BST260 - Final project

Sabine Friedrich

12/15/2022

Introduction

The dataset that I will analyze was assembled by Korean investigators for a cross-sectional retrospective research study aiming to evaluate accuracy of triage in the emergency department by the Korean Triage and Acuity Scale. The original study report was published in 2019 (1) and the dataset was made available on kaggle.com. This is a tidy dataset including 1267 records of adult patients who were admitted to the emergency department (ED) at two different hospitals between October 2016 and September 2017. It includes a variable detailing the disposition of each patient upon discharge from the ED. My initial plan to predict emergency surgery aiming to identify patients who may require emergency surgery early in order to reduce the time until start of the surgical procedure. However, there were only 22 patients (1.7%) who required emergency surgery upon exploratory analysis (Fig. 1). Accordingly, the aim of this project was adapted to predict inpatient admission (including mortality, or transfer to another hospital) in contrast to discharge home. The ability to predict hospital admission of ED patients may help guide and refine the triage process.

To predict inpatient admission, I will compare two approaches:

- **Clinical approach**: candidate predictors will be pre-selected based on clinical reasoning, then a stepwise forward selection using AIC will be used to build and train a logistic regression model
- **Machine learning approach**: using a random forest model (as many perdictors will be included)

Internal split will be used to derive a training (80% of observations) and a validation set (20%). The training set will be used to fit both models For the clinical model, model calibration will be assessed using a calibration plot comparing predicted risk and observed rates across deciles of predicted inpatient admission risk. To chose a cutoff for the binary prediction, sensitivity and specificity for different cutoffs of predicted inpatient risk will be evaluated.
Finally, performance of both models will be compared in the validation set using overall accuracy, sensitivity, and specificity.

Results

All cases with any missing values for potential predictors were removed and a total of 1228 cases remained in the complete case cohort of which 418 (34%) were admitted. This cohort was split into a training dataset (n=982) and a validation set (n=246). Characteristics of the complete case cohort by outcome status are summarized in Table 1 below.

The following predictors were pre-selected for the **clinical model**: sex, age, pain rating on numeric rating scale (0-10), triage rating by the nurse upon arrival, mode of arrival, if the patient had an injury, if the patient was in shock upon arrival (heart rate > systolic blood pressure), mental status and if the patient was hyperventilating and had fever upon arrival. Performing stepwise forward selection and logistic regression, the final model included triage rating (KTAS_RN), age, mode of arrival (2=ambulance, 3=private vehicle, 4= other, vs. 1=walking), fever (hypertherm), injury and shock and is summarized below.

```
##
## Call:
## glm(formula = admission ~ KTAS_RN + Age + as.factor(arrival) +
##     hypertherm + injury + shock, family = "binomial", data = ED.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0372  -0.8024  -0.4765   0.9187   2.4467
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.084733   0.564747  -0.150  0.88074
## KTAS_RN             -0.768416   0.102214  -7.518 5.57e-14 ***
## Age                  0.018588   0.004219   4.406 1.05e-05 ***
## as.factor(arrival)2  1.299313   0.397141   3.272  0.00107 **
## as.factor(arrival)3  0.557638   0.389871   1.430  0.15263
## as.factor(arrival)4 -0.260732   0.878573  -0.297  0.76664
## hypertherm           1.055091   0.339566   3.107  0.00189 **
## injury              -0.668763   0.242537  -2.757  0.00583 **
## shock                0.845204   0.389749   2.169  0.03011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1229.3  on 981  degrees of freedom
## Residual deviance: 1014.8  on 973  degrees of freedom
## AIC: 1032.8
##
## Number of Fisher Scoring iterations: 4
```

Calibration of the predicted inpatient admission risk using this model was accurate in the training set (Fig. 2), but the model overestimated the inpatient admission risk in the test set (Fig. 3).

Cutoffs of predicted inpatient risk ranging from 0.3 to 0.8 were evaluated in regards to sensitivity and specificity for correctly classifying inpatient admission from the ED (Table 2). A cutoff with a good balance between sensitivity and specificity with a slight emphasis on high sensitivity (low false negative rate) was preferred in order to identify patients who might be more likely to require inpatient admission. A predicted risk cutoff of 0.3 was chosen.

The **machine learning model** was fitted using random forest. The minimum node size was tuned and chosen based on accuracy in the training set. Figure 4 below shows the accuracy for different node sizes.

Table 3 and 4 show the confusion matrix stratified by predicted vs. observed inpatient admission in the validation set based on the clinical (Table 3) and random forest model (Table 4).

Accuracy, sensitivity and specificity for both models are summarized in Table 5. Overall accuracy is comparable for both models. While the random forest model achieves a very high specificity (91%), the sensitivity and balance between sensitivity and specificity is much better for the clinical compared to the machine learning model.

