Recidivism Forecasting Using XGBoost

Timothy Han

1. Introduction

The Recidivism Forecasting Challenge hosted by the National Institute of Justice (NIJ) asked participants to develop data-driven algorithms to accurately predict the probability that recently released prison parolees would recidivate at three different points in time: one year, two years, and three years after release from prison. To support such algorithms, NIJ provided a training dataset containing the recidivism outcomes of past prison parolees along with data fields or features that describe the behavior and history of each parolee.

I participated as an individual competitor under the small team category and made submissions for the two and three year forecasts. My modeling approach relied solely on the training dataset provided by NIJ and did not leverage any external datasets. I applied several feature engineering techniques to pre-process the data and then trained an XGBoost model in Python to make the final forecasts. This document will focus on the results and development of the model used to make predictions at the three year mark.

2. Data Pre-Processing

The training data provided by NIJ contains 18,028 records with recidivism outcomes as well as 48 numeric and categorical features which describe each parolee. The recidivism outcomes are stored in four target variable columns. Each target variable indicates the recidivism status of each individual at the one, two, and three year mark. The fourth target variable indicates any recidivism within three years of release.

2.1. Training Set Filters

At each stage of the challenge, the training dataset was filtered to only include individuals who had not already recidivated. For example, when training a model to make predictions for year three, the training data was filtered to remove all individuals who had recidivated in years one and two. The motivation behind this was to only train the model with data containing clear signals for what we wanted to predict. Including individuals who recidivated in year one might dilute the signals for year three recidivism predictions. Some experimentation was done by training models on data that included individuals who had already recidivated in previous years and down-weighting these samples. However, no performance gain was observed using this approach. When building the year three forecast model, the training set was filtered down to 9,398 records where 1,791 parolees recidivated in year three and 7,607 did not.

2.2. Target Variable Definition

To train supervised learning techniques, we must first identify a target variable which represents the event that we are trying to predict. For each stage of the challenge, the target variable was

defined by using the data fields which indicate recidivism arrest (*Recidivism_Arrest_Year1, Recidivism_Arrest_Year2, Recidivism_Arrest_Year3*). For example, when building a model to make recidivism predictions at year three, the binary data field *Recidivism_Arrest_Year3* was used as the target variable.

3. Variables

The following section describes the feature transformations applied to the training data. The choice of using these feature transformations was driven by the type of model used which was XGBoost. XGBoost can only handle numeric features, so any natively categorical features in the data had to be converted to numeric format. No additional features were brought in using external data.

3.1. Binary Features

Features like *Prior_Arrest_Episodes_DVCharges, Gender, Race*, or *Gang_Affiliated* take binary values. These types of features were simply transformed to contain binary integer values of 0 or 1. The exact mapping was not important, just so long as the original binary values were distinctly mapped to these new integers.

3.2. Ordinal Categorical Features

Several features contain character string values, yet they have a strong numeric meaning. For example, the feature *Residence_Changes* takes string values "1", "2", and "3 or more". Since these features have a clear numeric interpretation, they were simply converted from character representations of integers to actual integers. For example, "1" was converted to 1, "2" to 2, and "3 or more" to 3. Another feature that falls into this category is *Age_at_Release*. This feature takes string values that represent age ranges such as "18-22" and "23-27". These string values were converted to numeric form by taking the lower bound of the age range and converting it to an integer.

3.3. Nominal Categorical Features

Other features like *Supervision_Level_First* which do not have a clear numeric interpretation were converted to integer values using one-hot encoding. This method of feature encoding expands a single feature into multiple features where each new feature is a binary indicator of the occurrence of one of the categorical values the original feature takes. One-hot encoding was applied to the following features: *Supervision_Level_First, Education_Level, Prison_Offense,* and *Residence_PUMA*. The output columns of the one-hot encoding take the name of the input column concatenated with the value that the one-hot encoded feature is indicating. For example, the one-hot encoded feature that indicates the presence of *Residence_Puma* = 20 is called *Residence_Puma_20*.

