**Predictive Modeling of MIMIC III Admissions and Intensive Care Unit Length of Stays**

**Using Binomial Regression Model and Random Forest Classification**

**By**

**Hero Ozagho**

**Abstract**

**Objectives**:

The primary goal of this project is to build a design model capable of predicting patients’ LOS in the ICU, having as main predictor variables those commonly used in clinical practice for the assessment of the state of the patient. Emphasis has been placed to the first value recorded for each variable selected and time evolution is not contemplated. The selected variable indicators must be obtained during the first 24 hours of stay in ICU.

**Design**:

The design of the model starts with raw data collection, data preparation, data cleaning and exploratory data analysis. Two design models were utilized separately for predictive performance from the MIMIC III database.

**Measurement**s:

The data set for icustay only contains 100 patients but there are 136 entries in icustay table, which means that some patients have multiple icustay. During the first icustay, 37 patients did not survive, and the number for the last ICU stay is 36. The last icustay table was utilized as the data frame to do exploratory data analysis and build a prediction model. There are also a total of 758,355 chart events and 76,074 lab events during the 136 entries in icustay table.

**Results**:

The Binomial regression model has an AUC of 84% in training data , which is surprisingly good, while the AUC in testing data was approximately 56%. The Random Forest model predictive performance was better than the Binomial model.

**Conclusion**:

There are 9 variables from 758,355 chart events and 76,076 lab events that are identified and thus successfully built a Random Forest to predict patients that survives during icustay with an approximately 36% Out-Of-Bag error rate.

**1. Introduction**

This final project is an analysis from MIMIC-III database for admissions, intensive care unit (ICU) stay among others to evaluate two different predictive models for healthcare management. The incidence in mortality of the variability of possible observables of the patients such as the variability of glucose over time during an ICU stay, presence or not of comorbidities1 together with other clinical and demographic variables of the patients, it is investigated the relationship of this observables as being the cause or part of the cause to the ICU Length of Stay (LOS) for each single admission episode.

The required datasets were extracted and formatted as necessary and uploaded into the R software environment. The required R programming software packages were installed into the R environment and their respective libraries loaded to obtain and transform data from the original database and convert it in a more tractable way to perform analysis on that data. The R programming language provided insightful and valuable information that facilitates a lot the tasks associated with the data analysis and the development and implementation of predictive models.

**1.1 Data Preparation: Load/Check data, Installation of R Packages**

MIMIC-III is a large, freely available database comprising deidentified health-related data associated with over 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 . The MIMIC-III data is a clinical data that was downloaded from <https://physionet.org/content/mimiciii-demo/1.4/> for data exploration and analysis.

The following required downloaded data from MIMIC-III clinical database were uploaded into R environment as necessary to perform the predictive models:

admissions <- read.csv(file = "ADMISSIONS.csv")

caregivers <- read.csv(file = "CAREGIVERS.csv")

callout <- read.csv(file = "CALLOUT.csv")

patients <- read.csv(file ="PATIENTS.csv")

icustays <- read.csv(file = "ICUSTAYS.csv")

chartevents <- read.csv(file = "CHARTEVENTS.csv")

d\_items <- read.csv(file = "D\_ITEMS.csv")

labevents <- read.csv( file = "LABEVENTS.csv")

d\_labitems <- read.csv("D\_LABITEMS.csv")

The next approach is to get the necessary libraries and as such are: library(tidyverse) utilized for data wrangling and visualization, library(magrittr) meant for piping data and analysis chains, library(ggplot2) used for data visualization, library(rsample) is also required for data sampling, library(plotROC) is also utilized for data visualization, library(kableExtra) is also required for data wrangling, library(mice) is utilized for imputation and finally, library(randomForest) is a supervised classification algorithm. It is imperative to note that these libraries have their respective r packages that are automatically installed in the r programming environment.

> str(icustays)

'data.frame': 136 obs. of 12 variables:

$ row\_id : int 12742 12747 12749 12754 12755 12762 12763 12765 12767 12768 ...

$ subject\_id : int 10006 10011 10013 10017 10019 10026 10027 10029 10032 10033 ...

$ hadm\_id : int 142345 105331 165520 199207 177759 103770 199395 132349 140372 157235 ...

$ icustay\_id : int 206504 232110 264446 204881 228977 277021 286020 226055 267090 254543 ...

$ dbsource : chr "carevue" "carevue" "carevue" "carevue" ...

$ first\_careunit: chr "MICU" "MICU" "MICU" "CCU" ...

$ last\_careunit : chr "MICU" "MICU" "MICU" "CCU" ...

$ first\_wardid : int 52 15 15 7 15 33 12 33 52 33 ...

$ last\_wardid : int 52 15 15 7 15 33 12 33 52 33 ...

$ intime : chr "10/23/2164 21:10" "8/14/2126 22:34" "10/4/2125 23:38" "5/29/2149 18:52" ...

$ outtime : chr "10/25/2164 12:21" "8/28/2126 18:59" "10/7/2125 15:13" "5/31/2149 22:19" ...

$ los : num 1.63 13.85 2.65 2.14 1.29 ...

In the data.frame, it reveals 136 observations of 12 variables which implies that there are 136 patients but the truth is that there are only 100 patients. This means that some patients have multiple ICUSTAYS.

> first\_ICU\_stay <- icustays %>%

+ left\_join(admissions, by = c("subject\_id" = "subject\_id", "hadm\_id" = "hadm\_id")) %>%

+ group\_by(subject\_id) %>%

+ filter(intime == min(intime)) %>%

+ ungroup() %>%

+ select(subject\_id, hadm\_id, icustay\_id, intime, los, diagnosis, hospital\_expire\_flag)

> last\_ICU\_stay <- icustays %>%

+ left\_join(admissions, by = c("subject\_id" = "subject\_id", "hadm\_id" = "hadm\_id")) %>%

+ group\_by(subject\_id) %>%

+ filter(intime == max(intime)) %>%

+ ungroup() %>%

+ select(subject\_id, hadm\_id, icustay\_id, intime,los, diagnosis, hospital\_expire\_flag)

> t1 <- first\_ICU\_stay %>% filter(hospital\_expire\_flag == 1) %>% count()

> t2 <- last\_ICU\_stay %>% filter(hospital\_expire\_flag == 1) %>% count()

> kable(list(t1, t2)) %>% kable\_styling()

During the first icustay, n= 37 patients did not survive, and the number for the last ICU stay is n= 36. This implies that the last icustay table can be used as the data frame to perform the exploratory data analysis and then proceed to build the predictive model for the data.