Conclusion

Both approaches, predictor selection based on clinical reasoning combined with stepwise logistic regression as well as random forest, produced models with overall good accuracy. The sensitivity and specificity differed

for both models. In conclusion, of the two models I would prefer the clinical model in order to predict inpatient admission of patients in the ED. I was able to chose a risk cutoff for this model according to my performance preferences for this setting (higher sensitivity more important than good specificity and some balance between both metrics). Also, the clinical model allows healthcare providers in the ED to identify and pay more attention to the most important predictors.

Performance of the machine learning model could potentially be improved with choice of a different machine learning approach and fine tuning of modifiable parameters. The free text variables "chief complaint" and "ED diagnosis" contain important information. Extraction of this information and integration in the models would most likely enhance the predictive ability of both models.

References

Data source: https://www.kaggle.com/datasets/ilkeryildiz/emergency-service-triage-application

(1) Original analysis: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0216972

Appendix

Figures

Fig 1. Distribution of disposition location from ED

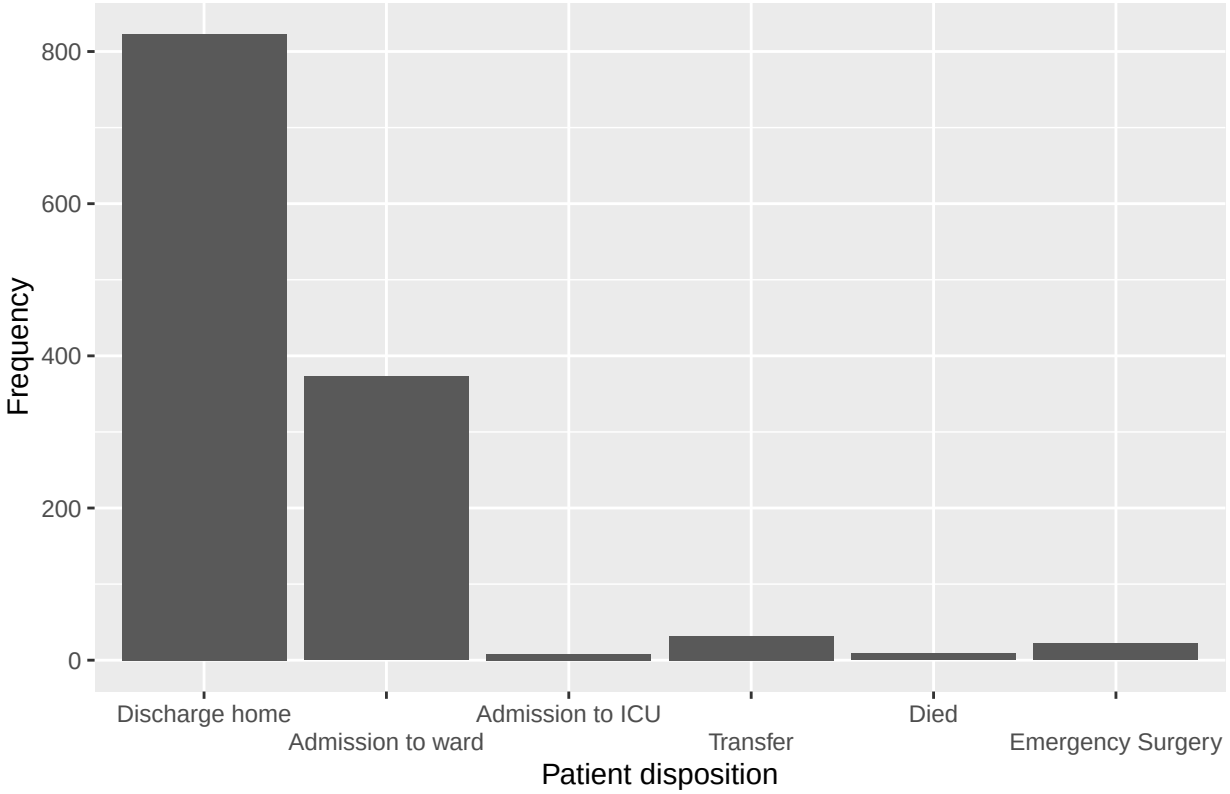


Fig.2 − Reliability Plot − Training Se

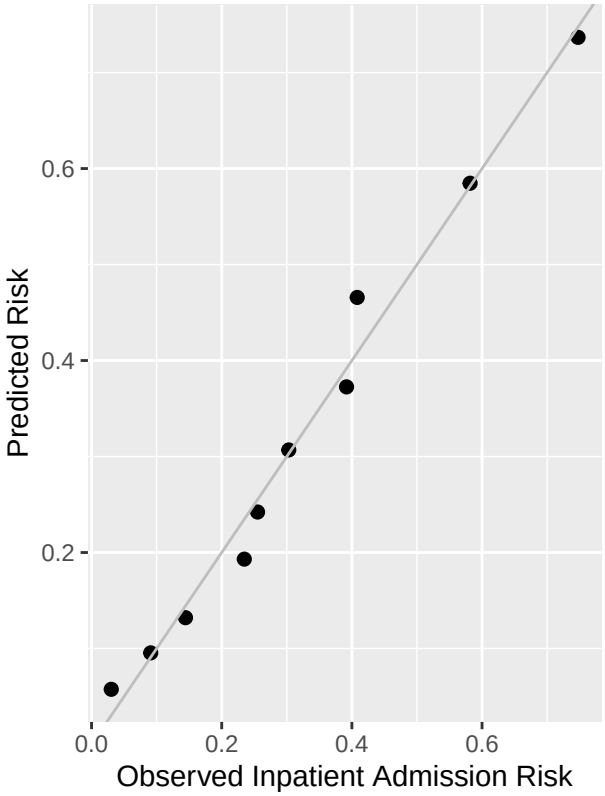


Fig.3 − Reliability Plot − Training Se

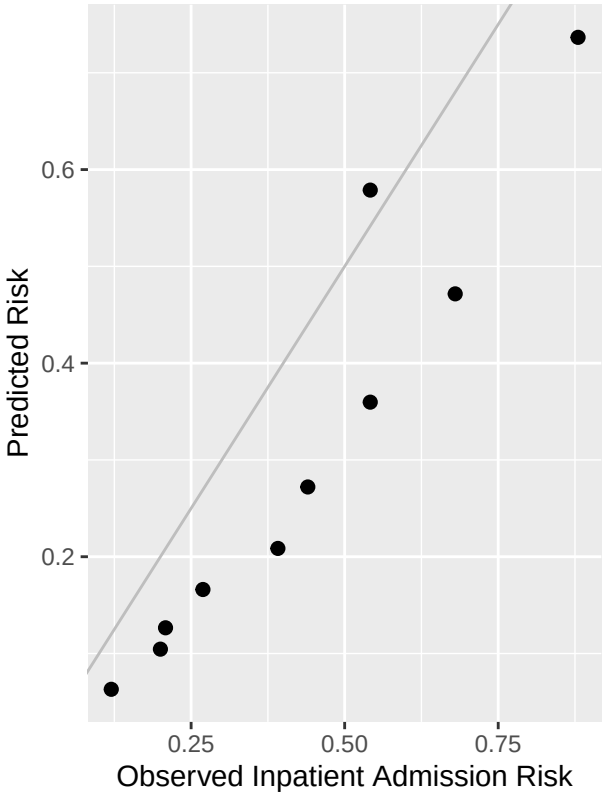
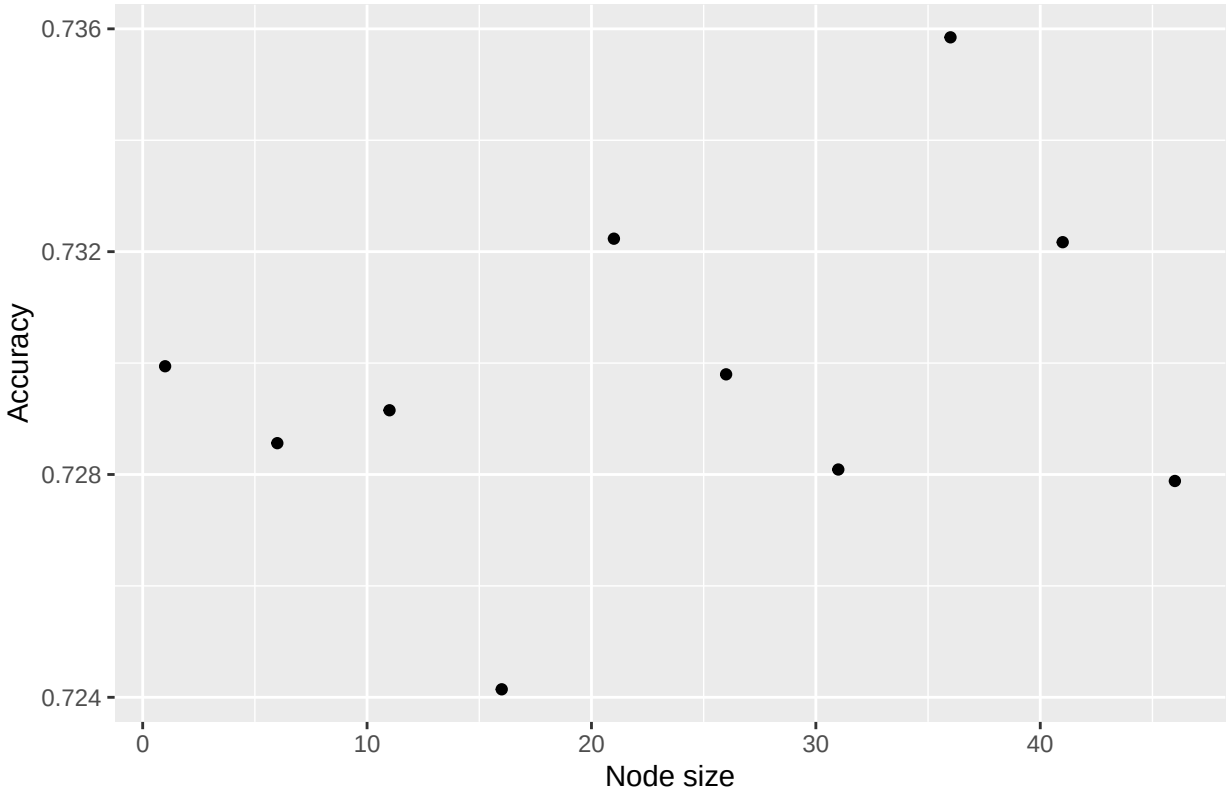


