paper

Group 5

2025-12-04

Abstract

have a summary of the paper

Background

here's our data and why we chose it here's where the data is from here are our research questions

Our data comes from kaggle, it contains three csv files; sentencing, person, and marks. Across these datasets we have a unique id for each prisoner assigned by the Illinois Department of Corrections. (Talk about data cleaning and the work put into making one final dataframe)

Research Questions

- What predictors have the strongest effect on sentencing length?
- How accurately can we predict sentencing length?
- What predictors have the strongest effect on offence category?
- How accurately can we predict offence categories?

Methods

KNN

Within the criminal justice system, it is expected that some patterns will exist for sentencing type, prior offenses, counts and severity of a charge (i.e., was it armed or aggravated), among other variables. However, there are some patterns or correlations that would ideally not exist, in a perfect criminal justice system (such as offense category hypothetically being correlated with number of scars). With these seemingly conflicting ideas in mind, this section of analysis focuses on using K-nearest neighbors (KNN) to attempt to predict offense category within the data. While predictive accuracy is one goal, the expectations raised in this section also indicate a desire to examine variable importance within prediction, to see if expected or unexpected variables are having large impacts on predictive accuracy in the model.

As such, in order to answer our second two research questions regarding prediction of offense category, as well as determine which predictors have the strongest effect on offense category, we decided to fit a KNN model. KNN stands for k-nearest neighbor, which is an algorithm that cycles through each point in a dataset, and identifies the k points that are closest to that point. This k value is adjustable and helps determine the classifier itself, so tuning k is critical for optimizing classification performance. I grouped offenses/charges into nineteen categories: controlled subst poss w/out prescription, burglary, murder, armed robbery, theft (identity or property), battery, sexual assault, forgery, kidnapping, illegal firearm/weapon/handgun use or possession, harassment, bribery, drug/meth manufacturing, vehicular hijacking/theft, DUI, child porn,

obstructing justice, home invasion, and other. It is important to note that these categories were created by reading through thousands of different individual charges, all unique and specific to each person. As such, these categories were the result of wide generalizations, and in no way serve as definite classifiers of charge or individual offense. With that said, the eighteen categories (besides other) captured a little more than 90 percent of all observations, meaning the vast majority of charges fell into one of the categories listed above.

Because this dataset is so large (even with an 80-20 training-test split), ten fold cross validation (which is the process of partitioning the entire data set in ten different folds, where each fold is the test set one time and the other nine folds are used for training) was extremely time-expensive (about 2 hours and 45 minutes to complete).

```
set.seed(445)
illinois_doc_data <- read.csv("../CSV Files/merged_data.csv")
cleaned_doc_data <- illinois_doc_data |> select(-c(last_name, name, id)) |>
  mutate(offense_category = as.factor(offense_category),
         adm_appx_age = appx_age,
         weight_lbs = as.numeric(weight),
         sex = as.factor(sex),
         height_in = as.numeric(height),
         race = as.factor(race),
         eye_color = as.factor(eyes),
         hair = as.factor(hair),
         class = as.factor(class),
         parent_institution = as.factor(parent_institution)) |>
  select(-c(eyes, height, weight, appx_age)) |> na.omit()
head(cleaned_doc_data)
```

```
##   num_scars num_tattoos num_other_marks                  hair  sex  race
## 1         2           1               0                 Brown Male White
## 2         1           1               0 Gray or Partially Gray Male White
## 3         1           0               0                 Black Male Black
## 4         3           0               0       Salt and Pepper Male Black
## 5         1           0               0                 Black Male Black
## 6         0           1               0                 Black Male Black
##                     parent_institution year_adm month_adm total_counts class
## 1            DIXON CORRECTIONAL CENTER     1983         2            1     2
## 2    PINCKNEYVILLE CORRECTIONAL CENTER     1988         2            1     4
## 3 WESTERN ILLINOIS CORRECTIONAL CENTER     2017        11            1     2
## 4           MENARD CORRECTIONAL CENTER     1988        12            3     X
## 5          PONTIAC CORRECTIONAL CENTER     1974         2            1     M
## 6    PINCKNEYVILLE CORRECTIONAL CENTER     1983         9            1     M
##      offense_category attempted aggravated armed current_sentence
## 1            BURGLARY     FALSE      FALSE FALSE              100
## 2 OBSTRUCTING JUSTICE     FALSE      FALSE FALSE               61
## 3               THEFT     FALSE      FALSE FALSE                3
## 4      SEXUAL OFFENSE     FALSE       TRUE  TRUE              100
## 5              MURDER     FALSE      FALSE FALSE              100
## 6              MURDER     FALSE      FALSE FALSE                0
##   time_sentenced_prior life_sentence death_sentence adm_appx_age weight_lbs
## 1              0.00000          TRUE          FALSE           34        185
## 2             82.00000         FALSE          FALSE           42        167
## 3             85.33333         FALSE          FALSE           63        245
## 4             15.00000          TRUE          FALSE           34        166
## 5            100.00000          TRUE          FALSE           18        195
```

