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

Karmen Hutchinson (kah771), Kelsey Markey (kcm312), Alene Rhea (akr435), Angela Teng (at2507)
New York University Center for Data Science, Fall 2019

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 illnesses will enable governments and other institutions to provide appropriate support to these people as early as possible. The project underscores the need for disparate impact tracking in such systems.

I. Introduction

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 (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 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 personnel). 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 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 (Center for Health and Justice at TASC, 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 individuals can be provided additional resources and support from initiation onward.

II. Data Understanding and Preparation

Our data comes from the December 2, 2019 updates of 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 is very populous, and has a well-kept open data portal that contains detailed metadata.

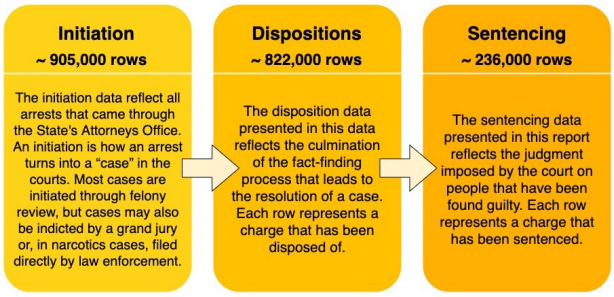


Figure 1: Descriptions of Cook County legal datasets used in this study.

The datasets contain multiple identification numbers that link the records between them. Because the project aims to make predictions at the level of the individual, the researchers used case_participant_id. Case_participant_id is a unique internal identifier assigned by Cook County to each person associated with a case. Each case_participant_id can be linked to multiple charges, with each charge appearing as a separate row in the datasets. A single charge can appear as multiple rows in the sentencing dataset 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. 14 of these columns are categorical, and 6 are time-based. A handful of columns and values were not completely interpretable to the researchers (e.g., "aoic"); 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 created a binary target variable "Mental Health Indicator (MHI)" to indicate whether or not an individual was identified to have a mental health disability. The researchers performed an exhaustive analysis of all the possible values in Sentencing and Disposition which might indicate a mental health-related outcome, and identified 15 relevant values in 6 columns (Table A2). After merging the sentencing and disposition 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 had a sentence_type = "Inpatient Mental Health Services", the row

was assigned an MHI of 1, otherwise it was assigned a 0. Since there may be multiple rows pertaining to a case_participant_id, we created a separate dataset that contained 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. Ultimately we assigned an MHI of 1 to 2212 unique case_participant_id's.

MHI Distribution

Base-rate analysis was performed on sensitive features to better understand their distribution with respect to our target variable (Figure A1). Since protected classes such as age, race, and gender could potentially introduce bias into the model, we examined their distribution in both positive and negative label instances. We found that age skewed higher in positive cases, women were almost four times as likely to have an MHI =1 than were men, people whose race was labeled 'biracial' or 'asian' had the highest rates of positive MHI across races, and people whose race was labeled 'hispanic' had the lowest rates of positive MHI across races. The implications of these findings will be discussed further in the Bias and Ethical Considerations section.

Filtering

The researchers removed all rows from the Initiation dataset with case_participant_id's that did not appear later in Disposition or Sentencing. This ensured that the team was working with only those case_participant_id's for which we would be able to assign an MHI.

Cleaning and Conversion

Cleaning the filtered dataset consisted of changing data types, filling in missing information, and verifying that inputs within a column are uniform. Specifically, we did the following:

- All string variables were converted to lowercase.
- Numeric columns (such as charge_count) were converted to integers.
- All missing or outlying numeric values were replaced with medians.
- All missing non-numeric inputs were replaced with "unknown" (because data exploration revealed this to be a common filler already being implemented by Cook County.)
- All gender values except "Male" and "Female" (i.e. null or "Male name, no gender given") were converted to "unknown"
- Ages were converted to integers and null and outlying values (greater than 100) were replaced with median age. 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.

- Dates converted to datetime and missing or unknown dates were assigned a filler value corresponding to midnight on January 1, 1900
- Since a correlation heatmap showed that ID numbers correlate with important features (see figure A2) all ID numbers were stripped from the data to prevent potential data leakage. Case_participant_id was set to the index, while case_id, charge_id, and charge_version_id were deleted entirely from the dataset.
- The researchers converted all categorical variables to binary dummy variables to allow for the use of parametric models and to prepare for the aggregation of rows (see Aggregation section.) This resulted in a sparse, high-dimensional dataset.

Aggregation

In order to turn the multiple rows per case_participant_id from the initiation dataset into the single row per case_participant_id required by our problem design, we needed to group our data by case_participant_id and then aggregate those rows. Since aggregation can only be performed on numeric variables, 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 A3). During aggregation, we took the median of the consistent categorical columns (which should be equal to the value of every row in the group), and we took the sum of the inconsistent categorical columns. We made a special case for charge_count, where we took the maximum value in order to express the total number of charges associated with the case_participant_id.

