Business Analytics ML Project – Ning (Nikki) Gong

Dataset Introduction

The data is an overview of how probation cases are processed in 32 urban and suburban jurisdictions in the US. Data was collected on offenders who were sentenced in 1986, and had committed one or more felony crimes including homicide, rape, robbery, aggravated assault, burglary, and others. During the latter half of 1989, probation history questionnaires were completed. The questionnaire provides information on offenders' information such as number of conviction charges, race, age, sex, marital status, education level, ethnicity, drug & alcohol use, as well as whether the probationer was arrested while on probation period.

In the dataset, the independent variables are the features or characteristics of the probationer; the dependent variable is whether he was arrested while on probation period.

Value Preposition

I propose that predicative analytics, classifying whether the probationer will recommit crimes during his probation period, has a major value opportunity.

Cost Saving through Cognitive Treatment: I can save costs by classifying these probationers for selective administration of cognitive treatment. Nowadays, the government has tried to develop and deliver cognitive treatment to offenders who committed felony crimes. Such cognitive treatment is a kind of Cognitive-Behavioral Therapy (CBT), which can effectively treat a wide range of mental disorders and has proven record to reduce recidivism. More recently, researchers have found that such cognitive treatment can help reduce recidivism by 25 to 35 percent.[1] Recidivism costs $31,286 per inmate per year on average based on incarnation estimates.[2] Full administration of cognitive treatment leads to waste and would not be beneficial. My hypothesis is that predictive analytics can help me to capture at least some of the value that perfect foresight might bring.

In this report, I will focus on estimating the value preposition given the confusion matrix and appropriate assumptions.

Methodology Walkthrough

Data Cleaning & Extraction, Filling in Missing Values

The raw data "probation-recidivism-raw.dta" obtained had to be converted into .csv in R.[3] Upon reading "probation-recidivism-raw.csv", the data contained 149 variables on 12,369 probations including their features/characteristics and if they had reoffended within probation period. Using 09574-0001-Codebook.txt , I identified the meaning of the variables and singled out the most relevant ones for predicting the probability of recidivism for a probationer. My next step was to replace the numeric names of the variables with labels for easier interpretability and analysis. The labels I chose and their respective meanings are listed below:

- **conv_num:** The number of prior convictions of the probationer.

[1] https://www.npr.org/sections/health-shots/2016/06/26/483091741/to-help-a-criminal-go-straight-help-him-change-how-he-thinks
[2] https://www.npr.org/sections/health-shots/2016/06/26/483091741/to-help-a-criminal-go-straight-help-him-change-how-he-thinks [3] https://www.icpsr.umich.edu/icpsrIb/NACJD/studies/9574/version/2

- **gender:** The gender of the probationer.
- **emp_perc:** The percentage of the time the probationer was employed prior to conviction.
- **wage_hr:** The hourly wage of the probationer.
- **mart_status:** The marital status of the probationer.
- **educ:** The level of education the probationer received.
- **drug_hist:** If the probationer has a drug abuse history.
- **felony_num:** Number of felony arrests of probationer.
- **supervis:** If probationer was put under supervision.
- **prob_length:** Length of probation sentence in months.
- **age:** The age of the probationer.
- **race:** The race of the probationer.
- **Offen_type:** The type of offence committed by the probationer.
- **risk_score:** The risk score allotted to the probationer.
- **arr_probation:** If the probationer was arrested while on probation.

The Y variable that I chose, the variable that I want to predict, is arr_probation, which is whether the probationer was arrested for recommitting a crime while on probation. After identifying the Y variable, and putting in the labels as names, my next step was to deal with the missing values ("NA") occurring in the dataset.

To overcome this challenge, I chose Multiple Imputation. **Multiple imputation** is a statistical technique for analyzing incomplete data sets, that is, data sets for which some entries are missing. Application of the technique requires three steps: imputation, analysis and pooling. A key assumption for using this technique is assuming the missing entries in the dataset appear at random, which, I believe, is valid in my case. This assumption is based on the fact that all individuals are not correlated to one another and do not miss details based on a pattern.

For imputation, I identified that the variables offence type and risk score had no NAs. I decided to remove them from the imputation to save computational cost. I ran imputation five times, upon which I got a complete dataset with filled in values.