3.4. Derived Features

New features were also generated based on the existing set of features. A set of missing value indicator features was derived from features which have missing values. These features with missing values include *Avg_Days_per_DrugTest, DrugTests_THC_Positive, DrugTests_Cocaine_Positive, DrugTests_Meth_Positive, DrugTests_Other_Positive, Percent_Days_Employed, Jobs_Per_Year, Gang_Affiliated, Supervision_Level_First, Prison_Offense,* and *Supervision_Risk_Score_First*. Each missing value indicator feature has the same name as the base feature with the addition of "_miss" at the end of the name. For example, the missing value indicator feature for *Jobs_Per_Year* is called *Jobs_Per_Year_miss*. When missing values occur, the mode value of the given feature is used to impute the missing values. Although we are able to impute these missing values, it is still useful to retain information that the feature had a missing value because the omission of data could also have predictive power.

Features that capture interactions between existing features were also generated and are previewed in Table 1. Table 3 in the appendix shows the full list of derived features along with their definitions. The features with names ending in "_Total" are simply summations of sets of related features. For example, the feature *Prior_Arrests_Episodes_Total* is the sum of the values of the prior arrest episode types. These types of features provide an aggregate view of how many past crimes the parolee committed.

The features with names ending in "_DISCREP" indicate contradictions in the data. For example, the feature *Residence_Changes_DISCREP* checks to see if an individual has had zero recorded residence changes (*Residence_Changes* = 0), yet somehow has a non-zero amount of violations for moving without permission (*Violations_MoveWithoutPermission* > 0). This type of contradiction in the data could indicate poor monitoring and reporting on the parolee which may be linked to low quality rehabilitative care from community corrections officers. Some of these features exhibit univariate lift. For example, parolees that have a residence change discrepancy are 1.5 times more likely to recidivate at the three year mark than the overall population.

Table 1. Definitions for a subset of the derived features used in the XGBoost model.

Derived Feature Name	Feature Description	Feature Logic
Prison_Years_Pct_Age	The percentage of the individual's lifetime spent in prison.	Prison_Years / Age_at Release
Age_at_Sentence	The age of the individual when they are released from prison.	Age_at_Release - Prison_Years
Prior_Arrest_Episodes_Total	The sum of the various prior arrest episode types	Prior_Arrest_Episodes_Felony + Prior_Arrest_Episodes_Misd + Prior_Arrest_Episodes_Violent + Prior_Arrest_Episodes_Property + Prior_Arrest_Episodes_Drug + Prior_Arrest_Episodes_PPViolationCharges + Prior_Arrest_Episodes_DVCharges + Prior_Arrest_Episodes_GunCharges
Residence_Changes_DISCREP	Parolee has 0 residence changes, yet they have more than 0 violations for moving without permission.	if Residence_Changes = 0 AND Violations_MoveWithoutPermission > 0 then 1; Otherwise 0;

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

3.5. Additional Feature Transformations

The numeric features *Avg_Days_per_DrugTest* and *Jobs_Per_Year* have distributions which are highly skewed, so the log transform was applied to compensate and make it easier for the XGBoost decision trees to make cut points. The features *DrugTests_THC_Positive, DrugTests_Cocaine_Positive, DrugTests_Meth_Positive, DrugTests_Other_Positive, Percent_Days_Employed,* and *Prison_Years_Pct_Age* contain high precision decimal values. These features were rounded to the nearest tenth because a model that takes into account the difference in drug test results at the hundredth or thousandth decimal place may be likely to overfit the training data.

4. Models

XGBoost was chosen as the modeling approach for this competition because it typically offers good performance out-of-the-box for tabular datasets. In order to save time, I chose not to explore alternative approaches and devoted more time to feature engineering and hyper parameter tuning.

4.1. Model Type

XGBoost or Extreme Gradient Boosting was used to build the models for all stages of prediction. XGBoost is a tree-based algorithm that uses gradient-boosting to build an ensemble of decision trees. It has become a popular algorithm to use on structured tabular data and is often successful in data science competitions and has become an industry standard in many fields. One of the reasons XGBoost has become so popular on tabular data is that it can be trained very fast using GPUs while retaining high predictive accuracy similar to or better than most other algorithms. Since it is a tree-based learning algorithm, it has reasonable interpretability qualities and can output a feature importance ranking. The feature selection process is also built into the XGBoost algorithm, which means time intensive manual feature selection is not critical.

