HS 650
Data Science and Predictive Analytics (UMich HS650)

Ashruti Tuteja | MHI | Fall 2023

Analysis on MIMIC II data, Guided by Prof. Ivo Dinov

Contents

Objective

I am planning to explore MIMIC data, to understand the trend and prevalence of chronic diseases in the presence of various physiological conditions amongst ICU admitted patients.
My aim is to contribute valuable information that can guide healthcare professionals in refining their prognostic assessments and, ultimately, enhancing patient outcomes.

Abstract

This project focuses on understanding and predicting the prognosis of Congestive Heart Failure (CHF) to improve patient care. The background research reveals a typical survival period of less than a year for CHF patients.
The project aims to provide valuable insights into the multifaceted aspects of CHF prognosis, aiding healthcare professionals in refining their assessments and optimizing care plans.

INTRODUCTION

- About Dataset
- Data Dictionary
- Data Glossary
- Data Overview
- Strength & Weakness of Dataset
- Health related Question
- Data Preparation

About Dataset
MIMIC

- The dataset is derived from MIMIC-II, the publicly-accessible critical care database.
- It contains a summary of clinical data and outcomes for 1,776 patients.
- The dataset in question was used throughout Chapter 16 (Data Analysis) by Raffa J. et al. **to investigate the effectiveness of indwelling arterial catheters in hemodynamically stable patients with respiratory failure for mortality outcomes.**

Data Dictionary

aline_flg	IAC used (binary, 1 = year, 0 = no)	censor_flg	censored or death (binary: 0 = death, 1 = censored)
icu_los_day	length of stay in ICU (days, numeric)	sepsis_flg	sepsis present (binary: 0 = no, 1 = yes -- absent (0) for all)
hospital_los_day	length of stay in hospital (days, numeric)	chf_flg	Congestive heart failure (binary: 0 = no, 1 = yes)
age	age at baseline (years, numeric)	afib_flg	Atrial fibrillation (binary: 0 = no, 1 = yes)
gender_num	patient gender (1 = male; 0=female)	renal_flg	Chronic renal disease (binary: 0 = no, 1 = yes)
weight_first	first weight, (kg, numeric)	liver_flg	Liver Disease (binary: 0 = no, 1 = yes)
bmi	patient BMI, (numeric)	copd_flg	Chronic obstructive pulmonary disease (binary: 0 = no, 1 = yes)
sapsi_first	first SAPS I score (numeric)	cad_flg	Coronary artery disease (binary: 0 = no, 1 = yes)
sofa_first	first SOFA score (numeric)	stroke_flg	Stroke (binary: 0 = no, 1 = yes)
service_unit	type of service unit (character: FICU, MICU, SICU)	mal_flg	Malignancy (binary: 0 = no, 1 = yes)
mort_day_censored	day post ICU admission of censoring or death (days, numeric)	hgb_first	first Hemoglobin (g/dL, numeric)

Data Dictionary

aline_flg	IAC used (binary, 1 = year, 0 = no)	censor_flg	censored or death (binary: 0 = death, 1 = censored)
service_num	service as a numeric (binary: 0 = MICU or FICU, 1 = SICU)	resp_flg	Respiratory disease (non-COPD) (binary: 0 = no, 1 = yes)
day_icu_intime	day of week of ICU admission (character)	map_1st	Mean arterial pressure (mmHg, numeric)
day_icu_intime_num	day of week of ICU admission (numeric, corresponds with day_icu_intime)	hr_1st	Heart Rate (numeric)
hour_icu_intime	hour of ICU admission (numeric, hour of admission using 24hr clock)	temp_1st	Temperature (F, numeric)
hosp_exp_flg	death in hospital (binary: 1 = yes, 0 = no)	spo2_1st	S_pO_2 (%, numeric)
icu_exp_flg	death in ICU (binary: 1 = yes, 0 = no)	abg_count	arterial blood gas count (number of tests, numeric)
day_28_flg	death within 28 days (binary: 1 = yes, 0 = no)	wbc_first	first White blood cell count (K/uL, numeric)
mort_day_censored	day post ICU admission of censoring or death (days, numeric)	hgb_first	first Hemoglobin (g/dL, numeric)