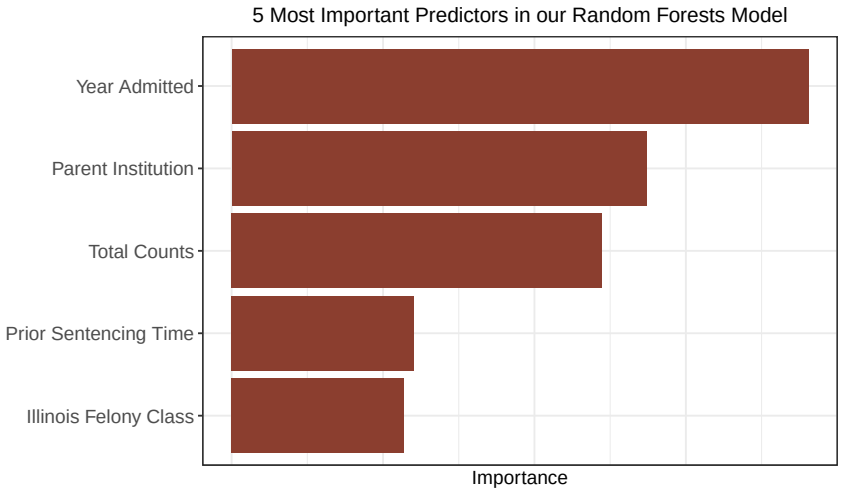
```
## 6             16.50000         FALSE          FALSE           30        180
##   height_in eye_color
## 1        67      Blue
## 2        69     Green
## 3        72     Brown
## 4        67     Brown
## 5        69     Brown
## 6        68     Brown
```

```
df_split <- initial_split(cleaned_doc_data, prop = 0.8)
df_train <- training(df_split)
df_test <- testing(df_split)

k_vals <- expand.grid(neighbors = c(10, 50, 100, 200, 500, 1000,
                                    2000, 5000, 10000))
knn_10foldcv <- vfold_cv(df_train, v = 10)
knn_spec <- nearest_neighbor(mode = "classification", neighbors = tune()) |>
  set_engine("kknn")
knn_rec <- recipe(offense_category ~ ., data = df_train) |>
  step_dummy(all_nominal_predictors()) |>
  step_normalize(all_numeric_predictors())
knn_wf <- workflow() |> add_recipe(knn_rec) |> add_model(knn_spec)
#tune_results <- tune_grid(knn_wf,
#                          resamples = knn_10foldcv,
#                          grid = k_vals)
#collect_metrics(tune_results)
#best_knn <- select_best(tune_results, metric = "accuracy")
#final_knn <- finalize_workflow(knn_wf, best_knn)
#final_knn_fit <- fit(final_knn, data = df_train)
#end_time <- Sys.time()
#elapsed <- end_time - start_time
#elapsed
# 2. Create a confusion matrix for the `test` data
#knn_preds <- predict(knn_fit, new_data = df_test)

#knn_test_res |>
#  conf_mat(truth = offense_category, estimate = .pred_class)

# test error
#test_res |>
 # accuracy(truth = offense_category, estimate = .pred_class) |>
#  mutate(error = 1 - .estimate) |>
#  pull(error)
```

