Predicting Parole Violations

ST 635 Group 1: Ethan Clabaugh, Jaime Fastino, Cayty Fitzgerald, Nicole Romangsuriat, Jessica Smith

Parole Violations Data and Research Questions

This analysis explores data regarding parole violations, conducting various analyses to determine cause-and-effect relationships and interactions between predictor variables. We believed that this topic was very different from others that we have looked into in this class and in other classes we have taken at Bentley. We agreed that it would be an interesting topic to focus our final project on. We found a dataset from Kaggle, a data science platform and forum for finding public datasets, that contains data on parole violations from 2004. This dataset contains 675 entries and was obtained from the US 2004 National Corrections Reporting Program, a nationwide census of parole releases in 2004 (Kaggle, 2019). The data includes paroles that have been served for at most 6 months and parolees whose prison sentences did not exceed 18 months (Kaggle, 2019).

In this dataset, we identified a categorical response variable with whether a parole was violated or not. This is a binary variable as data values are "yes" or "no", coded as 1 or 0. We identified eight predictor variables, including five categorical ones including "male" (referring to gender), "race", "state", "multiple.offenses", and "crime", and 3 quantitative ones including "age", "time served", and "max sentence". With these variables, we identified three research questions to base our analysis on. Our first question was "Which individual predictor has the greatest impact on parole violations?". We wanted to investigate whether demographic variables weigh heavier than conviction details, and examine the type of individual most likely to violate parole based on personal characteristics and criminal history. Our second question was "Which variable combinations are most predictive of a parole violation", which explores interaction terms within the predictors. Our third question was "Can we create a model that accurately predicts parole violations?", which compares various statistical models and compares accuracies.

Methodology

Our analysis focuses on four topics; logistic regression, ridge and LASSO, and tree classification. We started by loading the data in the form of a CSV file onto R, conducting exploratory data analysis using the str function to see the list of variables, the number of observations, and the types of variables we were working with. We proceeded with observing some summary statistics on the data with the summary function, and the mean data for the variables to give us an understanding of the average of each variable. Afterwards, we checked the data for any missing values, in which we found that none were missing. This was followed by unique methodologies in handling the various analyses, as explained in further sections.

To compare analysis and results across the three methods, we tested each created model for model utility by considering the area under the curve, goodness-of-fit and model assumptions using various model-specific techniques, and accuracy by using confusion matrices that utilized the same training and testing data split. To create confusion matrices, this required changing some variables into factors before running the confusion matrix function. We set our fitted values to be a binary value of 1 if they were greater than 0.5, and a value of 0 if it was less than 0.5. With this, we decided on the best model for making accurate parole violation predictions and answered our research questions.

Logistic Regression

The first analysis model was logistic regression. We created a logistic regression model by fitting the model with all the predictor variables and evaluated their respective p-values. The p-values for the variables male, age and crime were very high in the initial fitting of the model. We revised the model to exclude them, re-examining the p-values derived from the new model's variables, which were all small and indicated significant variables. The variables for our final logistic regression model were race, state, time served, max sentence and multiple offenses.

Ridge and LASSO

The second set of analyses were Ridge and LASSO, which are useful in reducing overfitting in a model or in cases where multicollinearity is present. Ultimately, with the Ridge and LASSO techniques, we are observing what impacts regularization and penalization have on our model's performance, if any. Although we did not have severe multicollinearity in our logistic regression model described above, we were able to divide the data into testing and training sets to assess the impacts of Ridge and LASSO on the mean squared error ('MSE') and area under the curve ('AUC') of the test set. We fitted two logistic regression models leveraging Ridge and LASSO regularization, and once each model was applied using the testing dataset, we calculated the AUCs for each and compared results. Additionally, we observed the optimal regularization parameters (lambda values) for both the Ridge and LASSO models to see if they were close to zero, indicating that the regularization parameter is not helpful for our models.

Tree Classification

Our third analysis tool is a classification tree. A classification tree is a useful tool for this data because the target variable, whether the subject violated their parole, is a binary variable, which classification trees require. Similar to the previous analyses, splitting the data into a training and testing set is beneficial to judge the accuracy of the model with new data after it is built. Another benefit of using a classification tree is that it acts as an automatic method of variable selection. We began by building a full tree with a CP of zero before pruning the tree to prevent overfitting that would result in worse predictions for future data. When we printed the CP table, there were only 3 trees; a full tree, a null tree with zero splits, and one in between. In observing the xerror column (the minimum xerror corresponds to the optimal CP), the smallest

value was the null tree. As this would not provide any sort of analysis, we chose to proceed with the next smallest value, which was the pruned tree in between the null tree and the full tree.

Summary of Results

As aforementioned, we noted each generated model's respective implications and results, considering model utility, goodness-of-fit, model assumptions, and accuracy.

