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

**Alene Rhea (akr435), Angela Teng (at2507), Karmen Hutchinson (kah771), Kelsey Markey (kcm312)**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, see Loveland et al, 2007) so as to avoid a continued involvement with the legal system.

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% (Behavioral Health Innovations, 2015). 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 (Center for Health and Justice at TASC, Accessed 2019). 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 Understanding and Preparation**

Our data comes from the Initiation, Dispositions, and Sentencing datasets available on the Cook County Open Data Portal ([https://datacatalog.cookcountyil.gov](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.

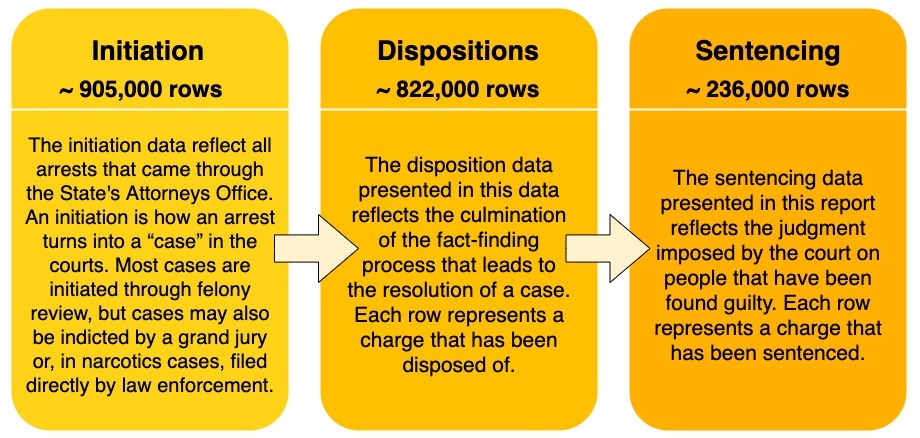


Figure 1: Descriptions of Cook County legal datasets used in this study.

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 (Table A1), in order to simulate the use case. Excluding age\_at\_initiation and the 6 of datetime columns, all 20 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 15 values in 6 columns which reveal an individual is mentally ill (link appendix here). After merging the two datasets and isolating the columns of interest, it was possible to identify which rows contained a proxy for MHI. 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.

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.

**MHI Distribution**

To better understand the distribution of MHI across protected features, or the effect of (in particular in protected classes such as age, race, gender (Table A1).

**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.

We paid close attention to how we dealt with both age and date/time features. Prior to converting ages and dates to their appropriate formats, integers and datetimes, it was necessary to represent unknown and missing values in a way that could also be formatted. Age contained outlying values that were greater than 100. Given that it is not likely to be arrested at such an old age, these values were replaced with the average. In order to preserve the usability of the column, all unknown or missing values were replaced with 0 before converting the entries to integer values. Missing and unknown dates were assigned a filler value corresponding to midnight on January 1, 1900. This is justified because....

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 (and in our case logistic regression and SVM), the researchers used StandardScaler to transform our data such that its distribution would 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.). Since extreme outliers can affect scaling, we made sure to examine feature distributions and handle outliers appropriately (for example in ages over 100, as mentioned above).

**Dummy Variables**

To deal with categorical variables, the team used one hot encoding. The method of taking dummy variables and converting categorical variables into binary features that represent the presence or absence of a category. Taking dummy variables allows the use of categorical variables in linear and parametric models (eg. non-tree based models such as logistic regression), while taking into account interactions.

**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 (Table A2). 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.