Scaling

Since standardization is a common requirement for many models (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. The researchers scaled all numerical features and all categorical dummy features which were summed during aggregation. 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).

Downsampling

The dataset is extremely class imbalanced. In order to help our models learn to identify positive classes in 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 always 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, hypothesizing that this would be the ideal ratio for our data and use case. In tuning our final model, we tested this hypothesis and confirmed that 50/50 was indeed ideal (see Figure A3). 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 70% of the data, a static validation set with 15% of the data, and a test set with 15% of the data, a ratio endorsed by many data scientists (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. Although we can not prevent one individual from appearing in both the training and test sets, we can at least ensure that we will not 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

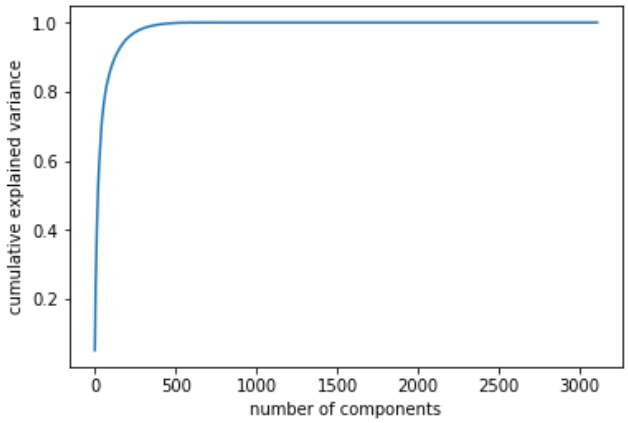
The distribution of MHI across time deciles bears out this hypothesis in that we see a notable dropoff in the top two deciles (see Table 1 & Figure A4). There also appears to be some concept drift in the opposite direction, which may be explained by expansion of the Cook County Mental Health Court Program (Isaacs, 2016). The implications of censoring bias and concept drift are discussed in the deployment section.

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

Table 1: MHI base rate across the datasets

Principal Component Analysis

To reduce the dimensionality of the data, which grew to almost 5000 features after implementation of dummy variables, the team opted to use PCA because of its simplicity, efficiency and non-parametric applications for extracting relevant information from datasets (Shlens, 2014). We used the classic application of PCA on our scaled training set (although sparse PCA should be investigated in future efforts), which yielded explained variance ratios shown in Figure 2. Considering these results, the researchers hypothesized that transforming the data with the most important latent features could improve our models. This hypothesis did not hold up when it was tested (see Section III). Given greater computational resources, the team would have liked to perform a test using 500 principal components, as Figure 2 indicates a sharp elbow at that point.



Principal Component	Explained_variance_ratio
1	0.030
2	0.010
3	0.009
4	0.008
5	0.007

Figure 2: Explained Variance Ratios from PCA

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 for

them is likely to incur only a marginal operational cost to the county. 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 poor 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 A5) 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 A6). In order to associate nearby regions, we geocoded incident city to latitude and longitude. 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. Since more than 80% of values were missing for incident_end_date (Figure A7), we also engineered a feature for incident length by calculating the distance between the incident begin and end dates (and setting this to 0 where incident_end_date was missing). Finally, since age at incident had nearly 4% missing values and approximately 40 (unrealistic) outlying 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. 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 (Table 2).

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

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 is likely to 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.

Support Vector Machine

Often viewed as the general-purpose algorithm for machine learning, we chose 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). Moreover, 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 feature 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. The team ultimately decided not to pursue logistic regression due to the difficulty in fitting a logistic regression to our dummy variables and the impressive results we were already getting using tree methods.

Understandable 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 governments 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 through 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. A visualization of the resultant tree can be found in Figure A8 of Appendix B. The tree is redundant and needs pruning; it could be reduced down to a total of only 10 leaf nodes.

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 A5 in Appendix B;

several engineered features appear in the former. 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 A4). 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.

Gradient Boosting

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 an 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 A5). 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 felt comfortable employing such a large number of estimators because this is known not to cause gradient boosting ensemble methods to overfit. 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 had achieved using a tuned random forest, so we decided to tune the gradient-boosted model further.

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 sensitivity = 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 the team was unable to beat the AUC we had already achieved, we decided not to use PCA during final tuning. At this point we selected as our final model a gradient-boosting ensemble algorithm with {max_depth = 3, min_samples_leafs = 10, learning_rate = 0.01, and n_estimators = 1000} (see Figure A9 for FPR and TPR).

Tuning Downsampling Ratio of Final Model

The final step in the tuning of our gradient boosting model was to experiment with various levels of downsampling. All previous tests relied on a downsampling ratio of 50% so that our training set had equal positive and negative label occurrences. First we tested the optimized 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 sensitivity of 0.28. When the model was run again with a downsampling ratio of 10% the sensitivity 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%. These results were consistent with our hypothesis about downsampling's effect on sensitivity.