Fig. 4 − Tune of node size



Tables

Table 1: Patient characteristics by discharge disposition

	Inpatient admission	Discharge home	Total
	(N=418)	(N=810)	(N=1228)
Sex			
Female	187 (44.7%)	400 (49.4%)	587 (47.8%)
Male	231 (55.3%)	410 (50.6%)	641 (52.2%)
Age			
Mean (SD)	60.6 (18.3)	50.7 (19.7)	54.1 (19.8)
Median [Min, Max]	64.0 [16.0, 96.0]	51.5 [16.0, 95.0]	56.0 [16.0, 96.0]
Pain rating (0-10)			
Mean (SD)	2.00 (2.28)	2.53 (2.29)	2.35 (2.30)
Median [Min, Max]	0 [0, 10.0]	3.00 [0, 10.0]	3.00 [0, 10.0]
Triage			
Resuscitation	6 (1.4%)	0 (0%)	6 (0.5%)
Emergent	123 (29.4%)	73 (9.0%)	196 (16.0%)
Urgent	184 (44.0%)	260 (32.1%)	444 (36.2%)
Less urgent	97 (23.2%)	402 (49.6%)	499 (40.6%)
Non-urgent	8 (1.9%)	75 (9.3%)	83 (6.8%)
Shock (HR/SBP > 1)			
Shock	24 (5.7%)	15 (1.9%)	39 (3.2%)
No	394 (94.3%)	795 (98.1%)	1189 (96.8%)
Hyperventilation (Respiratory rate > 25/min)			
Hyperventilation	10 (2.4%)	5 (0.6%)	15 (1.2%)
No	408 (97.6%)	805 (99.4%)	1213 (98.8%)
Fever (Temp > 37.5)			
Fever	33 (7.9%)	26 (3.2%)	59 (4.8%)
No	385 (92.1%)	784 (96.8%)	1169 (95.2%)
Mode of arrival			
Walking	16 (3.8%)	63 (7.8%)	79 (6.4%)
Ambulance	208 (49.8%)	185 (22.8%)	393 (32.0%)
Private vehicle	191 (45.7%)	551 (68.0%)	742 (60.4%)
other	3 (0.7%)	11 (1.4%)	14 (1.1%)
Injured			
Injured	42 (10.0%)	193 (23.8%)	235 (19.1%)
No injury	376 (90.0%)	617 (76.2%)	993 (80.9%)
Mental status			
Alert	388 (92.8%)	781 (96.4%)	1169 (95.2%)
Verbal response	19 (4.5%)	18 (2.2%)	37 (3.0%)
Pain response	9 (2.2%)	11 (1.4%)	20 (1.6%)
Unresponsive	2 (0.5%)	0 (0%)	2 (0.2%)

Table 2: Different cutoffs of predicted inpatient admission risk

	Sensitivity	Specificity	Accuracy
0.3	0.7316294	0.6651719	0.6863544
0.4	0.5495208	0.8086697	0.7260692
0.5	0.4472843	0.8804185	0.7423625
0.6	0.2939297	0.9431988	0.7362525
0.7	0.1757188	0.9745889	0.7199593
0.8	0.0319489	0.9970105	0.6894094

Table 3: Clinical model

	Discharge home	Inpatient admission
predicted discharge	104	39
predicted admission	37	66

Table 4: Random forest model

	Discharge home	Inpatient admission
predicted discharge	129	62
predicted admission	12	43

Table 5: Performance of both models in validation set

	Clinical model	Random forest
Specificity	0.7375887	0.9148936
Sensitivity	0.6285714	0.4095238
Accuracy	0.6910569	0.6991870

Code

```
#read in data
library(readr)
emergency <- read_delim("~/Library/Mobile
↪  Documents/com~apple~CloudDocs/Fall2/BST260/emergency.csv",
   delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

```
## Rows: 1267 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr  (9): Chief_complain, NRS_pain, SBP, DBP, HR, RR, BT, Saturation, Diagno...
## dbl (14): Group, Sex, Age, Patients number per hour, Arrival mode, Injury, M...
## num  (1): KTAS duration_min
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#data exploration
library(dplyr)
library(ggplot2)
# outcome
## Disposition: 1 = Discharge, 2 = Admission to ward, 3 = Admission to ICU, 4 =
↪  Discharge, 5 = Transfer, 6 = Death, 7 = Surgery
emergency$Disposition <- as.factor(ifelse(emergency$Disposition == 1 |
↪  emergency$Disposition == 4, 1, emergency$Disposition))