**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.

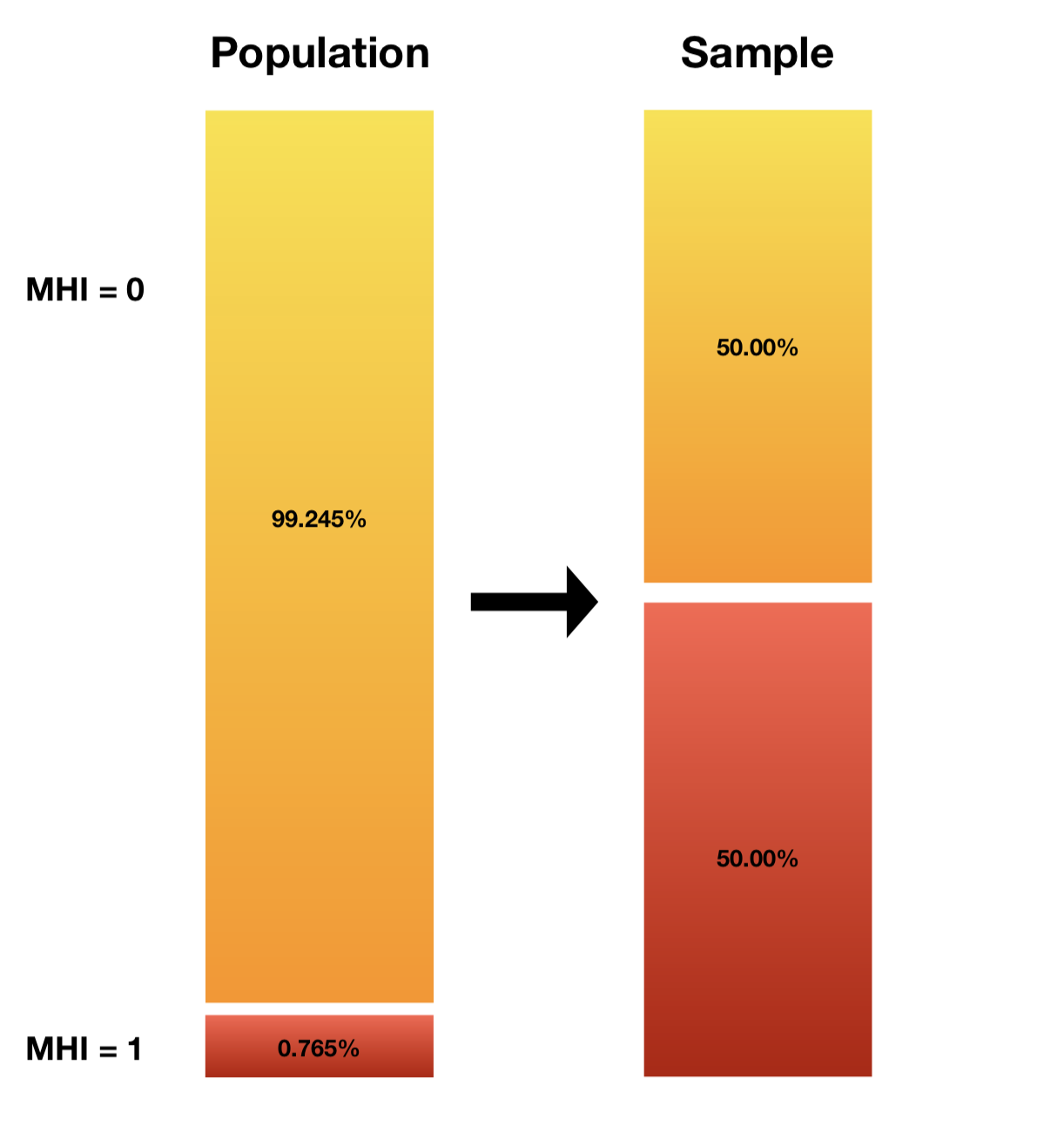


Figure 1: Class probabilities in the dataset population and after downsampling

**Training, Validation, and Test Sets**

The researchers chose to split the dataset into a training set with 70% of the data, a static validation set with 15% of the data, and a test set with 15% of the data (Shah, 2017). 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.

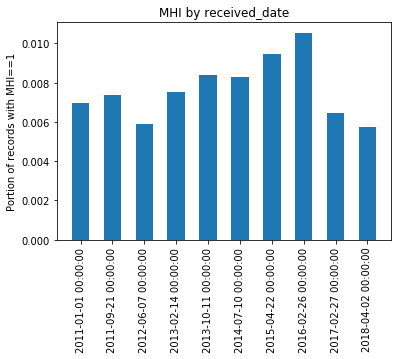
Partitioning based on time also helps prevent data leakage. This is especially important because the researchers have no way of linking the records of individuals who have multiple cases -- a person is assigned a new case\_participant\_id every time they re-enter the system with a new case. Ideally we 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 in that we see a notable dropoff in the top two deciles (see table X). There also appears to be some more gentle concept drift in the opposite direction, which may be explained by expansion of the Cook County Mental Health Court Program. The implications of censoring bias and concept drift will be discussed in the deployment section.

|  |  |
| --- | --- |
| Training base rate | 0.0077 |
| Validation base rate | 0.0092 |
| Testing base rate | 0.0060 |

Table 2: MHI base rate across the split datasets



**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.” (Li et al., 2017) 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 datasets (Shlens, 2014). After applying PCA to the training set, the researchers found that 96.4% of the total variance was explained in one component. This signaled the researchers that modeling using different combinations of principal components may yield beneficial results (although these models were not selected as the champion model, they are discussed in further detail in Section III). Thus, the researchers explored feature selection via the method of Functional PCA. Additionally, another method of PCA that would similarly be applicable to this dataset is 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). Although the researchers implemented the classic application of PCA, sparse PCA is a future application that they would like to explore.

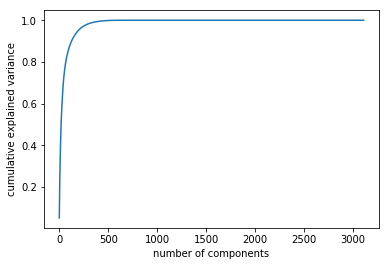


Chart 2: Principal Components and Cumulative Explained Variance

**III. 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, 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 (Tutz & Berger, 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 (Figure A2) 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). As such, the researchers engineered a new indicator variable to encode if the section column contains other ‘402’ sections outside of just 402(c) (see Figure A3). 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.

|  |  |  |
| --- | --- | --- |
|  | **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, after more consideration, the logistic regression model was ruled out because we were unable to find a metric that could be used to compare this model to others. Given that a logistic regression directly estimates probabilities, we couldn’t justify using an accuracy measure that would require grouping the probabilities into correct and incorrect sets. Although it’s possible to adjust the threshold at which the probabilities are divided, we would essentially only be manipulating the AUC by manually choosing a cutoff.

**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 shown in Figure A1 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 0.82 with sensitivity = 0.78 (Table 2). We then decided to tune 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. Ultimately, no combination of PCA components and hyperparameters could beat the model performance without PCA.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **AUC** | **Recall** | **PCA components** | **Tree Parameters** |
| **Highest AUC without PCA** | 0.82 | 0.78 | 0 | max\_depth=None, min\_samples\_leafs = 2, and n\_estimators = 1000 |
| **Highest AUC with PCA** | 0.80 | 0.70 | 3 | max\_depth=10, min\_samples\_leafs=2, n\_estimators= 100 |

Table 2: Random forest tuning and performance on validation set

**Gradient Boosting Model:**

Considering how well our tree ensemble methods performed, we decided to include gradient boosting as one of our exploratory algorithms. We found that 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.

Tuning tree-based and learning-based hyperparameters at the same time proved too costly, so we chose to optimize tree-based parameters first, and then use the selected parameters to tune learning-based parameters. We began by tuning our tree-based hyperparameters by looping through all possible combinations of max\_depth = [None, 3, 10, 30] and min\_samples\_leafs = [1, 2, 10, 500] (Table A3). We optimized AUC under these conditions at max\_depth = 3 and min\_samples\_leafs = 10 with an AUC = 0.82 and a sensitivity = 0.78. We then used these parameters to iterate through n\_estimators = [10, 100, 500, 1000] and learning\_rates = [0.25, 0.1, 0.01, 0.0001]. We found that AUC was optimized here with AUC = 0.82 and sensitivity = 0.79 when learning\_rate = 0.01 and n\_estimators = 1000. This sensitivity was slightly better than the one we’d achieved using a tuned random forest, so we decided to tune the gradient-boosted model further.

Since we found that 94.6% of our variance was explained by one principal component, we decided to explore how various numbers of PCA components affected the performance of our gradient boosting model. We again looped through all hyperparameters also varying number of components between 1, 3 and 100. In evaluating models using fewer components, we were able to create one large loop to optimize the number of principal components, tree-based hyperparameters, and learning-based hyperparameters. All hyperparameters remained the same, except for adding an additional min\_samples\_leafs of 100 to improve granularity of analysis between 10 and 500. Optimal AUC was found at AUC = 0.81 and recall = 0.73 at 100 principal components. Sensitivity was optimized at 0.91 with 1 component, however that model only had a 0.55 AUC.