Random Forest

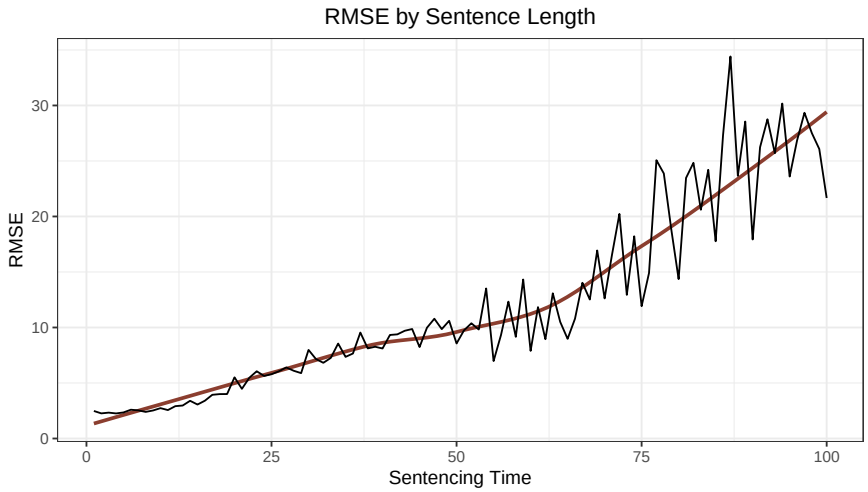
We made some decisions for this model, filtering out people with the hair color "Sandy" as there were very few observations (5). We also decided to not predict for people who were sentenced to zero years, as there were too many data entry issues around these rows which made no sense. This still left us with a large dataset at 59388 observations, where we decided random foresting would be a good model to run our data through for predicting a prisoner's sentencing time in years.The use of random forests allows for several trees to be fit with random subsets of our data and then combines results to better assess predictions. We found random foresting to be a good option because of its built-in out-of-bag feature which allows for us to run the

model on all predictors and not worry about cross-validation. We didn't want the correlated trees you would expect in boosting, random forests don't care for the most important features it learns from tree to tree. It builds its trees on random subsets of predictors each time, while boosting begins to "boost" predictions by further finding optimal splits with the important features it learns as it goes. We went with 500 trees, and 4 predictors for each tree because it's common practice to choose the square root of how many predictors you have which is $\sqrt{20} = 4.47$. It's also important we note that we kept a similar training split for this model at 70:30 ($\sim$ 17 41,200:17,600). Once this model was fit, we were eager to look into the feature importance plot to see if it was any different to our previous models.



The predictor with the most importance in this case is the year the prisoner was admitted, which instantly tells us based off our model that the time period they were sentenced plays the biggest role in sentence length. It's interesting to think about and hypothesize why this may be; Has the way certain charges have been punished drastically changed throughout the years? Another predictor that stood out from this variable importance plot was total counts, which makes sense looking at it from a judge's point of view. It makes sense that a person with more charges historically will often get a longer sentence.

We checked model performance first by looking at the out-of-bag RMSE which measures prediction error on the training split by evaluating each tree on the subset of data not used during its training. We got a value of 12.11, which up to this point was the best, but still not ideal at first glance. We also looked at the test RMSE which checks the model performance on the test split and we found a very similar value at 12.18, meaning our model generalized well. We took a further step to look into why our predictions were off that much by grouping by sentencing length predicted and then calculating RMSE per sentencing length. The motivation here was that we feared our model was no good if we were off by 12 years for people with sentencing lengths of 1-5 years. As you can see below, we were very happy to find that this was not the case. We don't get into that 12 RMSE region until just about 50 predicted sentencing years. With such a broad range from 1-100 on our response variable, I would actually consider our RMSE not that bad with all things considered.



