Final Project

Data Mining: Fall 2020

Brooke O’Donnell

December 13, 2020

This final project used data from a births data set collected in 2018 by the Center for Disease Control and Prevention (CDC). This dataset is accessible using the vital statistics online data portal on the CDC’s website. Or you can [**Click here**](https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm) to access more information regarding the dataset presented in the following project. “Method of Delivery” is a variable in the births 2018 dataset that recorded the mode of delivery for each birther; ‘vaginal birth’ or ‘cesarean birth’. “Method of Delivery” was recoded into a binary factor variable in attempt to predict whether the birthers in this dataset will have a vaginal birth or a cesarean section birth based on selected feature variables. The selected feature variables are listed below:

* Birth.provider = attendant at birth [medicalized (OBGYN) or midwife]
* Birth.location = (hospital or not in hospital)
* Prev.csection = reported a previous cesarean section (yes or no)
* Born U.S. = (yes or no)
* Num.prev.births = Number of previous births
* Age = birthers age
* PREVIS= number of prenatal visits
* SEX = baby’s reported sex (male or female)
* FRACE = Father’s race (white or non-white)
* MHISP = the birther identified as Hispanic (yes or no)
* Pay = payment method (government assistance, self-pay, private insurance)
* Risk = risks reported before birth (yes or no)

Part A:

1. First, what is the proportion of the chosen predictor varaible “Method of Delivery”?

meth<-select(final.df, Method.delivery, Birth.provider)

library(qwraps2)  
options(qwraps2\_markup = "markdown")  
meth <- as.data.frame(meth)  
summary\_statistics.meth <-  
 list(  
 "Method of Delivery"=  
 list(  
 "Vaginal Birth"= ~qwraps2::n\_perc(na.omit(Method.delivery) %in% "vaginal"),  
 "Cesarean Section" =~qwraps2::n\_perc(na.omit(Method.delivery) %in% "c.section"),  
 "Missing"=~sum(is.na(Method.delivery))  
 )  
 )  
summary\_table(meth, summary\_statistics.meth)

|  |  |
| --- | --- |
| **Method of Delivery** | **(N = 836,696)** |
| Vaginal Birth | 566,316 (67.68%) |
| Cesarean Section | 270,380 (32.32%) |

* There are two categories without classifying the cases. (68%) vaginal births and (32%) cesarean section births. Without any feature variables the GINI impurity measure for “Method of Delivery” is (.44).

head(G.meth)

## [1] 0.4367015

print(qwraps2::summary\_table(  
 dplyr::group\_by(meth, Birth.provider),  
 summary\_statistics.meth  
),  
rtitle = "Method of Delivery by Birth Provider"  
)

|  | |
| --- | --- |
| **Method of Delivery by Birth Provider** | | **Medicalized (N = 738274)** | **Midwife (N = 98422)** |
| **Method of Delivery** | |  |  |
| Vaginal Birth | | 469,109 (63.54%) | 97,207 (98.77%) |
| Cesarean Section | | 269,165 (36.46%) | 1,215 (1.23%) |

1. Does the variable “Birth Provider” lead to information gain if used as a feature variable?
   * The category ‘Midwife’ may lead to information gain is used as a feature variable.

set.seed(16)  
births.sample<-sample\_n(final.df, 9000)

tab1<-table(births.sample$Birth.provider, births.sample$Method.delivery)  
head(tab1)

##   
## vaginal c.section  
## Medicalized 5062 2885  
## Midwife 1039 14

gini.midwife<-1-((1005/(1005+19))^2+(19/(1005+19))^2)  
head(gini.midwife)

## [1] 0.03642082

gini.medicalized<-1-((4982/(4982+2907))^2+(2907/(4982+2907))^2)  
head(gini.medicalized)

## [1] 0.4654091

wgini.prov<-((4982+2907/9000)\*(0.4654091) + ((1005+19)/9000)\*0.03642082)  
round(wgini.prov, 3)