Since we were unable to beat the AUC we’d already achieved, we decided not to use PCA during final tuning. At this point we selected a gradient-boosting ensemble model with {max\_depth = 3, min\_samples\_leafs = 10, learning\_rate = 0.01, and n\_estimators = 1000} as our final model (see Figure A4 for FPR and TPR).

**Tuning the downsampling of our winning model**

The final step in the tuning of our gradient boosting model was to experiment with various levels of downsampling. Originally, we relied on a downsampling ratio of 50% so that our training set had equal positive and negative label occurrences. We began by testing the gradient boosting model with a downsampling ratio of 20% (meaning 20% of the training set was comprised of positive instances) and found that AUC remained nearly the same but with an extremely low recall of 0.28. When the model was run again with a downsampling ratio of 10% the recall worsened again to 0.10, so we concluded that ratios less than 50% worsened our model performance and that downsampling ratio was optimized at 50%.

At this stage, we chose not to scale any of our features, as scaling should not affect tree models (Li, Ting, et al. 2017). To test this assumption, we reran the final model with a 50% downsample ratio on the unscaled dataset. This produced the same AUC and sensitivity that we’d found for scaled data, confirming that scaling was an unnecessary use of computing power.

**Final model performance**

Using the results of our previous tuning experimentation, we trained the final gradient boosting model without PCA or scaled features, and with a downsampling ratio of 50%. We used the highest performing hyperparameters of max\_depth = 3, min\_samples\_leaf = 10, learning\_rate = 0.01, and n\_estimators = 1000, and evaluated the model on the combined test and validation data. Ultimately, this final model beat our baseline (and all other tested models) with an AUC of 0.84 while still maintaining a high sensitivity = 0.76.

**VI. Bias and Deployment**

**Bias and ethical considerations**

There are a number of bias-related issues that need addressing prior to deployment of this model. First, we are predicting based on cases of mental illnesses that have been identified by the courts, giving the court full control over what’s considered a mental health disability. In this way, MHI is only a loose proxy for mental illness and it is possible that there may exist a selection bias towards certain types or presentations of mental illness within those records. There may also be a disadvantage for demographics that either cannot general afford mental healthcare or are systematically misdiagnosed by healthcare providers, as they will not have documented disabilities. Prior research shows that certain demographics are less likely to be taken seriously by healthcare professionals (Hoffman, et al., 2016), and thus some individuals belonging to a protected group may not be linked to mental health issues that they indeed have. In order to capture these effects of implicit bias, the entries for race were altered as little as possible. We did not combine groups or filter particular entries, as how an individual’s race is perceived may give insight into how that individual is able to navigate the judicial system (Maryfield, 2018), especially in terms of their mental health.

Another shortcoming in our model is that the training dataset did not link participant\_ids across disparate cases (likely due to privacy concerns) and as such this model does not take into account an individual’s history within the Cook County judicial system.

We were surprised to see that features related to domestic violence were not reflected in our models, since research has shown that there is a strong relationship between domestic violence and mental health disorders (Tsirigotis & Luczak, 2017; Behavioral Health Innovations, 2015). It was also unexpected that our understandable decision tree found that having an updated offense category associated with narcotics was an important feature for not having MHI = 1. This seems inconsistent to the researchers, as a large portion of Mental Health court programs involves drug support.

To mitigate societal bias that’s introduced into the model, we could employ discrimination unit tests to identify where the problem lies (d’Alessandro et al., 2019). Verifying which features are correlated directly and indirectly with protected attributes would allow one to identify which features to remove from a model.

**Deployment**

Our model was developed using only publically available data. If Cook County becomes interested in pursuing this project, we could work with them to develop unbiased early identification systems using their more extensive data (including people’s criminal records). If Cook County is not interested, our model still holds promise for non-governmental organizations looking to offer services to individuals within the legal system. The data used here is specific to Cook County, so directly exporting a model to another jurisdiction will not be possible, but this project could easily serve as a blueprint for similar projects elsewhere.

As described above in the Training, Validation, and Test Sets section above, concept drift is likely to be an issue in deployment. As mental health court continues to expand, we can expect an increase in base rate over time, which could eventually degrade model performance. Hence, there ought to be careful monitoring of the MHI base rate and of the legal and policy factors which may influence it. Periodic re-training may be necessary as models become out of date. Model custodians may also decide to exclude older data from training to mitigate concept drift -- this could be tested empirically, and would need to be considered in conjunction with the effects of censoring bias. Type-1 censoring bias is likely to have the opposite effect on our model, and mitigating it would require developing a heuristic cut-off point for the age of cases to be included in training (e.g., only training models on cases which have been in the system at least 6 months). Monitoring model sensitivity in deployment may prove challenging under censoring, but developing the aforementioned heuristic would give custodians a set time at which to evaluate participant’s MHI.

Ultimately, this is not currently an implementable model because of its reliance on protected classes (race, age, gender) for prediction. Additional efforts need to be put into testing model performance while removing these protected classes, however since many of the other features in the dataset could reconstruct the sensitive features, the researchers chose to keep the protected features in for modeling.

# 

# **Bibliography**

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.

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.

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.

Loveland, David, and Michael Boyle. “Intensive Case Management as a Jail Diversion Program for People With a Serious Mental Illness.” *International Journal of Offender Therapy and Comparative Criminology*, vol. 51, no. 2, 2007, pp. 130–150., doi:10.1177/0306624x06287645.

Li, Ting, et al. “Adaptive Scaling .” *ArXiv:1709.00566v1 [Stat.ML] 2 Sep 2017*, 2017.

Hoffman, Kelly M, et al. “Racial Bias in Pain Assessment and Treatment Recommendations, and False Beliefs about Biological Differences between Blacks and Whites.” *Proceedings of the National Academy of Sciences of the United States of America*, National Academy of Sciences, 19 Apr. 2016, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4843483/>.

Levinson, Justin D., et al. “Implicit Racial Bias.” *Implicit Racial Bias Across the Law*, pp. 9–24., doi:10.1017/cbo9780511820595.002.

Maryfield, Bailey. “Implicit Racial Bias.” *Justice Research and Statistics Association*, 2018.

Tsirigotis, Konstantinos, and Joanna Łuczak. “Resilience in Women Who Experience Domestic Violence.” *The Psychiatric Quarterly*, Springer US, Mar. 2018, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5807488/>.

Shah, Tarang. “About Train, Validation and Test Sets in Machine Learning.” *Medium*, Towards Data Science, 10 Dec. 2017, <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>.

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>.