During this round of evaluation, 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 had 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 scaling, 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_estimator = 1000, and evaluated the model on the combined test and validation data. Our final model achieved an AUC of 0.84 while still maintaining a high sensitivity of 0.76 (Figure 3). The Receiver Operating Characteristic Curve demonstrates that the model achieves a recall of over 90% fairly quickly, and then flattens out. The false positive rate at that point is just over 40%. The researchers posit that this point represents the classification threshold best suited to the business case.

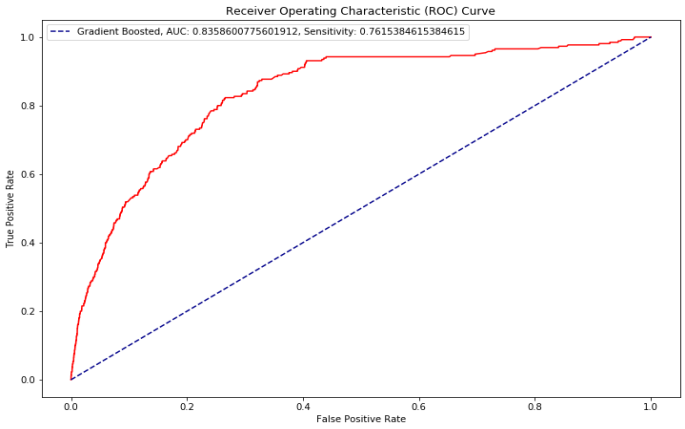


Figure 3: ROC curve for final model on test set

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 and foremost, we must address the societal bias that our model has learned from the data. Our feature importances are brimming with discriminatory features, which is unsurprising given the distribution of MHI across these classes. Discrimination unit tests could be employed to identify where the problem lies (d'Alessandro et al., 2019). Verifying which features are correlated directly and indirectly with protected attributes would allow researchers to identify which features to remove from the model.

Because our model can only learn from cases of mental illnesses that have been identified by the courts, it effectively gives the court full control over what is considered a mental health disability. In this way, MHI is only a loose proxy for mental illness and it is possible that there exists bias towards certain types or presentations of mental illness within those records. There may also be a disadvantage for demographics that either cannot generally 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.

Researchers were surprised to see that features related to domestic violence were not reflected in our models, since research has shown a strong relationship between domestic violence and mental health disorders in Cook County (Tsirigotis & Luczak, 2017; Behavioral Health Innovations, 2015). It is possible that domain-informed feature engineering could capture this relationship; it is also possible that these cases of mental illness commonly go without identification by the courts.

Deployment

Our model was developed using only publically available data. If Cook County becomes interested in pursuing this project, the researchers could work with them to develop a fair and unbiased early identification system for mental illness. Such a system could make use of information which is not publically available, such as an individual's history within the Cook County judicial system and their Health Department 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; however, this project could easily serve as a blueprint for similar projects elsewhere.

As described above in the Training, Validation, and Test Sets section, concept drift is likely to be an issue in deployment. As the Cook County mental health court program 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 for 6 months of longer). 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 an individual's MHI.

Our model is not currently implementable because of its reliance on protected classes (race, age, gender) for prediction. Further efforts need to be put into testing model performance while removing these protected classes and the features able to predict them. The researchers decided to keep the sensitive features in our final model in order that the model not be misconstrued as fair, and to underscore the need for disparate impact tracking.

Works Cited

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

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.

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

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

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/.

“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, et al. “Conscientious Classification: A Data Scientist's Guide to Discrimination-Aware Classification.” *ArXiv.org*, 21 July 2019, https://arxiv.org/abs/1907.09013.

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

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/.

Isaacs, Mike. “County’s Mental Health Court Opts for Offender Treatment over Jail.” chicagotribune.com. Accessed December 9, 2019. https://www.chicagotribune.com/suburbs/skokie/ct-skr-mental-health-court-tl-0526-20160523-story.html.

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.

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

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, 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.

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

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

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

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

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

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/.

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

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.

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

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.

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

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

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/.

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.

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

Appendix A: Team Member Responsibilities

Karmen Hutchinson: Geocoding incident city feature, logistic regression, integrating 12/2/19 data update

Kelsey Markey: Setting up APIs to read in data, engineering MHI target variable, feature engineering age and datetime features, scaling, PCA, random forest pseudo-baseline, tuning random forest hyperparameters

Alene Rhea: Project formulation, filtering, cleaning and aggregating data, building time-based split, PCA, understandable decision tree model, tuning gradient boosting model with PCA, final model tuning with downsampling, training and testing final model on holdout set, MHI distributions

Angela Teng: downsampling function, one hot encoding, PCA, 402 feature engineering, out-of-the-box baseline models, gradient-boosting hyperparameter tuning

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

Dataset	Column	Possible Entries
Sentencing	charge_disposition	FNG Reason Insanity, Finding Guilty But Mentally Ill, Plea of Guilty But Mentally Ill, Verdict Guilty But Mentally Ill, Sexually Dangerous Person
	commitment_type	Mental Health Probation, Inpatient Mental Health Services
	charge_disposition_reason	Mental Health Graduate
	sentence_type	Inpatient Mental Health Services
Disposition	charge_disposition_reason	Mental Health Graduate
	charge_disposition	FNG Reason Insanity, Finding Guilty But Mentally Ill, Plea of Guilty But Mentally Ill, Verdict Guilty But Mentally Ill, Sexually Dangerous Person