4.2. Performance Metric

Forecast submissions were evaluated using the Brier Score to compare the predicted probabilities against the true recidivism outcomes. The Brier Score is the mean squared error between the target variable and the predicted probability:

$$Brier\ Score = \frac{1}{n}\sum_{t=1}^{n}(f_t - A_t)^2$$

To reduce the potential for racial bias in models, the competition judging criteria imposes a fairness penalty which is defined as one minus the absolute value of the difference in false positive rate between white and black parolees. To determine false positives, predicted probabilities greater than or equal to a threshold of .5 are converted to “Yes” predictions of the parolee having recidivated. The fairness penalty is combined with the Brier Score to produce a final “Fair and Accurate” score defined as:

$$FP = 1 - |FP_{Black} - FP_{White}|$$

$$Fair\ and\ Accurate = (1 - BS)(FP)$$

When training and tuning the models, this fair and accurate metric was used to evaluate each candidate model. During the hyperparameter tuning phase described in the next section, I found that when tuning against this metric, the best models favored predictions that very rarely exceeded the .5 probability threshold which led to very low false positive rates in both white and black parolees. This meant that the fairness penalty was not very meaningful in preventing racial bias, as there were almost never any false positives for either black or white parolees.

4.3. Parameter Tuning

To help guard against overfitting, the training set was first randomly split into a new training and hold-out set where 33% of the data was allocated to the hold-out set. For convenience, these new datasets will be referred to as the dev training set and the hold-out set respectively. XGBoost requires the tuning of several hyperparameters which control how robust the model is against overfitting. The dev training set was used to tune these hyperparameters, while the hold-out set was only used later for performance evaluation of candidate models.

I applied a grid search of the following XGBoost hyperparameters on the dev training set: *N_estimators, Depth, Min Child Weight, Learning Rate, Gamma,* and *Colsample by Tree*. To evaluate each set of hyperparameters, I randomly divided the dev training set into a sub-training and sub-test set with 33% of the data used for the sub-test set. The model was trained on this sub-training set and then evaluated against the sub-test set where the "fair and accurate" metric defined in the previous section was used to measure the model's performance. When evaluating each set of hyperparameters, I randomized the sub-training and sub-test set ten separate times and averaged the model's "fair and accurate" performance value on each randomized sub-test set. This repeated randomization approach was done to ensure that the best hyperparameters on average were chosen.

After completing a grid search, the hyperparameters with the best average performance were chosen as the final model hyperparameters. I then trained a model using these hyperparameter values on the dev training set and evaluated it on the hold-out set. This method provided an unbiased estimate of the model performance, since the hold-out set was never used in the tuning phase. These are the final hyperparameter values chosen for the XGBoost model:

- *N_estimators* = 800
- *Depth* = 4
- *Min Child Weight* = 20
- *Learning Rate* = .005
- *Gamma* = .001
- *Colsample by Tree* = .9
- *Booster* = 'gbtree'

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

4.4. Model Performance

After tuning the XGBoost hyperparameters, a model was trained on the dev training set and then evaluated on the hold-out set. Figure 1 displays the ROC curve for the model performance on the hold-out set where an AUC of .689 was achieved.

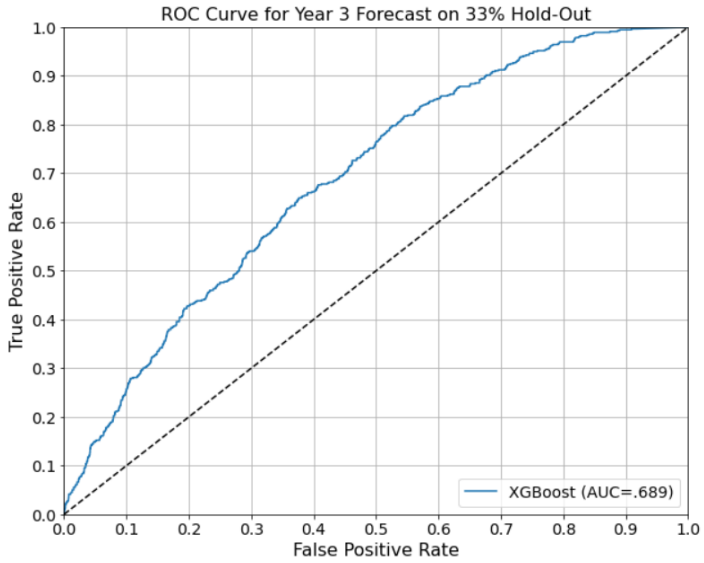

Figure 1. ROC curve for an XGBoost model trained on the dev training set and evaluated on hold-out set. Model achieves AUC=.689.