**2. Methodology**

**2.1 Predictive Assessment in the MIMIC III Dataset**

There are four basic tasks in this study to serve as benchmark for the machine learning algorithms. These are: length of stays, diagnosis, mortality rates in ICU and admissions.

**2.1.1 Length of Stays**

This is one of the most crucial tasks that requires accurate predictions for identifying the cost of admissions in the ICU with respect to staff managements and other related items. The duration of stays of patients in the ICU for clinical decisions may play a key role in cost management adjustment. To investigate the predictive performance for length of stay, the AUC was employed for a definitive and accurate results.

**2.1.2 Diagnosis**

To evaluate the predictive performance for diagnosis, the AUC also played a vital role in assessing the diagnosis type case count and diagnosis type by death.

**2.1.3 Admissions**

The type of admissions to the ICU was carefully examined and re-admissions of patients were also considered in the predictive performance using the AUC and ROC.

**2.2 Data Analysis and Model Design Evaluations**

This section comprises of the application of feature engineering for machine learning and data analytics to differentiate between categorical and numeric variable in the data frame. The selection of the variables is not that intuitive as one would assume for especially ICU mortality rate and length of stay in the ICU. The variations in ICU mortality rate can be easily observed in patients based on their characteristics during admissions. The ICU risk adjusted mortality need to be investigated carefully for patient severity of illness as well as the ICU length of stay (LOS) in addition to patient risk factors.

The structure for the chartevents is given below:

> str(chartevents)

'data.frame': 758355 obs. of 15 variables:

$ row\_id : int 5279021 5279022 5279023 5279024 5279025 5279026 5279027 5279028 5279029 5279030 ...

$ subject\_id : int 40124 40124 40124 40124 40124 40124 40124 40124 40124 40124 ...

$ hadm\_id : int 126179 126179 126179 126179 126179 126179 126179 126179 126179 126179 ...

$ icustay\_id : int 279554 279554 279554 279554 279554 279554 279554 279554 279554 279554 ...

$ itemid : int 223761 224695 220210 220045 220179 220180 220181 220046 220047 223751 ...

$ charttime : chr "2/4/2130 4:00" "2/4/2130 4:25" "2/4/2130 4:30" "2/4/2130 4:32" ...

$ storetime : chr "2/4/2130 4:35" "2/4/2130 5:55" "2/4/2130 4:43" "2/4/2130 4:43" ...

$ cgid : int 19085 18999 21452 21452 21452 21452 21452 19085 19085 19085 ...

$ value : chr "95.9" "2222221.7" "15" "94" ...

$ valuenum : num 9.59e+01 2.22e+06 1.50e+01 9.40e+01 1.63e+02 ...

$ valueuom : chr "?F" "cmH2O" "insp/min" "bpm" ...

$ warning : int 0 0 0 0 0 0 0 0 0 0 ...

$ error : int 0 0 0 0 0 0 0 0 0 0 ...

$ resultstatus: chr "" "" "" "" ...

$ stopped : chr "" "" "" "" ...

> head(chartevents)

row\_id subject\_id hadm\_id icustay\_id itemid charttime storetime cgid value valuenum

1 5279021 40124 126179 279554 223761 2/4/2130 4:00 2/4/2130 4:35 19085 95.9 95.9

2 5279022 40124 126179 279554 224695 2/4/2130 4:25 2/4/2130 5:55 18999 2222221.7 2222221.7

3 5279023 40124 126179 279554 220210 2/4/2130 4:30 2/4/2130 4:43 21452 15 15.0

4 5279024 40124 126179 279554 220045 2/4/2130 4:32 2/4/2130 4:43 21452 94 94.0

5 5279025 40124 126179 279554 220179 2/4/2130 4:32 2/4/2130 4:43 21452 163 163.0

6 5279026 40124 126179 279554 220180 2/4/2130 4:32 2/4/2130 4:43 21452 81 81.0

valueuom warning error resultstatus stopped

1 ?F 0 0

2 cmH2O 0 0

3 insp/min 0 0

4 bpm 0 0

5 mmHg 0 0

6 mmHg 0 0

The structure for labevents is provided in the below for in r programming environment:

> str(labevents)

'data.frame': 76074 obs. of 9 variables:

$ row\_id : int 6244563 6244564 6244565 6244566 6244567 6244568 6244569 6244570 6244571 6244572 ...

$ subject\_id: int 10006 10006 10006 10006 10006 10006 10006 10006 10006 10006 ...

$ hadm\_id : int NA NA NA NA NA NA NA NA NA NA ...

$ itemid : int 50868 50882 50893 50902 50912 50931 50960 50970 50971 50983 ...

$ charttime : chr "9/24/2164 20:21" "9/24/2164 20:21" "9/24/2164 20:21" "9/24/2164 20:21" ...

$ value : chr "19" "27" "10" "97" ...

$ valuenum : num 19 27 10 97 7 126 2.3 5.6 4.3 139 ...

$ valueuom : chr "mEq/L" "mEq/L" "mg/dL" "mEq/L" ...

$ flag : chr "" "" "" "" ...

There are a total of 758355 observations of 15 variables in the chartevents and 76074 observations of 9 variables in the labevents during the 134 entries in icustays.

**2.3 Diagnosis**

> last\_ICU\_stay <- last\_ICU\_stay %>% mutate(diagnosis\_type = case\_when(

+ grepl("(.)\*CANCER|(.) CA|(.)\*LEUKEMIA", diagnosis, ignore.case = TRUE) ~ "CANCER",

+ grepl("(.)\*SEPSIS|INFECTION|CHOLANGITIS|ABSCESS|FEVER|PNEUMONIA", diagnosis, ignore.case = TRUE) ~ "INFECTION",

+ grepl("(.)\*MI|VF ARREST|STROKE(.)\*|CHEST PAIN|MYOCARDIAL", diagnosis, ignore.case = TRUE) ~ "ACUTE CARDIAC",

+ grepl("(.)\*ACCIDENT|FRACTURE|(.)\*FALL|SEIZURE|BREATH|BLEED|HEMATOMA|OVERDOSE|SYNCOPE(.)\*|TACHYPNEA|ACUTE",diagnosis, ignore.case = TRUE) ~ "ACUTE",