Table A2: Columns used for the assignment of MHI

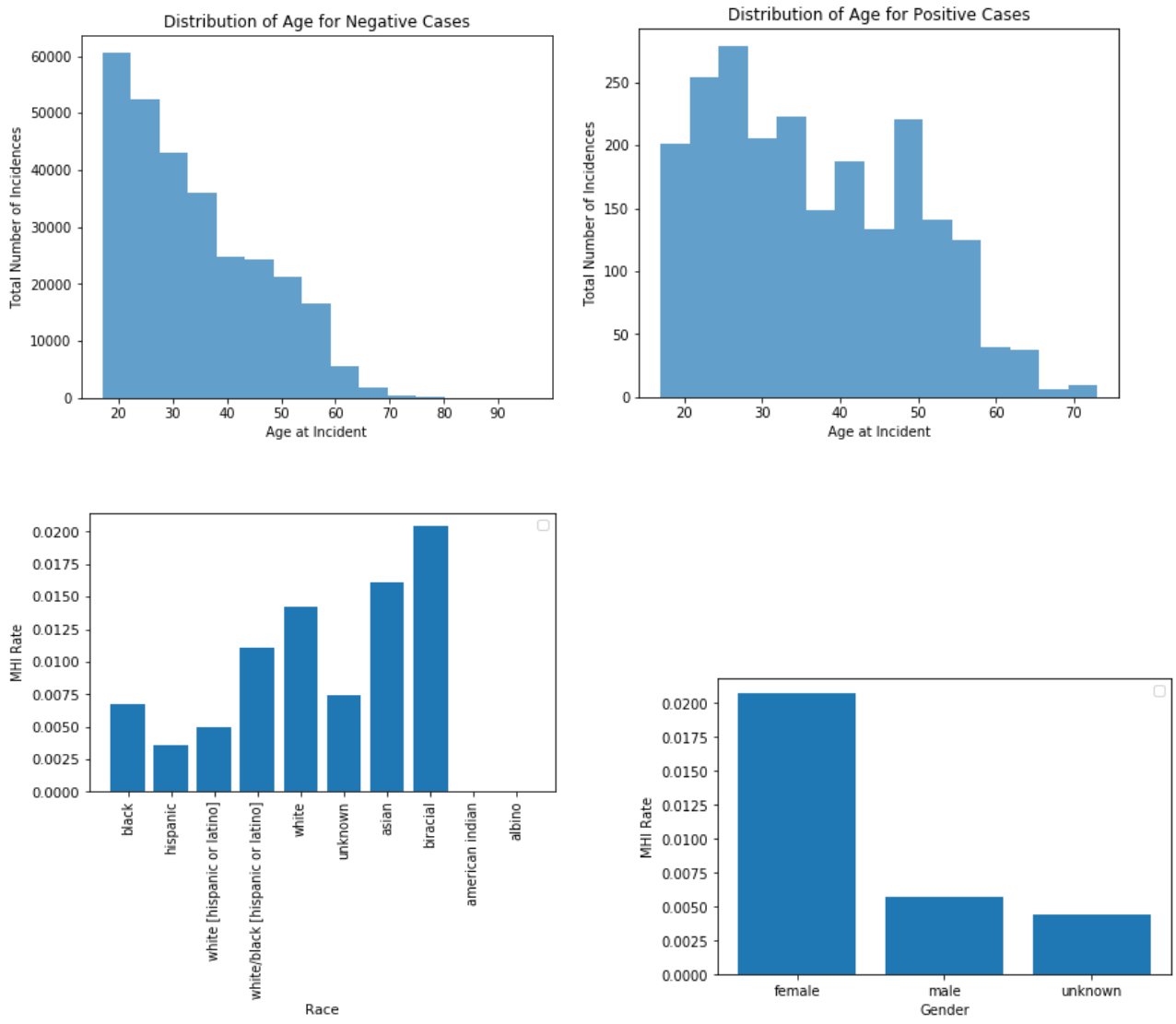


Figure A1: Distribution of MHI Across Features

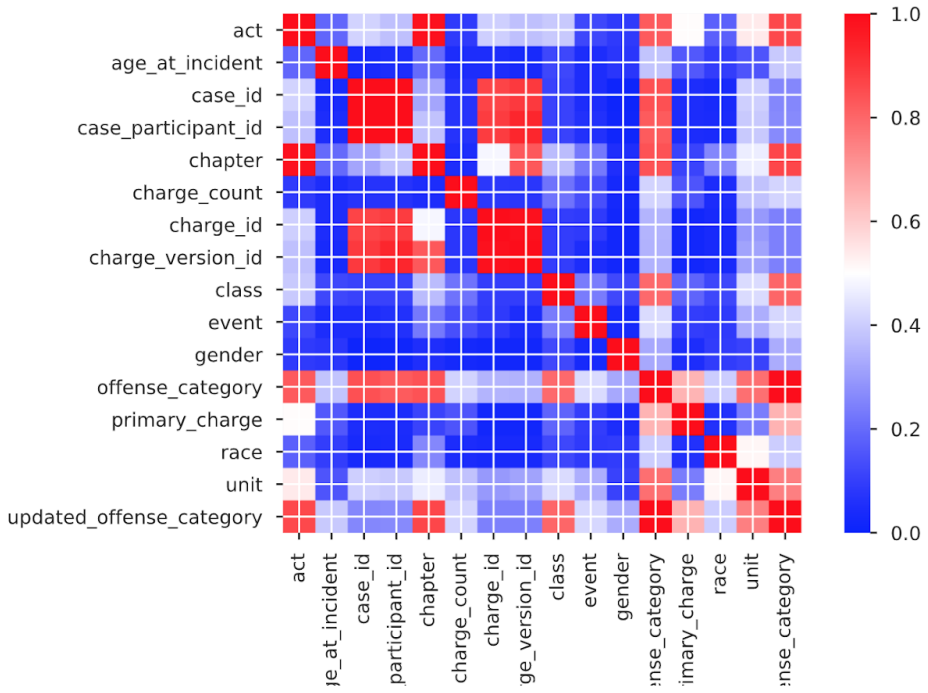


Figure A2: Feature correlations (phi coefficient)

Same columns	Different columns
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 A3: Features that remained the same (left) and varied (right) during aggregation by case_participant_id