“Center for Health and Justice at TASC: Connecting Policy, Research, and Practice.” *Center for Health and Justice at TASC | Connecting Policy, Research, and Practice.*, <http://www2.centerforhealthandjustice.org/>.

d'Alessandro, Brian, Cathy O'Neil, and Tom LaGatta. "Conscientious classification: A data scientist's guide to discrimination-aware classification." Big data 5.2 (2017): 120-134.

**Appendix A: Team member responsibilities**

All team members worked on data cleaning and model validation. Additional responsibilities are outlined below:

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

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

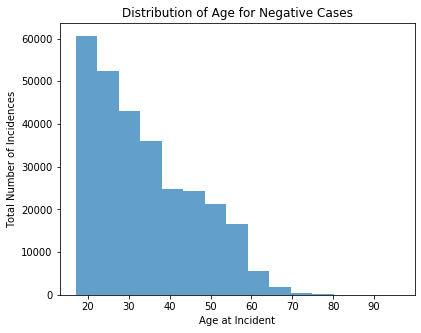
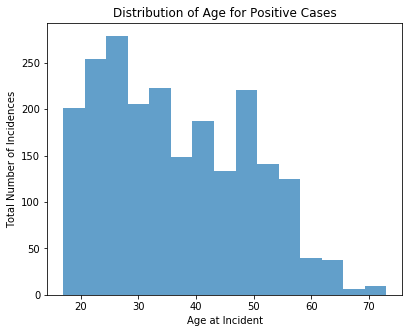
**Alene Rhea**: Project formulation, aggregating data, building time-based split, PCA, understandable decision tree model, tuning gradient boosting model, final model tuning and testing, MHI distribution charts

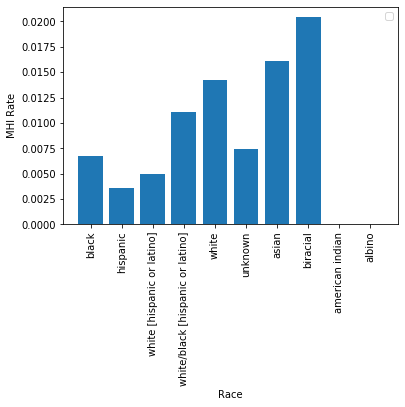
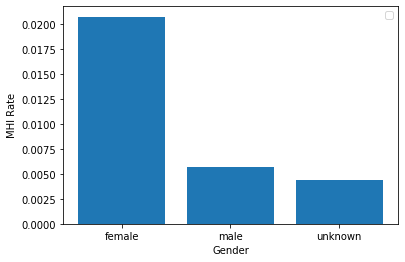
**Amber Teng**: PCA, Baseline models (decision tree, logistic regression, support vector machine, , gradient-boosted model and hyperparameter tuning, downsampling function

**Appendix B: Supplementary Tables and Visualizations**

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| CASE\_ID | Internal unique identifier for each case |
| CASE\_PARTICIPANT\_ID | Internal unique identifier for each person associated with a case |
| OFFENSE\_CATEGORY | Broad offense categories before specific charges are filed on a case |
| PRIMARY\_CHARGE | A flag for the top charge, usually the way the case is referred to |
| CHARGE\_ID | Internal unique identifier for each charge filed |
| CHARGE\_VERSION\_ID | Internal unique identifier for each version of a charge associated with charges filed |
| CHAPTER | The legal chapter for the charge |
| ACT | The legal act for the charge |
| SECTION | The legal section for the charge |
| CLASS | The legal class of the charge |
| AOIC | Administrative Office of the Illinois Courts ID for law of the charge |
| EVENT | The way the charge was brought about |
| EVENT\_DATE | The date the charges were brought about |
| AGE\_AT\_INCIDENT | Recorded age at the time of the incident |
| GENDER | Recorded gender of the defendant |
| RACE | Recorded race of the defendant |
| INCIDENT\_BEGIN\_DATE | Date of when the incident began |
| INCIDENT\_END\_DATE | Date of when the incident ended (this will be blank for incidents that did not go more than one day) |
| ARREST\_DATE | Date and time of arrest |
| LAW\_ENFORCEMENT\_AGENCY | Law enforcement agency associated with the arrest |
| UNIT | The law enforcement unit associated with the arrest |
| INCIDENT\_CITY | The city where the incident took place |
| RECEIVED\_DATE | Date when felony review received the case |
| ARRAIGNMENT\_DATE | Date of the arraignment |
| UPDATED\_OFFENSE\_CATEGORY | ​This field is the offense category for the case updated based upon the top charge for the primary offender. It can differ from the first offense category assigned to the case in part because cases evolve. |
| CHARGE\_COUNT | The charge count of the charged offense. |

**Table A1: Columns in Initiation Dataset**

** **

** **

**Figure A1: Distribution of MHI Across Features**

|  |  |
| --- | --- |
| **Same** | **Different** |
| 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 | primary\_charge, charge\_id, charge\_version\_id, charge\_offense\_title, chapter, act, section, class, aoic, charge\_count, 402 |