+ grepl("(.)\*PULMONARY|RESPIRATORY|LUNG|ASTHMA|LIVER|HEPATI(TIS|IC)|ESOPHAGEAL|OA|ARTHRITIS|(HYPO|HYPER)TENSION|HEADACHE|DISTRESS|MENTAL|HEAD|ANGINA|BRAIN|HEART FAILURE|(.)\*EFFUSION|FAILURE TO THRIVE|"

+ , diagnosis, ignore.case = TRUE) ~ "CHRONIC", TRUE ~ "OTHERS"))

> ggplot(last\_ICU\_stay, aes(x=diagnosis\_type, fill = factor(hospital\_expire\_flag)))+

+ geom\_bar(stat = "count", position = "dodge")+

+ theme\_bw()+

+ theme(axis.text.x = element\_text(size = 8, angle = 90))+

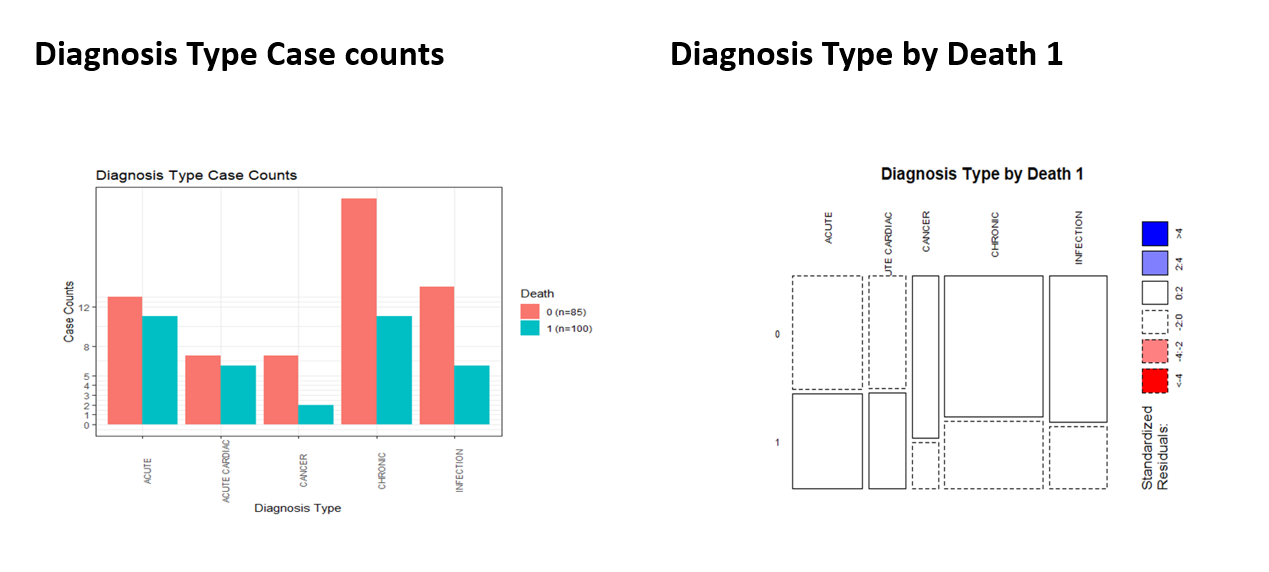
+ labs(x= "Diagnosis Type", y="Case Counts", title = "Diagnosis Type Case Counts")+

+ scale\_y\_continuous(breaks = c(0,1,2,3,4,5,8,12))+

+ scale\_fill\_discrete(name="Death", labels=c("0 (n=85)","1 (n=100)"))

> mosaicplot(table(last\_ICU\_stay$diagnosis\_type, last\_ICU\_stay$hospital\_expire\_flag),

+ main = "Diagnosis Type by Death 1", shade = TRUE, las = 2)



**Figure 1 – Diagnosis type counts and death**

**2.4 Heart Rate and Blood Pressure During ICUSTAY**

> d\_items %>%

+ filter(str\_detect(label, pattern = regex("heart rate", ignore\_case = TRUE))) %>%

+ print()

row\_id itemid label abbreviation dbsource linksto category

1 212 211 Heart Rate carevue chartevents

2 1078 3494 Lowest Heart Rate carevue chartevents

3 12712 220045 Heart Rate HR metavision chartevents Routine Vital Signs

4 12713 220046 Heart rate Alarm - High HR Alarm - High metavision chartevents Alarms

5 12714 220047 Heart Rate Alarm - Low HR Alarm - Low metavision chartevents Alarms

unitname param\_type conceptid

1 NA

2 NA

3 bpm Numeric NA

4 bpm Numeric NA

5 bpm Numeric NA

> heart\_rate <- chartevents %>%

+ filter(itemid %in% c(211, 220045)) %>%

+ select(subject\_id, hadm\_id, icustay\_id, charttime, valuenum) %>%

+ group\_by(subject\_id, hadm\_id, icustay\_id) %>%

+ summarise(max\_heart\_rate = max(valuenum, na.rm = TRUE))

`summarise()` has grouped output by 'subject\_id', 'hadm\_id'. You can override using the `.groups` argument.

> #Lowest blood pressure

> d\_items %>%

+ filter(str\_detect(label, pattern = regex("diastolic(blood pressure)?|NBP|Arterial BP", ignore\_case = TRUE))) %>%

+ print()