Data Dictionary

platelet_first	first Platelets (K/u, numericL)
sodium_first	first Sodium (mEq/L, numeric)
potassium_first	first Potassium (mEq/L, numeric)
tco2_first	first Bicarbonate (mEq/L, numeric)
chloride_first	first Chloride (mEq/L, numeric)
bun_first	first Blood urea nitrogen (mg/dL, numeric)
creatinine_first	first Creatinine (mg/dL, numeric)
po2_first	first PaO_2 (mmHg, numeric)
iv_day_1	input fluids by IV on day 1 (mL, numeric)
pco2_first	first PaCO_2 (mmHg, numeric)

Data Overview

A glimpse of the dataframe

Data Sample

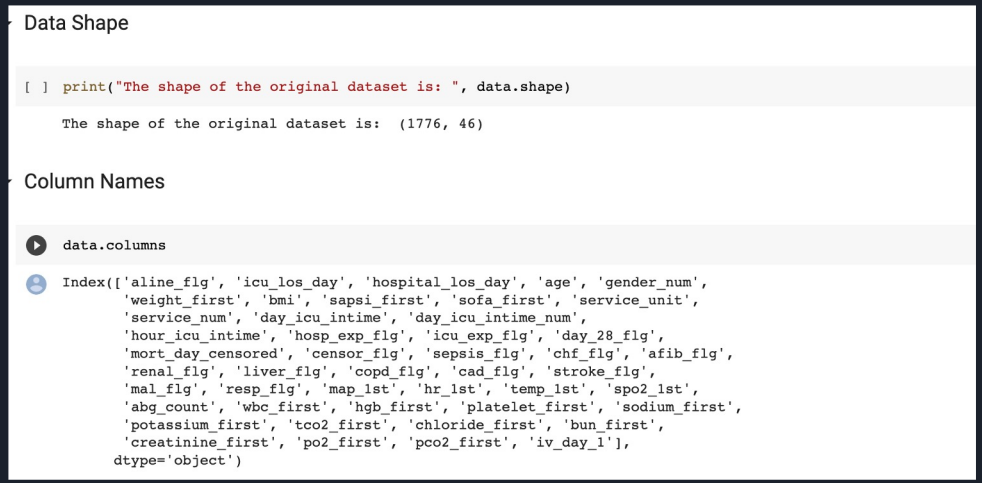
```
data.head(5)
```

	aline_flg	icu_los_day	hospital_los_day	age	gender_num	weight_first	bmi	sapsi_first	sofa_first	service_unit	...	platelet_first	sodium_firs
0	1	7.63	13	72.36841	1.0	75.0	29.912791	15.0	9.0	SICU	...	354.0	138
1	0	1.14	1	64.92076	0.0	55.0	20.121312	NaN	5.0	MICU	...	NaN	Na
2	0	2.86	5	36.50000	0.0	70.0	27.118272	16.0	5.0	MICU	...	295.0	144
3	1	0.58	3	44.49191	0.0	NaN	NaN	21.0	7.0	SICU	...	262.0	139
4	1	1.75	5	23.74217	1.0	95.2	28.464563	18.0	7.0	SICU	...	22.0	146

5 rows × 46 columns

46 columns:
Each column is a physiological factor for each patient amongst 1776 patients in the dataset.

Size of the Dataset



Data Shape

```
[ ] print("The shape of the original dataset is: ", data.shape)
```

```
The shape of the original dataset is:  (1776, 46)
```

Column Names

```
data.columns
```

```
Index(['aline_flg', 'icu_los_day', 'hospital_los_day', 'age', 'gender_num',
       'weight_first', 'bmi', 'sapsi_first', 'sofa_first', 'service_unit',
       'service_num', 'day_icu_intime', 'day_icu_intime_num',
       'hour_icu_intime', 'hosp_exp_flg', 'icu_exp_flg', 'day_28_flg',
       'mort_day_censored', 'censor_flg', 'sepsis_flg', 'chf_flg', 'afib_flg',
       'renal_flg', 'liver_flg', 'copd_flg', 'cad_flg', 'stroke_flg',
       'mal_flg', 'resp_flg', 'map_1st', 'hr_1st', 'temp_1st', 'spo2_1st',
       'abg_count', 'wbc_first', 'hgb_first', 'platelet_first', 'sodium_first',
       'potassium_first', 'tco2_first', 'chloride_first', 'bun_first',
       'creatinine_first', 'po2_first', 'pco2_first', 'iv_day_1'],
      dtype='object')
```

