# DustSafe North America Household Pb Prediction

Matthew Dietrich

8/19/2021

## Contents

```
#'Packages needed for logistic regression and filtering data
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.3.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

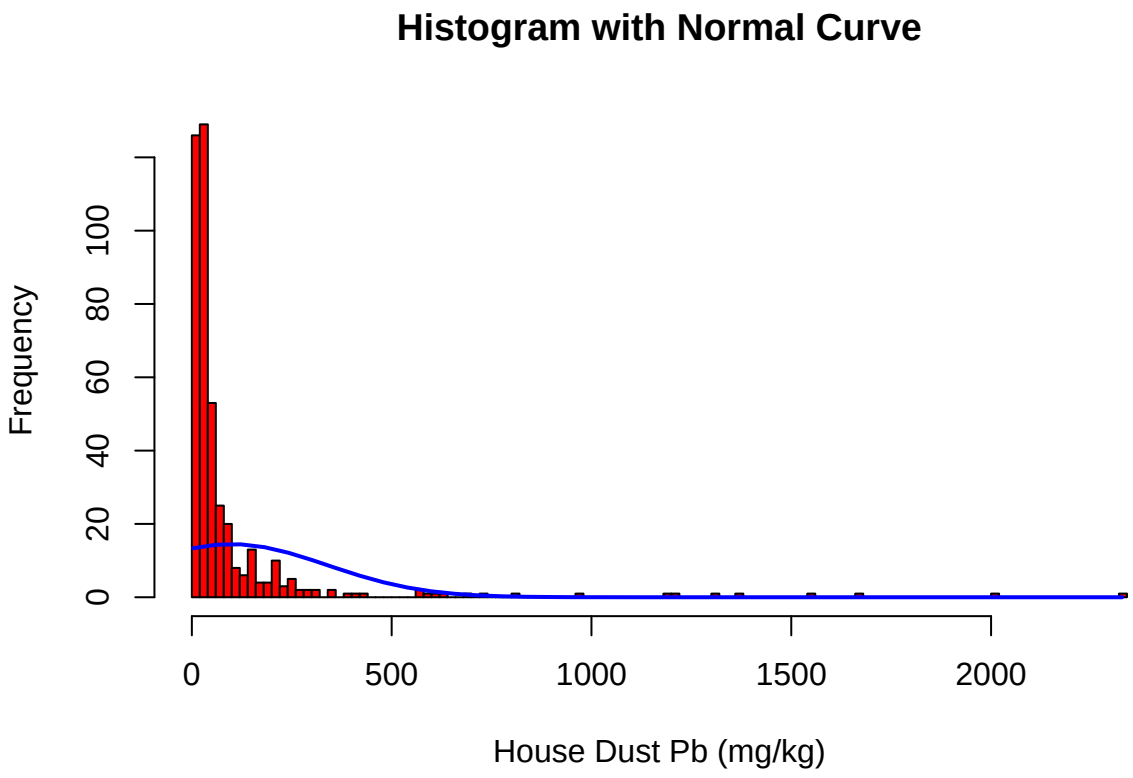
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(readxl)
```

```r
#' Read in Dec. 2020-July 2021 DustSafe Data
Indy1 <- read_excel("MME_NA_DustSafe_2021.xlsx",
    sheet = "Plotting Data")
```

```r
#' Add a Normal Curve to histogram (Thanks to Peter Dalgaard)
#' Household dust Pb
x <- na.omit(Indy1$Pb)
h<-hist(x, breaks=100, col="red", xlab="House Dust Pb (mg/kg)",
        main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```




```
#'Correlation between independent variables to test if violate assumptions of multicollinearity
p <- cor.test(Indy1$InteriorPeeling,Indy1$Housing, method=c("pearson"))
(p)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Indy1$InteriorPeeling and Indy1$Housing
## t = 6.4973, df = 355, p-value = 2.767e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2299905 0.4157322
## sample estimates:
##       cor
## 0.3260039
```

```
#'Filter for missing data of potentially useful variables in initial model
IndyPredict <- na.omit(Indy1[,c(4, 24:25, 30, 28)])

