons with Mental Illness in Felony Bond Court. Report prepared for the Administrative Office of the Illinois Courts, July 2015.**

**Blandford, A. & Osher, F. (2012). *A checklist for implementing evidence-based practices and programs (EBPs) for justice-involved adults with behavioral health disorders.* Delmar, NY: SAMHSA’s GAINS Center for Behavioral Health and Justice Transformation.**

**Eno Louden, J., & Skeem, J. (2011). Parolees with mental disorder: Toward evidence-based practice. Bulletin of the Center for Evidence-Based Corrections, 7(1), 1-9.**

**Haneberg, R., & Watts, K. “Stepping Up” to beat the mental health crisis in U.S. jails. Criminal Justice/Corrections. New York, NY: Council of State Governments Justice Center. Retrieved from http://knowledgecenter.csg.org/kc/system/files/Haneberg Watts 2016.pdf**

**Tutz, Gerhard, and Moritz Berger. “Tree-Structured Modelling of Categorical Predictors in Generalized Additive Regression.” *Advances in Data Analysis and Classification*, vol. 12, no. 3, 2017, pp. 737–758., doi:10.1007/s11634-017-0298-6.**

**Appendix A: Team member responsibilities**

**Kelsey:** Setting up APIs to read in data, feature engineering age and datetime features, baseline models, random forest models

**Karmen**: Geoencoding incident city, logistic regression

**Alene**: Project/idea formulation, filtering data, cleaning data, aggregating data, building time-based split, PCA, understandable decision tree model, tuning gradient boosting model, final model tuning and testing, MHI distribution charts

