Predicting readmission probability for diabetes inpatients

STAT 471/571/701, Fall 2017

Trevor Wexner

April 6, 2017

I: Executive Summary

Since the Centers for Medicare and Medicaid Services will not reimburse hospitals for cases that have an associated readmission within 30 days of discharge, it is in the best interest of hosptals to minimize the number of readmissions that occur within this timeframe. So, it was the focus of this project to determine how to minimize the number of these readmissions in an actionable way - before patients are discharged. For the remainder of this report, "readmission" refers to readmission within 30 days of discharge. Specifically, our study focused on how we could effectively lower the number of readmissions for diabetes incidents, and, in turn lower hospital costs. To answer this question, we studied analyzed data containing information for over 100,000 seperate diabetes hospital admissions from 130 different hospitals across the US. These encounters occured between 1999 and 2008, and our dataset contains the corresponding patient's demographic and medical information for each one. (Dataset courtesy of the Center for Clinical and Translational Research at Virginia Commonwealth University). Overall, 12.73% of encounters resulted in readmission.

In order to effectively reduce the number of readmissions due to diabetes, we used the dataset to build a model that predicts, given a set of variables, whether or not someone will be readmitted within 30 days. Four models were considered, with a total of 123 predictor variables were considered for input. Only the most effective and practically applicable one was chosen: a 14-variable model built using logistic regression, where all variables were statistically significant. The classifier was built on the assumption that it costs twice as much to incorrectly say that someone will be readmitted as it does to miss someone who will be readmitted. Thus, people with whose probability of readmission was predicted to be greater than 1/3 were deemed to be readmitted. We then tested our model out of sample. Of those people predicted to be readmitted, 40% were indeed readmitted. Our model had an overall misclassification rate of 10% and a weighted misclassification rate of 21% (weighted by the 2:1 ratio of costs above). While this model is not at all perfect, using it would certainly help improve identifying the people who will be readmitted before they are actually discharged. Thus, we can actionably reduce the number of readmissions by giving more care and attention to those deemed "higher risk" by the model. That is, it may be a good idea for people identified as "going to be readmitted" by our model, along with people of similar risk features to have more frequent mandatory follow up sessions for the following 30 day period and possibly have them stay in the hospital for slightly longer. These people that the model identified as more likely to be readmitted are those whose encounters are more severe, who have been to the ER within the past year, who are older, and who have other severe diagnoses chronic - particularly renal failure, unspecified congestive heart failure, occulsion of cerebral arteries, or diabetes with neurological manifestations. By following up with these people more closely, hospitals can certainly reduce the number of readmissions. While this model certainly helps to identify some people who are likely to be redmitted, it does not identify all. In fact, only 6% of those who were readmitted were accurately classified by the model. Thus, while it is advisable to treat those whom the model classifies as at risk with care, it is also important that this model be used in conjunction with the expertise of a medical professional when determining to whom the hospital should devote more resources. Moreover, perhaps more important than the raw classification produced by the model is the insight into those factors which are important in predicting readmission - namely the severity of the current incident, additional severe conditions, and overall health and age.

II: Methods and Results

i) Data Summary

The dataset in question contains information about hospital admissions due to diabetes, and was obtained from 130 different hostpitals across the US from the years 1999 - 2008. The original dataset contained 101766 admissions for 71518 different patients. However, 3622 of these entrees had missing values, so they were removed from the analysis. The resulting dataset used for study contained information for 98144 hospital admissions. For each admission, we know whether or not the patient was readmitted within 30 days along with his/her demographic information, medical history, and test results at the time of admission. Of the 29 varibles considered, most variables were categorical, with only some previous medical information being numerical. (E.g. number of medications taken and time in hospital). Of the hospital visits considered, 12.73% resulted in readmission within 30 days. Below is a complete list of all input variables along with their associated types. Detailed descriptions for each of these variables can be found in Appendix (1).

```
##                     Variable             type     
## race                "race"               "factor" 
## gender              "gender"             "factor" 
## time_in_hospital    "time_in_hospital"   "integer"
## num_lab_procedures  "num_lab_procedures" "integer"
## num_procedures      "num_procedures"     "integer"
## num_medications     "num_medications"    "integer"
## number_outpatient   "number_outpatient"  "integer"
## number_emergency    "number_emergency"   "integer"
## number_inpatient    "number_inpatient"   "integer"
## number_diagnoses    "number_diagnoses"   "integer"
## max_glu_serum       "max_glu_serum"      "factor" 
## A1Cresult           "A1Cresult"          "factor" 
## metformin           "metformin"          "factor" 
## glimepiride         "glimepiride"        "factor" 
## glipizide           "glipizide"          "factor" 
## glyburide           "glyburide"          "factor" 
## pioglitazone        "pioglitazone"       "factor" 
## rosiglitazone       "rosiglitazone"      "factor" 
## insulin             "insulin"            "factor" 
## change              "change"             "factor" 
## diabetesMed         "diabetesMed"        "factor" 
## disch_disp_modified "disch_disp_modified" "factor"
## adm_src_mod         "adm_src_mod"        "factor" 
## adm_typ_mod         "adm_typ_mod"        "factor" 
## age_mod             "age_mod"            "factor" 
## diag1_mod           "diag1_mod"          "factor" 
## diag2_mod           "diag2_mod"          "factor" 
## diag3_mod           "diag3_mod"          "factor" 
## readmitted          "readmitted"         "factor" 
```