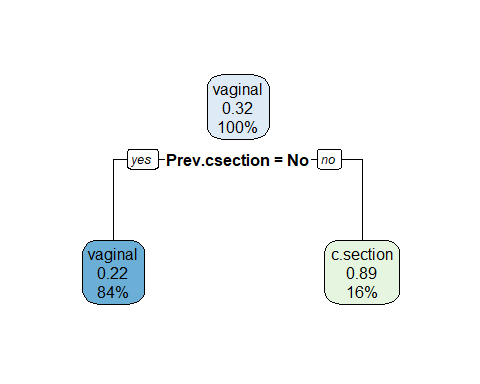
* + If “Birth Provider” was used as a feature varaible, birth providers classified as ‘Midwife’ would provide more informaiton than the birth providers classified as ‘Medicalized’. This is according to the lower GINI inpurity measue of (0.03642082) compared to the GINI impurity measure of (0.4654091) produced by the ‘Medicalized’ category. Indicating ‘Medicalized’ birth providers are almost split evenly half and half.

1. Split the data into (70%) training and (30%) testing sets. Build a decision tree for the target “Method of Delivery” by including all feature variables and setting the cp value to (0.01), (0.005), and (0.001).

set.seed(16)  
split<-sample.split(births.sample$Method.delivery, SplitRatio=0.7)  
train.df<-subset(births.sample, split==TRUE)  
test.df<-subset(births.sample, split==FALSE)

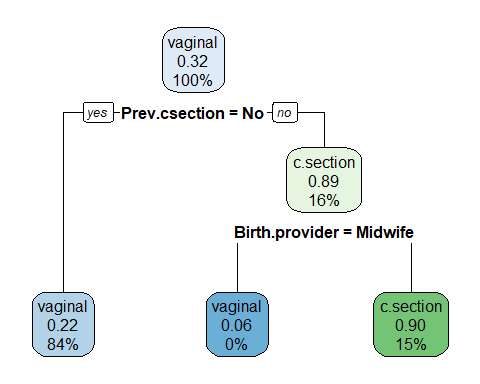
d.tree.01<-rpart(Method.delivery~.,  
 data=train.df,  
 method="class",  
 minsplit=2,  
 minbucket=1,  
 cp=0.01)  
printcp(d.tree.01)  
## Classification tree:  
## rpart(formula = Method.delivery ~ ., data = train.df, method = "class",   
## minsplit = 2, minbucket = 1, cp = 0.01)  
## Variables actually used in tree construction:  
## [1] Prev.csection  
## Root node error: 2029/6300 = 0.32206   
## n= 6300   
## CP nsplit rel error xerror xstd  
## 1 0.37703 0 1.00000 1.00000 0.018279  
## 2 0.01000 1 0.62297 0.62297 0.015666

rpart.plot(d.tree.01)



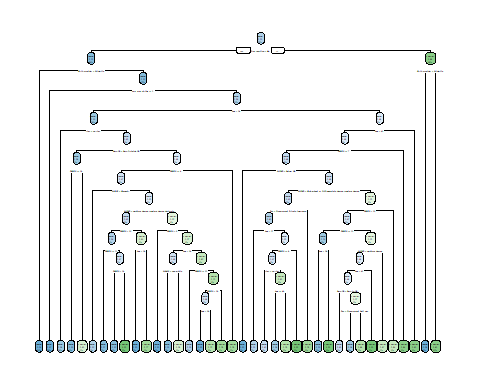
d.tree.005<-rpart(Method.delivery~.,  
 data=train.df,  
 method="class",  
 minsplit=2,  
 minbucket=1,  
 cp=0.005)  
printcp(d.tree.005)  
summary(d.tree.005)

rpart.plot(d.tree.005)



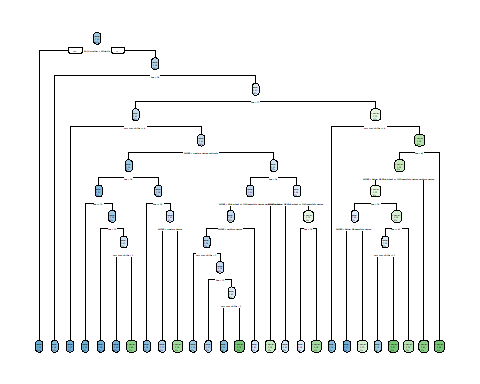
d.tree.001<-rpart(Method.delivery~.,  
 data=train.df,  
 method="class",  
 minsplit=2,  
 minbucket=1,  
 cp=0.001)  
printcp(d.tree.001)  
summary(d.tree.001)

rpart.plot(d.tree.001)