Logistic Regression

The logistic regression model received an AUC of 0.767 which is significantly higher than 0.5 and indicates usefulness. We executed the likelihood ratio test to evaluate goodness-of-fit and found that value to be 2.021985e-17, or close to 0, also good as it is less than 5%. We further tested model assumptions including linearity, which holds as our crPlots all show rather straight and aligned pink and blue dotted lines on each plot. There is also no multicollinearity as our values from the vif function are all close to 1. In regards to accuracy, the confusion matrix illustrated a value of 88.3%, determined that 596 observations were correctly identified, and showed a misclassification rate of 11.7%. We also can see that the model has more specificity (0.98995) and less sensitivity (0.064103).

Ridge/LASSO

The models with Ridge and LASSO regularization had AUC values of 0.6909 and 0.6959, respectively. These values are higher than 0.5, indicating a moderately good model fit. Given that these values are similar to one another and are lower than the one from our logistic regression model, this implies that there is no severe overfitting or multicollinearity present in our logistic regression model and the penalization techniques were not useful. Additionally, the optimal regularization parameters (lambda values) showed little impact on our model, as the values in both of our Ridge and LASSO models were close to zero (0.04 and 0.006).

Classification Tree

The pruned classification tree had an AUC of 0.799, also higher than the baseline of 0.5 and indicating model usefulness. The generated confusion matrix using the testing data had an accuracy of 87.68%, another high result that is indicative of a good model. This was compared to the full tree which had an AUC of 0.758 and an accuracy of 85.22%. With a higher AUC and accuracy result, the pruned tree that included age, multiple offenses, race, and state is evidently better at correctly predicting parole violators.

Discussion of Findings: Model Comparison

In analyzing our various analysis outcomes, we started by comparing the highest obtained AUC values for logistic regression, ridge and LASSO, and decision tree. Ridge and LASSO achieved the lowest but still higher than 0.5 value of 0.696. This was followed by logistic regression which was utilized as a baseline for comparison with a value of 0.767, which was

beaten by the decision tree, specifically the classification tree which had a value of 0.799. This illustrates that the classification tree which utilized age, multiple offenses, race, and state had the best model out of the three in regards to model utility. We proceeded by comparing accuracy measures for all four models. This considered the confusion matrix accuracy values for the aforementioned three models for comparison. Logistic regression had a value of 88.3%, classification tree had a value of 87.7%, ridge and LASSO both had a value of 88.8%. The ridge and LASSO models were most accurate, though by merely 1%.

Conclusions and Recommendations

To conclude our research and analysis, we answer our initial research questions. The generated classification tree that utilizes age, multiple offenses, race, and state as predictors was the best overall model out of the four as we consider utility, goodness-of-fit, and accuracy. However, due to all three analyses generate three accurate and successful models, all three could have been suitable as binary models for further analysis and predictions. Despite classification tree being the best method for this, the logistic regression and ridge and LASSO models can somewhat accurately predict parole violations as well, with good accuracies of roughly 87% to 89%. With this, the aforementioned predictor variables of age, multiple offenses, race, and state composes the best variable combination most predictive of a parole violation. Furthermore, multiple offenses was the individual predictor that had the greatest impact on parole violation, as determined by having the highest variable importance value that was found in the decision tree summary analysis.

With our classification tree, we created an "average" person from the data and predicted whether they will violate parole. Using the means of the 4 key variables, we created a person who is white, 34.5 years old, from Louisiana, with multiple offenses. This person is predicted to have a 0.71 probability of not violating their parole, assigning them to the non-violator class.

References

Predicting parole violators. (2019, December 28). Kaggle. https://www.kaggle.com/datasets/econdata/predicting-parole-violators/data