**Amber**: Baseline models gradient-boosted model, downsampling function,

**Appendix B: Model Comparison**

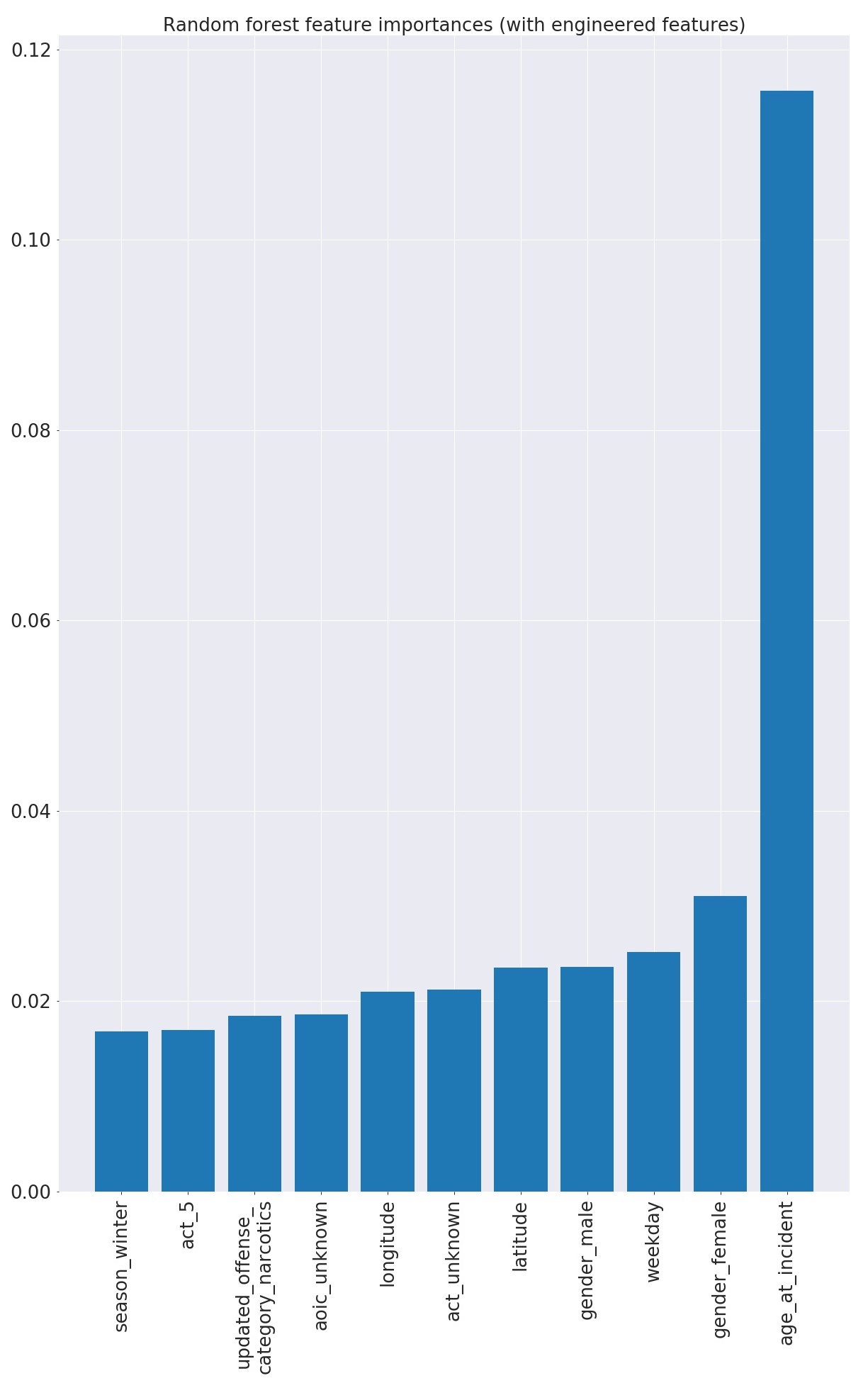
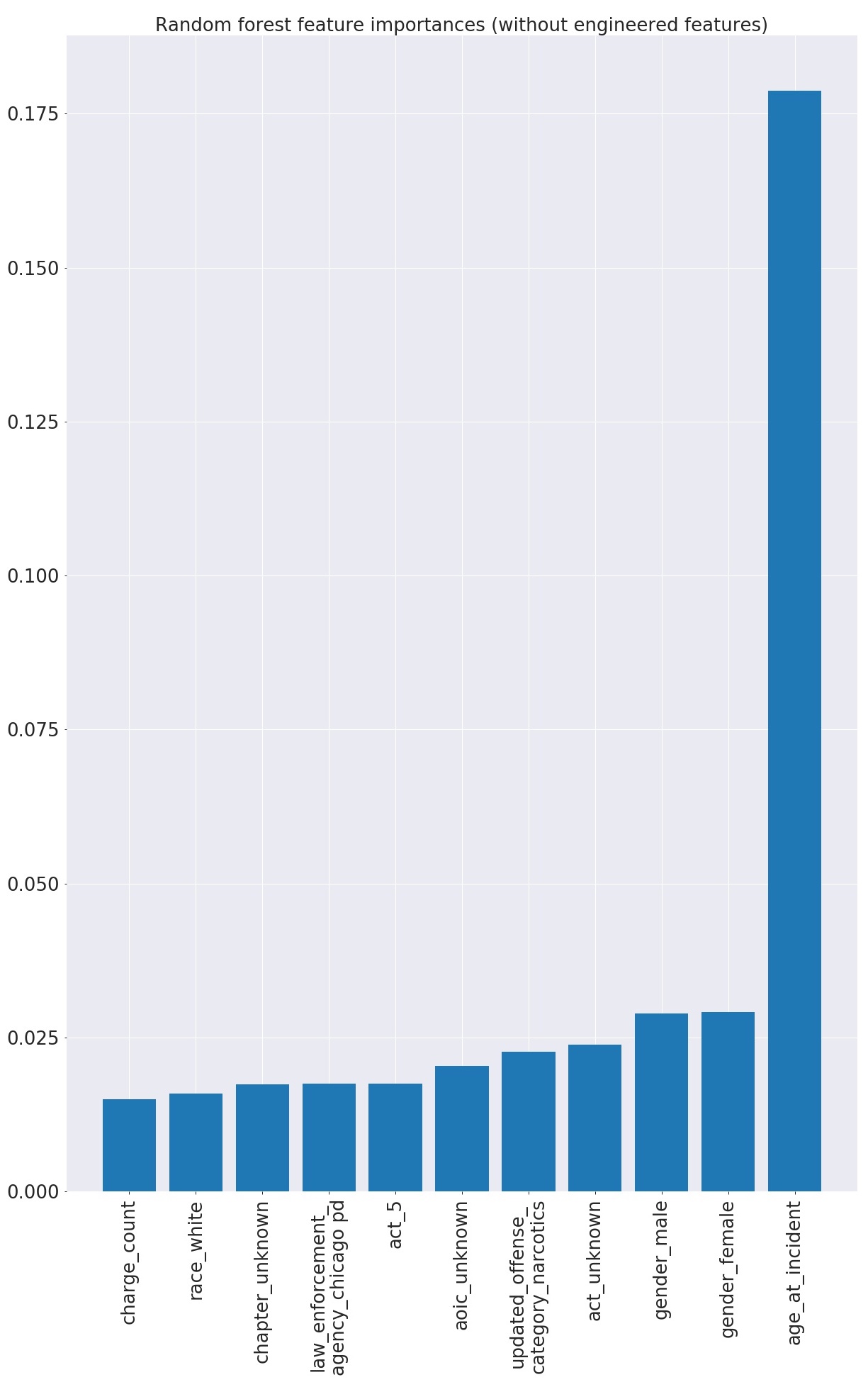