row\_id itemid label abbreviation dbsource

1 8 1449 Arterial BP(Rad) carevue

2 57 51 Arterial BP [Systolic] carevue

3 58 52 Arterial BP Mean carevue

4 154 153 Diastolic Unloading carevue

5 419 455 NBP [Systolic] carevue

6 420 456 NBP Mean carevue

7 1920 5817 NBP Alarm [Low] carevue

8 4325 6701 Arterial BP #2 [Systolic] carevue

9 4326 6702 Arterial BP Mean #2 carevue

10 4433 6926 Arterial BP #3 carevue

11 4827 8364 ABP [Diastolic] carevue

12 4831 8368 Arterial BP [Diastolic] carevue

13 4903 8440 Manual BP [Diastolic] carevue

14 4904 8441 NBP [Diastolic] carevue

15 4907 8444 Orthostat BP sitting [Diastolic] carevue

16 4908 8445 OrthostatBP standing [Diastolic] carevue

17 4909 8446 Orthostatic BP lying [Diastolic] carevue

18 4911 8448 PAP [Diastolic] carevue

19 4953 8502 BP Cuff [Diastolic] carevue

20 4954 8503 BP Left Arm [Diastolic] carevue

21 4955 8504 BP Left Leg [Diastolic] carevue

22 4956 8505 BP PAL [Diastolic] carevue

23 4957 8506 BP Right Arm [Diastolic] carevue

24 4958 8507 BP Right Leg [Diastolic] carevue

25 4959 8508 BP UAC [Diastolic] carevue

26 4995 8551 NBP Alarm [High] carevue

27 5017 8555 Arterial BP #2 [Diastolic] carevue

28 12717 220051 Arterial Blood Pressure diastolic ABPd metavision

29 12722 220060 Pulmonary Artery Pressure diastolic PAPd metavision

30 12735 220180 Non Invasive Blood Pressure diastolic NBPd metavision

31 13424 224643 Manual Blood Pressure Diastolic Left Manual BPd L metavision

32 13688 225310 ART BP Diastolic ART BP Diastolic metavision

33 14616 227240 NBP Alarm Source NBP Alarm Source metavision

34 14618 227242 Manual Blood Pressure Diastolic Right Manual BPd R metavision

35 14630 226851 RV diastolic pressure(PA Line) RV diastolic pressure(PA Line) metavision

36 14632 226853 PA diastolic pressure(PA Line) PA diastolic pressure(PA Line) metavision

37 15338 228151 Aortic Pressure Signal - Diastolic Aortic Pressure Signal - Diastolic metavision

linksto category unitname param\_type conceptid

1 chartevents NA

2 chartevents NA

….

28 chartevents Routine Vital Signs mmHg Numeric NA

29 chartevents Hemodynamics mmHg Numeric NA

30 chartevents Routine Vital Signs mmHg Numeric NA

31 chartevents Routine Vital Signs mmHg Numeric NA

32 chartevents Routine Vital Signs mmHg Numeric NA

33 chartevents Alarms Text NA

34 chartevents Routine Vital Signs mmHg Numeric NA

35 chartevents PA Line Insertion mmHg Numeric NA

36 chartevents PA Line Insertion mmHg Numeric NA

37 chartevents Impella None Numeric NA

> blood\_pressure <- chartevents %>%

+ filter(itemid %in% c(227242, 224643, 220180, 220051,455, 8441, 5817, 8551, 8368)) %>%

+ select(subject\_id, hadm\_id, icustay\_id, charttime, valuenum) %>%

+ group\_by(subject\_id, hadm\_id, icustay\_id) %>%

+ summarise(min\_blood\_pressure = min(valuenum, na.rm = TRUE))

` summarise()` has grouped output by 'subject\_id', 'hadm\_id'. You can override using the `.groups` argument.

> last\_ICU\_stay <- last\_ICU\_stay %>%

+ left\_join(heart\_rate, by = c("subject\_id", "hadm\_id", "icustay\_id")) %>%

+ left\_join(blood\_pressure, by = c("subject\_id", "hadm\_id", "icustay\_id"))

> last\_ICU\_stay %>% filter(is.na(last\_ICU\_stay$max\_heart\_rate))

# A tibble: 2 x 11

subject\_id hadm\_id icustay\_id intime los diagnosis hospital\_expire\_~ diagnosis\_type min\_GCS

<int> <int> <int> <chr> <dbl> <chr> <int> <chr> <dbl>

1 10067 160442 236674 10/6/2130 ~ 0.106 S/P MOTORCYCLE ~ 1 ACUTE Inf

2 10120 193924 268282 5/12/2115 ~ 2.65 LIVER FAILURE 1 CHRONIC Inf

# ... with 2 more variables: max\_heart\_rate <dbl>, min\_blood\_pressure <dbl>

> ggplot(last\_ICU\_stay, aes(x= as.character(hospital\_expire\_flag),

+ y= max\_heart\_rate,

+ fill = factor(hospital\_expire\_flag)))+

+ geom\_boxplot(position = "dodge")+ theme\_bw()+

+ labs(x= "Death", y="Heart Rate(BPM)",

+ title = "Maximum Heart Rate During the ICU Stay")+

+ scale\_fill\_discrete(name="Death", labels=c("0 (n=100)","1 (n=120)"))

Warning message:

Removed 2 rows containing non-finite values (stat\_boxplot).

> ggplot(last\_ICU\_stay, aes(x= as.character(hospital\_expire\_flag),

+ y= min\_blood\_pressure,

+ fill = factor(hospital\_expire\_flag)))+

+ geom\_boxplot(position = "dodge")+ theme\_bw()+

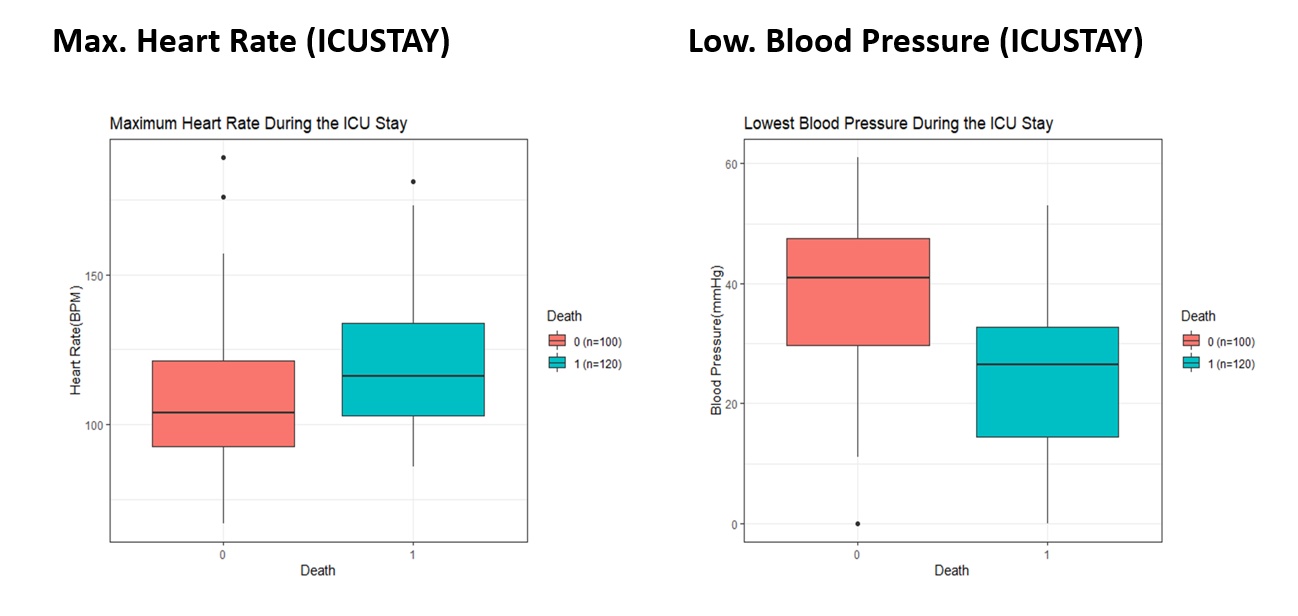
+ labs(x= "Death", y="Blood Pressure(mmHg)",