Additionally, the following gives sumaries of all numerical variables:

```
##  time_in_hospital num_lab_procedures num_procedures num_medications
##  Min.   : 1.000   Min.   :  1.00     Min.   :0.00   Min.   : 1.00  
##  1st Qu.: 2.000   1st Qu.: 31.00     1st Qu.:0.00   1st Qu.:11.00  
##  Median : 4.000   Median : 44.00     Median :1.00   Median :15.00  
##  Mean   : 4.421   Mean   : 43.15     Mean   :1.35   Mean   :16.12  
##  3rd Qu.: 6.000   3rd Qu.: 57.00     3rd Qu.:2.00   3rd Qu.:20.00  
##  Max.   :14.000   Max.   :132.00     Max.   :6.00   Max.   :81.00  
##  number_outpatient number_emergency  number_inpatient  number_diagnoses
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 3.000  
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 6.000  
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 8.000  
##  Mean   : 0.3762   Mean   : 0.2024   Mean   : 0.6469   Mean   : 7.511
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 9.000
##  Max.   :42.0000   Max.   :76.0000   Max.   :21.0000   Max.   :16.000
```

Prior to analysis, each factor level is converted to be a distinct variable, creating 123 explanatory variables and one response, “readmitted” (0/1).

ii) Analyses

Before building any models, we also needed to remove the column `diag3_modV45` due to missing values. We then randomly split the data into testing and training sets, where 3/4 of the data were used for training and 1/4 were used for testing. Once this pre-processing was done, we then used the training data to build our models. We fit four models to predict the likelihood of readmission. The models are as follows:

First, we built `Fit 1` by running a logistic regression on all 123 variables in the dataset, then using backwards elimination until we obtained a model where all predictors were significant at the .05 level. The result of this procedure was a 46 variable model.

Next, we built `Fit 2` by running a LASSO on the dataset, where the minimizing criteria used was deviance. We used 10-fold cross-validation to select the model whose tuning parameter was `lambda1se` (becuase it would yield less variables than `lambda_min`). The result was a 16 variable model.

Third, we built `Fit 3` by running a logistic regression on the variables selected by the above LASSO. One variable was insignificant, so we eliminated it and obtained a 15-variable model where all predictors were significant at the .05 level.

Lastly, we ran a random forest for classification using deviance minimization as the splitting criterion. Due to the size of the datast, we limited the procedure to 100 trees. The expected testing error plot shows all 100 trees are necessary to minimize the false negative rate (see Appendix 2). The variables used for prediction in each of the models can be found in Appendix (3).

For all four models, we build classifiers using the optimal cutoff to minimiize weighted misclassification error ($threshold = \frac{a_{01}}{a_{01}+a_{10}} = \frac{1}{3}$). We then compared these models to see how they performed out of sample, on our test set. All four classifiers have essentially the same weighted misclassification rate out of sample (21%). So, for the sake of being parsimonious, we will use the smallest model. This is the 14-variable model given by running a logistic regression on the variables selected by LASSO.

We then used other model selection criteria to validate our decision. Our models all still performed effectively the same among these other criteria. All fits had out-of-sample misclassification rates (without costs) of about 10% and testing AUC values of roughly .64. Consequently, their corresponding ROC curves were essentially overlapping (see Appendix (2)). These results for how the four models performed are summarized in the table below.

```
##                              Fit 1                  Fit 2  
## Weighted Error Rate          "0.213"                "0.212"
## Unweighted Error Rate        "0.112"                "0.109"
## AUC                          "0.641"                "0.639"
## Number of Variables Included "46"                   "16"   
## Model Used                   "Backwards Elimination" "LASSO"
##                              Fit 3                                   
## Weighted Error Rate          "0.211"                                 
## Unweighted Error Rate        "0.111"                                 
## AUC                          "0.64"                                  
## Number of Variables Included "15"                                    
## Model Used                   "Logistic Regression with LASSO Variables"
##                              Fit 4  
## Weighted Error Rate          "0.215"
## Unweighted Error Rate        "0.107"
## AUC                          "0.64" 
## Number of Variables Included "123 (All)"
## Model Used                   "Random Forest"
```

Given these equal performances, our choice of the smallest model is indeed valid. Our final classifier becomes: predict that someone will be readmitted if they have over a 1/3 probability of being readmitted, where this probability is estimated by fit 3. This is given by the following formula

$\hat{P}(readmitted) = \frac{e^k}{1-e^k}$ where k is given by the sum of the coefficients in the following table:

```
##                                                  coefficients
## (Intercept)                                            -3.015
## time_in_hospital                                        0.014
## num_medications                                         0.004
## number_emergency                                        0.054
## number_inpatient                                        0.266
## number_diagnoses                                        0.044
## diabetesMedYes                                          0.185
## disch_disp_modifiedDischarged.Transferred.to.SNF        0.392
## disch_disp_modifiedOther                                0.364
## diag1_mod434                                            0.408
```

While this model had a very low false positive rate (1.1%), it had a very high false negative rate (94%), and a relatively high false discovery rate (60%) out of sample. Given that our goal is identifying people who are likely to be readmitted, this model is not a great fit. Still, the model did adequately explain some of the readmittance out of sample, so it is better to use this model than no model at all.

iii) Conclusion

Our final model is to say someone will be readimitted if the predicted probability that they will be readmitted exceeds 1/3. Here the predicted probability is given by $\hat{P}(readmitted) = \frac{e^k}{1-e^k}$ where k is given by the sum of the coefficients in the table above. Two things can be taken from this classifier.