Figure 1: Feature importances of random forest models (with out of the box parameters) before (left) and after (right) engineered features

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Random Forest Model Hyperparameter Tuning on Validation Set** | **AUC** | **Recall** | **PCA n\_components** | **max\_depth** | **min\_samples\_leafs** | **n\_estimators** |
| **Highest AUC** | 0.801 | 0.698 | 3 | 10 | 2 | 100 |
| **Highest sensitivity** | 0.538 | 0.836 | 1 | 3 | 100 | 1000 |
| **Highest combined AUC & sensitivity** | 0.794 | 0.736 | 100 | None | 2 | 1000 |

Table 1: Random forest tuning and performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Gradient Boosting Model Hyperparameter Tuning on Validation Set** | | | | | |
| **AUC** | **Sensitivity** | **max\_depth** | **min\_samples\_leaf** | **Learning\_rate** | **N\_estimators** |
| 0.82 | 0.78 | 3 | 10 | Default | Default |
| 0.81 | 0.78 | 10 | 2 | Default | Default |
| 0.82 | 0.79 | 3 | 10 | 0.01 | 1000 |
| 0.72 | 0.87 | 3 | 10 | 0.01 | 10 |

**Appendix C: Supplementary Visualizations**

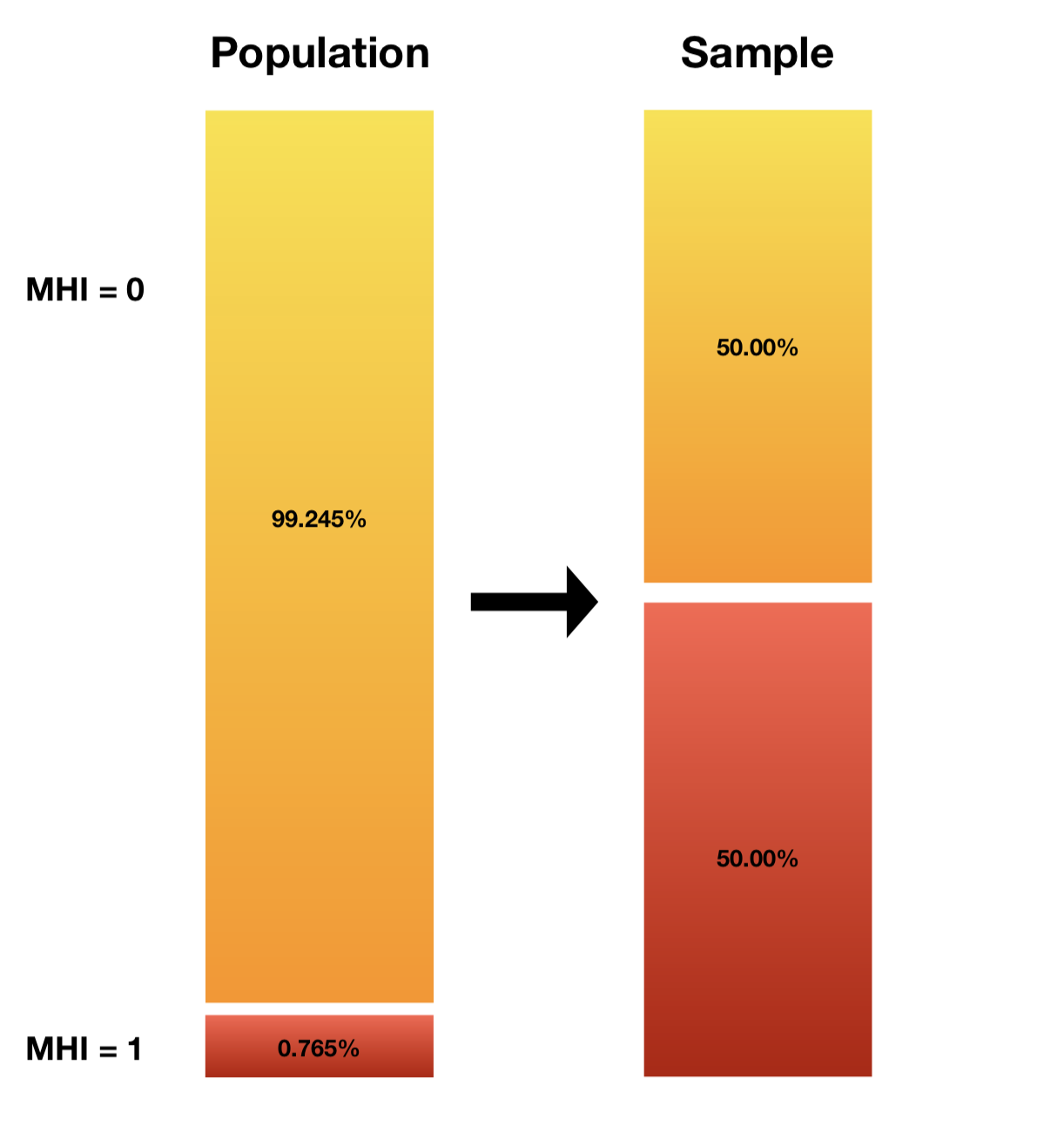


Figure 1: Class probabilities in the dataset population and after downsampling

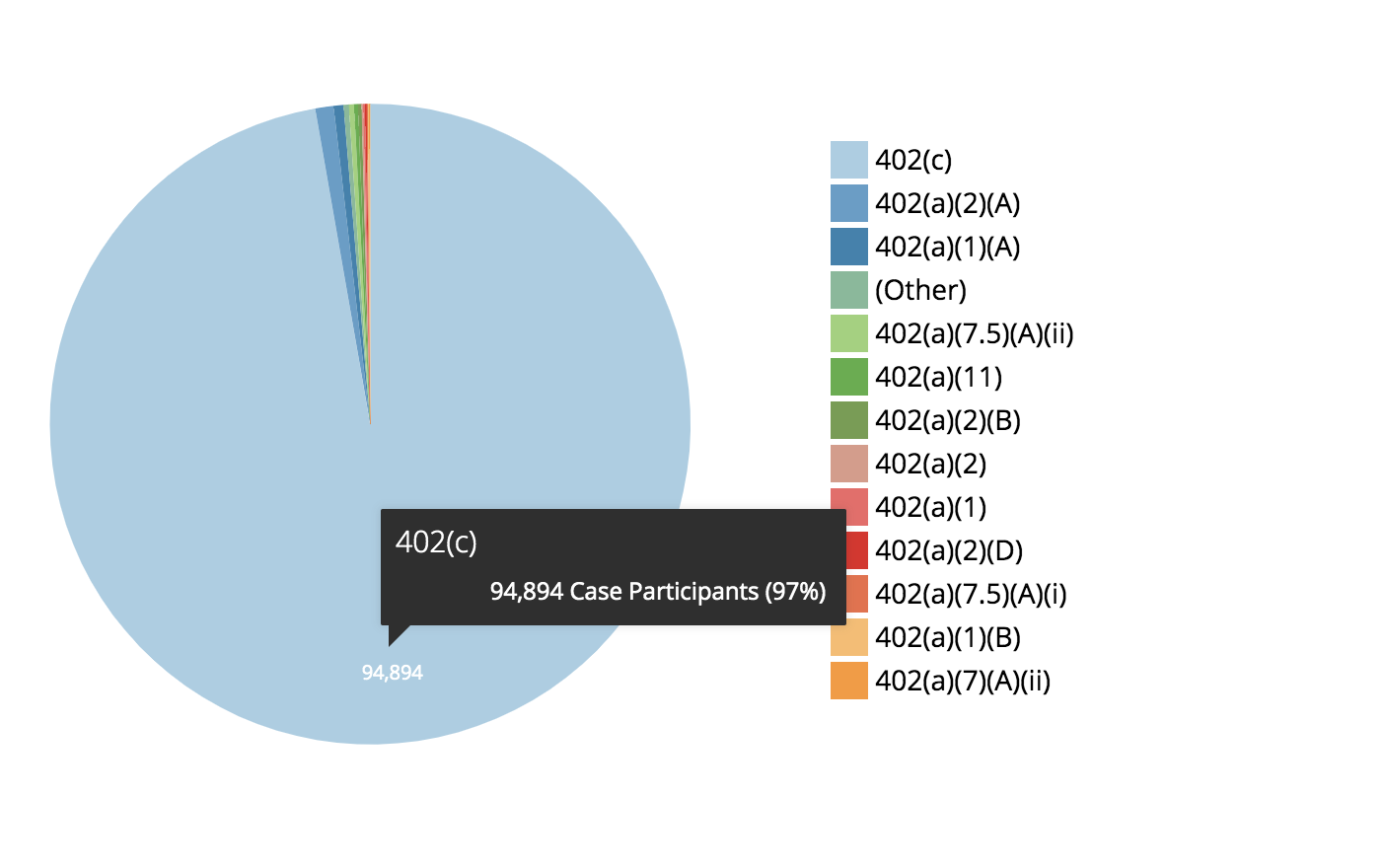
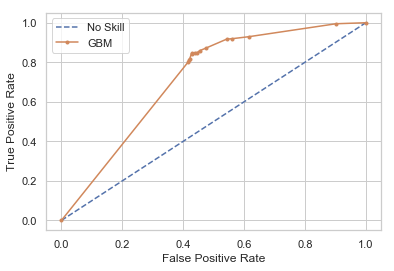
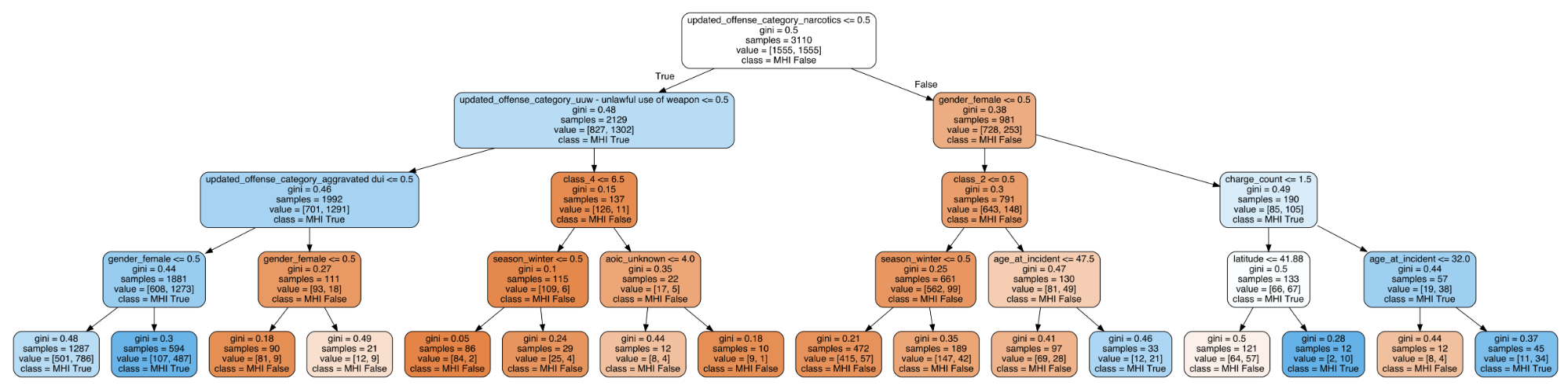