+ title = "Lowest Blood Pressure During the ICU Stay")+

+ scale\_fill\_discrete(name="Death", labels=c("0 (n=100)","1 (n=120)"))

Warning message:

Removed 2 rows containing non-finite values (stat\_boxplot).



**Figure 2 – Heart rate and blood pressure**

**2.5 Lowest Blood Glucose During Admissions and Length of ICUSTAY**

It is ideal to investigate whether the ICU length of stay is relevant for survival of patients admitted into ICU or not. Though, it is known that blood glucose greater than or equal to 80mg/dl could play a factor to survival rate. Below is a r code to analyze blood glucose and length of stay with their respective output:

> d\_labitems %>%

+ filter(str\_detect(label, pattern = regex("Glucose", ignore\_case = TRUE))) %>%

+ print()

row\_id itemid label fluid category loinc\_code

1 10 50809 Glucose Blood Blood Gas 2339-0

2 43 50842 Glucose, Ascites Ascites Chemistry 2347-3

3 132 50931 Glucose Blood Chemistry 2345-7

4 214 51014 Glucose, CSF Cerebrospinal Fluid (CSF) Chemistry 2342-4

5 222 51022 Glucose, Joint Fluid Joint Fluid Chemistry 2348-1

6 234 51034 Glucose, Body Fluid Other Body Fluid Chemistry 2344-0

7 253 51053 Glucose, Pleural Pleural Chemistry 2346-5

8 284 51084 Glucose, Urine Urine Chemistry 2350-7

9 678 51478 Glucose Urine Hematology 5792-7

10 729 51529 Estimated Actual Glucose BLOOD CHEMISTRY

> min\_BG<- labevents %>%

+ filter(itemid %in% c(50809, 50931)) %>%

+ group\_by(subject\_id, hadm\_id) %>%

+ summarise(min\_BG = min(valuenum, na.rm = T))

`summarise()` has grouped output by 'subject\_id'. You can override using the `.groups` argument.

> last\_ICU\_stay <- last\_ICU\_stay %>%

+ left\_join(min\_BG, by=c("subject\_id", "hadm\_id"))

> #the length of ICU stay

> last\_ICU\_stay %>% group\_by(hospital\_expire\_flag) %>%

+ summarise(mean=mean(los))

# A tibble: 2 x 2

hospital\_expire\_flag mean

<int> <dbl>

1 0 4.09

2 1 5.82

> ggplot(last\_ICU\_stay, aes(x= as.character(hospital\_expire\_flag),

+ y= min\_BG,

+ fill = factor(hospital\_expire\_flag)))+

+ geom\_boxplot(position = "dodge")+ theme\_bw()+

+ labs(x= "Death", y="Blood Glucose(mg/dL)",

+ title = "Lowest Blood Glucose During the Hospital Admission")+

+ scale\_fill\_discrete(name="Death", labels=c("0 (n=100)","1 (n=140)"))

> ggplot(last\_ICU\_stay, aes(x= as.character(hospital\_expire\_flag),

+ y= los,

+ fill = factor(hospital\_expire\_flag)))+

+ geom\_boxplot(position = "dodge")+ theme\_bw()+

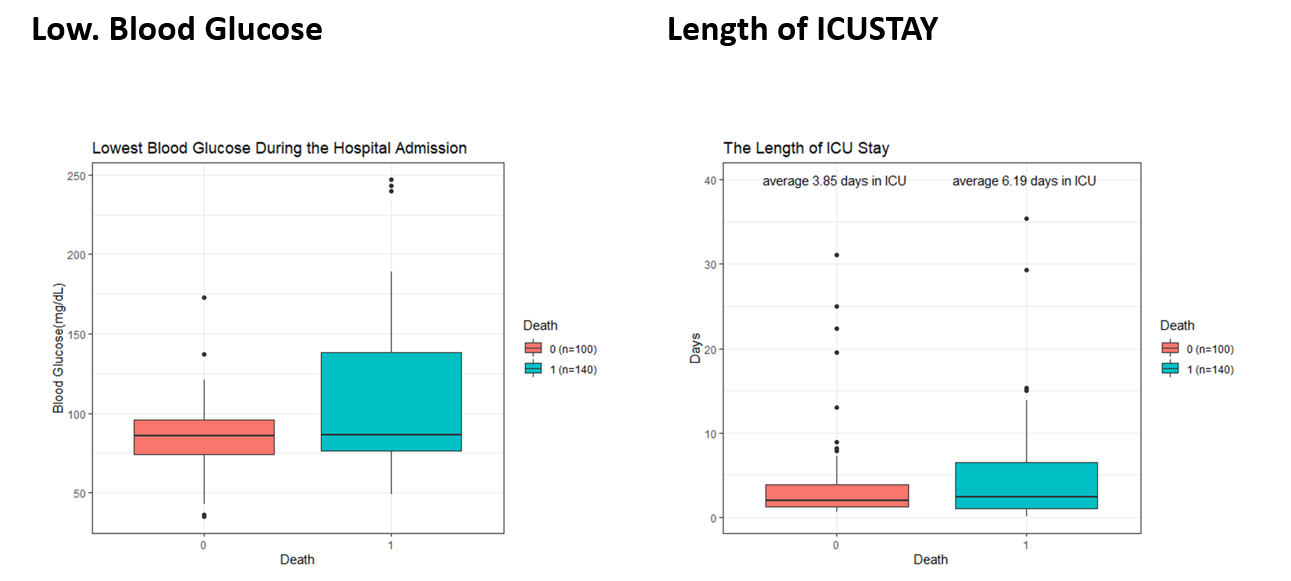
+ labs(x= "Death", y="Days",

+ title = "The Length of ICU Stay")+

+ scale\_fill\_discrete(name="Death", labels=c("0 (n=100)","1 (n=140)"))+

+ annotate("text", x=1, y=40, label="average 3.85 days in ICU ")+

+ annotate("text", x=2, y=40, label="average 6.19 days in ICU ")