NIJ released the final competition results which include each winning participant's Brier scores on the official test sets. Table 2 shows the model's Brier scores on the official test set for the year three forecast. These scores are compared against the model's Brier scores on the hold-out set used during model development. The table shows the comparison for the male, female, and combined segments. I did not place in the top five ranking for the female segment, so I do not have the Brier score on the official test set for female parolees.

Table 2. Brier scores on hold-out set and year three forecast official test set.

Segment	Brier Score on Hold-Out Set for Year 3 Forecast	Brier Score on Year 3 Forecast Official Test Set
Male	0.1450	0.1526
Female	0.1019	-
Combined	0.1384	0.1359

The combined Brier scores on the hold-out and official test sets are fairly close to one another, which suggests that the model did not overfit the training set. The difference in Brier score on the

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

Male segment is more pronounced, however this could simply be a result of the Male hold-out set segment not being large enough to produce a reliable Brier score estimate.

4.5. Feature Importance

The importance or impact of each feature in an XGBoost model can be evaluated using several different metrics. One of these metrics is the "gain" of the feature which measures the average reduction in loss when trees make splits on the feature. This provides a good measurement of the overall impact of a feature on an ensemble of trees. Features with high gain values generally have high predictive power and are important to a model's performance. Figure 2 ranks the features in terms of their gain value in descending order for the XGBoost model used in the year three forecast. Features at the top of the list are considered more important in the model than the features near the bottom of the list.

The top ranked feature, *Jobs_Per_Year_miss*, is a derived feature that indicates that the feature *Jobs_Per_Year* had a missing value. It is difficult to understand why this yields high predictive power. A better understanding of the data intake process may help shed some light on this. From a purely data-drive perspective, this missing indicator feature is ranked highly because it can provide a decision tree with a split that produces a leaf node where all parolees on the leaf do not recidivate. An examination of the year three forecast training data shows that all 534 parolees with a missing value for *Jobs_Per_Year*, do not recidivate in year three. Another highly ranked feature is *Gang_Affiliated* which is a binary indicator of whether or not the parolee is gang affiliated. The training data shows that when a parolee is gang affiliated, they are 1.69 times more likely to recidivate than the general population.

Scanning further down the list of feature importance, it is worth noting that several other missing indicator features like *Gang_Affiliated_miss* and *Avg_Days_per_DrugTest_miss* are within the top 20 ranked features. These indicator features also exhibit univariate discriminative power, however unlike the *Gang_Affiliated* feature which identifies high risk parolees, these features point to lower risk populations. For example, when the *Gang_Affiliated* feature has a missing value (*Gang_Affiliated_miss* = 1), the parolee is around 30% less likely to recidivate than the overall population.