#hist and provide histogram
label <- c("Discharge home", "Admission to ward", "Admission to ICU", "Transfer", "Died",
↪  "Emergency Surgery")
emergency$disposition <- factor(emergency$Disposition, levels = c(1, 2, 3, 5, 6, 7),
↪  labels = label)
#emergency |> ggplot() + geom_bar(aes(disposition)) + xlab("Patient disposition") +
↪  ylab("Frequency") + ggtitle("Fig 1. Distribution of disposition location from ED") +
↪  scale_x_discrete(guide = guide_axis(n.dodge=2))
# binary outcome: discharge - disposition 1 of 4
emergency$admission <- ifelse(emergency$Disposition == 1 | emergency$Disposition == 4, 0,
↪  1)
##data cleaning

# use as is
## sex: 1 female, 2 male
## age: continuous in years
## mental: 1 = Alert, 2 = Verbal Response, 3 = Pain Response, 4 = Unresponsive
## chief complaint: text
## pain: yes=1, no = 0
## SBP: systolic blood pressure
## DBP: diastolic blood pressure
## HR: heart rate
## KTAS_RN: 1 = resuscitation, 2 = emergent, 3 = urgent, 4 = less urgent, 5 = non-urgent

# delete
```

```
## group, which hospital; patients number per hour; saturation - many missing, and
↪  available values range from 90 to 100, not very pathologic, no high predictive value
↪  to be expected
emergency <- emergency |> select(-Group, -`Patients number per hour`, -Saturation,
↪  -KTAS_expert, -Error_group, -mistriage, -`KTAS duration_min`)

#clean/rename
emergency$sbp <- as.numeric(emergency$SBP)
```

```
## Warning: NAs introduced by coercion
```

```
emergency$dbp <- as.numeric(emergency$DBP)
```

```
## Warning: NAs introduced by coercion
```

```
emergency$hr <- as.numeric(emergency$HR)
```

```
## Warning: NAs introduced by coercion
```

```
emergency$resp <- as.numeric(emergency$RR)
```

```
## Warning: NAs introduced by coercion
```

```
emergency$temp <- as.numeric(emergency$BT)
```

```
## Warning: NAs introduced by coercion
```

```
emergency <- emergency |> select(-SBP, -DBP, -HR, -RR, -BT)

# recategorize
##arrival mode: 1 = Walking, 2 = Public Ambulance, 3 = Private Vehicle, 4 = Private
↪  Ambulance, 5,6,7 = Other]
## -> 1 = walking, 2 = ambulance, 3 = private vehicle, 4 = other
emergency$arrival <- ifelse(emergency$`Arrival mode`==2 | emergency$`Arrival mode`==4, 2,
↪  emergency$`Arrival mode`)
emergency$arrival <- ifelse(emergency$arrival==5 | emergency$arrival==6 |
↪  emergency$arrival==7, 4, emergency$arrival)
## injury: 2=yes, 1=no -> 1 yes, 0 no
emergency$injury <- ifelse(emergency$Injury==2, 1, 0)
emergency <- emergency |> select(-Injury, -`Arrival mode`)
##NRS_pain: replace missing as 0, if they did not have pain
emergency$NRS_pain <-  as.numeric(emergency$NRS_pain)
```

```
## Warning: NAs introduced by coercion
```

```r
emergency$NRS_pain <-  ifelse(is.na(emergency$NRS_pain) & emergency$Pain==0, 0,
↪  emergency$NRS_pain)

# generate additional variables out of existing predictors:
## shock index: HR/SBP
emergency$shock_index <- emergency$hr/emergency$sbp
## shock: shock index > 1
emergency$shock <- ifelse(emergency$shock_index > 1, 1, 0)
## hyperventilation: respiratory rate > 25
emergency$hyperventilation <- ifelse(emergency$resp > 25, 1, 0)
## fever based on body temperature?
emergency$hypertherm <- ifelse(emergency$temp > 37.5 & !is.na(emergency$temp), 1, 0)

#fix some column_names
emergency$ED_diagnosis <- emergency$`Diagnosis in ED`
emergency$ED_LOS_min <- emergency$`Length of stay_min`
emergency$chief_complaint <- emergency$Chief_complain
emergency$mental_status <- emergency$Mental
emergency$pain_yn <- emergency$Pain
emergency <- emergency |> select(-`Diagnosis in ED`, -`Length of stay_min`,
↪  -Chief_complain, -Mental, -Pain)

## create a complete case cohort -> drop anyone with any missings as all variables kept
↪  in the dataset will be used for machine learning approach
ED_complete <- na.omit(emergency)
#creating a validation set with 20% of data
smp_size <- floor(0.80 * nrow(ED_complete))

## set the seed
set.seed(2404)
train_ind <- sample(seq_len(nrow(ED_complete)), size = smp_size)

ED.train <- ED_complete[train_ind, ]
ED.test <- ED_complete[-train_ind, ]