**Figure 3 – Blood glucose and ICU length of stay**

**2.6 Gender and Age of Patients**

> library(lubridate)

> gender <- patients %>%

+ select(subject\_id, gender) %>%

+ mutate(male = case\_when(gender == "M" ~1,

+ TRUE ~0)) %>%

+ right\_join(last\_ICU\_stay) %>%

+ select(subject\_id, hadm\_id, icustay\_id, male, hospital\_expire\_flag)

> date\_of\_birth <- patients %>%

+ select(subject\_id, dob) %>%

+ right\_join(last\_ICU\_stay) %>%

+ mutate(

+ dob = mdy\_hm(dob),

+ intime = mdy\_hm(intime),

+ age\_intake\_ICU = round(as.numeric(intime - dob)/365.25)

+ )%>%

+ select(subject\_id, hadm\_id, icustay\_id, age\_intake\_ICU, hospital\_expire\_flag)

> table(is.na(date\_of\_birth$age\_intake\_ICU))

FALSE TRUE

92 8

> last\_ICU\_stay <- last\_ICU\_stay %>%

+ left\_join(gender) %>%

+ left\_join(date\_of\_birth)

> analysis\_data <- last\_ICU\_stay

> factor\_vars <- c('los','diagnosis\_type', "min\_GCS", "max\_heart\_rate"

+ , "min\_blood\_pressure", "min\_BG", "male")

> last\_ICU\_stay[factor\_vars] <- lapply(last\_ICU\_stay[factor\_vars], function(x) as.factor(x))

> set.seed(20210116)

> library(mice)

> mice\_mod <- mice(last\_ICU\_stay[, !names(last\_ICU\_stay) %in% c('subject\_id',

+ 'hadm\_id', 'icustay\_id', 'intime', 'hospital\_expire\_flag')], method='rf')

iter imp variable

1 1 max\_heart\_rate min\_blood\_pressure

1 2 max\_heart\_rate min\_blood\_pressure

1 3 max\_heart\_rate min\_blood\_pressure

1 4 max\_heart\_rate min\_blood\_pressure

1 5 max\_heart\_rate min\_blood\_pressure

2 1 max\_heart\_rate min\_blood\_pressure

2 2 max\_heart\_rate min\_blood\_pressure

2 3 max\_heart\_rate min\_blood\_pressure

2 4 max\_heart\_rate min\_blood\_pressure

2 5 max\_heart\_rate min\_blood\_pressure

3 1 max\_heart\_rate min\_blood\_pressure

3 2 max\_heart\_rate min\_blood\_pressure

3 3 max\_heart\_rate min\_blood\_pressure

3 4 max\_heart\_rate min\_blood\_pressure

3 5 max\_heart\_rate min\_blood\_pressure

4 1 max\_heart\_rate min\_blood\_pressure

4 2 max\_heart\_rate min\_blood\_pressure

4 3 max\_heart\_rate min\_blood\_pressure

4 4 max\_heart\_rate min\_blood\_pressure

4 5 max\_heart\_rate min\_blood\_pressure

5 1 max\_heart\_rate min\_blood\_pressure

5 2 max\_heart\_rate min\_blood\_pressure

5 3 max\_heart\_rate min\_blood\_pressure

5 4 max\_heart\_rate min\_blood\_pressure

5 5 max\_heart\_rate min\_blood\_pressure

Warning message:

Number of logged events: 3

> mice\_output <- complete(mice\_mod)

> par(mfrow=c(1,2))

> hist(last\_ICU\_stay$age\_intake\_ICU, freq = F, main = "Age: Original Data",

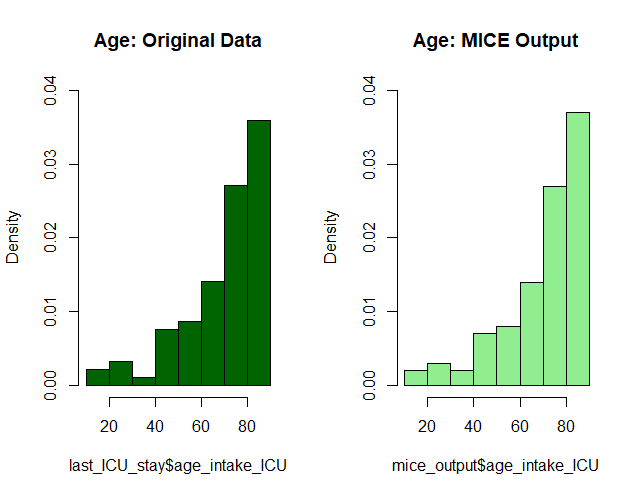
+ col = 'darkgreen', ylim = c(0, 0.04))

> hist(mice\_output$age\_intake\_ICU, freq = F, main = "Age: MICE Output",

+ col = 'lightgreen', ylim = c(0, 0.04))

>

> mice\_output <- complete(mice\_mod)



**Figure 4 – Age of Patients**

**3. Results:**

**3.1 Receiver Operator Curve (ROC) and Area Under Curve (AUC)**

ROC curve is an important classification evaluation metric. It tells us how well the model has accurately predicted. The ROC curve shows the sensitivity of the classifier by plotting the rate of true positives to the rate of false positives. If the classifier is outstanding, the true positive rate will increase, and the area under the curve will be close to one. If the classifier is like random guessing, the true positive rate will increase linearly with the false positive rate. The better the AUC measure, the better the model.

The ICU risk adjustment was applied to the models for patients with respect to the features or variables in the data. Each model has maintained adequate data training and testing with no patient discrimination in the length of stay in the ICU. To improve the quantitative validation of the model, the ROC was properly integrated to ensure accurate prediction.

