

DustSafe North America Household Pb Prediction

Matthew Dietrich

8/17/2021

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()
```

```
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
```

```
library(readxl)
```

```
#' Read in Dec. 2020-July 2021 DustSafe Data
```

```
Indy1 <- read_excel("MME_NA_DustSafe_2021.xlsx",  
  sheet = "Plotting Data")
```

```
#' 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")
```

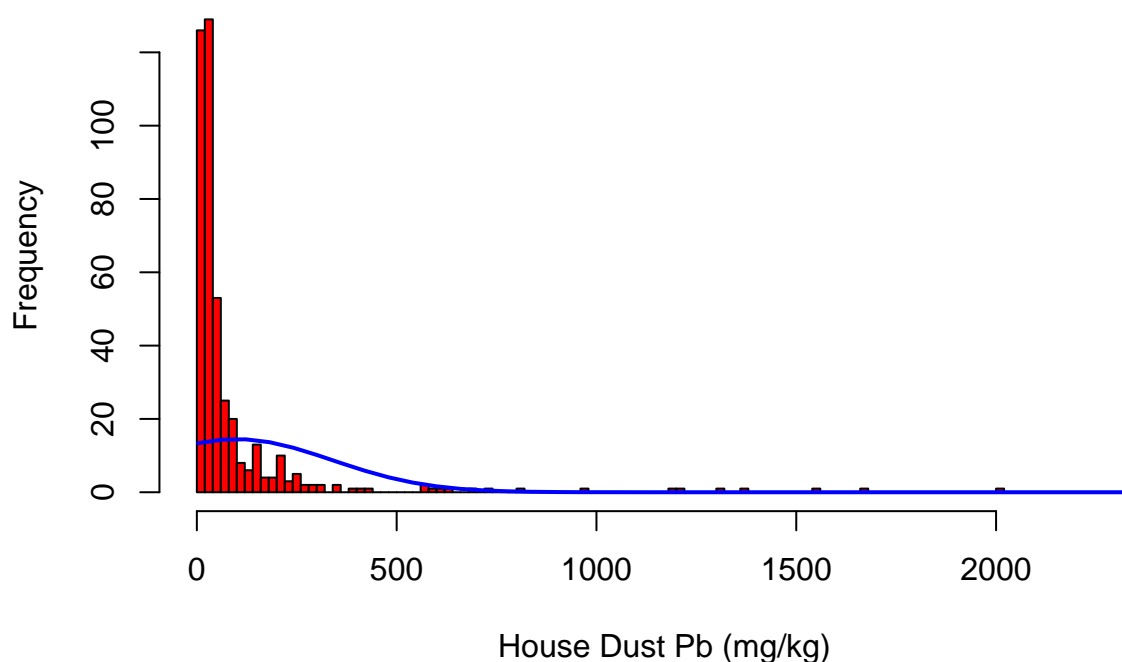
```
xfit<-seq(min(x),max(x),length=40)
```

```
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)
```

Histogram with Normal Curve



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

```
##
## Pearson's product-moment correlation
##
## data: Indy1$RecentRenovation and Indy1$Housing
## t = 0.98353, df = 357, p-value = 0.326
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.05180089 0.15465739
## sample estimates:
## cor
## 0.05198369
```

```
#'Filter for missing data of potentially useful variables in initial model
IndyPredict <- na.omit(Indy1[,c(4, 22,24:25, 30, 21, 23, 26, 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)
```

```
#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, ]
```

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

```
##
## Call:
## glm(formula = Pb_level_cat ~ Housing + RecentRenovation, family = binomial,
##      data = train.data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3471   0.3627   0.3627   0.5726   1.5617
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.6886     0.3480   7.725 1.12e-14 ***
## Housing        -0.9632     0.1547  -6.228 4.73e-10 ***
## RecentRenovation -0.6682     0.3680  -1.816  0.0694 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 233.35  on 219  degrees of freedom
## Residual deviance: 184.27  on 217  degrees of freedom
## AIC: 190.27
##
## Number of Fisher Scoring iterations: 5
```