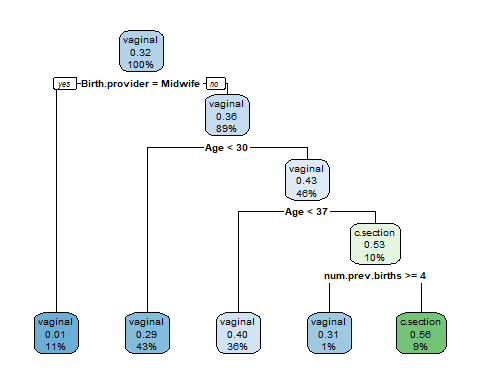


1. Identify problems with the models before tuning and pruning.
   * The first two decsion trees were set to cp(0.01) and cp(0.005); these parameters did not split enough nodes. The third decision tree ran into the problem of overfitting. In order to help both issues I decided to create two new models with key variables listed from the decision tree summaries above and set the cp values to (0.001) and (0.005).

d.tree.vi.001<-rpart(Method.delivery~Birth.provider+MEDUC+num.prev.births+Age,  
 data=train.df,  
 method="class",  
 minsplit=2,  
 minbucket=1,  
 cp=0.001)  
printcp(d.tree.vi.001)  
summary(d.tree.vi.001)

rpart.plot(d.tree.vi.001) ## Warning: labs do not fit even at cex 0.15, there may be some overplotting

d.tree.vi.005<-rpart(Method.delivery~Birth.provider+MEDUC+num.prev.births+Age,  
 data=train.df,  
 method="class",  
 minsplit=2,  
 minbucket=1,  
 cp=0.0055)  
printcp(d.tree.vi.005)  
summary(d.tree.vi.005)

rpart.plot(d.tree.vi.005) 

* + The key variables selected and displayed in the pruned decision tree were “Birth Provider”, “Birther’s Level of Education”, “Number of Previous Births” and “Age”. Reading the terminal node in the far-left corner indicates that (11%) of vaginal births in this dataset were among birthers who reported ‘yes’ to having a ‘Midwife’ as their “Birth Provider”. (43%) of vaginal births were among birthers who were less than 30 years old and did not report having a ‘Midwife’ as their “Birth Provider”. Finally, the far-right terminal node explains that (9%) of cesarean sections were among birthers who had greater than or equal to 4 previous births, were greater than age 37 and who did not have a ‘Midwife’ as their “Birth provider”. The decision tree indicates that almost (10%) of cesarean sections reported in this dataset were among a specific population of birthers who were at an advanced birthing age (over 37), experienced numerous births (greater than 4) and who did not have a midwife as their “Birth Provider”. This indicates that this population of birthers show an increased risk for a medicalized cesarean birth. This finding could encourage childbirth health professionals to implement programs aimed to decrease the risk for cesarean births among this population of birthers.

1. Obtain a ConfusionMatrix on the test dataset for the decision tree models with the selected key varaibles and cp values of (0.005) and (0.001).

test.pred.005<-predict(d.tree.vi.005, test.df, type="class")  
confusionMatrix(test.pred.005, test.df$Method.delivery)

## Confusion Matrix and Statistics  
## Reference  
## Prediction vaginal c.section  
## vaginal 1700 734  
## c.section 130 136   
## Accuracy : 0.68   
## 95% CI : (0.662, 0.6976)  
## No Information Rate : 0.6778   
## P-Value [Acc > NIR] : 0.4113   
## Kappa : 0.1043   
## Mcnemar's Test P-Value : <2e-16   
## Sensitivity : 0.9290   
## Specificity : 0.1563   
## Pos Pred Value : 0.6984   
## Neg Pred Value : 0.5113   
## Prevalence : 0.6778   
## Detection Rate : 0.6296   
## Detection Prevalence : 0.9015   
## Balanced Accuracy : 0.5426   
## 'Positive' Class : vaginal

test.pred.001<-predict(d.tree.vi.001, test.df, type="class")  
confusionMatrix(test.pred.001, test.df$Method.delivery)