**3.1 Prediction – Binomial Regression Model**

> library(rsample)

> set.seed(2021)

> data\_split <- initial\_split(analysis\_data, prop = 7/10)

> training\_data <- training(data\_split)

> testing\_data <- testing(data\_split)

> model\_bin <- training\_data %>%

+ glm(formula = hospital\_expire\_flag ~ los+diagnosis\_type

+ +max\_heart\_rate+min\_blood\_pressure+male+age\_intake\_ICU,

+ family = 'binomial',na.action = na.omit)

> summary(model\_bin)

Call:

glm(formula = hospital\_expire\_flag ~ los + diagnosis\_type + max\_heart\_rate +

min\_blood\_pressure + male + age\_intake\_ICU, family = "binomial",

data = ., na.action = na.omit)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.3928 -0.8309 -0.4283 1.0465 2.1282

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.45443 3.39163 -1.019 0.3084

los 0.03528 0.06395 0.552 0.5812

diagnosis\_typeACUTE CARDIAC -0.48649 0.92732 -0.525 0.5998

diagnosis\_typeCANCER -3.17787 1.53799 -2.066 0.0388 \*

diagnosis\_typeCHRONIC -1.39977 0.82103 -1.705 0.0882 .

diagnosis\_typeINFECTION -1.81710 0.94747 -1.918 0.0551 .

max\_heart\_rate 0.02357 0.01804 1.306 0.1915

min\_blood\_pressure -0.04106 0.02459 -1.670 0.0949 .

male 0.46198 0.66267 0.697 0.4857

age\_intake\_ICU 0.03278 0.02472 1.326 0.1847

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 91.422 on 68 degrees of freedom

Residual deviance: 71.381 on 59 degrees of freedom

AIC: 91.381

Number of Fisher Scoring iterations: 5

> training\_roc <- training\_data %>%

+ ggplot(aes(m= predicted\_outcome, d=hospital\_expire\_flag))+

+ labs(title = "ROC Plot for Training Data")+

+ geom\_roc(n.cuts = 10, labels = F, labelround = 4)+

+ style\_roc(theme = theme\_grey)

> calc\_auc(training\_roc)$AUC\*100

[1] 100

> training\_data$predicted\_outcome <- predict(model\_bin, training\_data,

+ type = "response")

> training\_roc <- training\_data %>%

+ ggplot(aes(m= predicted\_outcome, d=hospital\_expire\_flag))+

+ labs(title = "ROC Plot for Training Data")+

+ geom\_roc(n.cuts = 10, labels = F, labelround = 4)+

+ style\_roc(theme = theme\_grey)

> calc\_auc(training\_roc)$AUC\*100

[1] 84.07871

> testing\_data$predicted\_outcome <- predict(model\_bin, testing\_data,

+ type = "response")

> testing\_roc <- testing\_data %>%

+ ggplot(aes(m= predicted\_outcome, d= hospital\_expire\_flag))+

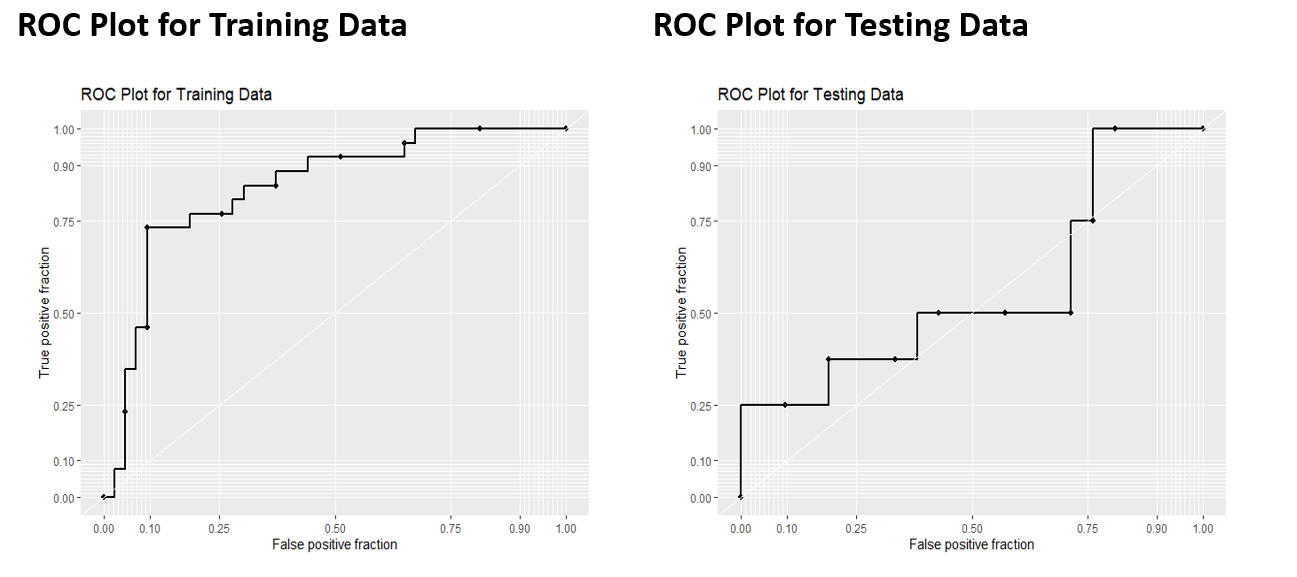
+ labs(title = "ROC Plot for Testing Data")+

+ geom\_roc(n.cuts = 10, labels = F, labelround = 4) +

+ style\_roc(theme = theme\_grey)

> calc\_auc(testing\_roc)$AUC\*100

[1] 55.95238

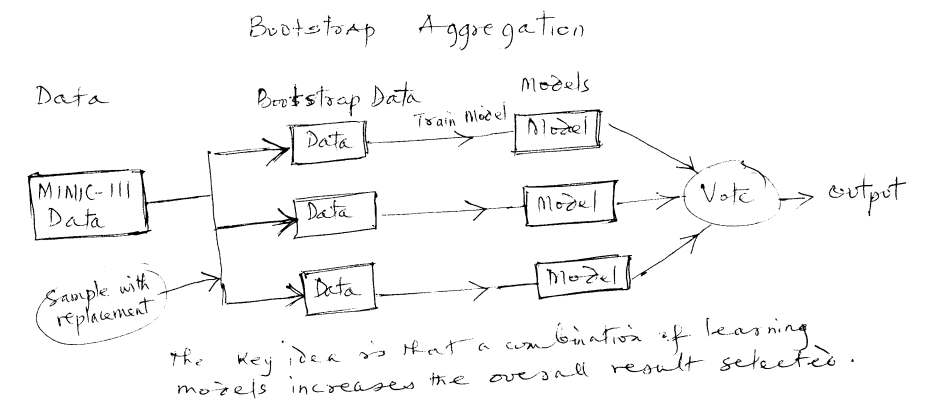


**Figure 5- Binomial Regression Model ROC Plots**

The Binomial regression model has an AUC of 84% in training data, which is surprisingly good, while the AUC in testing data was approximately 56%.