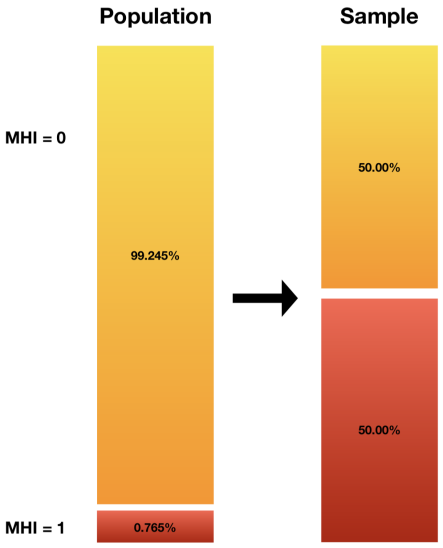


Figure 3: Class probabilities in the dataset population and after downsampling

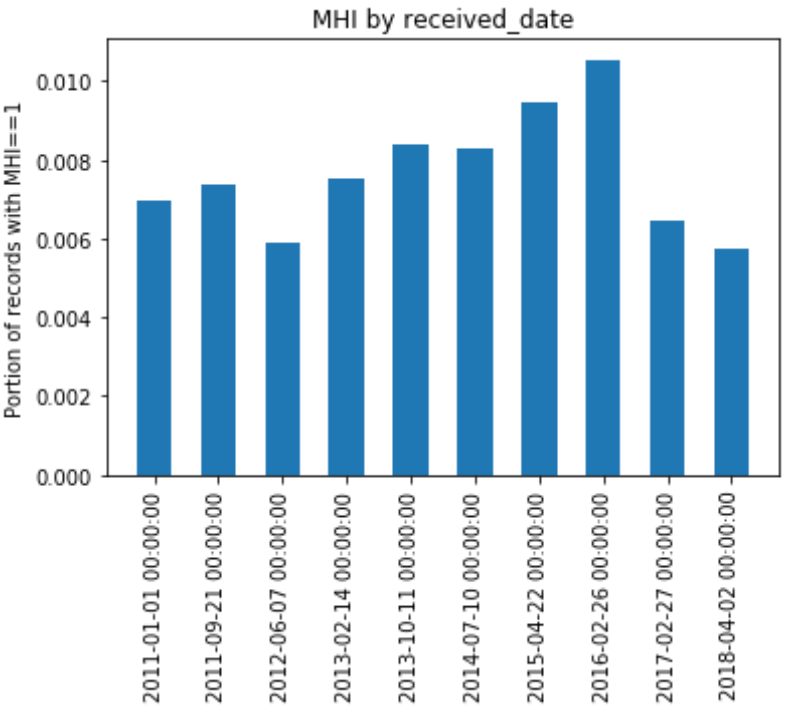


Figure A4: Positive instances of MHI by received_date