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

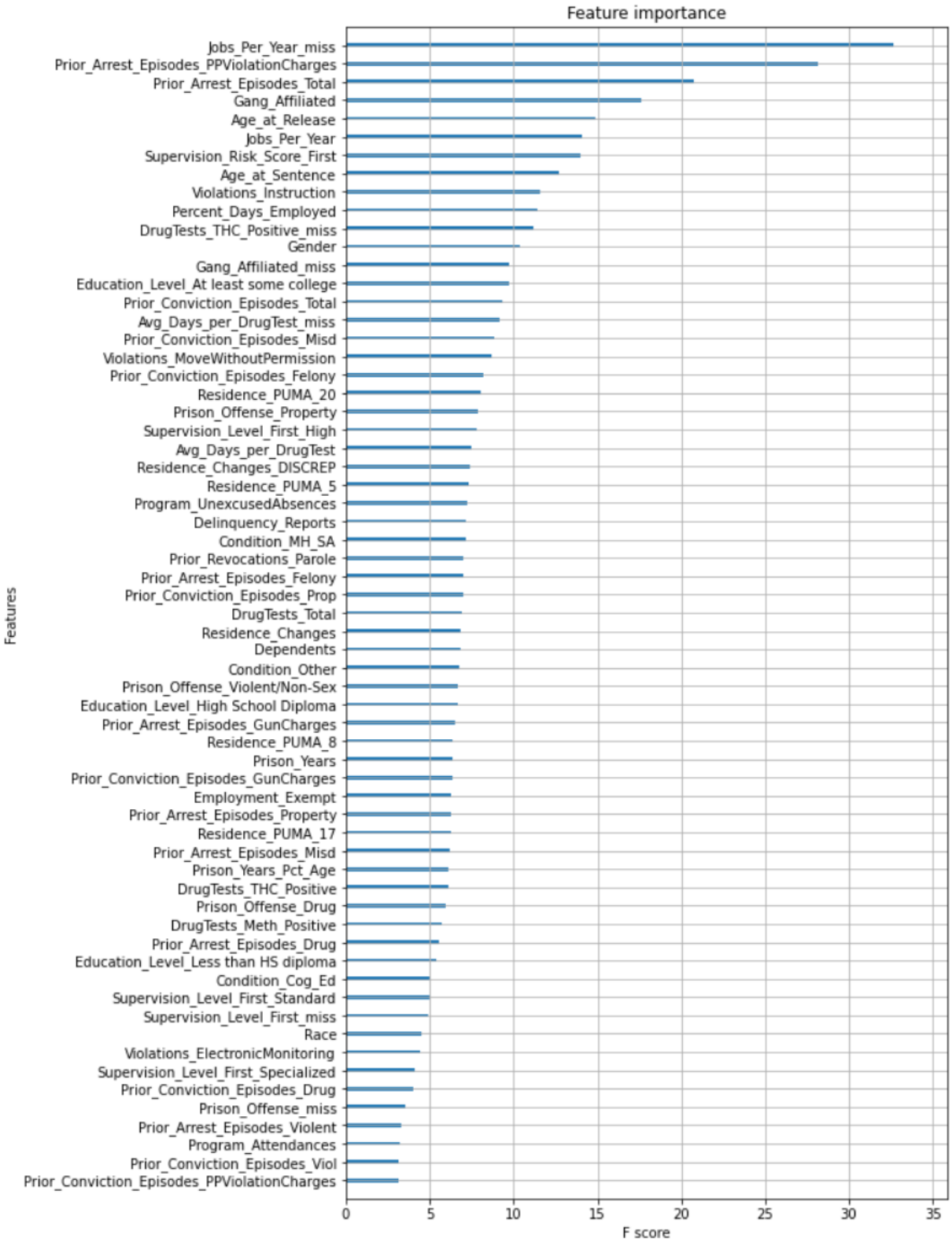

Figure 2. Feature importance ranking of the final XGBoost model used for Year 3 predictions. Features listed with a larger F score (gain), have a higher importance or impact in the model.

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

5. Future Considerations

5.1.Alternative Performance Metrics

The main performance metric chosen for this challenge is the Brier score which is the mean squared error between the target variable and the predicted probability. This metric provides a wholistic view of how well a model performs on a given dataset. However, for tasks like monitoring and rehabilitation of parolees, a different performance metric may be more appropriate.

Community corrections officers responsible for providing support and outreach to parolees have limited time and resources and may carry a large caseload of parolees that they cannot realistically manage. Given these resource constraints, it is important to develop predictive models that help corrections officers prioritize which parolees to focus on in order to maximize their impact of preventing recidivism. A performance metric that captures how large of an impact a corrections officer can have given limited time and resources, may be better at optimizing predictive models that solve the real-world business need. One such metric would be the true positive rate within the top N percentile of probabilities output by a model. To explain why this would be of use, let us imagine that a corrections officer has a caseload of 100 parolees, but they only have enough time and resources to diligently manage 20 parolees. A model that can place the parolees most likely to recidivate at the top 20% of the corrections officer's priority list would maximize the impact that the corrections officer can have at preventing recidivism. In this scenario, we only care about placing as many high risk parolees within the top 20% of model probability scores because these are the individuals who will actually be given attention. A model that is good at minimizing the Brier score for the entire population of parolees may not have as much of a real-world impact given the resource constraints and high workloads that corrections officers face. The threshold of N would need to be calibrated based on the typical caseloads of each community corrections department.

5.2. Practical Findings

One learning from this competition is that given the relatively small training data size, it is important to use cross validation or some other repeated and randomized evaluation of hyperparameters. I found that a given set of hyperparameters could yield significantly different model performance when the training and hold-out set are randomized multiple times.

Engineered features that identify contradictions in the data can have predictive power and high impact in a model. For example, the feature *Residence_Changes_DISCREP* has a univariate lift of 1.5 in the year three forecast training set. This means that parolees that do have this data contradiction are 1.5 times more likely to recidivate than the general population. This feature also ranked in the top half of the overall feature importance list, meaning that it has significant impact to the model.