**3.2 Prediction – Random Forest Classification Model**

Random forest classifier is an ensemble algorithm based on bagging such as bootstrap aggregation. Ensemble methods combines more than one algorithm of the same or different kind for classifying objects (i.e., an ensemble of SVM, naive Bayes or decision trees, as a typical example)



**Figure 6 – Random Forest Classification Model**

Deep decision trees may suffer from overfitting, but random forests prevent overfitting by creating trees on random subsets. The main reason is that it takes the average of all the predictions, which cancels out the biases. Random forest adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

> model\_rf <- randomForest(factor(hospital\_expire\_flag) ~ los+diagnosis\_type

+ +max\_heart\_rate

+ +min\_blood\_pressure+male+age\_intake\_ICU,

+ data = analysis\_data, na.action = na.exclude)

> print(model\_rf)

Call:

randomForest(formula = factor(hospital\_expire\_flag) ~ los + diagnosis\_type + max\_heart\_rate + min\_blood\_pressure + male + age\_intake\_ICU, data = analysis\_data, na.action = na.exclude)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 2

OOB estimate of error rate: 35.71%

Confusion matrix:

0 1 class.error

0 51 13 0.2031250

1 22 12 0.6470588

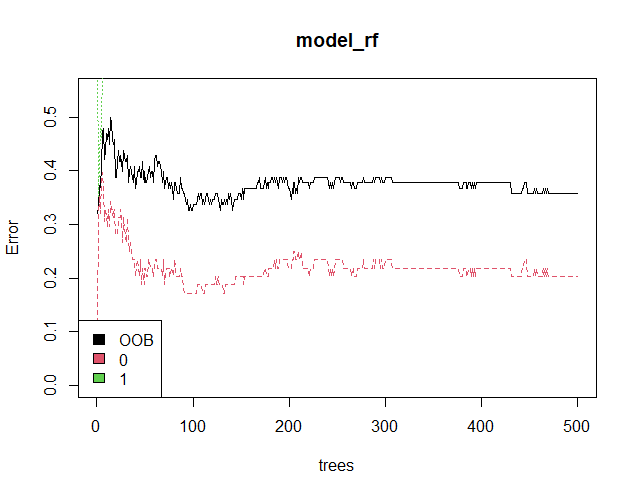
> par(mfrow=c(1,1))

Warning message:

In diff.default(xscale) : reached elapsed time limit

> plot(model\_rf, ylim=c(0,0.55))

> legend("bottomleft", colnames(model\_rf$err.rate), col = 1:3, fill = 1:3)



**Figure 7 – Random Forest Model Out-Of Bag Results**

The black plot reveals the overall out-of-bag error rate which falls approximately 36%. The red and green plots show the error rate for patients that survived and died, respectively. This implies that one can easily predict survival rate than the death rate.

> library(ggplot2)

> importance <- importance(model\_rf)

> varImprtance <- data.frame(variables = row.names(importance),

+ Importance = round(importance[, 'MeanDecreaseGini'], 2))

> rankImportance <- varImprtance %>%

+ mutate(Rank = paste0('#', dense\_rank(desc(Importance))))

>

> ggplot(rankImportance, aes(x= reorder(variables, Importance),

+ y = Importance, fill= Importance))+

+ geom\_bar(stat = 'identity') +

+ geom\_text(aes(x= variables, y=0.5, label = Rank),

+ hjust=0, vjust=0.55, size=4, color='red')+

+ labs(x='Variables')+

+ coord\_flip()+

+ theme\_bw()

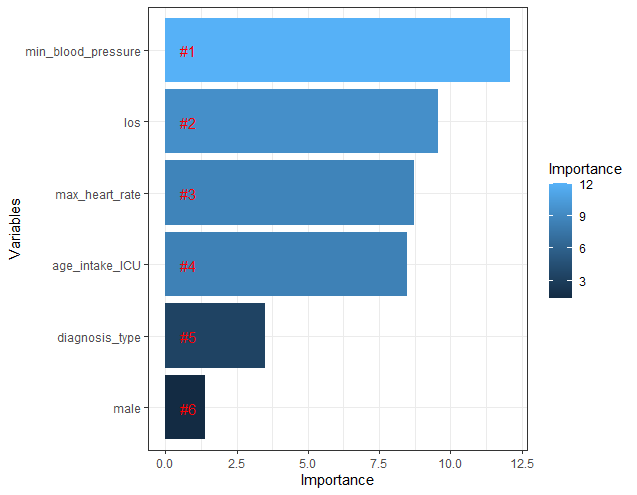


Figure 8 -

The minimum blood pressure and los are the top highest relative importance out of all the predictor variables. Meanwhile, the gender (male) is the least relevant variable and thus regarded as an unimportant (insignificant) variable.

**4. Conclusion**

The MIMIC III dataset is a semi-structured that allows certain variables to be access without real system for validation. The use of r programming language is justifiable with the availability of libraries and many models. The scalability and stability characteristics of models are very suitable when working with large datasets in the r programming environment.

The predictive models revealed that the model with the best results is the Random Forest Classifier which obtain the best predictions in the complete dataset and in the different groups. There are 9 variables from 758,355 chart events and 76,076 lab events that are identified and thus successfully built a Random Forest to predict patients that survives during icustay with an approximately 36% out-of-bag (OOB) error rate. More so, it should be noted that variables with analytic outcomes are more significant in terms of the dataset and different group as related to los. Finally, the ICU admissions and the predictive model can provide an insightful and valuable estimate to los categorically in as much as a cutoff benchmark is defined. This approach can help Doctors to simply the problems even though the dataset is complex.

**5. References**

1. E. de Jonge and M. van der Loo. An introduction to data cleaning with R. Statistics Netherlands, Discussion Paper, 2013, pp. 7

2. T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, Inc., 2003, pp. 99–137.

3. Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. Scientific data, 3, 160035.

4. <https://physionet.org/content/mimiciii-demo/1.4/>