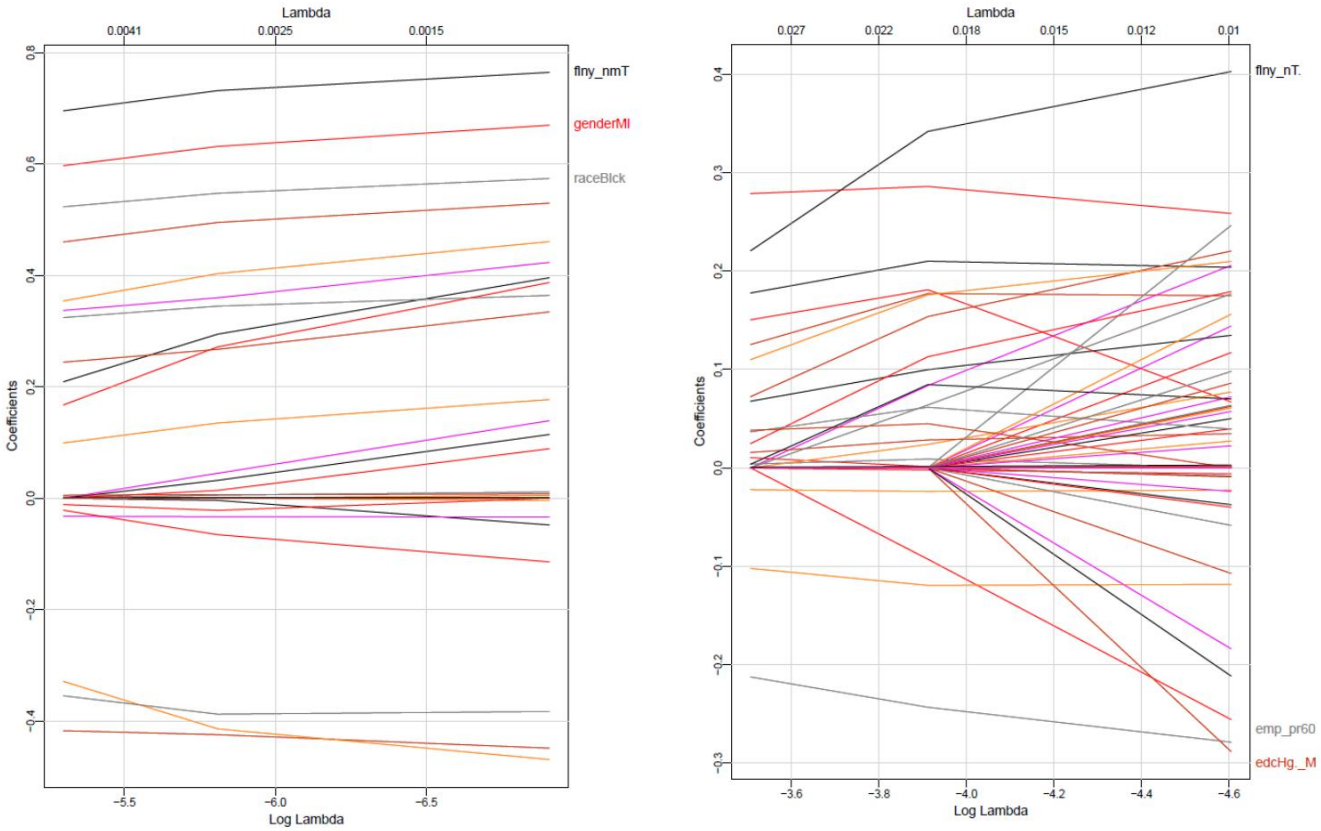
Nonracial Considerations

Besides finding the best model for the prediction, I am interested in figuring out how significant racial factors are playing out in my prediction model. Many past algorithms in fact have been criticized for being discriminatory towards certain races by predicting for them a higher recidivism probability.[4]

An ideal result would be that racial factors are not essentially causing the outcome, but contingently relate to the outcome. More specifically, racial factors are highly correlated with other factors like education, wage level, drug use which are causal factors of whether the criminal will recommit during the probation period. In other words, keeping all other factors constant (i.e. they got same education reslces, same marital status and so on), the race of the person won't affect his probability of recidivism during probation period.

[4] http://fairness-measures.org/Pages/Datasets/Compas.html

I first center the continuous variables for an average interpretation[5]. I suspect that after adding interaction terms, the race variables, their polynomials, and interactions will not be as significant as before.



As shown above, the left graph represents when I run lasso on logistic regression on the dataset of variables and their polynomials only, while the right represents that on logistic regression using the dataset with variables, their polynomials, and interaction terms. The rightmost axis is the cutoff for optimal respective lambda. The effect of number of more than two prior felonies on recidivism is perhaps self-explanatory. However, it is revealing to find while the lasso without interactions showed race & gender variables (genderMale & raceBlack) to be significant, including interactions enabled me to uncover the true causes, which are in fact employment, high school education, and previous convictions of murder.