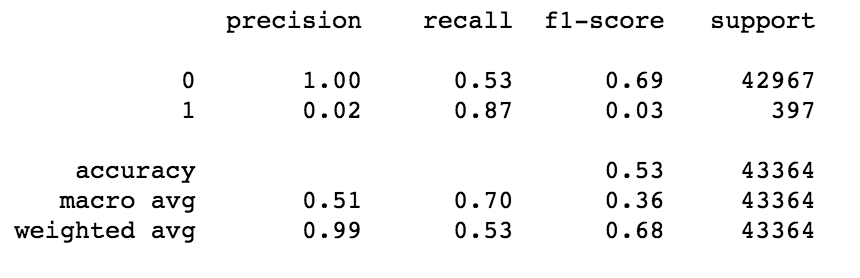
CHART XXXXX: FREQUENCY OF CASE PARTICIPANTS (INITIATION DATASET) BY 402 SECTION (<https://datacatalog.cookcountyil.gov/Courts/Initiation/7mck-ehwz>)



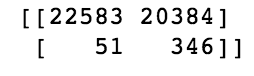
Understandable Decision Tree



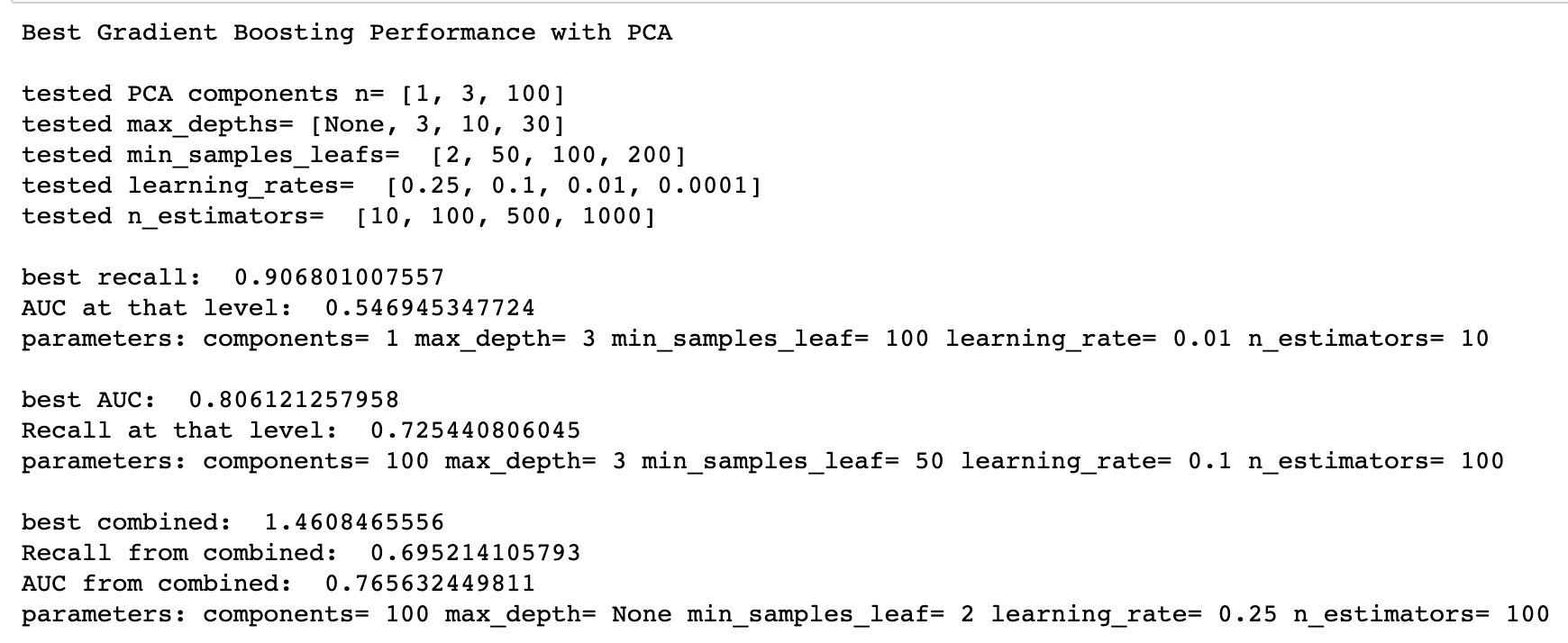
ROC AUC For Optimized Gradient Boosting Model (No PCA) on Validation Set