#create a table displaying characteristics by outcome including the preselected variables
library(table1)

ED_complete_table <- ED_complete

ED_complete_table$admission <-
  factor(ED_complete_table$admission, levels=c(1,0), labels=c("Inpatient admission",
  ↪  "Discharge home"))
ED_complete_table$KTAS_RN <-
  factor(ED_complete_table$KTAS_RN, levels=c(1,2,3,4,5), labels=c("Resuscitation",
  ↪  "Emergent", "Urgent", "Less urgent", "Non-urgent"))
ED_complete_table$Sex <-
  factor(ED_complete_table$Sex, levels=c(1,2), labels=c("Female", "Male"))
ED_complete_table$shock <-
  factor(ED_complete_table$shock, levels=c(1,0), labels=c("Shock", "No"))
ED_complete_table$hyperventilation <-
  factor(ED_complete_table$hyperventilation, levels=c(1,0), labels=c("Hyperventilation",
  ↪  "No"))
```

```r
ED_complete_table$hypertherm <-
  factor(ED_complete_table$hypertherm, levels=c(1,0), labels=c("Fever", "No"))
ED_complete_table$injury <-
  factor(ED_complete_table$injury, levels=c(1,0), labels=c("Injured", "No injury"))
ED_complete_table$arrival <-
  factor(ED_complete_table$arrival, levels=c(1,2,3,4), labels=c("Walking", "Ambulance",
  ↪  "Private vehicle", "other"))
ED_complete_table$mental_status <-
  factor(ED_complete_table$mental_status, levels=c(1, 2, 3, 4), labels=c("Alert", "Verbal
  ↪  response", "Pain response", "Unresponsive"))

label(ED_complete_table$NRS_pain)       <- "Pain rating (0-10)"
label(ED_complete_table$KTAS_RN)       <- "Triage"
label(ED_complete_table$sbp)       <- "Systolic blood pressure, mmHg"
label(ED_complete_table$dbp)     <- "Diastolic blood pressure, mmHg"
label(ED_complete_table$shock_index) <- "Shock index"
label(ED_complete_table$shock) <- "Shock (HR/SBP > 1)"
label(ED_complete_table$hr) <- "Heart rate, bpm"
label(ED_complete_table$resp) <- "Respiratory rate (RR), per minute"
label(ED_complete_table$hyperventilation) <- "Hyperventilation (Respiratory rate >
↪  25/min)"
label(ED_complete_table$temp) <- "Body tmeperature, celsius"
label(ED_complete_table$hypertherm) <- "Fever (Temp > 37.5)"
label(ED_complete_table$arrival) <- "Mode of arrival"
label(ED_complete_table$injury) <- "Injured"
label(ED_complete_table$mental_status) <- "Mental status"

#table1(~ Sex + Age + NRS_pain + KTAS_RN + shock + hyperventilation + hypertherm +
↪  arrival + injury + mental_status | admission, data=ED_complete_table,
↪  overall="Total", caption="Patient characteristics by discharge disposition")
#training the clinician model: sex, age, NRS_pain, KTAS_RN, arrival, injury, shock,
↪  hyperventilation, mental status, hyperthermia/fever
library(caret)

#stepwise forward regression with AIC as criterion
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
Fitall.tr <- glm(admission ~ as.factor(Sex) + Age + NRS_pain + KTAS_RN +
↪  as.factor(arrival) + injury + shock + hyperventilation + as.factor(mental_status) +
↪  hypertherm, family="binomial", data= ED.train)
Fitstart <- glm(admission ~ 1, family="binomial", data= ED.train)
set.seed(2024)
m_clin <- step(Fitstart, scope=formula(Fitall.tr), direction="forward", k=2, trace=0)
summary(m_clin)
```

```
##
```

```
## Call:
## glm(formula = admission ~ KTAS_RN + Age + as.factor(arrival) +
##     hypertherm + injury + shock, family = "binomial", data = ED.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0372  -0.8024  -0.4765   0.9187   2.4467
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.084733   0.564747  -0.150  0.88074
## KTAS_RN             -0.768416   0.102214  -7.518 5.57e-14 ***
## Age                  0.018588   0.004219   4.406 1.05e-05 ***
## as.factor(arrival)2  1.299313   0.397141   3.272  0.00107 **
## as.factor(arrival)3  0.557638   0.389871   1.430  0.15263
## as.factor(arrival)4 -0.260732   0.878573  -0.297  0.76664
## hypertherm           1.055091   0.339566   3.107  0.00189 **
## injury              -0.668763   0.242537  -2.757  0.00583 **
## shock                0.845204   0.389749   2.169  0.03011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1229.3  on 981  degrees of freedom
## Residual deviance: 1014.8  on 973  degrees of freedom
## AIC: 1032.8
##
## Number of Fisher Scoring iterations: 4
```