**Table A2: Features that remained the same (left) and varied (right) during aggregation by case\_participant\_id**

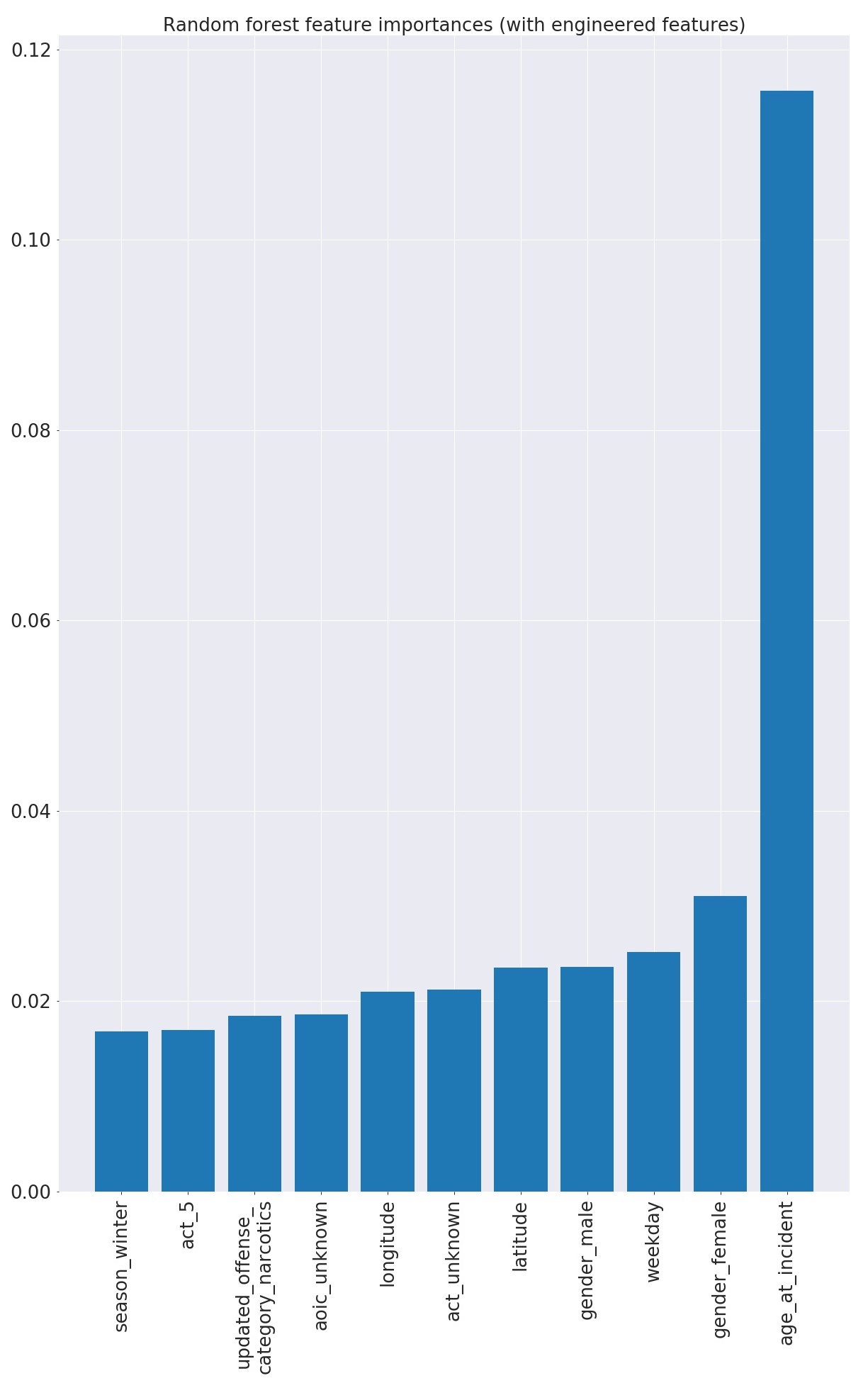
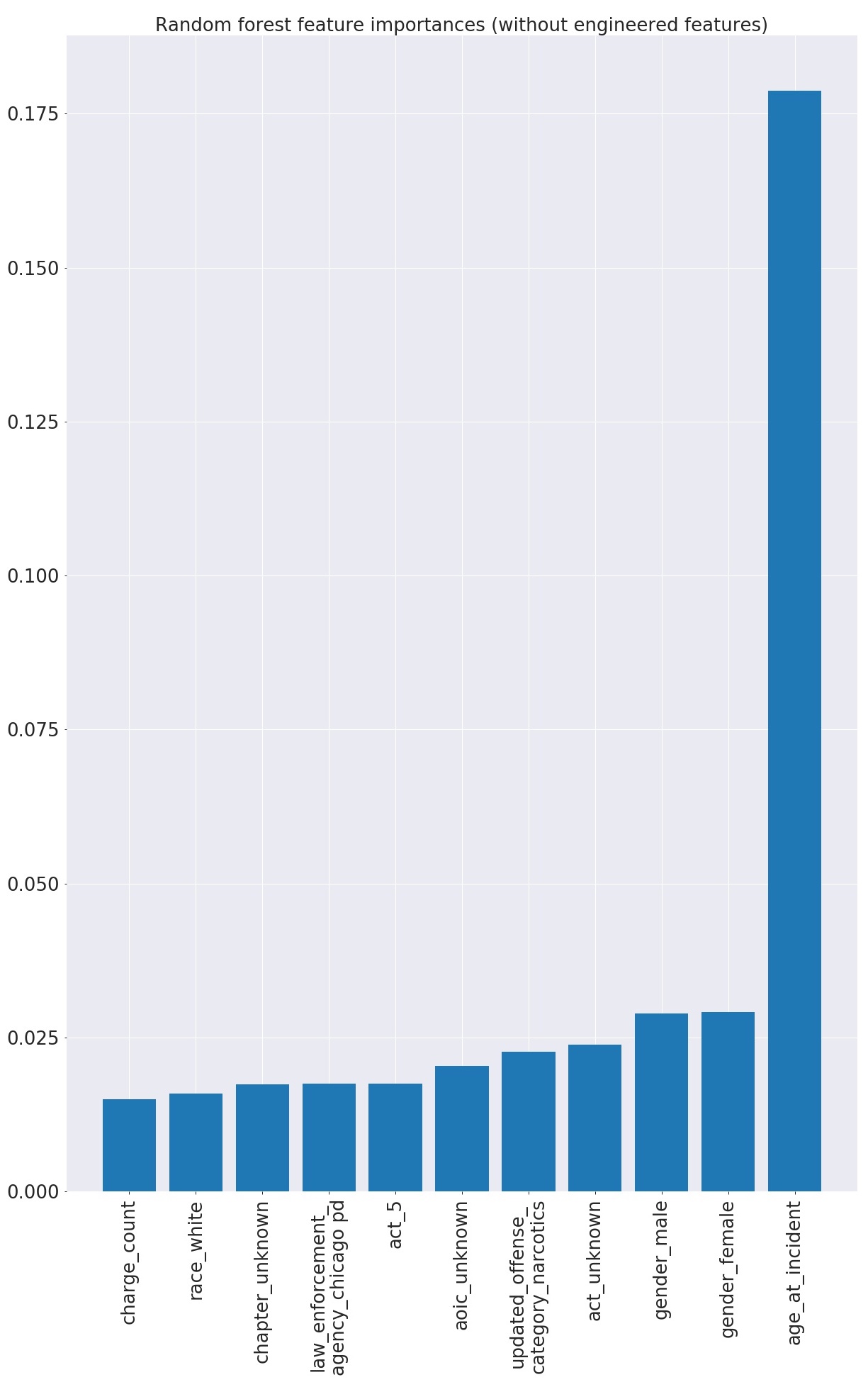
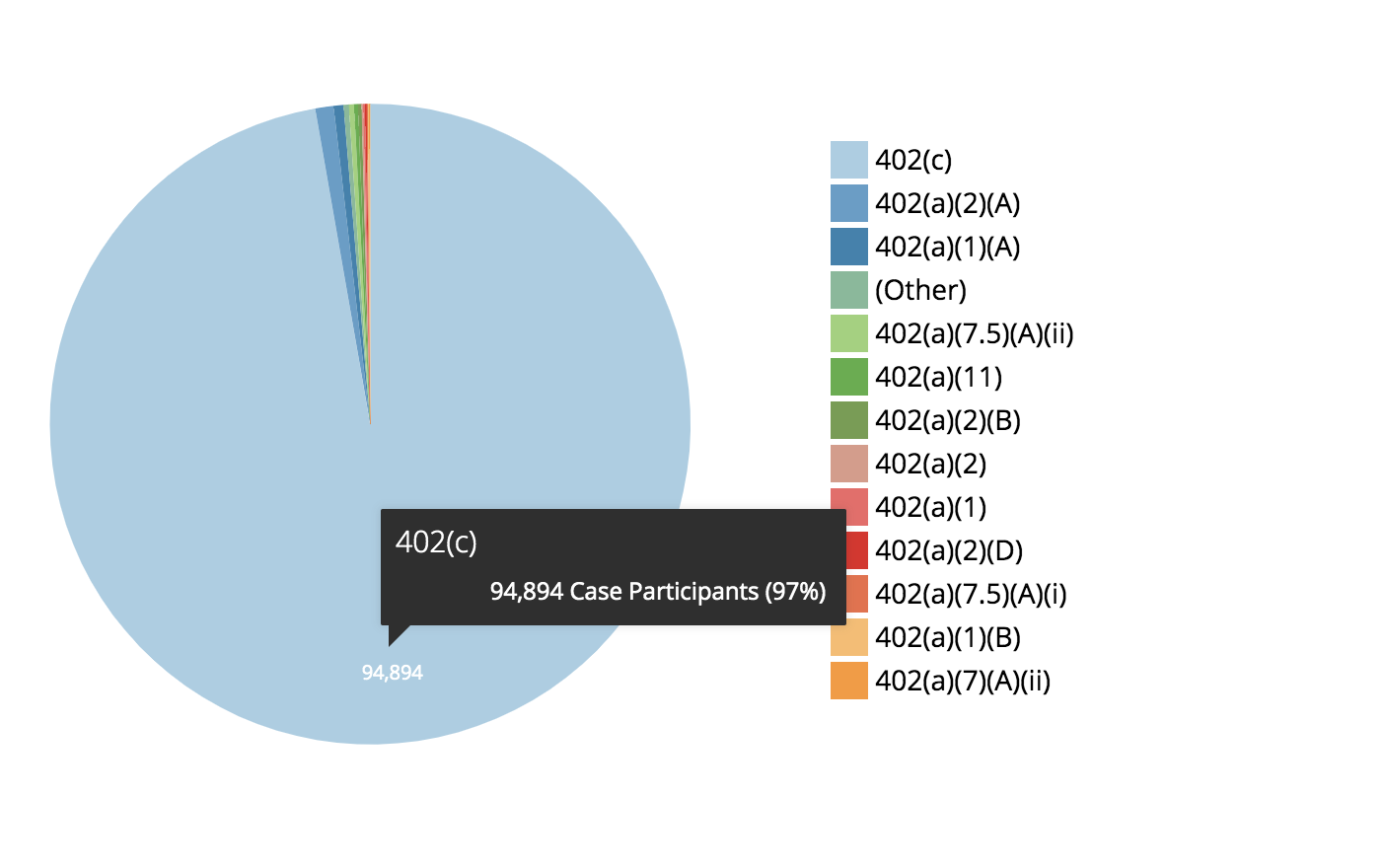