## Confusion Matrix and Statistics

Reference  
## Prediction vaginal c.section  
## vaginal 1633 696  
## c.section 197 174   
## Accuracy : 0.6693   
## 95% CI : (0.6512, 0.687)  
## No Information Rate : 0.6778   
## P-Value [Acc > NIR] : 0.8335   
## Kappa : 0.1087   
## Mcnemar's Test P-Value : <2e-16   
## Sensitivity : 0.8923   
## Specificity : 0.2000   
## Pos Pred Value : 0.7012   
## Neg Pred Value : 0.4690   
## Prevalence : 0.6778   
## Detection Rate : 0.6048   
## Detection Prevalence : 0.8626   
## Balanced Accuracy : 0.5462   
## 'Positive' Class : vaginal

* + Both models preform similarly in accuracy and sensitivity. The model with the cp value of (.001) produced a higher specificity however the model with the cp value of (.005) is the accepted model because it is less likely to run into the problem of overfitting.

|  | **Accuracy** | **Sensitivity** | **Specificity** |
| --- | --- | --- | --- |
| **Models with key variables** |  |  |  |
| cp(.001) | 67% | .89 | .20 |
| cp(.005) | 68% | .93 | .16 |

Part B:

* 1. Build a tuning procedure for the hyperparameter of K.

set.seed(16)  
births.sample<-sample\_n(final.df, 1616)

* + split data into training and testing sets (70%-30% split)

set.seed(16)  
subset.split<-initial\_split(births.sample, prop=.7, strata=Method.delivery)  
subset.train<-training(subset.split)  
subset.test<-testing(subset.split)

* + specify a re-sampling procedure (10-fold cross-validation)

k\_fold<-vfold\_cv(subset.train)

model.rec<-recipe(Method.delivery~., subset.train) %>% step\_range(all\_numeric()) %>% step\_dummy(all\_nominal(),-all\_outcomes())

* + create a recipe and preprocessing steps

model.rec %>% prep() %>% juice() %>% summary()

* + specify the metrics that will eventually be used

m.metrics<-yardstick::metric\_set(accuracy, sens, roc\_auc, spec)

* + save the predicted value using model control grid

m\_control<-control\_grid(save\_pred=TRUE)

* + set the mode and engine for the tuning process

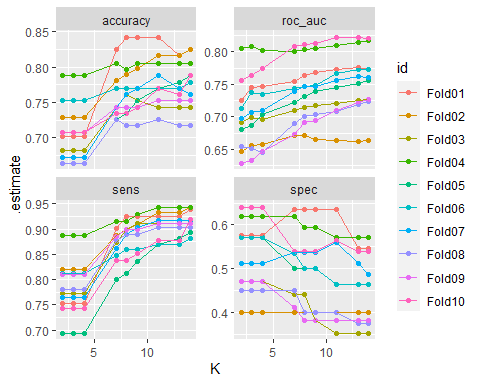
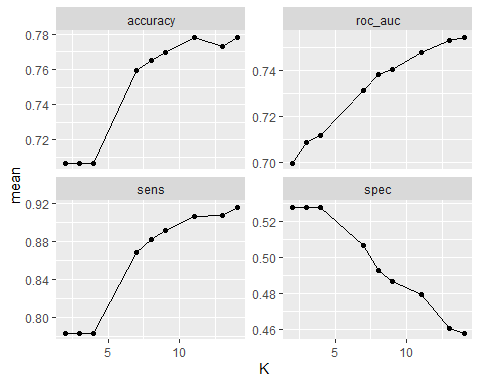
set.seed(16)  
knn.model<-nearest\_neighbor(neighbors=tune("K")) %>%   
 set\_mode("classification") %>%   
 set\_engine("kknn")

1. Tune K in the KNN model and obtain plot for all mean metrics. Which metric show a clear pattern when K increases? What is the best K if you are to base your decision on the mean “accuracy”, or “roc\_auc” alone?