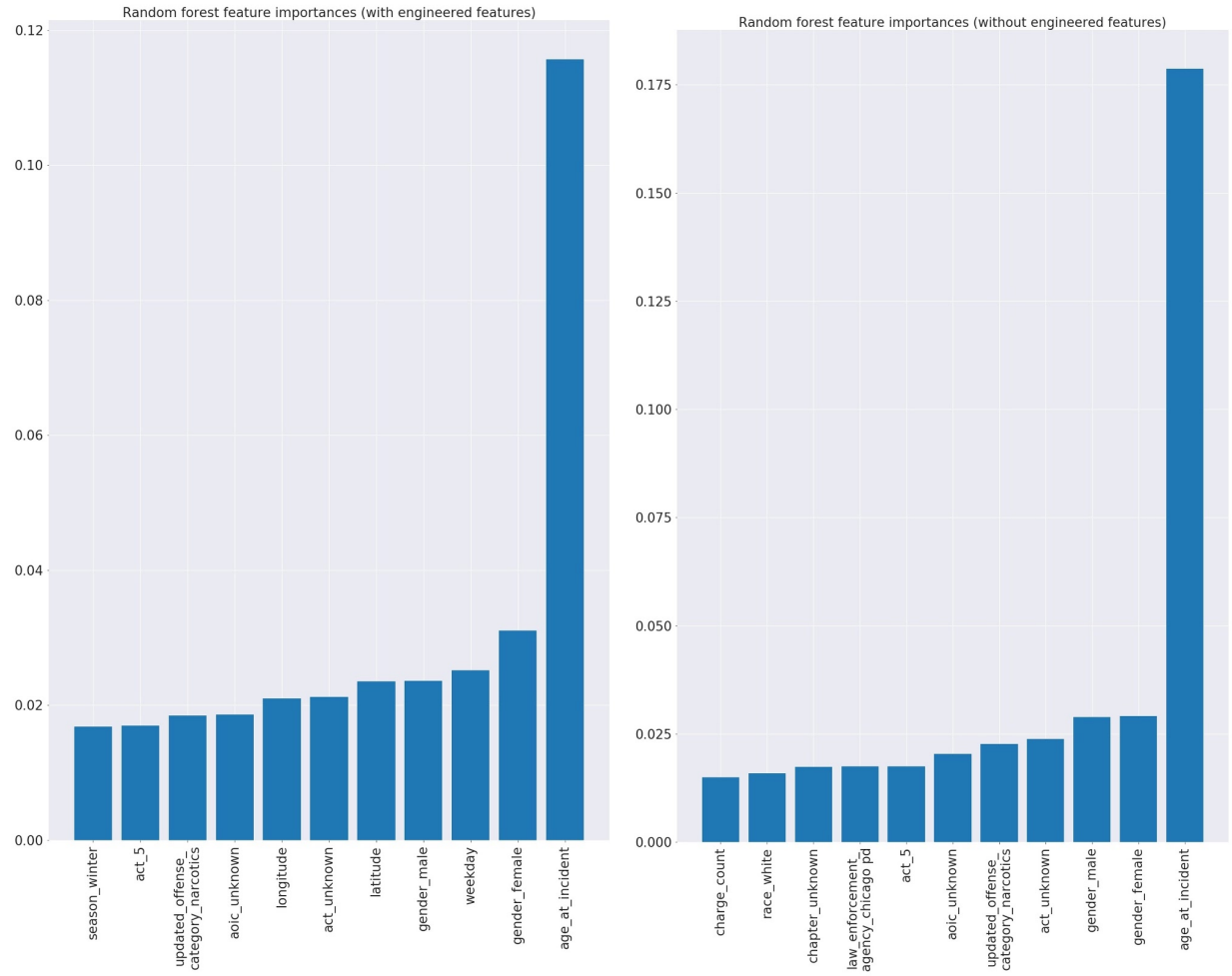


Figure A5: Top 11 feature importances of random forest models (with out of the box parameters) before (left) and after (right) engineered features