Classification Report for Optimized Gradient Boosting Model (No PCA)

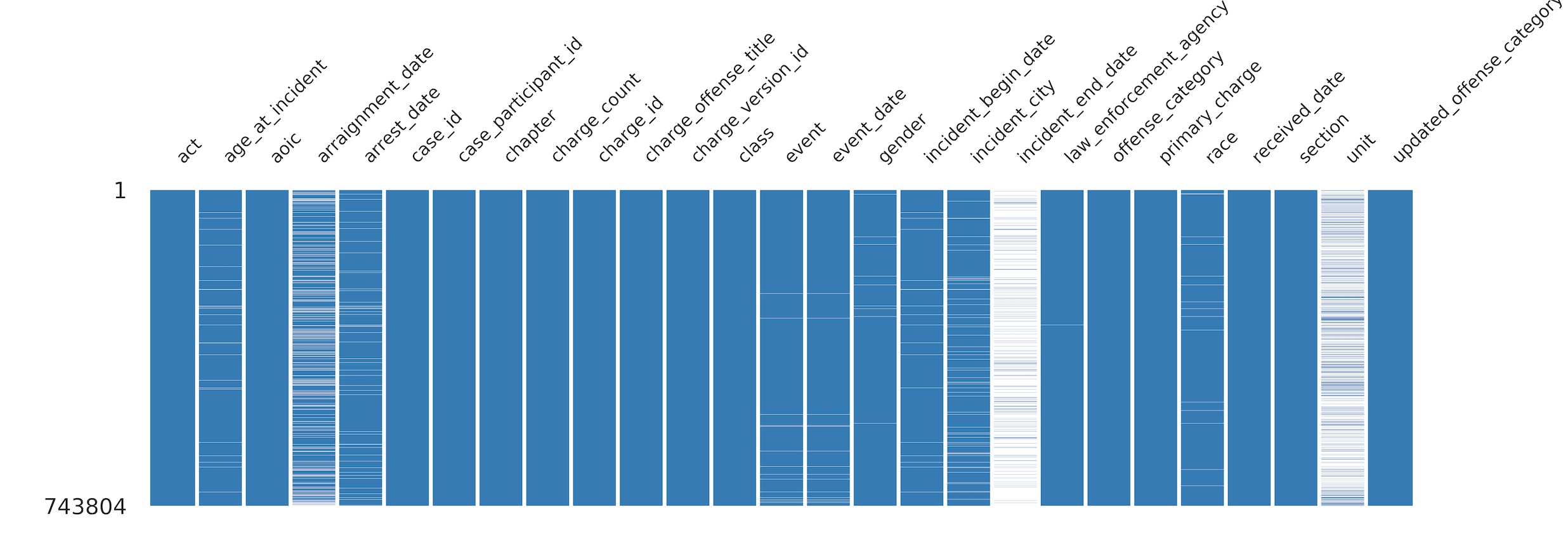
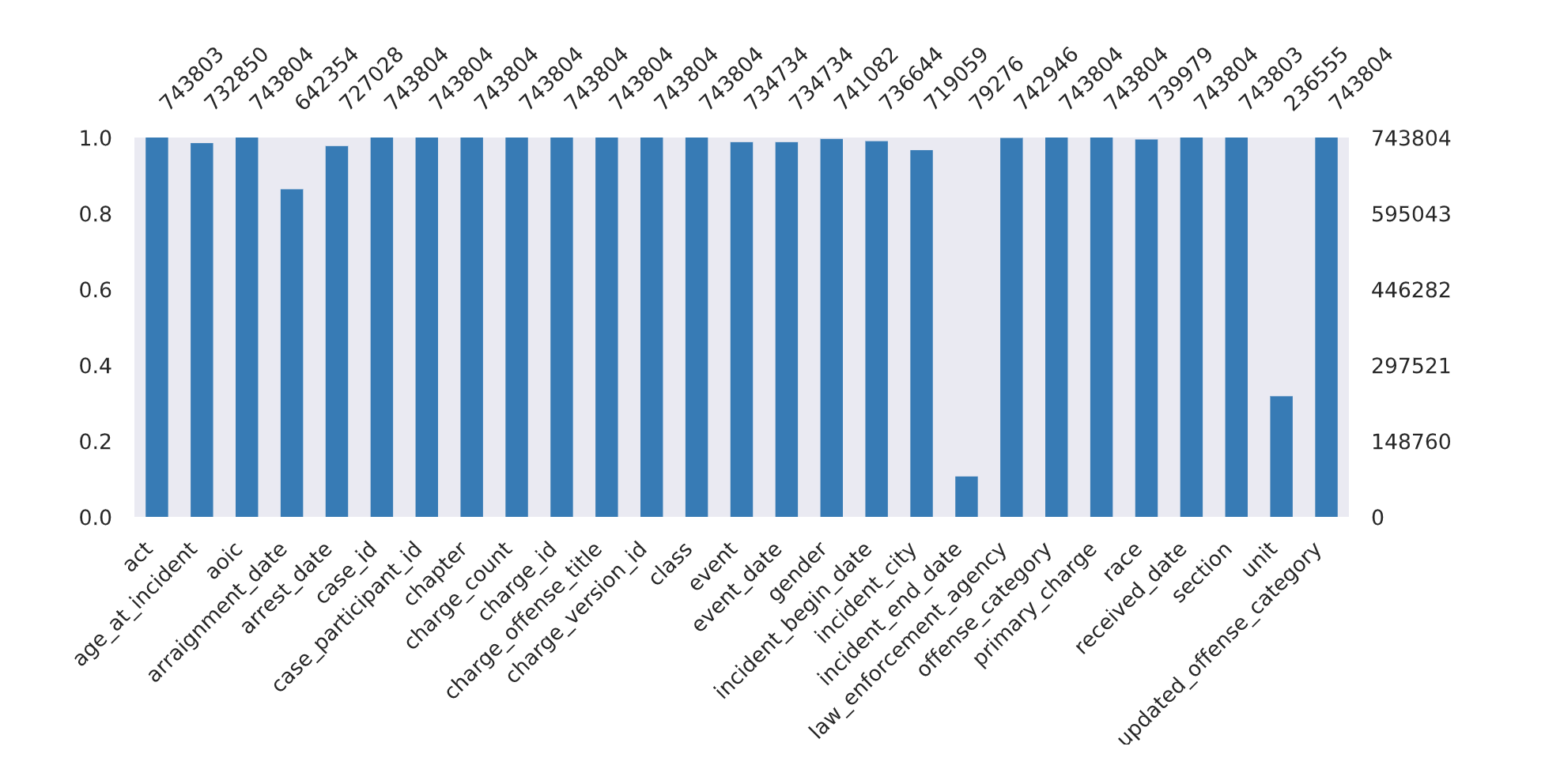