First, this classifier can be used to quantitatively predict who will and will not be readmitted.

Second, this classifier gives insight into who is likely to be readmitted. These insights are all statistically valid, since all variables in the model are significant at the .05 level. As expected, patients who visit the hospital for a more severe incident are more likely to be readmitted. That is, patients with longer time in the hospital, more medications prescribed, had diabetes medication prescribed, and more diagnoses made for a given encounter are more likely to be readmitted. This makes sense, since it is more difficult to make a full recovery from a more severe condition. Also, people with a higher frequency of recent emergency visits were more likely to be readmitted. This agrees with intuition, since people in poorer health are likely to need to go to the hospital more frequently. Additionally, people who were discharged to places other than their homes were more likely to be readmitted. This is likely because these people go to nursing homes or other health facilities, and are therefore either older or in worse health. So, we can use discharge location as a proxy for age. Similarly, people with chronic renal failure, unspecified congestive heart failure, occulsion of cerebral arteries, and diabetes with neurological manifestations were all more likely to be readimitted. This makes sense, considering that all of these are serious health issues associated with the kidneys, cardiovascular system, and neurological system. Conversely, people with less serious diagnoses of shortness of breath and diabetes mellitus were less likely to be readmitted.

Given these insights and our model's predictive ability, the following actionable plan can be implemented to reduce readmission rates:

(1) Before a given patient is going to be discharged, run his/her information through our model. If he/she is classified as going to be readmitted, increase the length of stay by a day or two, shorten the time between followup visits by a factor of at least 1/2, and/or schedule more frequent followup visits up to 30 days out. This will ensure that the patient is more stable both upon release and throughout the following 30 day period. Although our classifier only has an estimated 40% positive prediction rate out

of sample, our goal is to reduce readmittances. Thus, taking action on predicted positives would reduce this 40%.

(2) Regardless of if someone is predicted to be readimitted, institute the following:

- If someone is set to be discharged to a location other than a home, schedule more regular followup visits with them. This will help reduce the higher readmission risk of people living in these facilites.
- People with more severe incidents should be kept slightly longer and/or required to follow up more quickly and more regularly for the next month. This includes people with more severe diagnoses, relatively high numbers of diagnoses, who spend long amounts of time in the hospital, and who have been perscribed a high number of medications during a given visit. For example, people with more than 9 diagnoses on a given visit, who spend more than 6 days in the hospital, and who are perscribed more than 20 medications are all in the upper quartiles for these respective categories, and should be could be considered to be a "more severe incident". Thus, it may be advisable for such people to have more frequent followups or be kept for longer. Ultimately, the disgression of a medical professional should also be used to truly determine what characterizes a "severe" incident.

-Particularly, if the patient has chronic renal failure, unspecified congestive heart failure, occulsion of cerebral arteries, or diabetes with neurological manifestations, they should have required regular followup visits with their respective neurologist, nefrologist, or neurologist for the next 30 day period.

- If someone has had at least one emergency visit in the prior year, it would be advisable to schedule at at least one additional followup, given that this was an important predictor and that over 75% of people did not have emergency visits in the year prior.

While our model was by no means excellent (with a 21% weighted error rate), it still did predict a non-trivial amount of the readmissions out of sample. Thus, following the above guidelines should still reduce readmissions.

III: Appendices

(1) Complete descriptions of all variables considered for input into the model

a) Patient identifiers

a. `encounter_id`: unique identifier for each admission
b. `patient_nbr`: unique identifier for each patient

b) Patient Demographics:

`race`, `age`, `gender`, `weight` cover the basic demographic information associated with each patient. `Payer_code` is an additional variable that identifies which health insurance (Medicare /Medicaid / Commercial) the patient holds.

c) Admission and discharge details:

a. `admission_source_id` and `admission_type_id` identify who referred the patient to the hospital (e.g. physician vs. emergency dept.) and what type of admission this was (Emergency vs. Elective vs. Urgent).
b. `discharge_disposition_id` indicates where the patient was discharged to after treatment.

d) Patient Medical History:

a. `num_outpatient`: number of outpatient visits by the patient in the year prior to the current encounter
b. `num_inpatient`: number of inpatient visits by the patient in the year prior to the current encounter
c. `num_emergency`: number of emergency visits by the patient in the year prior to the current encounter

e) Patient admission details:

a. `medical_specialty`: the specialty of the physician admitting the patient

b. `diag_1`, `diag_2`, `diag_3`: ICD9 codes for the primary, secondary and tertiary diagnoses of the patient. ICD9 are the universal codes that all physicians use to record diagnoses. There are various easy to use tools to lookup what individual codes mean (Wikipedia is pretty decent on its own)
c. `time_in_hospital`: the patient's length of stay in the hospital (in days)
d. `number_diagnoses`: Total no. of diagnosis entered for the patient
e. `num_lab_procedures`: No. of lab procedures performed in the current encounter
f. `num_procedures`: No. of non-lab procedures performed in the current encounter
g. `num_medications`: No. of distinct medications prescribed in the current encounter

f) Clinical Results:

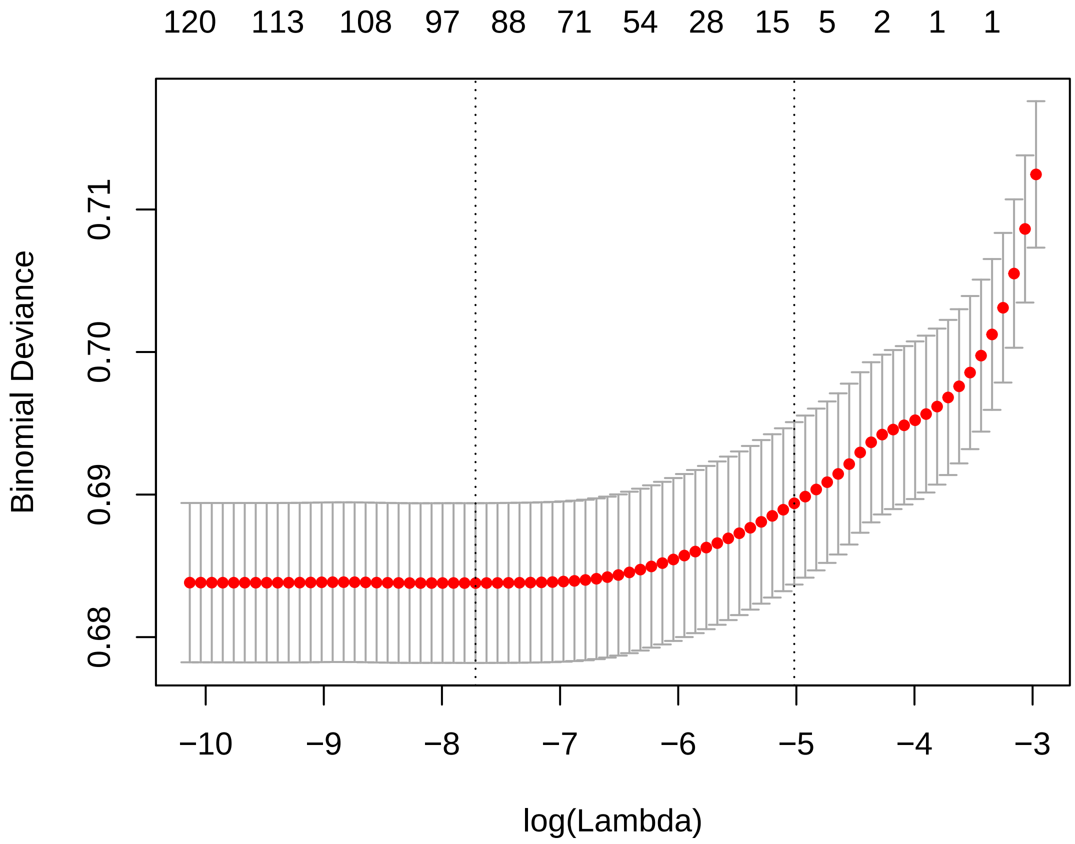
a. `max_glu_serum`: indicates results of the glucose serum test
b. `A1Cresult`: indicates results of the A1c test

g) Medication Details:

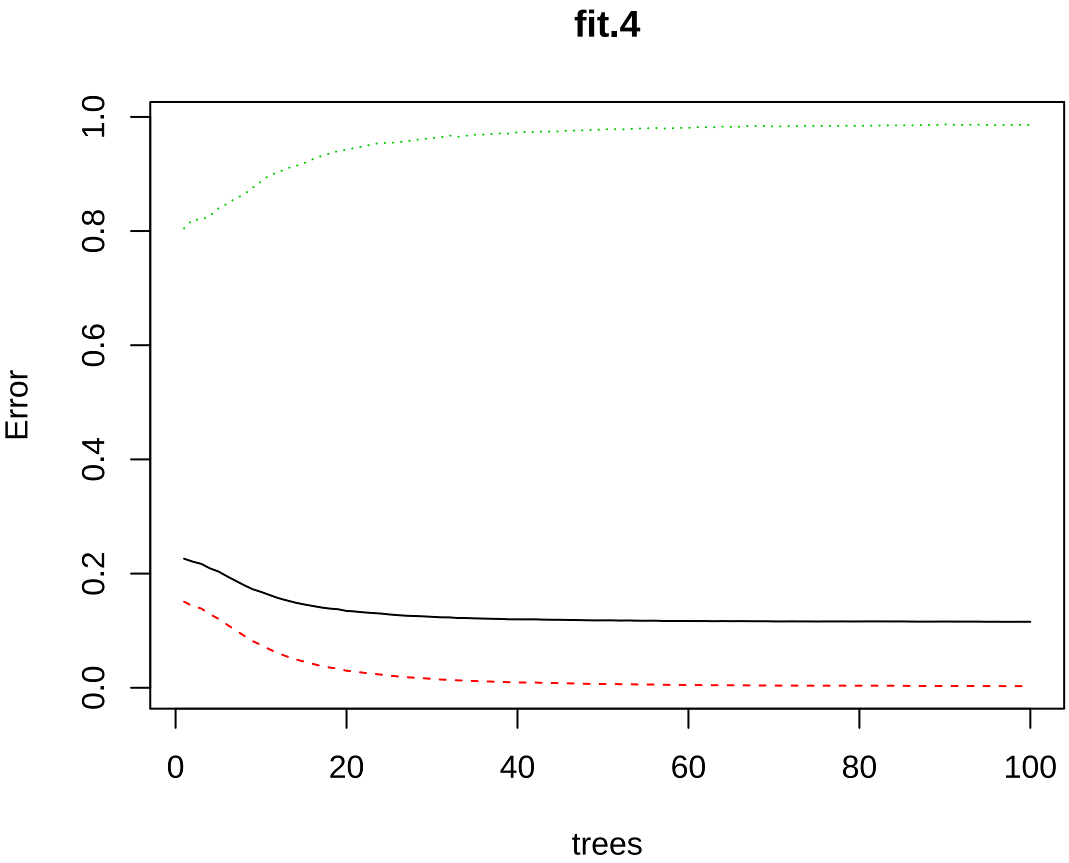
a. `diabetesMed`: indicates if any diabetes medication was prescribed
b. `change`: indicates if there was a change in diabetes medication
c. `24 medication variables`: indicate whether the dosage of the medicines was changed in any manner during the encounter