```
#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
## 0.8488174 0.8829174 0.2954001 0.2954001 0.4498990 0.4498990
```

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

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

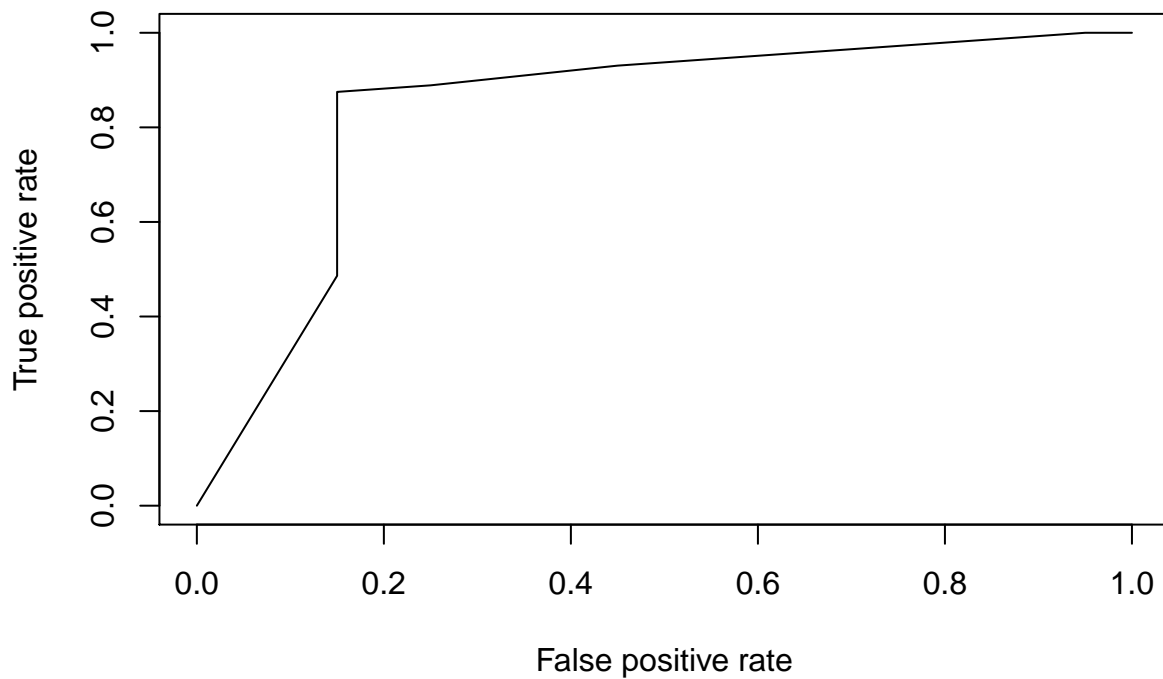
```
#'ROC curve to help set predictive thresholds  
library(ROCR)
```

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

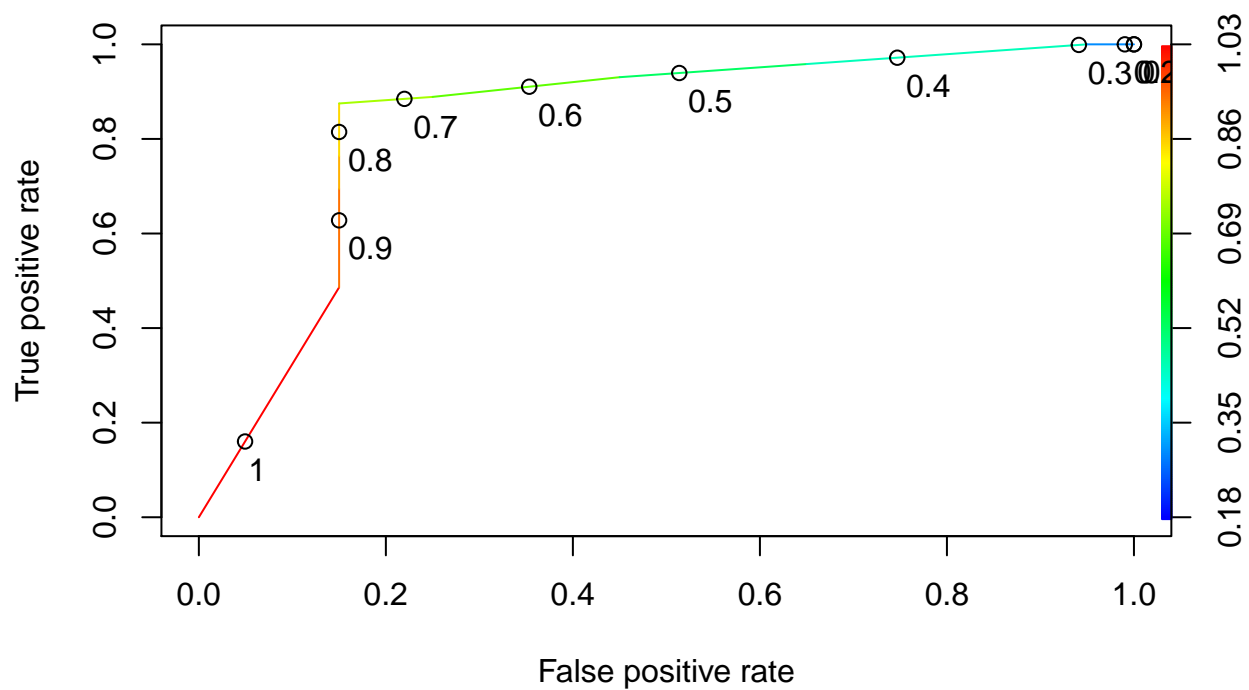
```
ROCRpred <- prediction(glm.probs, test.data$Pb_level_cat)
```

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

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



```
#' Add colors  
plot(ROCRperf, colorize=TRUE)  
#' Add threshold labels  
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), 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 > 0.7, "Low", "High")
#'Confusion matrix
table(glm.pred, test.data$Pb_level_cat)
```

```
##
## glm.pred High Low
##   High   17   9
##   Low    3  63
```

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

```
## [1] 0.8695652
```