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

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

- basic explanation of model In order to answer our second two research questions regarding prediction of offense category, as well as determining 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 18 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.
- results Because our dataset is so large (even with a 70-30 training-test split), I expect the cross-validation to take a very long time to run, and I have temporarily commented out that code. I may need to make modifications to the training-test split, as well as the number of possible k-values, to increase efficiency. Once complete, I will create a confusion matrix to assess the model (with the optimal k) on the test set.

```
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,
```

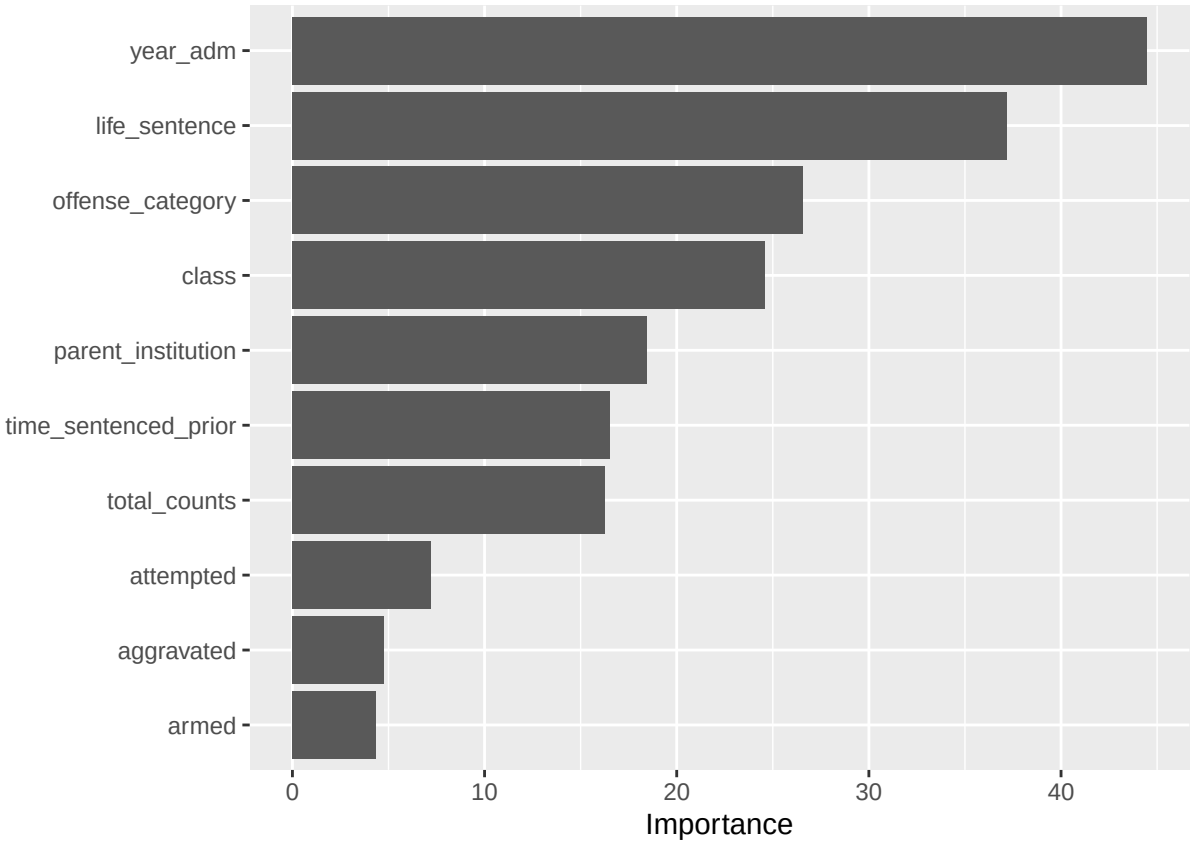
```
         weight_lbs = as.numeric(weight),
         sex = as.factor(sex),
         height_in = as.numeric(height),
         race = as.factor(race),
         eye_color = as.factor(eyes),
         hair = as.factor(hair),
         class = as.factor(class),
         parent_institution = as.factor(parent_institution)) |>
  select(-c(eyes, height, weight, appx_age)) |> na.omit()
df_split <- initial_split(cleaned_doc_data, prop = 0.7)
df_train <- training(df_split)
df_test <- testing(df_split)
# FULL EXPECTED RUNTIME: 6+ hours?
#start_time <- Sys.time()
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

(Want to use this in intro) We have taken 3 datasets from kaggle 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)

With such a large dataset at 59388, we decided random foresting would be a good model to run our data through for predicting a prisoners sentencing time in years. The use of random forests allows for several

trees to be made with random susbsets of our data and then combine to better assess predictions. We found random foresting to be a good option as well 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 can also pull the variable importance from the model and see which predictors had the most importance in predicting sentencing time.



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? (Look and plot effect of year admitted to better understand.) A couple of other predictors that stood out from this variable importance plot are total counts, which makes sense looking at it from a judges point of view. A person with more charges historically will often get a harsher sentencing. There is also our categorical predictor for life sentence, which we were positive would play a big role in predicting for sentencing length because of how we included it in our code. (explain earlier in code that we dealt with anyone getting sentenced "LIFE" by giving them 100 years, and capping anyone sentenced above 100 years to 100.) (Make sure to double check vip plot for all data as this is a subset of the data for paper to knit faster.)

While the subset of our data (3000 rows) used for our rough draft got a OOB rmse of 12.8 years and a training rmse of 12.1 years

- variable selection
- results

Linear Regression

- basic explanation of model quantitative response: current sentence

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

- basic explanation of model
- results

Conclusion

here's why we pick x model over y model

References

kaggle data set