(2) Plots

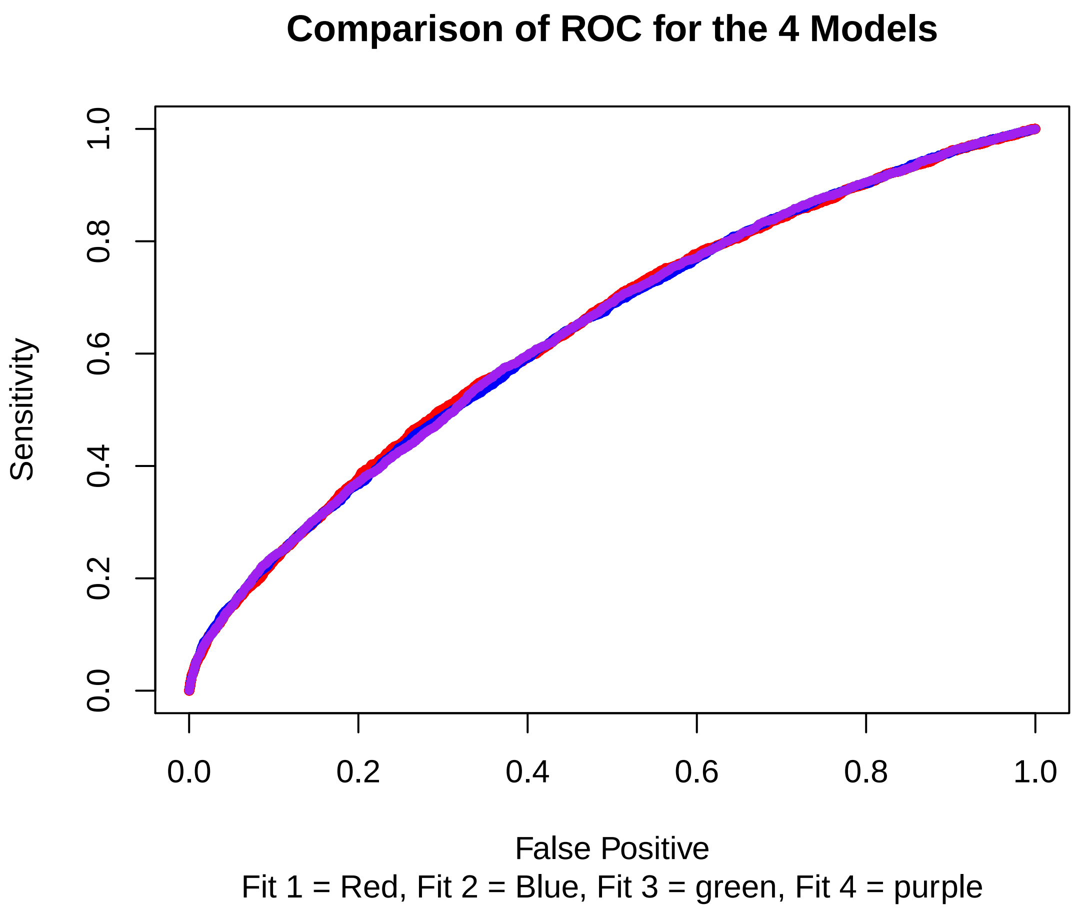
LASSO Cross-Validation Plot



Random Forest Misclassification Rates Plot



ROC Curves for the 4 Models



```
## numeric(0)
```

(3) Input Variables for of the Four Models

```
## [[1]]
##       Fit 1 Variables
##  [1,] "(Intercept)"
##  [2,] "raceAfricanAmerican"
##  [3,] "time_in_hospital"
##  [4,] "num_procedures"
##  [5,] "num_medications"
##  [6,] "number_emergency"
##  [7,] "number_inpatient"
##  [8,] "number_diagnoses"
##  [9,] "A1CresultNone"
## [10,] "metforminSteady"
## [11,] "metforminUp"
## [12,] "glipizideNo"
## [13,] "glipizideSteady"
## [14,] "rosiglitazoneNo"
## [15,] "diabetesMedYes"
## [16,] "disch_disp_modifiedDischarged.to.home.with.Home.Health.Service"
## [17,] "disch_disp_modifiedDischarged.Transferred.to.SNF"
## [18,] "disch_disp_modifiedOther"
## [19,] "adm_src_modOther"
## [20,] "adm_src_modTransfer.from.Home.Health"
## [21,] "age_mod20.59"
## [22,] "age_mod60.79"
## [23,] "age_mod80."
## [24,] "diag1_mod250.8"
## [25,] "diag1_mod38"
## [26,] "diag1_mod410"
## [27,] "diag1_mod414"
## [28,] "diag1_mod427"
## [29,] "diag1_mod435"
## [30,] "diag1_mod486"
## [31,] "diag1_mod491"
## [32,] "diag1_mod493"
## [33,] "diag1_mod518"
## [34,] "diag1_mod599"
## [35,] "diag1_mod682"
## [36,] "diag1_mod780"
## [37,] "diag1_mod786"
## [38,] "diag1_modOther"
## [39,] "diag2_mod250.01"
## [40,] "diag2_mod518"
## [41,] "diag2_modOther"
## [42,] "diag3_mod250.02"
## [43,] "diag3_mod250.6"
## [44,] "diag3_mod403"
## [45,] "diag3_mod428"
## [46,] "diag3_mod496"
## [47,] "diag3_mod585"
## [48,] "diag3_mod707"
## [49,] "diag3_modOther"
##
## [[2]]
##       Fit 2 Variables
##  [1,] "(Intercept)"
##  [2,] "time_in_hospital"
##  [3,] "num_medications"
##  [4,] "number_emergency"
##  [5,] "number_inpatient"
##  [6,] "number_diagnoses"
##  [7,] "insulinNo"
##  [8,] "diabetesMedYes"
##  [9,] "disch_disp_modifiedDischarged/Transferred to SNF"
## [10,] "disch_disp_modifiedOther"
## [11,] "diag1_mod434"
##
## [[3]]
##       Fit 3 Variables
##  [1,] "(Intercept)"
##  [2,] "time_in_hospital"
##  [3,] "num_medications"
##  [4,] "number_emergency"
##  [5,] "number_inpatient"
##  [6,] "number_diagnoses"
##  [7,] "diabetesMedYes"
##  [8,] "disch_disp_modifiedDischarged.Transferred.to.SNF"
##  [9,] "disch_disp_modifiedOther"
## [10,] "diag1_mod434"
##
## [[4]]
## [1] "Fit 4 Used All Variables in the Cleaned Dataset"
```