```
#apply to validation set
ED.train$phat_clin <- predict(m_clin, type="response", newdata=ED.train)
ED.test$phat_clin <- predict(m_clin, type="response", newdata=ED.test)

#Calibration Plot -training set
##create risk deciles on predicted risk
cuts <- quantile(ED.train$phat_clin, prob=c(.1,.2,.3,.4,.5,.6,.7,.8,.9), na.rm=T)
ED.train$risk_decile <-cut(ED.train$phat_clin, breaks=c(0, cuts, 1))
dec<-c(1:10) #for plot
#observed proportion of difficult hearing in risk deciles
t1.train<-table(ED.train$risk_decile, ED.train$admission)
#addmargins(t1.train)
t2.train <- prop.table(t1.train, 1)
obs.train <- t2.train[,2] #for plot
#mean predicted risk in risk deciles
deciles.train <- ED.train %>% group_by(risk_decile) %>% summarise(mean=mean(phat_clin))
pred.train <- deciles.train$mean #for plot
cali_train<-data.frame(dec, obs.train, pred.train) # for plot
p1 <- ggplot(cali_train, aes(x=obs.train, y=pred.train)) + geom_point(size=2) +
↪  xlab("Observed Inpatient Admission Risk") + ylab("Predicted Risk") + ggtitle("Fig.2 -
↪  Reliability Plot - Training Set") + theme(plot.title = element_text(hjust = 0.5)) +
↪  geom_abline(intercept = 0, slope = 1, color="grey")

#Calibration Plot -validation set
```

```
##create risk deciles on predicted risk
cuts <- quantile(ED.test$phat_clin, prob=c(.1,.2,.3,.4,.5,.6,.7,.8,.9), na.rm=T)
ED.test$risk_decile <-cut(ED.test$phat_clin, breaks=c(0, cuts, 1))
dec<-c(1:10) #for plot
#observed proportion of difficult hearing in risk deciles
t1.test<-table(ED.test$risk_decile, ED.test$admission)
#addmargins(t1.test)
t2.test <- prop.table(t1.test, 1)
obs.test <- t2.test[,2] #for plot
#mean predicted risk in risk deciles
deciles.test <- ED.test %>% group_by(risk_decile) %>% summarise(mean=mean(phat_clin))
pred.test <- deciles.test$mean #for plot
cali_test<-data.frame(dec, obs.test, pred.test) # for plot
p2 <- ggplot(cali_test, aes(x=obs.test, y=pred.test)) + geom_point(size=2) +
↪  xlab("Observed Inpatient Admission Risk") + ylab("Predicted Risk") + ggtitle("Fig.3 -
↪  Reliability Plot - Training Set") + theme(plot.title = element_text(hjust = 0.5)) +
↪  geom_abline(intercept = 0, slope = 1, color="grey")
#check sensitivity and specificity for different cutoffs of predicted risk in training
↪  set
ED.train$admission <- as.factor(ED.train$admission)
## 0.3
ED.train$yhat_clin03 <- as.factor(ifelse(ED.train$phat_clin > 0.3, 1, 0))
cm_clinical_train03 <- confusionMatrix(ED.train$yhat_clin03, ED.train$admission,
↪  positive="1")
## 0.4
ED.train$yhat_clin04 <- as.factor(ifelse(ED.train$phat_clin > 0.4, 1, 0))
cm_clinical_train04 <- confusionMatrix(ED.train$yhat_clin04, ED.train$admission,
↪  positive="1")
## 0.5
ED.train$yhat_clin05 <- as.factor(ifelse(ED.train$phat_clin > 0.5, 1, 0))
cm_clinical_train05 <- confusionMatrix(ED.train$yhat_clin05, ED.train$admission,
↪  positive="1")
##0.6
ED.train$yhat_clin06 <- as.factor(ifelse(ED.train$phat_clin > 0.6, 1, 0))
cm_clinical_train06 <- confusionMatrix(ED.train$yhat_clin06, ED.train$admission,
↪  positive="1")
##0.7
ED.train$yhat_clin07 <- as.factor(ifelse(ED.train$phat_clin > 0.7, 1, 0))
cm_clinical_train07 <- confusionMatrix(ED.train$yhat_clin07, ED.train$admission,
↪  positive="1")
##0.8
ED.train$yhat_clin08 <- as.factor(ifelse(ED.train$phat_clin > 0.8, 1, 0))
cm_clinical_train08 <- confusionMatrix(ED.train$yhat_clin08, ED.train$admission,
↪  positive="1")

#performance parameters for different cutoffs
rownames <- c("0.3","0.4","0.5", "0.6", "0.7", "0.8")
Specificity <- c(cm_clinical_train03$byClass["Specificity"],
↪  cm_clinical_train04$byClass["Specificity"],
↪  cm_clinical_train05$byClass["Specificity"],
↪  cm_clinical_train06$byClass["Specificity"],
↪  cm_clinical_train07$byClass["Specificity"],
↪  cm_clinical_train08$byClass["Specificity"])
```