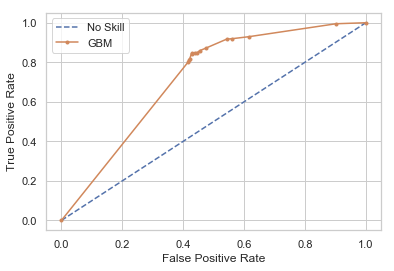
Figure A2: Top 11 feature importances of random forest models (with out of the box parameters) before (left) and after (right) engineered features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |

Table A3: Gradient Boosting Model Hyperparameter Tuning on Validation Se**t**



**Figure A3: Frequency of Case Participants with “402” Sections**



**Figure A4: False Positive Rates of Parameter-Tuned Gradient Boosting Model**

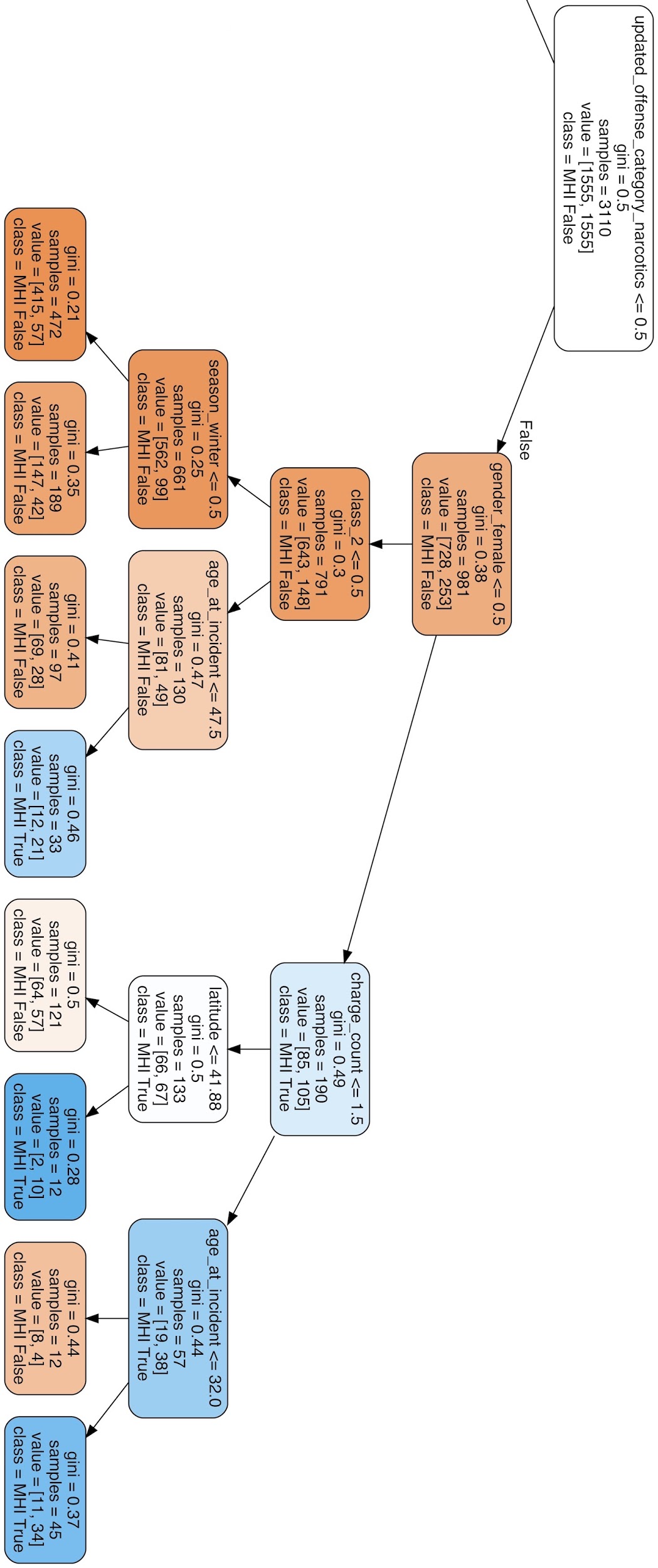
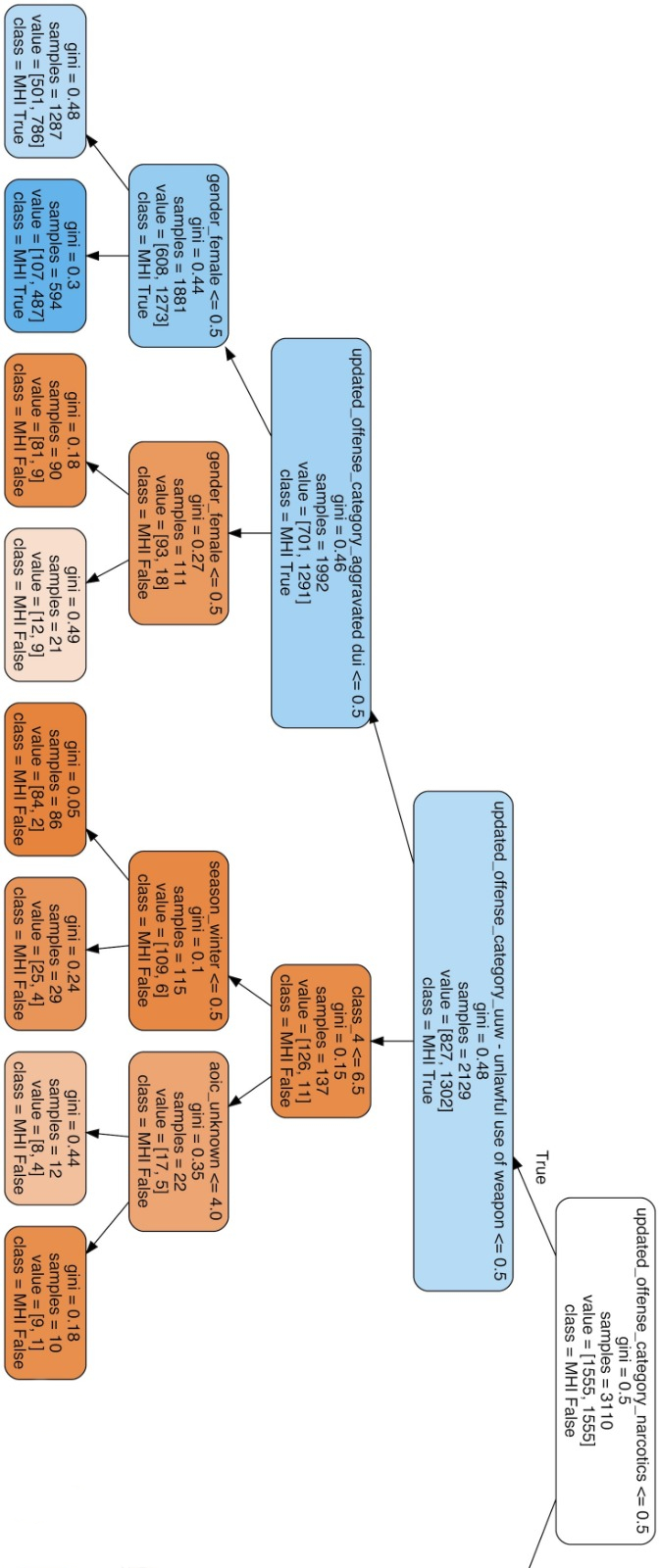