1776 rows (each patient)
and
46 columns (physiological and anatomical factors)

Understanding the data types of the columns in the dataframe

- Most of the columns seem to contain either integer or float values.
- **Continuous data values** make it easier for analysis.

```
data.dtypes
```

```
aline_flg               int64
icu_los_day           float64
hospital_los_day        int64
age                   float64
gender_num            float64
weight_first          float64
bmi                   float64
sapsi_first           float64
sofa_first            float64
service_unit           object
service_num             int64
day_icu_intime         object
day_icu_intime_num      int64
hour_icu_intime         int64
hosp_exp_flg            int64
icu_exp_flg             int64
day_28_flg              int64
mort_day_censored     float64
censor_flg              int64
sepsis_flg              int64
chf_flg                 int64
afib_flg                int64
renal_flg               int64
liver_flg               int64
copd_flg                int64
cad_flg                 int64
stroke_flg              int64
mal_flg                 int64
resp_flg                int64
map_1st               float64
hr_1st                  int64
temp_1st              float64
spo2_1st                int64
abg_count               int64
wbc_first             float64
hgb_first             float64
platelet_first        float64
sodium_first          float64
potassium_first       float64
tco2_first            float64
chloride_first        float64
bun_first             float64
creatinine_first      float64
po2_first             float64
pco2_first            float64
iv_day_1              float64
```

Statistics - A description of the dataset

Basic Statistics

```
data.describe()
```

	aline_flg	icu_los_day	hospital_los_day	age	gender_num	weight_first	bmi	sapsi_first	sofa_first	service_num	...	platelet_first	so
count	1776.000000	1776.000000	1776.000000	1776.000000	1775.000000	1666.000000	1310.000000	1691.000000	1770.000000	1776.000000	...	1768.000000	
mean	0.554054	3.346498	8.110923	54.379660	0.577465	80.075948	27.827316	14.136606	5.820904	0.552928	...	246.083145	
std	0.497210	3.356261	8.157159	21.062854	0.494102	22.490516	8.210074	4.114302	2.334666	0.497331	...	99.865469	
min	0.000000	0.500000	1.000000	15.180230	0.000000	30.000000	12.784877	3.000000	0.000000	0.000000	...	7.000000	
25%	0.000000	1.370000	3.000000	38.247318	0.000000	65.400000	22.617307	11.000000	4.000000	0.000000	...	182.000000	
50%	1.000000	2.185000	6.000000	53.678585	1.000000	77.000000	26.324846	14.000000	6.000000	1.000000	...	239.000000	
75%	1.000000	4.002500	10.000000	72.762992	1.000000	90.000000	30.796551	17.000000	7.000000	1.000000	...	297.000000	
max	1.000000	28.240000	112.000000	99.110950	1.000000	257.600000	98.797134	32.000000	17.000000	1.000000	...	988.000000	

8 rows × 44 columns

- **Count:** Some values are missing from various columns, as count varies in each columns
- **Mean:** age is 54, SAPS score is 14, SOFA score 5.8
- **Min & max:** reveals the extreme values for various physiological factors, like for platelet 7 (min), 988 (max)
- **The quantiles:** if plotted will boxplot will help us determining outliers for each of the physiological factors.

Data Glossary

SAPS score: Estimates the probability of mortality for ICU patients on admission.

SOFA score: The Sequential Organ Failure Assessment (SOFA) score is a scoring system that assesses the performance of several organ systems in the body (neurologic, blood, liver, kidney, and blood pressure/hemodynamics).

REFERENCE:
https://www.mdcalc.com/calc/10403/simplified-acute-physiology-score-saps-3#:~:text=Estimates%20the%20probability%20of%20mortality%20for%20ICU%20patients%20on%20admission.&text=The%20SAPS%203%20Score%20predicts,physiologic%20derangement%20upon%20ICU%20admission.

Strengths & Weaknesses of the Dataset

Strengths:

- Reliable dataset – MIMIC is reputed and open data source for medical data
- Extensive – incorporating many attributes (~46 columns)
- Meaningfulness - Data dictionary is self-explanatory
- Completeness – Less missing values

Weakness of your dataset

- Less instances – 1776 rows depicting 1776 patients
- Validity – source is unknown

HEALTH RELATED QUESTION

To determine the mortality rate among individuals with SAPS scores ranging from 5 to 15, stratified by age groups, among patients admitted to the ICU, considering both those with and without heart disease.