```
Sensitivity <- c(cm_clinical_train03$byClass["Sensitivity"],
↪  cm_clinical_train04$byClass["Sensitivity"],
↪  cm_clinical_train05$byClass["Sensitivity"],
↪  cm_clinical_train06$byClass["Sensitivity"],
↪  cm_clinical_train07$byClass["Sensitivity"],
↪  cm_clinical_train08$byClass["Sensitivity"])
Accuracy <- c(cm_clinical_train03$overall["Accuracy"],
↪  cm_clinical_train04$overall["Accuracy"], cm_clinical_train05$overall["Accuracy"],
↪  cm_clinical_train06$overall["Accuracy"], cm_clinical_train07$overall["Accuracy"],
↪  cm_clinical_train08$overall["Accuracy"])
Table_cutoff <- data.frame(row.names=rownames, Sensitivity, Specificity, Accuracy)
# best trade-off between sensitvity and secificity: cutoff: 0.3
#knitr::kable(Table_cutoff, caption = "Different cutoffs of predicted inpatient admission
↪  risk")

#Accuracy in test set
ED.test$yhat_clin03 <- as.factor(ifelse(ED.test$phat_clin > 0.3, 1, 0))
ED.test$admission <- as.factor(ED.test$admission)
cm_clinical_test <- confusionMatrix(ED.test$yhat_clin03, ED.test$admission, positive="1")
#random forest
detach("package:MASS", unload = TRUE)
```

```
## Warning: 'MASS' namespace cannot be unloaded:
##   namespace 'MASS' is imported by 'ipred' so cannot be unloaded
```

```
library(randomForest)

y_train <- ED.train$admission
x_train <- ED.train |> select(Sex, Age, NRS_pain, KTAS_RN, sbp, dbp, hr, resp, temp,
↪  arrival, injury, shock_index, mental_status)

#tuning nodesize
set.seed(2404)
nodesize <- seq(1, 50, 5)
acc <- sapply(nodesize, function(ns){
  train(data.frame(x_train), factor(y_train), method = "rf",
              tuneGrid = data.frame(mtry = 5),
              nodesize = ns)$results$Accuracy
})
#qplot(nodesize, acc,  main = "Fig. 4 - Tune of node size", xlab = "Node size", ylab =
↪  "Accuracy",)


#fit random forest model
set.seed(2333)
fit_rf <- randomForest(data.frame(x_train), factor(y_train),
                       mtry = 5, nodesize = nodesize[which.max(acc)])

#random forest model performance in internal validation set
set.seed(2134)
ED.train$yhat_ML <- predict(fit_rf, type="response", newdata=ED.train)
ED.test$yhat_ML <- predict(fit_rf, type="response", newdata=ED.test)
#Clinical model performance in test set
```

```
ED.test$yhat_clin03 <- as.factor(ifelse(ED.test$phat_clin > 0.3, 1, 0))
ED.test$admission <- as.factor(ED.test$admission)
cm_clinical_test <- confusionMatrix(ED.test$yhat_clin03, ED.test$admission, positive="1")

#Machine Learning model performance in test set
ED.test$yhat_ML <- as.factor(ED.test$yhat_ML)
cm_ML_forest_test <- confusionMatrix(factor(ED.test$yhat_ML), factor(ED.test$admission),
↪  positive="1")

table3 <- cm_clinical_test$table
rownames(table3) = c("predicted discharge", "predicted admission")
#knitr::kable(table3, caption = "Clinical model", col.names = c("Discharge home",
↪  "Inpatient admission"))

table4 <- cm_ML_forest_test$table
rownames(table4) = c("predicted discharge", "predicted admission")
#knitr::kable(table4, caption = "Random forest model", col.names = c("Discharge home",
↪  "Inpatient admission"))
#Clinical model performance in test set
ED.test$yhat_clin03 <- as.factor(ifelse(ED.test$phat_clin > 0.3, 1, 0))
ED.test$admission <- as.factor(ED.test$admission)
cm_clinical_test <- confusionMatrix(ED.test$yhat_clin03, ED.test$admission, positive="1")

#Machine Learning model performance in test set
ED.test$yhat_ML <- as.factor(ED.test$yhat_ML)
cm_ML_forest_test <- confusionMatrix(factor(ED.test$yhat_ML), factor(ED.test$admission),
↪  positive="1")

rownames <- c("Specificity","Sensitivity","Accuracy")
Clinical_model <- c(cm_clinical_test$byClass["Specificity"],
↪  cm_clinical_test$byClass["Sensitivity"], cm_clinical_test$overall["Accuracy"])
Random_forest_model <- c(cm_ML_forest_test$byClass["Specificity"],
↪  cm_ML_forest_test$byClass["Sensitivity"], cm_ML_forest_test$overall["Accuracy"])

Table5 <- data.frame(row.names=rownames, Clinical_model, Random_forest_model)
#knitr::kable(Table5,  caption = "Performance of both models in validation set",
↪  col.names = c("Clinical model", "Random forest"))
```