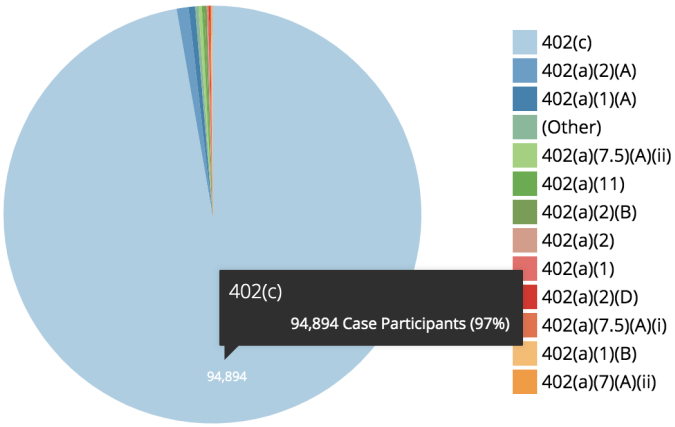


Figure A6: Frequency of Case Participants with "402" Sections

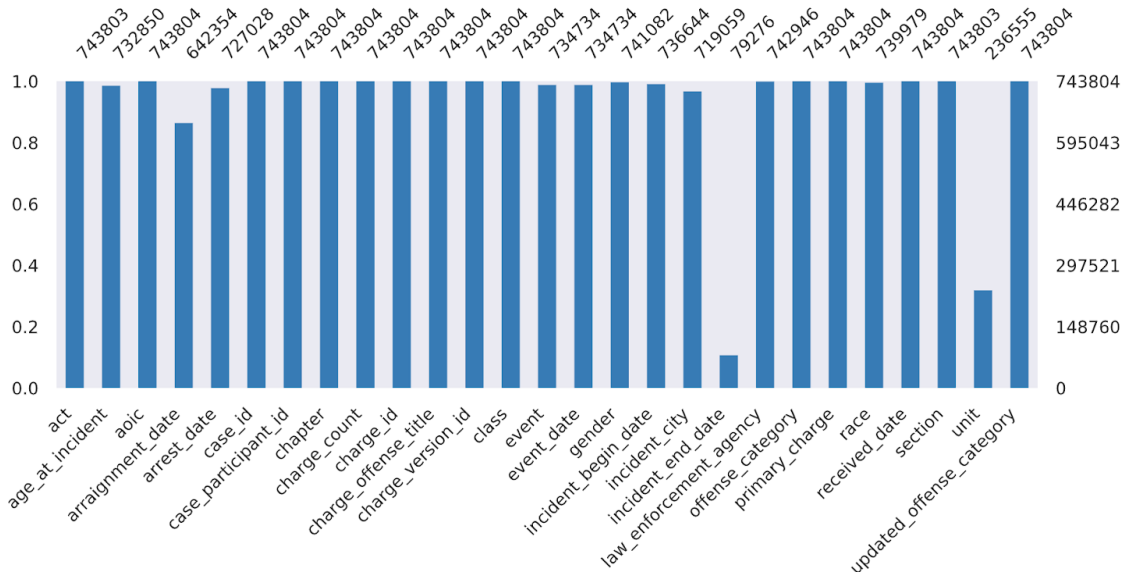


Figure A7: Missing values of dataset features

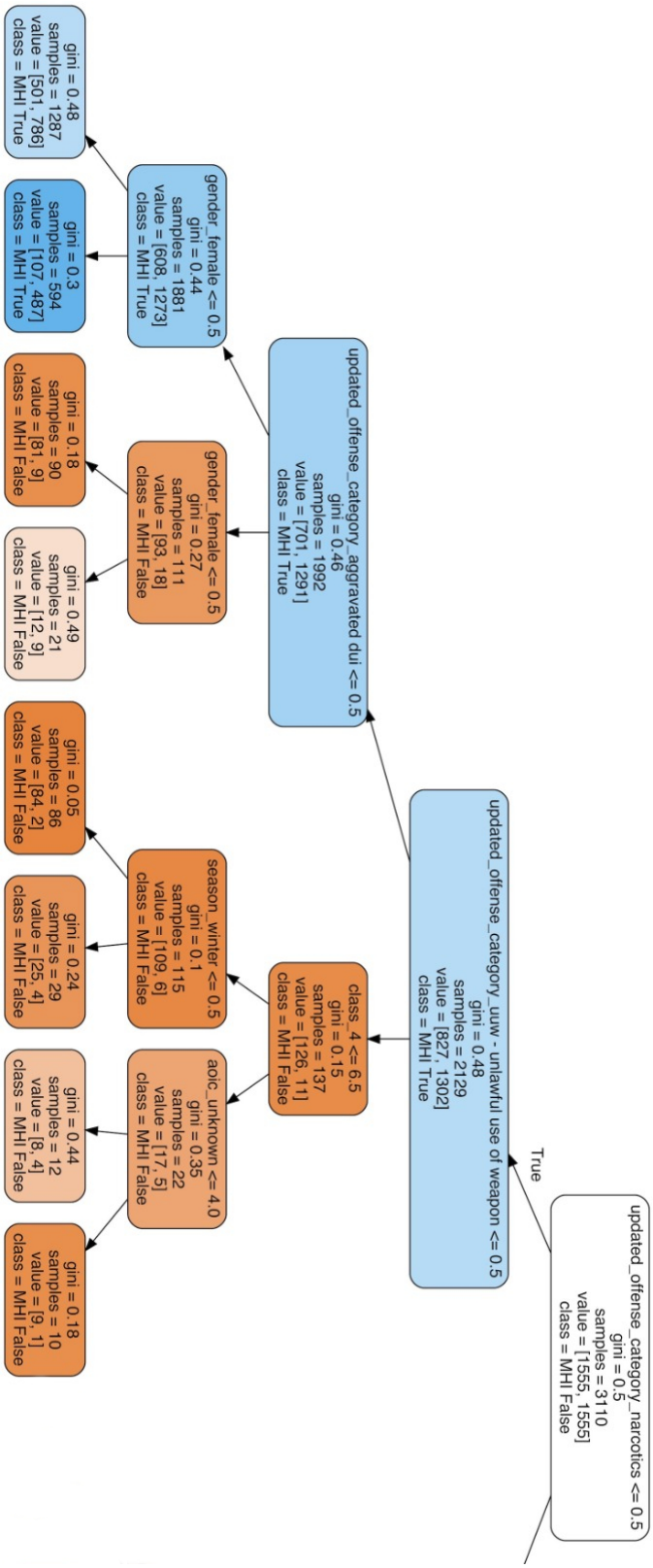


Figure A8: Decision Tree (part 1)