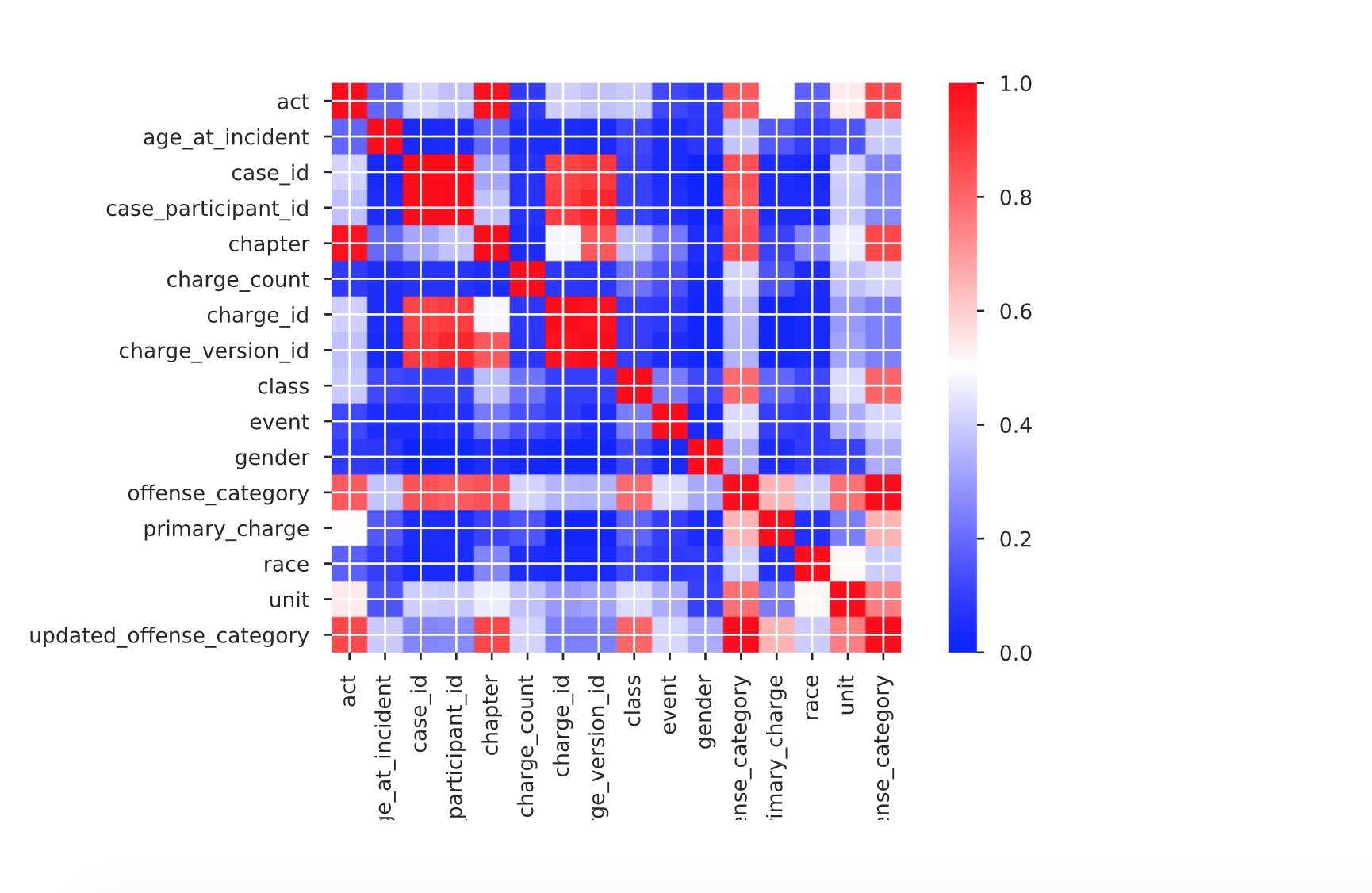
Confusion Matrix for Optimized Gradient Boosting Model (No PCA)



**Here are some things I finally got from Pandas.profiling on init\_filtered let me know any you think we should include or remake - KCM**

Missing values:

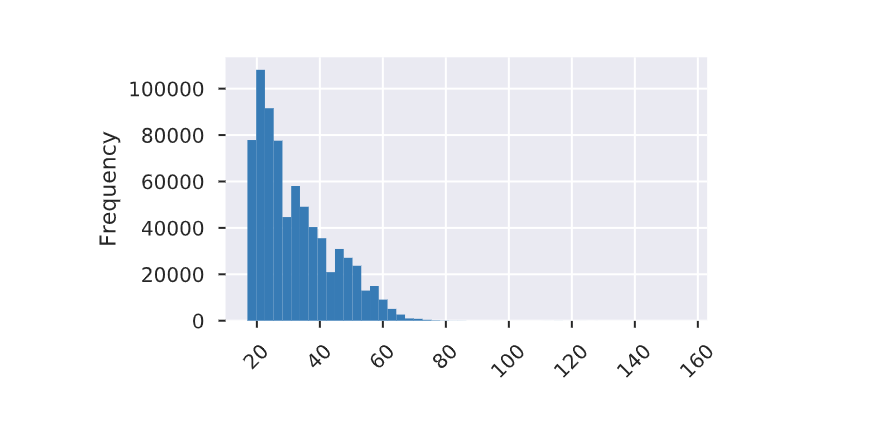


Correlations:

Some other thoughts:

1. Age is definitely not normal (standard scalar problem?)  
   Karm: if this is on init\_filtered, then we haven’t used the scalar yet? We use the scalar in init\_clean.

KCM: Right, but StandardScalar wants a normal distribution before it scales. And we don’t have one :/



1. Act 5 was a big feature on pseudo-baseline RF but turns out ~75% of data have act=5. Wonder what it means?