#'Change to factor data
#'"Low" (< 80 mg/kg Pb) and "High" (> 80 mg/kg Pb)
IndyPredict$Pb_level_cat <- as.factor(IndyPredict$Pb_level_cat)
```

```r
#'Split the data into training and test set with 80 mg/kg for high dust Pb threshold
set.seed(123)
training.samples <- IndyPredict$Pb_level_cat %>%
  createDataPartition(p=0.7, list=FALSE)
train.data  <- IndyPredict[training.samples, ]
test.data <- IndyPredict[-training.samples, ]
```

```r
#'Multiple logistic regression
glm.fit <- glm(Pb_level_cat ~ Housing + InteriorPeeling, data = train.data, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Pb_level_cat ~ Housing + InteriorPeeling, family = binomial,
##     data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1225   0.4714   0.4714   0.5818   2.0489
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.1413     0.2591   8.264  < 2e-16 ***
## Housing          -0.4506     0.1216  -3.704 0.000212 ***
## InteriorPeeling  -1.1535     0.4334  -2.662 0.007774 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 250.88  on 239  degrees of freedom
## Residual deviance: 216.96  on 237  degrees of freedom
## AIC: 222.96
##
## Number of Fisher Scoring iterations: 4
```

```r
#'Model probability of success for binomial factor variable
glm.probs <- glm.fit %>% predict(test.data,type = "response")
head(glm.probs)
```

```
##         1         2         3         4         5         6
## 0.6311577 0.7286526 0.6877262 0.6877262 0.6877262 0.6877262
```

```r
#'Checking the dummy coding
contrasts(test.data$Pb_level_cat) #So a probability of 0.92 means 92% chance of low dust Pb (<80 mg/kg)
```