knn\_grid<-grid\_regular(parameters(knn.model), levels=5)

set.seed(16)  
knn.tune<-tune\_grid(  
 knn.model,  
 model.rec,  
 resamples=k\_fold,  
 control=m\_control,  
 metrics=m.metrics  
)set.seed(16)  
knn.tune %>%   
 collect\_metrics() %>%   
 ggplot(aes(x=K, y=mean))+  
 geom\_point()+  
 geom\_line()+  
 facet\_wrap(~.metric, scales="free\_y")

set.seed(16)  
knn.tune %>% select(id, .metrics) %>%   
 unnest(.metrics) %>%   
 ggplot(aes(x=K, y=.estimate, color=id))+  
 geom\_point()+  
 geom\_line()+  
 facet\_wrap(~.metric, scales="free\_y")



* + “accuracy”, “roc\_auc”, and “sensitivity” all show a clear pattern when K increases these metrics also increase. If the best K was based solely on the accuracy and roc\_auc alone I would pick the highest point which is k=14 or k=15. Specificity shows a mild decreasing pattern.

1. “Collect” the predicted values that have resulted from the tuning process and obtain a confusion matrix. What is the sample size according to the confusion matriz? Explain why it is much larger than the size of the training set.

pred.data<-knn.tune %>%   
 collect\_predictions %>%   
 mutate(pred=if\_else(.pred\_vaginal>=0.5, "vaginal", "c.section"))  
pred.data<-pred.data%>%mutate(pred=recode\_factor(pred, "1"="vaginal", "2"="c.section"))

pred.data %>% conf\_mat(Method.delivery,pred)

## Truth  
## Prediction vaginal c.section  
## vaginal 6083 1545  
## c.section 1009 1551

* + The process used the training set sample size ten times that is why the sample size is larger with 10188 observations.

1. What are the mean accuracy and mean sensitivity that have resulted from the tuning process?

pred.data %>% accuracy(Method.delivery, pred)

| .metric | .estimator | .estimate |
| --- | --- | --- |
| accuracy | binary | 0.7493129 |

pred.data %>% sens(Method.delivery, pred)

| .metric | .estimator | .estimate |
| --- | --- | --- |
| sens | binary | 0.857727 |

* + The “sensitivity” indicates that I will be correct 86% of the time when predicting positive cases in this dataset (vaginal births).

1. Decide on the best K based on the mean “accuracy”, “sensitivity”, “specificity” and “roc\_auc”.

knn.tune %>% collect\_metrics() %>% filter(.metric=="accuracy") %>% top\_n(mean, n=1)

knn.tune %>% collect\_metrics() %>% filter(.metric=="roc\_auc") %>% top\_n(mean, n=1)

knn.tune %>% collect\_metrics() %>% filter(.metric=="sens") %>% top\_n(mean, n=1)

knn.tune %>% collect\_metrics() %>% filter(.metric=="spec") %>% top\_n(mean, n=1)

| K | .metric | .estimator | mean | n | std\_err | .config |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | accuracy | binary | 0.7781866 | 10 | 0.01113655 | Preprocessor1\_Model9 |

| K | .metric | .estimator | mean | n | std\_err | .config |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | roc\_auc | binary | 0.7542363 | 10 | 0.01485241 | Preprocessor1\_Model9 |

| K | .metric | .estimator | mean | n | std\_err | .config |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | sens | binary | 0.9164704 | 10 | 0.00652919 | Preprocessor1\_Model9 |

| K | .metric | .estimator | mean | n | std\_err | .config |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | spec | binary | 0.528206 | 10 | 0.02503929 | Preprocessor1\_Model1 |
| 3 | spec | binary | 0.528206 | 10 | 0.02503929 | Preprocessor1\_Model2 |
| 4 | spec | binary | 0.528206 | 10 | 0.02503929 | Preprocessor1\_Model3 |

* + The “accuracy”, “roc\_auc”, and “sens” all indicate that k=14 is the best decision.