(4) References Interpetation of Diagnoses Codes - https://en.wikipedia.org/wiki/List_of_ICD-9_codes_240%E2%80%93279:_endocrine,_nutritional_and_metabolic_diseases,_and_immunity_disorders - https://en.wikipedia.org/wiki/List_of_ICD-9_codes_390%E2%80%93459:_diseases_of_the_circulatory_system - https://en.wikipedia.org/wiki/List_of_ICD-9_codes_580%E2%80%93629:_diseases_of_the_genitourinary_system - https://en.wikipedia.org/wiki/List_of_ICD-9_codes_780%E2%80%93799:_symptoms,_signs,_and_ill-defined_conditions

Background of Various Diseases - Diabetes Mellitus: https://en.wikipedia.org/wiki/Diabetes_mellitus - Hypertensive renal disease: https://en.wikipedia.org/wiki/Hypertensive_kidney_disease

(5) R Code

```
# read in data
data <- read.csv("/Users/TrevorWexner/Documents/Stat 471/Datasets/readmission.csv")
```

i) Data Summary

```
## data formatting
str(data)
# convert encounter_id & patient_nbr to factors
data$encounter_id <- as.factor(data$encounter_id)
data$patient_nbr <- as.factor(data$patient_nbr)
# change readmitted to Y/N (it doen't matter if readmission occurs more than
# 30 days out)
readmitted <- rep("0", nrow(data))
readmitted[which(data$readmitted == "<30")] <- "1"
data$readmitted <- as.factor(readmitted)
str(data)
# remove patient ID & visit ID from the dataset, they're not necessary for
# the analysis
data <- data[, !colnames(data) %in% c("encounter_id", "patient_nbr")]
str(data)
# problems with data: discard the following values gender ==
# 'Invalid/Unkown', race == '?',diag3_mod == '?'
data.clean <- subset(data, (gender != "Unknown/Invalid" & race != "?" & diag3_mod !=
    "?"))

## plots and tables
library(xtable)
table(data.clean$readmitted)
for (i in 1:ncol(data.clean)) {
    print(colnames(data.clean)[i])
    print(table(data.clean[, i]))
}
type <- c(unlist(lapply(data.clean[, colnames(data.clean)], class)))
variable.summary.table <- cbind(colnames(data.clean), type)
colnames(variable.summary.table)[1] <- "Variable"
```

```
# All variable names and types
variable.summary.table
```