```
##      Low
## High   0
## Low    1
```

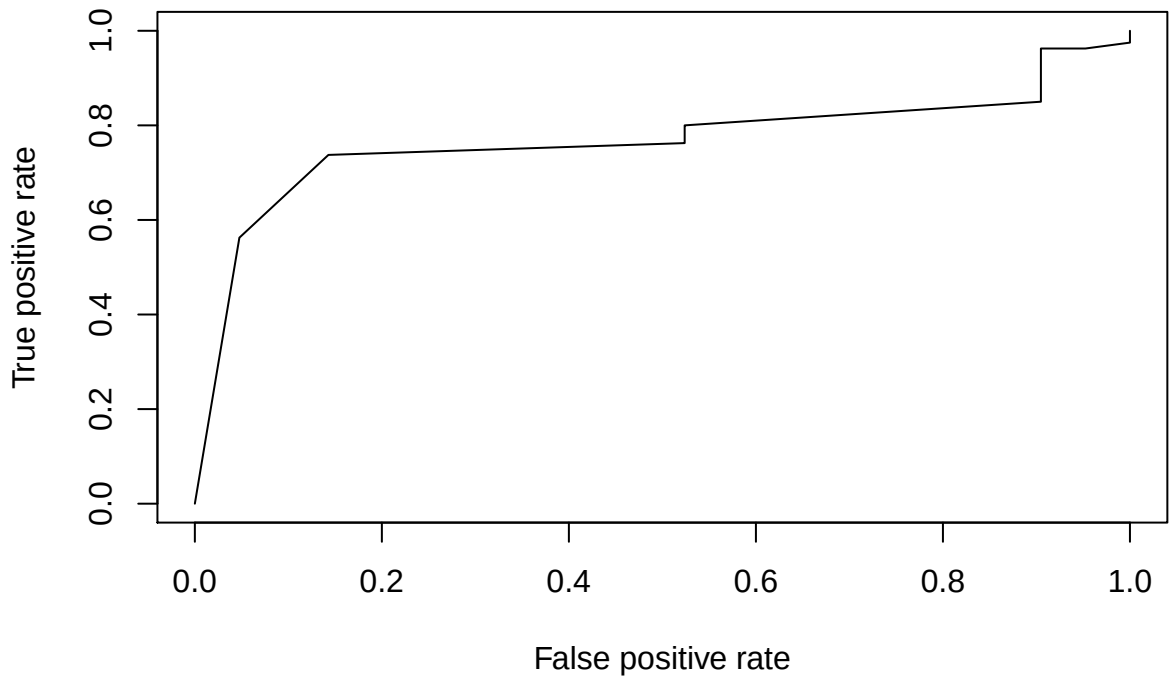
```r
#'ROC curve to help set predictive thresholds
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.0.3
```

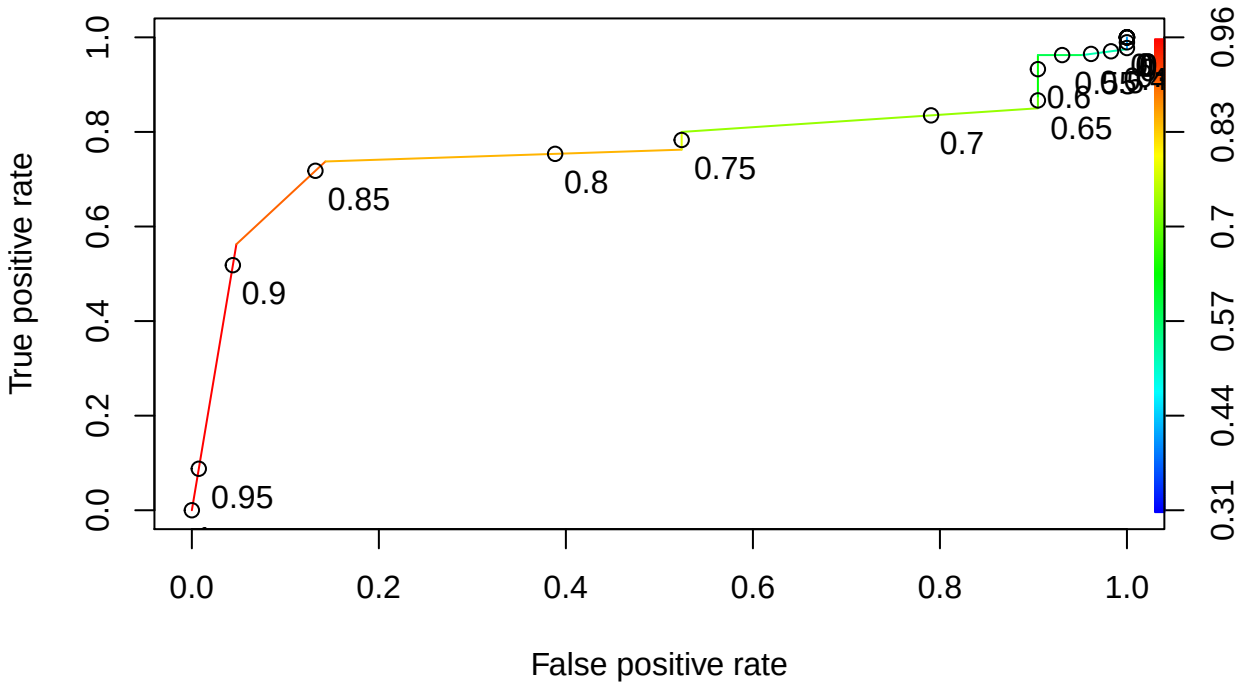
```r
ROCRpred <- prediction(glm.probs, test.data$Pb_level_cat)

#' Performance function
ROCRperf <- performance(ROCRpred, "tpr", "fpr")

#' Plot ROC curve
plot(ROCRperf)
```




```r
#' Add colors
plot(ROCRperf, colorize=TRUE)
#' Add threshold labels
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.05), text.adj=c(-0.2,1.7))
```




```
#'Classify if high or low dust Pb based on probability of predictive power from model
glm.pred <- ifelse(glm.probs > 0.8, "Low", "High")
#'Confusion matrix
table(glm.pred, test.data$Pb_level_cat)
```

```
##
## glm.pred High Low
##     High   18  21
##     Low     3  59
```

```
#'Mean proportion of correct predictions
mean(glm.pred == test.data$Pb_level_cat)
```

```
## [1] 0.7623762
```