Appendix: R code

```
#Load all required packages
library(car)
library(caret)
library(pROC)
library(mlbench)
library(glmnet)
```

```
library(tidyverse)
library(rpart)
library(rattle)
library(RColorBrewer)

#Logistic Regression

#Load data
data <- read.csv("C:/Users/18605/Downloads/sqldeveloper-21.4.3.063.0100-
x64/ST635/parole.csv")
str(data)
summary(data)
is.na(data)
data$violator <- as.factor(data$violator)
data$race <- as.factor(data$race)
data$state <- as.factor(data$state)
data$multiple.offenses <- as.factor(data$multiple.offenses)

#Split dataset
set.seed(1) # for reproducibility
training_rows <- createDataPartition(data$violator, p = 0.8, list = FALSE)
train_data <- data[training_rows, ]
test_data <- data[-training_rows, ]

lrmodel1 <- glm(violator~., data = train_data, family = binomial(link="logit"))
summary(lrmodel1)
lrmodel <- glm(violator~race + state + time.served + max.sentence + multiple.offenses, data =
train_data, family = binomial(link="logit"))
summary(lrmodel)

## Confusion matrix
predviolator <- test_data$violator
predviolator[lrmodel$fitted.values>=0.5] <- 1
predviolator[lrmodel$fitted.values<0.5] <- 0
confusionMatrix(predviolator , test_data$violator, positive = '1')

## ROC & AUC
roc(test_data$violator, lrmodel$fitted.values, plot = T, print.auc = T)

## Likelihood ratio test
```

```
pchisq(lrmodel$null.deviance - lrmodel$deviance, 1, lower.tail = F)

# Diagnosis
summary(lrmodel)
crPlots(lrmodel)
vif(lrmodel)
coef(lrmodel)
predict(lrmodel, newdata = data.frame(race = 1, state = 2, time.served = 5.5, max.sentence =10,
multiple.offenses = 1),type = "response")

#Ridge % Lasso Regression

#Regularized Models
x_train <- as.matrix(train_data[,1:8])
y_train <- train_data$violator

# Lasso
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1)
lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = cv_lasso$lambda.min)

# Ridge
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0)
ridge_model <- glmnet(x_train, y_train, alpha = 0, lambda = cv_ridge$lambda.min)

# 3. Model Evaluation
x_test <- as.matrix(test_data[,1:8])
y_test <- test_data$violator

# Baseline
baseline_preds <- predict(baseline_model, newdata = test_data, type = "response")
baseline_roc <- roc(y_test, baseline_preds)

# Lasso
lasso_preds <- predict(lasso_model, newx = x_test, type = "response", s = cv_lasso$lambda.min)
lasso_roc <- roc(y_test, lasso_preds[,1])

# Ridge
ridge_preds <- predict(ridge_model, newx = x_test, type = "response", s =
cv_ridge$lambda.min)
ridge_roc <- roc(y_test, ridge_preds[,1])
```

```
# AUC values
auc(baseline_roc)
auc(lasso_roc)
auc(ridge_roc)

#PCA
predictors <-
c('male','race','age','state','time.serverd','max.sentence','multiple.offense','crime','violator')
predictors.PCA <- setdiff(predictors, 'Violator')
PCA <- prcomp(train_data[, colnames(train_data)%in%predictors.PCA],center = T, scale. = T)
summary(PCA)

#Calculate Variance
var_explain <- PCA$sdev^2 /sum(PCA$sdev^2)

#Create a scree plot
plot(var_explain, xlab = "Principal Component", ylab = "Proportion of Variance Explained",
type = "b", pch= 18, main = "Scree Plot of the Parole Dataset"
    ,ylim= c(0,1))
cum_var_explain <- cumsum(var_explain)
lines(cum_var_explain, col= "red", type= "b", pch= 18)
#Loadings (the importance of each variable to the principal components)
loadings <- PCA$rotation
print(loadings)

#Extract scores from first two PCs
PCA_scores <- data.frame(PCA$x[, 1:5])
summary(PCA_scores)

# Fit a linear regression model
PCA_scores$violator <- train_data$violator
model_PCA <- lm(violator~ ., data = PCA_scores)
# Summary of the regression model
summary(model_PCA)
# Compare adjusted R^2 with the full linear model
model_LM <- lm(violator ~ ., data = train_data)
summary(model_LM)
```

```
#Decision Trees

#Unpruned tree with all predictors
tree1 = rpart(violator ~ ., data = train_data, method = 'class')
printcp(tree1)
fancyRpartPlot(tree1)

#Prune the tree
tree.prune.m <- prune(tree1, cp = 0.02)
printcp(tree.prune.m)
fancyRpartPlot(tree.prune.m)

#Confusion Matrix
predClass.p <- predict(tree.prune.m, newdata = test_data, type = 'class')
confusionMatrix(predClass.p, test_data$violator)

#AUC Plot
predProb.p = predict(tree.prune.m, test_data, type="prob")[, 2]
roc(test_data$violator, predProb.p, plot = TRUE, print.auc = TRUE)

#Comparing Pruned vs. Full Tree
predClass.f <- predict(tree1, newdata = test_data, type = 'class')
confusionMatrix(predClass.f, test_datat$violator)

predProb.f = predict(tree1, test_data, type="prob")[, 2]
roc(test_datat$violator, predProb.f, plot = TRUE, print.auc = TRUE)
summary(tree.prune.m)

# “Average” person prediction
mean(parole$multiple.offenses)
mean(parole$race)
mean(parole$state)
mean(parole$age)

newdat = data.frame(multiple.offenses = 1, race = 1, age = mean(parole$age), state = 3, male =
0, time.served = 0, max.sentence = 0, crime = 0)
predict(tree.prune.m, newdata = newdat, type = "prob")
predict(tree.prune.m, newdata = newdat, type = "class")
```