Figure A5: Decision Tree

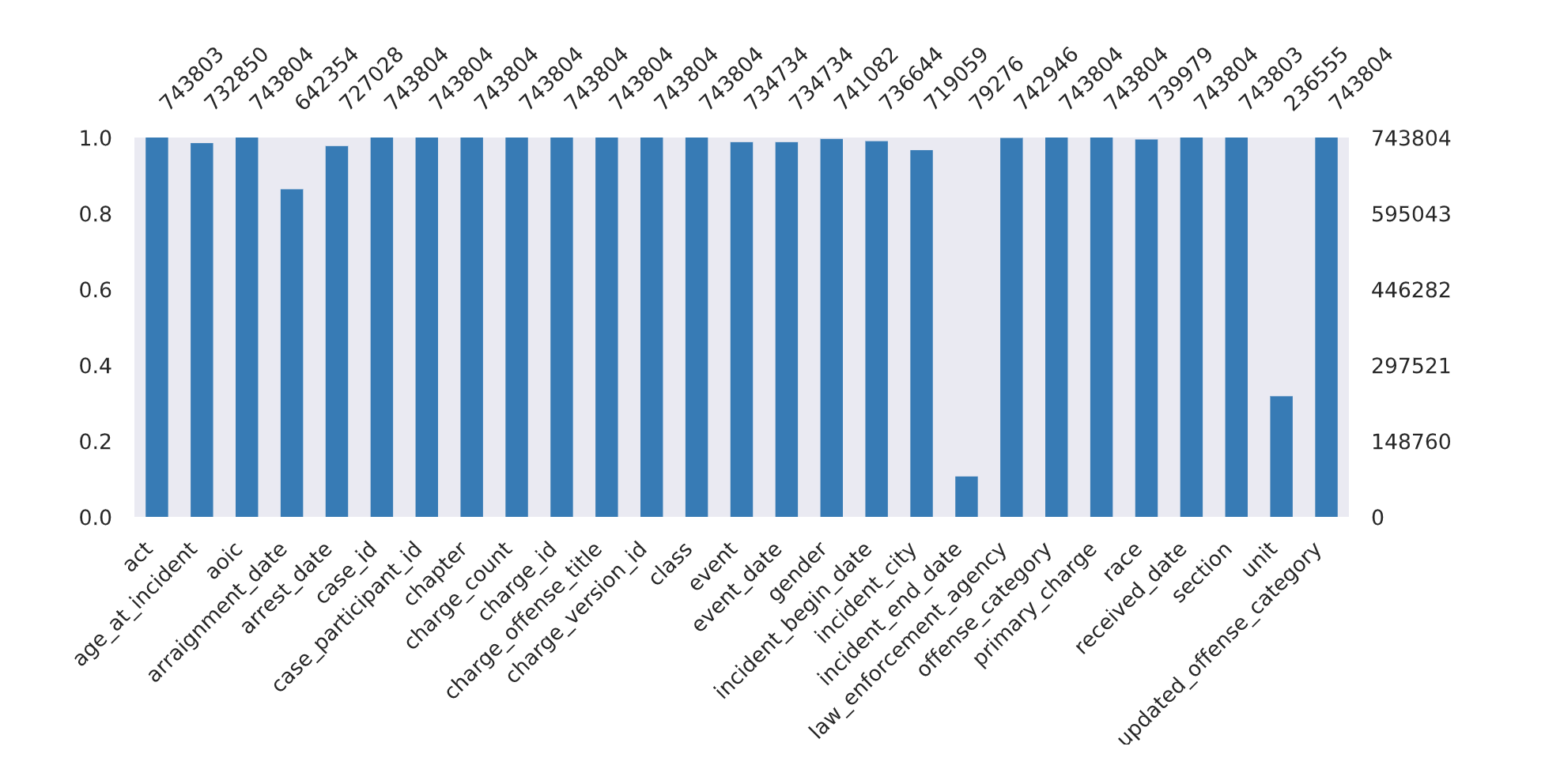


Figure A6: Missing values from features

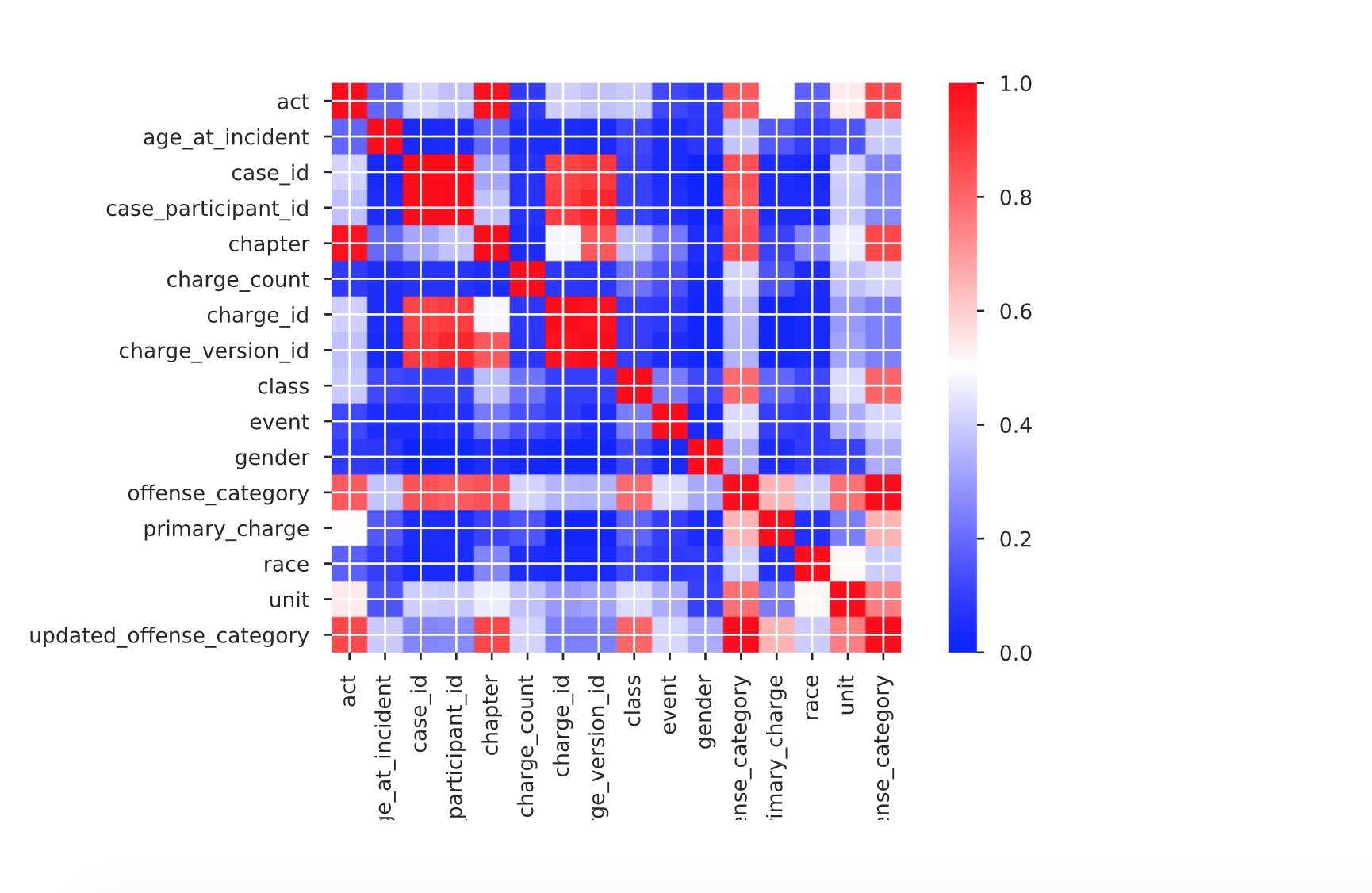


Figure A7: Feature correlations (phi coefficient)