Additionally, the following gives sumaries of all numerical variables:

```
# Summary Table
summary(data.clean[, type == "integer"])
```

ii) Analyses

```
library(glmnet)
library(bestglm)
library(car)
library(randomForest)
library(pROC)
### Run a logistic regression to estimate the probability that someone will be
### readmitted using the following methods:

## First, we syphen off 3/4 of the data for training and 1/4 for testing
set.seed(12)
training.indeces <- sample.int(3 * nrow(data.clean)/4)
data.train <- data.clean[training.indeces, ]
data.test <- data.clean[-training.indeces, ]
X <- data.frame(model.matrix(readmitted ~ ., data.train)[, -1])
# Remove NA values
X <- X[, !colnames(X) %in% c("raceOther", "genderUnknown.Invalid", "diag3_modV45")]
Y <- data.train[, "readmitted"]
Xy <- data.frame(cbind(X, Y))
colnames(Xy)[ncol(Xy)] <- "readmitted"

# (1) Do a logistic regression on all variables. Do backward elimination
# until all variables are significant Do a backwards elimination here using
# the design matrix #### TO DO

# backwards selection function for logistic regression
backwards.select <- function(y, data) {
    p <- ncol(data) - 1  # start with all variables
    data.new <- data
    fit.temp <- glm(as.formula(paste(y, "~.", sep = "")), data = data.new, family = binomial)
    largestp <- max(coef(summary(fit.temp))[2:p + 1, 4])  # largest p-values of all the predictors

    while (largestp > 0.05) {
        p <- p - 1
        # get var with largest p value & remove from model
        var.remove <- rownames(subset(coef(summary(fit.temp)), coef(summary(fit.temp))[,
            4] == largestp))
        data.new <- data.new[, !(names(data.new) == var.remove)]
        fit.temp <- glm(as.formula(paste(y, "~.", sep = "")), data = data.new,
            family = binomial)
        largestp <- max(coef(summary(fit.temp))[2:p + 1, 4])  # largest p-values of all the predictors
    }
    return(fit.temp)
}

# Get all variables significant at .05 level
fit1 <- backwards.select(y = "readmitted", data = Xy)
summary(fit1)  # We have a 46 variable model
```

```
# (2) Do cv.glmnet and do LASSO for glm --> take lambda1se, using deviance
X <- model.matrix(readmitted ~ ., data.train)[, -1]
fit.lasso.cv <- cv.glmnet(X, Y, alpha = 1, family = "binomial", nfolds = 10,
    type.measure = "deviance")
coef.1se <- coef(fit.lasso.cv, s = "lambda.1se")
coef.1se <- coef.1se[which(coef.1se != 0), ]
lasso.names <- rownames(as.matrix(coef.1se))
fit.2 <- fit.lasso.cv

# 16 variables included in this model

# (3) Run a regular logistic regression on LASSO output variables & remove
# any non-significant variables
Xy <- model.matrix(readmitted ~ . + 0, data = data.train)
Xy <- cbind(data.frame(Xy[, colnames(Xy) %in% rownames(as.matrix(coef.1se))]),
    data.train$readmitted)
colnames(Xy)[ncol(Xy)] <- "readmitted"
fit.3 <- glm(readmitted ~ ., data = Xy, family = "binomial")
summary(fit.3)  #insulinNo not significant, so remove
Xy <- Xy[, !colnames(Xy) == "insulinNo"]
fit.3 <- glm(readmitted ~ ., data = Xy, family = "binomial")
summary(fit.3)  # 15 variable model: all variables significant

# (4) Random Forest for classification, cutoff = 1/3. We limit ntree to be
# 100 due to the size of the dataset. Also, we build our classification
# trying to minimize weighted total cost, so we make
fit.4 <- randomForest(readmitted ~ ., data.train, ntree = 100, type = "Prob")
plot(fit.4)  # Examining the plot

# Variables included in each model
fit1.vars <- cbind(names(fit1$coefficients))
colnames(fit1.vars) <- "Fit 1 Variables"
fit2.vars <- cbind(rownames(as.matrix(coef.1se)))
colnames(fit2.vars) <- "Fit 2 Variables"
fit3.vars <- cbind(names(fit.3$coefficients))
colnames(fit3.vars) <- "Fit 3 Variables"

## COMPARING EACH MODEL AS A CLASSIFIER, USING WEIGHTED MISCLASSIFICATION
## ERROR Weighted cost information: a_01 = (1/2)a_10 <--> a_10 = 2a_01
## Optimal Cutoff = a_01/(a_01 + a_10) = 1/3

# (1) Predict the values of the testing data using each model
X.test <- data.frame(model.matrix(readmitted ~ ., data.test)[, -1])
X.matrix <- model.matrix(readmitted ~ ., data.test)[, -1]
# Remove NA values
X.test <- X.test[, !colnames(X.test) %in% c("raceOther", "genderUnknown.Invalid",
    "diag3_modV45")]
Y.test <- data.test[, "readmitted"]
Xy.test <- cbind(X.test, Y.test)
fit1.predicted <- predict(fit1, Xy.test, type = "response")
fit2.predicted <- predict(fit.2, X.matrix, s = "lambda.1se", type = "response")
fit3.predicted <- predict(fit.3, Xy.test, type = "response")