Sampling Data and Model Selection

First, I randomly divide *rec_data_interactions_cen* and *rec_data_polys_cen* into training (50%), validation (25%), and test (25%) sets.

$$rec\_data\_interactions\_cen \begin{cases} train\_data\_interactions \\ valid\_data\_interactions \\ test\_data\_interactions \end{cases}$$

$$rec\_data\_polys\_cen \begin{cases} train\_data\_polys \\ valid\_data\_polys \\ test\_data\_polys \end{cases}$$

[5] It can be proved that by centering, the coefficient of raceBlack, for instance, can be directly interpreted in an interactions regression. This is when other dummy variables are switched off, and the continuous variables are centered i.e. for a probationer with average characteristics. See enclosed Interpreting Interactions & Centering.pdf.

I then want pick the best modelling to depict the relation between the probationer's background with his probability of recidivism during the probation period. More specifically, I will run different sorts of parametric and non-parametric modellings to fit in the training data

Here I will run Logistic Regression with Lasso, Decision Tree, LDA, and KNN. I will then pick the one that has the lowest classification error on the validation set, and retrain the model on the entire training and validation (75%) set.

❖ *Logistic Regression Model*

I use a logistic model on train_data_polys to gauge, and found out that raceBlack, was significant with a low p-value. The results of the regression are presented below.

```
glm.fit.polys = glm(arr_probation~.,data=train_data_polys,family=binomial)
summary(glm.fit.polys)
```

```
supervis1              0.20635016  0.13484648   1.530           0.125952
raceBlack              0.84221043  0.21689467   3.883           0.000103 ***
raceIndian             0.43870219  0.34867448   1.258           0.208319
```

I then use a logistic model on train _data_interactions, and found that raceBlack, other race variables, their polynomials, and interaction terms are not some of the most relevant predictors. Adding interaction terms can address the problem of racial bias as seen here, and in my lasso graphs comparison earlier.

```
glm.fit.interactions = glm(arr_probation~.,data=train_data_interactions,family=binomial)
summary(glm.fit.interactions)
```

```
supervis1                              17.87969684 2399.54490489   0.007     0.9941
raceBlack                              15.80223888 2399.54539231   0.007     0.9947
raceIndian                              4.13088377 3466.39757359   0.001     0.9990
```

To select my relevant predictors from the vast number, I train the model on train_valid_ interactions using 10-fold Cross-Validation and Lasso to optimize lambda parameter.

```
library(glmnet)
x = model.matrix(arr_probation~.,train_data_interactions)
y = train_data_interactions$arr_probation
grid = 10^(-4:4)
cv.out = cv.glmnet(x,y,type.measure="mse",alpha=1,lambda=grid,family="binomial",nfolds=10)
bestlam = cv.out$lambda.min
bestlam
```

The best lambda is 0.01. Then retrain the model on train_data_interactions using best lambda.

```
lasso.mod = glmnet(x,y,alpha=1,lambda=bestlam,family="binomial")
summary(lasso.mod)
library(plotmo)
glm.lam = glmnet(x,y,alpha=1,lambda=c(0.01,0.02,0.03),family ="binomial")
plot_glmnet(glm.lam,label=5,xvar="rlambda",grid.col ="lightgray")
```

I can see that with best lambda 0.01, raceBlack is not one of the most relevant predictors. I run the interactions model on valid_data_interactions to get confusion matrix and classification error of 0.331.

```
x1 = model.matrix(arr_probation~.,valid_data_interactions)
y1 = valid_data_interactions$arr_probation
pred = predict(lasso.mod,x1,type="class")
table(pred,y1)
lasso_error = mean(pred!=y1)
lasso_error
```

❖ *Decision Tree*

I train decision tree on *train_ data_polys* and use Cross-Validation to optimize the prune parameter, which is 2. I then prune the tree into the best size.

```
library(tree)
tree.fit=tree(arr_probation~.,data=train_data_polys)
set.seed(1)
cv.fit = cv.tree(tree.fit)
plot(cv.fit$size,cv.fit$dev,type="b")
prune.fit = prune.tree(tree.fit,best=2)
plot(prune.fit)
text(prune.fit,pretty=0)
```