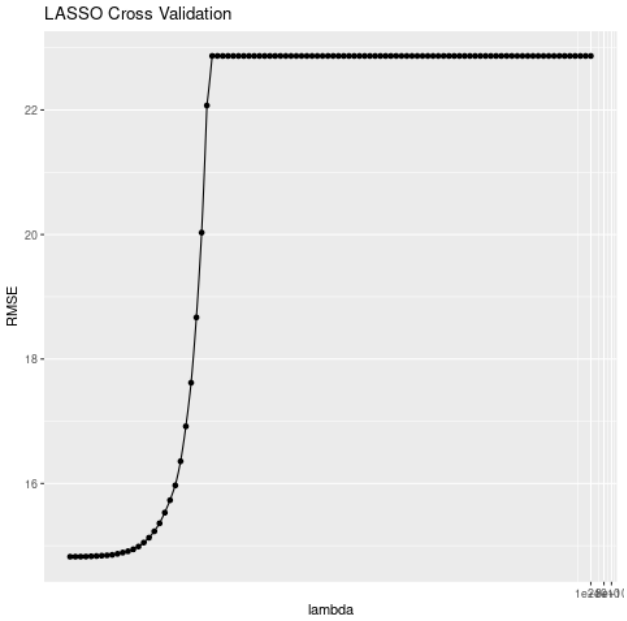
Linear Regression

To explore the relationship between prison sentence and our other variables, we chose to analyze a variety of linear models for their superior interpretability over non-parametric models. Because we were interested in identifying the most important variables for prison sentence prediction, we chose not to fit either an ordinary least squares regression model or a Ridge regression model. Both of these models would have assigned a non-zero coefficient to every predictor in the model, requiring every variable to be incorporated into the interpretation of a given prison sentence. We suspected that not every variable would be important so we focused on models that have a component of feature selection, including LASSO and subset selection models.

The first model we fit was the LASSO model, which selects optimal predictor coefficients by minimizing the combination of residual sum of squares and a penalty shrinking the size of those coefficients according to the formula below.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS } + \lambda\sum_{j=1}^{p}|\beta_j|$$

The penalty parameter λ was chosen according to 10 fold cross validation using 100 possible values of λ between 0.01 and 10×10^10. As shown on the lambda vs. RMSE graph, cross validation selected $\lambda = 0.01$.



Unfortunately, due to the number of categorical predictors in our dataset, LASSO was not as helpful as we would have liked. Categorical predictors must be reference encoded for LASSO to compute coefficients, turning our 20 predictors into 88. Of these 88 terms, LASSO only selected out the following five variables:

term	estimate
height	0
hair_Salt.and.Pepper	0
sex_Male	0
race_Bi.Racial	0
race_White	0