5.3. Suggestions for Future Competitions

One suggestion for future competitions is to give participants access to more upstream forms of the data before it has been aggregated and processed. This will give data scientists the chance to engineer more complex features which may yield higher predictive power. Feature engineering can often times be more impactful than the actual modeling technique used so allowing as much exploration on the underlying feature engineering may lead to better model performance.

6. Appendix

Table 3. Definitions for all derived features used in the XGBoost model.

Derived Feature Name	Feature Description	Feature Logic
Prison_Years_Pct_Age	The percentage of the Parolee's lifetime spent in prison.	Prison_Years / Age_at Release
Age_at_Sentence	The age of the Parolee when they are released from prison.	Age_at_Release - Prison_Years
Prior_Arrest_Episodes_Total	The sum of the various prior arrest episode types	Prior_Arrest_Episodes_Felony + Prior_Arrest_Episodes_Misd + Prior_Arrest_Episodes_Violent + Prior_Arrest_Episodes_Property + Prior_Arrest_Episodes_Drug + Prior_Arrest_Episodes_PPViolationCharges + Prior_Arrest_Episodes_DVCharges + Prior_Arrest_Episodes_GunCharges
Prior_Conviction_Episodes_Total	The sum of the various prior conviction episode types	Prior_Conviction_Episodes_Felony + Prior_Conviction_Episodes_Misd + Prior_Conviction_Episodes_Prop + Prior_Conviction_Episodes_Drug + Prior_Conviction_Episodes_Viol + Prior_Conviction_Episodes_PPViolationCharges + Prior_Conviction_Episodes_DomesticViolenceCharges + Prior_Conviction_Episodes_GunCharges
DrugTests_Total	The sum of the various drug test results	DrugTests_THC_Positive + DrugTests_Cocaine_Positive + DrugTests_Meth_Positive + DrugTests_Other_Positive
Percent_Days_Employed_DISCREP	Parolee has 0% days of employment, yet they have more than 0 jobs per year on average.	if Percent_Days_Employed = 0 AND Jobs_Per_Year > 0 then 1; Otherwise 0;
Residence_Changes_DISCREP	Parolee has 0 residence changes, yet they have more than 0 violations for moving without permission.	if Residence_Changes = 0 AND Violations_MoveWithoutPermission > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_Felony_DISCREP	Parolee has 0 prior felony arrests, yet they have more than 0 felony convictions.	if Prior_Arrest_Episodes_Felony = 0 AND Prior_Conviction_Episodes_Felony > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_Misd_DISCREP	Parolee has 0 prior misdemeanor arrests, yet they have more than 0 misdemeanor convictions.	if Prior_Arrest_Episodes_Misd = 0 AND Prior_Conviction_Episodes_Misd > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_Violent_DISCREP	Parolee has 0 prior violent arrests, yet they have more than 0 violent convictions.	if Prior_Arrest_Episodes_Violent = 0 AND Prior_Conviction_Episodes_Viol > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_Property_DISCREP	Parolee has 0 prior property arrests, yet they have more than 0 property convictions.	if Prior_Arrest_Episodes_Property = 0 AND Prior_Conviction_Episodes_Prop > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_Drug_DISCREP	Parolee has 0 prior drug arrests, yet they have more than 0 drug convictions.	if Prior_Arrest_Episodes_Drug = 0 AND Prior_Conviction_Episodes_Drug > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_PPViolationCharges_DISCREP	Parolee has 0 prior PP violation charge arrests, yet they have more than 0 PP violation charge convictions.	if Prior_Arrest_Episodes_PPViolationCharges = 0 AND Prior_Conviction_Episodes_PPViolationCharges > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_DVCharges_DISCREP	Parolee has 0 prior DV charge arrests, yet they have more than 0 DV charge convictions.	if Prior_Arrest_Episodes_DVCharges = 0 AND Prior_Conviction_Episodes_DomesticViolenceCharges > 0 then 1; Otherwise 0;
Prior_Arrest_Episodes_GunCharges_DISCREP	Parolee has 0 prior gun charge arrests, yet they have more than 0 gun charge convictions.	if Prior_Arrest_Episodes_GunCharges = 0 AND Prior_Conviction_Episodes_GunCharges > 0 then 1; Otherwise 0;

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.