I then run the pruned tree model on valid_data_polys to get confusion matrix and classification error of 0.379.

```
tree.pred = predict(prune.fit,newdata=valid_data_polys,type="class")
table(valid_data_polys$arr_probation,tree.pred)
tree_error = mean(tree.pred!=valid_data_polys$arr_probation)
tree_error
```

❖ *Linear Discriminant Analysis (LDA)*

I train LDA on train_data_polys and run on valid_data_polys to get confusion matrix and classification error of 0.341.

```
library(MASS)
lda.fit = lda(arr_probation~.,data=train_data_polys)
lda.fit
lda.pred=predict(lda.fit,valid_data_polys)
lda.class = lda.pred$class
table(valid_data_polys$arr_probation,lda.class)
lda_error = mean(lda.class!=valid_data_polys$arr_probation)
lda_error
```

❖ *K-Nearest Neighbors*

I scale all the continuous variables to get train_polys_scale and valid_polys_scale.

I use Train-Validation to optimize best K, which is 10. I then train and test the KNN model to get confusion matrix and classification error of 0.377.

```
library(class)
library(caret)
predQuality = vector("numeric",10)
for (nn in 1:10){
  KNNpred = knn(x_train_polys_scale,x_valid_polys_scale,y_train_polys_scale, k = nn)
  predQuality[nn] = mean(KNNpred == y_valid_polys_scale)
}
best_k=which.max(predQuality)
print(best_k)
KNNpred_validation = knn(x_train_polys_scale,x_valid_polys_scale,y_train_polys_scale,k = 8)
table(y_valid_polys_scale,KNNpred_validation)
mean(KNNpred_validation != y_valid_polys_scale)
```

I removed all race related variables to take into account racial effects, and repeated KNN to get best K equals 10, and also confusion matrix and classification error of 0.379.
By comparing, classification rate is not improved by adding race variables. I can conclude including race variables will not add significant predictive power in KNN.

Confusion Matrix and Value Opportunity Calculation

In order to find the best threshold, I first have to estimate the benefits and loss when I make a right or wrong judgement about whether the probationer will recommit in the future. I derive the assumptions on which I can build a confusion matrix.

Assumptions	
Probability of recidivism reduction	25%[6]
Cost of recidivism to society ($)	31,286[7]
Cost of Cognitive-Behavioral Therapy (CBT) package ($)	3,200[8]

The confusion matrix will be used to generate the optimal threshold that converts probability into a decision.

```
cost = c(c(0,31286),c(3200,26664.5))
pr = seq(from=0.05, to=0.75, by=0.01)
glm_cost = vector()
for (p in pr)
{
  decision = ifelse(pred_prob_test>p,'Y','N')
  classificationTable = table(y_test,decision)
  glm_cost = c(glm_cost,sum(classificationTable*cost))
}
glm_cost
min(glm_cost)
best_cost_ind=which.min(glm_cost)
pr[best_cost_ind]
##best threshold = 0.4 and minimum cost is 35,721,538
decision_best = ifelse(pred_prob_test>pr[best_cost_ind],'Y','N')
table(y_test,decision_best)
test_error = mean(decision_best!=y_test)
test_error# test misclass rate is 0.333
```

Once I decide on the optimal threshold, which is 0.4, I calculate the cost savings of predictive analytics I can deliver to the government. This is a total of US$ 1,571,375 saved on taxpayer money for a sample test size of 3,092 probationers, compared to the case if Cognitive-Behavioral Therapy has not been used.

Use CBT with predictive analytics

	Treatment		
Outcome	Y	N	Total
Y	743	449	1,192
N	582	1,318	1,900
Total	1,325	1,767	3,092
Classificaiton Error:		*33.3%*	
True Positive Rate:		*62.3%*	
False Positive Rate:		*30.6%*	
Total Cost ($):	**35,721,538**		
Total Cost Savings ($):		**1,571,375**	

	Treatment	
Outcome	Y	N
Y	26,665	31,286
N	3,200	0

[6] Conservative estimate from 25 - 30%: https://www.npr.org/sections/health-shots/2016/06/26/483091741/to-help-a-criminal-go-straight-help-him-change-how-he-thinks

[7] Mainly based on incarceration (imprisonment) costs: https://www.npr.org/sections/health-shots/2016/06/26/483091741/to-help-a-criminal-go-straight-help-him-change-how-he-thinks

[8] About US$ 100 per session: http://www.academyofct.org/page/LowCost
Recommended weekly sessions for eight months: https://positivepsychologyprogram.com/cbt-cognitive-behavioral-therapy-techniques-worksheets/