fit4.predicted <- predict(fit.4, data.test)
```

```
# (2) Make a classifier out of each model, using the optimal cutoff to
# minimize cost If P(Y = 1) > 1/3, we say Y-hat = 1
fit1.classes <- rep("0", nrow(data.test))
fit2.classes <- rep("0", nrow(data.test))
fit3.classes <- rep("0", nrow(data.test))
fit4.classes <- rep("0", nrow(data.test))
fit1.classes[fit1.predicted > 1/3] <- "1"
fit2.classes[fit2.predicted > 1/3] <- "1"
fit3.classes[fit3.predicted > 1/3] <- "1"
fit4.classes[fit4.predicted > 1/3] <- "1"


# (3) Compute the Weighted Testing Error for Each Classifer and Compare.
# Take the best one to be our final model
MCE.fit1 <- (sum(2 * (fit1.classes[data.test$readmitted == "1"] != "1")) + sum(fit1.classes[data.test$re
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit2 <- (sum(2 * (fit2.classes[data.test$readmitted == "1"] != "1")) + sum(fit2.classes[data.test$re
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit3 <- (sum(2 * (fit3.classes[data.test$readmitted == "1"] != "1")) + sum(fit3.classes[data.test$re
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit4 <- (sum(2 * (fit4.classes[data.test$readmitted == "1"] != "1")) + sum(fit4.classes[data.test$re
    "0"] != "0"))/length(data.test$readmitted)

# The 4 weighted misclassification rates are 0.2120150 0.2112814 0.2116074
# 0.2148679 (respectively) These are all very close, so for the sake of
# being parsimonious we will take the smallest model, fit.3
weighted.mce <- sapply(c(MCE.fit1, MCE.fit2, MCE.fit3, MCE.fit4), round, 3)
model.names <- c("Fit 1", "Fit 2", "Fit 3", "Fit 4")
model.used <- c("Backwards Elimination", "LASSO", "Logistic Regression with LASSO Variables",
    "Random Forest")

# (4) Compare the ROC curves of each of the 5 classifiers (for visualization
# purposes) Give these plots in Appendix 2
roc.1 <- roc(response = data.test$readmitted, predictor = fit1.predicted, plot = T)
roc.2 <- roc(response = data.test$readmitted, predictor = fit2.predicted, plot = T)
roc.3 <- roc(response = data.test$readmitted, predictor = fit3.predicted, plot = T)
roc.4 <- roc(response = data.test$readmitted, predictor = fit3.predicted, plot = T)

roc.plot <- plot(1 - roc.1$specificities, roc.1$sensitivities, col = "red",
    pch = 16, cex = 0.7, xlab = "False Positive", ylab = "Sensitivity") + points(1 -
    roc.2$specificities, roc.2$sensitivities, col = "blue", pch = 16, cex = 0.6) +
    points(1 - roc.3$specificities, roc.3$sensitivities, col = "green", pch = 16,
        cex = 0.6) + points(1 - roc.4$specificities, roc.4$sensitivities, col = "purple",
    pch = 16, cex = 0.6) + title(main = "Comparison of ROC for the 4 Models",
    sub = "Fit 1 = Red, Fit 2 = Blue, Fit 3 = green, Fit 4 = purple")
# Get AUC's
auc <- round(c(auc(roc.1), auc(roc.2), auc(roc.3), auc(roc.4)), 3)
```

```
# (5) As a side note, show overall misclassification rates for each
MCE.fit1 <- (sum((fit1.classes[data.test$readmitted == "1"] != "1")) + sum(fit1.classes[data.test$readmi
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit2 <- (sum((fit2.classes[data.test$readmitted == "1"] != "1")) + sum(fit2.classes[data.test$readm
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit3 <- (sum((fit3.classes[data.test$readmitted == "1"] != "1")) + sum(fit3.classes[data.test$readm
    "0"] != "0"))/length(data.test$readmitted)

MCE.fit4 <- (sum((fit4.classes[data.test$readmitted == "1"] != "1")) + sum(fit4.classes[data.test$readm
    "0"] != "0"))/length(data.test$readmitted)

mce <- sapply(c(MCE.fit1, MCE.fit2, MCE.fit3, MCE.fit4), round, 3)
weighted.mce.table <- rbind(weighted.mce, mce, auc, c("46", "16", "15", "123 (All)"),
    model.used)
colnames(weighted.mce.table) <- model.names
rownames(weighted.mce.table) <- c("Weighted Error Rate", "Unweighted Error Rate",
    "AUC", "Number of Variables Included", "Model Used")

# print weighted mce table
weighted.mce.table

# get coefficients of final model, rounded to 3 decimal places
rounded.coeffs <- round(fit.3$coefficients, 3)
rounded.coeffs

# print final model
k <- as.data.frame(rounded.coeffs)
colnames(k) <- "coefficients"
k

# evaluating performance of final model
cm <- table(fit3.classes, data.test$readmitted)  # confusion matrix:
false.positive <- cm[2, 1]/sum(data.test$readmitted == "0")  #false positive rate of .011
false.negative <- cm[1, 2]/sum(data.test$readmitted == "1")  # false negative rate of .94
true.positive <- cm[2, 2]/sum(data.test$readmitted == "1")  # .06
positive.prediction <- 159/(238 + 159)
```