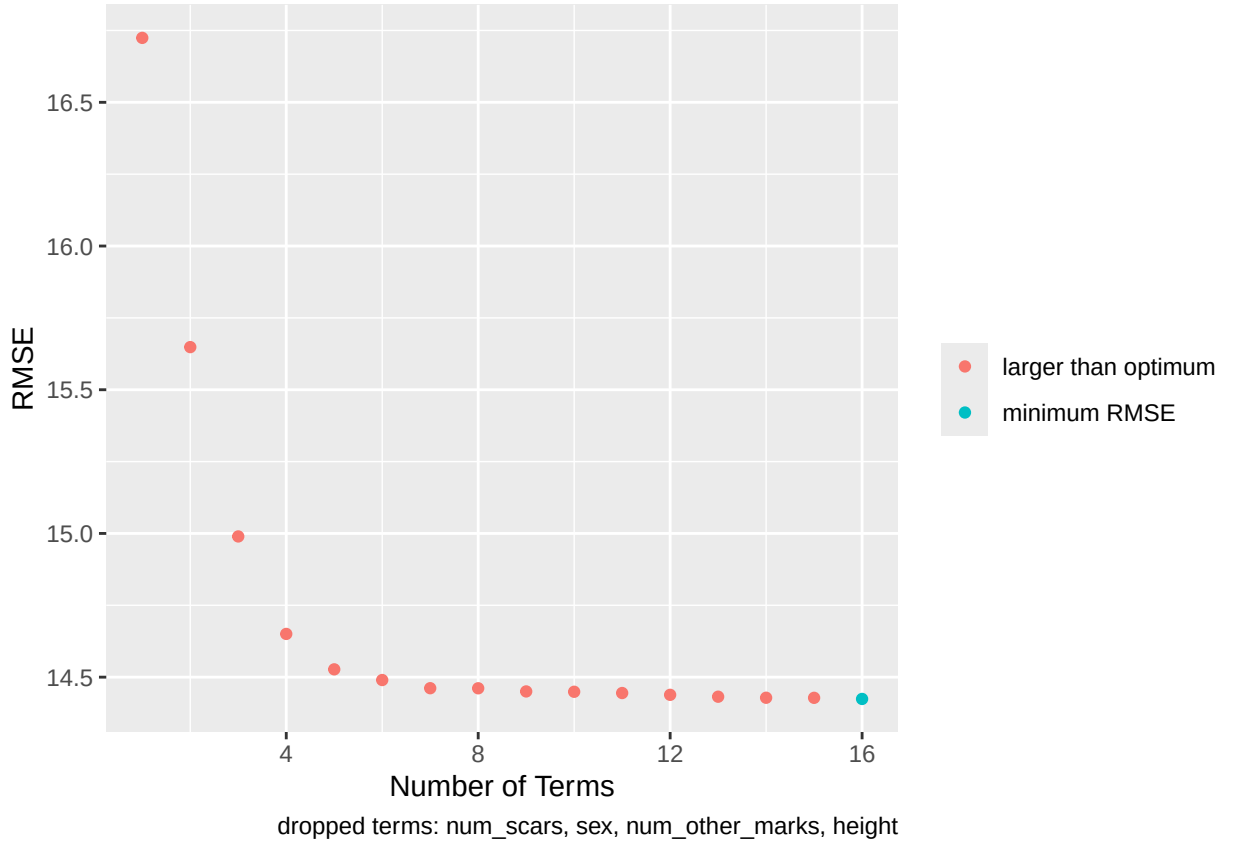
This tells us that height is not a significant predictor of prison sentence length. The other four variables, however, are individual categories from hair color, sex, and race respectively. These do not encapsulate whole categorical variables, so this information is not meaningfully interpretable, as it is not possible to only drop some categories from a categorical variable in linear regression.

Since LASSO did not produce as interprative a model as we had hoped, we moved on to subset selection. We elected not to perform best subset selection, as this method of subset selection requires fitting a model with every possible combination of variables and we decided that fitting approximately 2^20 models would be too computationally expensive. Instead we fit both forward and backward subset selection. Using the metric of adjusted R^2, forward and backward subset selection selected the same combination of variables, excluding the number of scars an individual has, their sex, number of other marks, and height, and including the following variables in order of importance.

step	variable	step (cont)	variable (cont)
1	year_adm	9	armed
2	class	10	attempted
3	parent_institution	11	race
4	total_counts	12	weight
5	offense_category	13	month_adm
6	aggravated	14	time_sentenced_prior
7	appx_age	15	hair

step	variable	step (cont)	variable (cont)
8	eyes	16	num_tattoos

```
rmse <- read.csv("LinearRegressionOutputs/rmse_v_terms.csv")

ggplot(rmse) + geom_point(aes(num_terms, RMSE, color = RMSE == min(RMSE))) + scale_color_discrete(labels
```



if the relationship is approximately linear, least squares will have low bias if we have many data points, n » p, also have low variance – do have many data points limitations to linear model: linearity is an approximation

use LASSO, forward, backward to obtain more interpretable model, drop predictors not associated with the response

not doing best subset because there are over 20 predictors and fitting approximately 2^20 models would be too computationally expensive; not doing principle component regression because interpretability again; ridge doesn't eliminate does not eliminate predictors and I care about interpretation

Ridge regression: RSS + $\lambda \sum_{j=1}^{p} \beta_j^2$ where λ is the tuning parameter - variable selection - LASSO - Ridge - results

GAM

We decided to use fit a generalized additive model(GAM) to see how well it can predict sentencing length and what predictors have the strongest effect on sentencing length. GAMs are an extension of linear regression that are used to show relationships between the predictors that can be better explained non-linearly while being easily interpretable. We then used 10 fold Cross-Validation to evaluate the model

When fitting a model that includes physical descriptors and a model without any of the physical descriptors. The model that was able to best predict sentencing length was:

This model shows that:

- results

Conclusion

here's why we pick x model over y model

References

kaggle data set