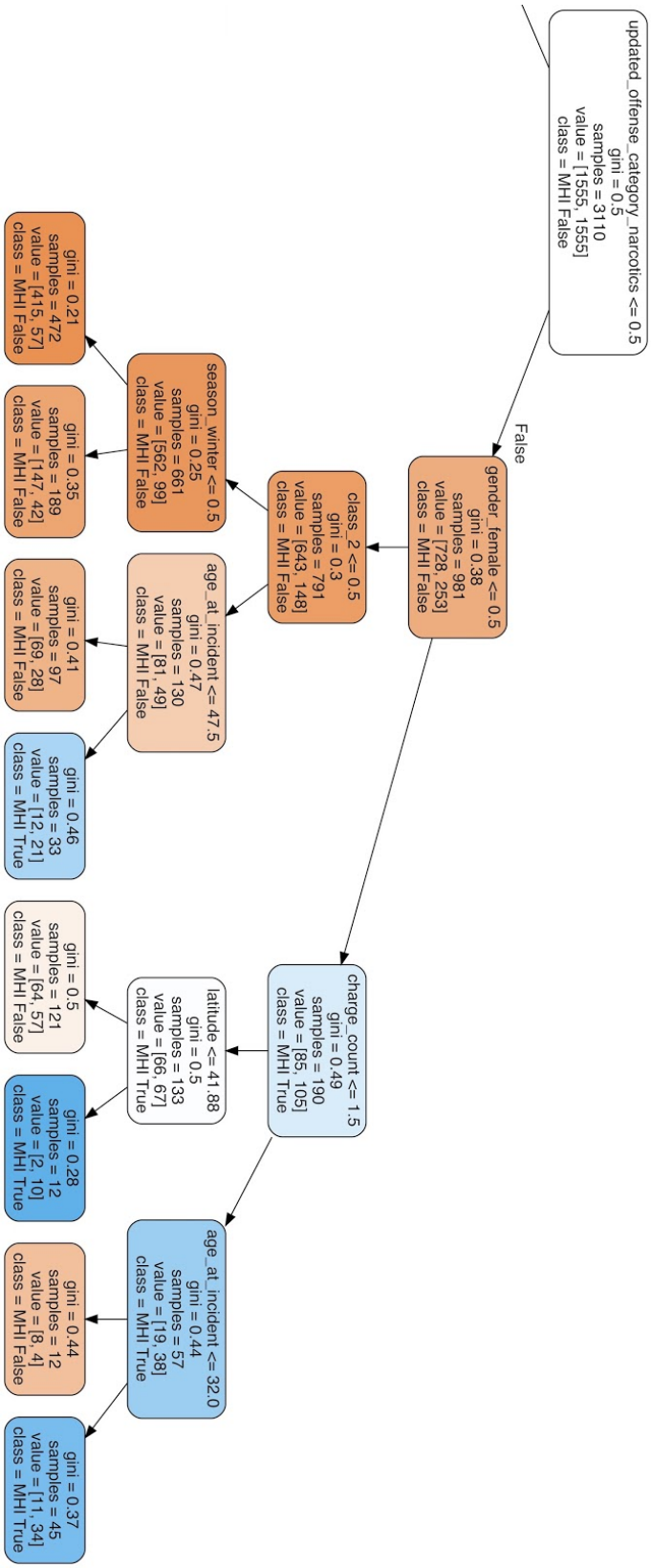


Figure A8: Decision Tree (part 2)

	AUC	Sensitivity	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 A4: Random forest tuning and performance 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

Table A5: Gradient Boosting Model Hyperparameter Tuning on Validation Set

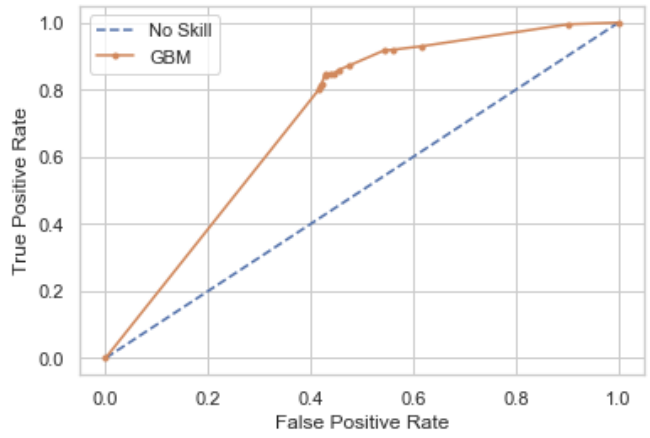


Figure A9: ROC of Parameter-Tuned Gradient Boosting Model