1. Use the best k and estimate the best model. What are the accuracy and roc\_auc of this model? Compared with the decision tree models, does the KNN model yield better or worse metrics?

model.best<-nearest\_neighbor(neighbors=14) %>%   
 set\_mode("classification") %>%   
 set\_engine("kknn")  
  
best.model<-workflow() %>%   
 add\_model(model.best) %>%   
 add\_recipe(model.rec)  
last\_fit(best.model, subset.split) %>% collect\_metrics()

| .metric | .estimator | .estimate | .config |
| --- | --- | --- | --- |
| accuracy | binary | 0.8161157 | Preprocessor1\_Model1 |
| roc\_auc | binary | 0.7971396 | Preprocessor1\_Model1 |

* + The decision tree model in Part A produced an optimal accuracy of .68. I was able to improve the accuracy by using a knn model. Compared to the decision tree model the knn model improved the accuracy by almost 14%. Therefore I would accept the knn model over the decision tree model.

Cleaning Data:

library(naniar)  
library(dplyr)  
library(rpart)  
library(rpart.plot)  
library(caTools)  
library(caret)  
library(e1071)

setwd("C:/Data Mining/Data Sets")

births2018<-read.csv("births2018.csv", header=TRUE, sep=",", fileEncoding="UTF-8-BOM")

df<-df %>% replace\_with\_na(replace = list(Method.delivery=9))  
df<-df %>% replace\_with\_na(replace = list(Method.delivery="NA"))  
df<-df%>%mutate(Method.delivery=recode\_factor(Method.delivery, "1"="vaginal", "2"="vaginal","3"="c.section", "4"="c.section","5"="vaginal","6"="c.section"))

df<-df %>% replace\_with\_na(replace = list(Birth.provider=9))  
df<-df %>% replace\_with\_na(replace = list(Birth.provider=5))  
df<-df %>% replace\_with\_na(replace = list(Birth.provider="NA"))  
df<-df%>%mutate(Birth.provider=recode\_factor(Birth.provider, "1"="Medicalized", "2"="Medicalized", "3"="Midwife","4"="Midwife"))

df<-df %>% replace\_with\_na(replace = list(Birth.location=7))  
df<-df %>% replace\_with\_na(replace = list(Birth.location=9))  
df$Birth.location<-as.factor(df$Birth.location)  
df<-df %>% replace\_with\_na(replace = list(Birth.location="NA"))  
df<-df%>%mutate(Birth.location=recode\_factor(Birth.location, "1"="hospital", "2"="not in hospital", "3"="not in hospital","4"="not in hospital", "5"="not in hospital","6"="hospital"))

df$LI<-as.factor(df$LI)  
df<-df %>% replace\_with\_na(replace = list(LI="U"))  
df<-df %>% replace\_with\_na(replace = list(LI="NA"))  
df<-df%>%mutate(LI=recode\_factor(LI, "N"="no.LI", "Y"="yes.LI", "U"="no.LI"))

df<-df %>% replace\_with\_na(replace = list(risk=9))  
df<-df %>% replace\_with\_na(replace = list(risk="NA"))  
df<-df%>%mutate(risk=recode\_factor(risk, "1"="no.risk", "0"="yes.risk"))

df<-df %>% replace\_with\_na(replace = list(MRACE="NA"))  
df<-df%>%mutate(MRACE=recode\_factor(MRACE, "1"="white", "2"="non.white", "3"="non.white","4"="non.white", "5"="non.white","6"="non.white","7"="non.white","8"="non.white", "9"="non.white", "10"="non.white", "11"="non.white","12"="non.white", "13"="non.white", "14"="non.white", "15"="non.white"))

df<-df %>% replace\_with\_na(replace = list(num.prev.births="99"))  
df<-df %>% replace\_with\_na(replace = list(PREVIS="99"))

df$PREVIS<-as.numeric(df$PREVIS)  
df$Age<-as.numeric(df$Age)  
df$num.prev.births<-as.numeric(df$num.prev.births)

final.df<-select(df, Method.delivery, MRACE, Birth.provider, Birth.location, Prev.csection, Born.US, MEDUC, num.prev.births, Age, PREVIS, SEX, FRACE,MHISP, Pay, risk)

final.df<-na.omit(final.df)