Population

MIMIC data, represents a population with a SAPS score between '5-15' identifying risk of mortality of the patient in the ICU based on the severity of the disease condition.

Comparison

Comparing patients with congestive heart failure and without congestive heart failure

Intervention or Exposure Variable

Congestive heart failure (chf_flg) which is a binary variable where 0 indicates the negative outcome and 1 indicates the positive outcome.

Outcome Variable

"censor_flg" is the outcome variable that indicates 'censored (0) or death (1)'.

It is medically observed that having chronic kidney disease (CKD) implies a greater chance of having heart disease. CKD can cause heart disease, and heart disease can cause CKD. **Renal disease is a confounder** that can affect or impact both the exposure variable of heart disease and the outcome variable of mortality.

Data Preparation

```r
knitr::opts_chunk$set(echo=T, warning = F, message = F)
```

```r
#load packages:
library(tidyverse);library(igraph);library(plotly)
```

```r
#Required Package
library(tidyverse)
library(dplyr)
library(skimr)
library(ggplot2)
library(plotly)
```

```r
patient_data = read_csv("full_cohort_data.csv")
```

After initial data importing and setup, I sorted the data, cleaned it, and checked for missingness using the `naniar` package. I performed exploratory data analysis on various variables (both categorical and continuous) to hypothesize the question.

Subset the data

```
```{r, echo=FALSE}
#Select required columns
filter_SAPS = patient_data %>%
  select(age,gender_num,sapsi_first,chf_flg,censor_flg,renal_flg,wbc_first,hgb_first,icu_los_day,hospital_los_day)
%>%
  filter(sapsi_first >=15)

without_conf = filter_SAPS %>%
  mutate(agegroup = if_else(age >60, 'Above 60', 'Below 60')) %>%
  mutate(heart_failure = if_else(chf_flg ==0, 'Without heart disease', 'With heart disease')) %>%
  mutate(mortality = if_else(censor_flg == 0, 'dead', 'alive'))
```
```

Finding Missingness

```
# 4.1 Finding Missigness
``{r, echo=FALSE}
ibrary(naniar)
_var_miss(without_conf)
``
```

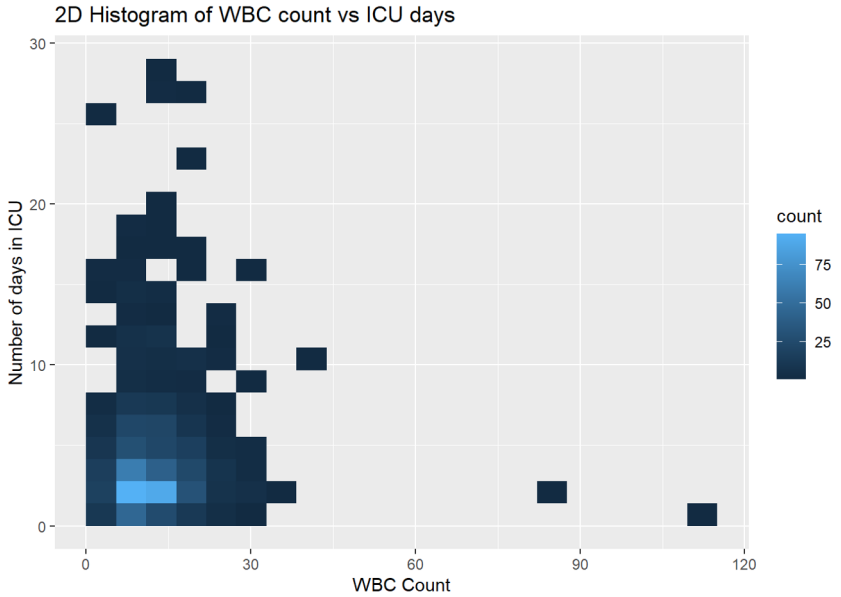
- No missingness - no imputations required

ANALYSIS, APPROACH, INTERPRETATION

- 2-D Histogram
- Correlational Plot
- 3-D Scatter Plot
- K-means Clustering
- PCA
- Feature selection
- Modeling
  - Rpart
  - K-Nearest Neighbors
  - Logistic Regression

2-D Histogram

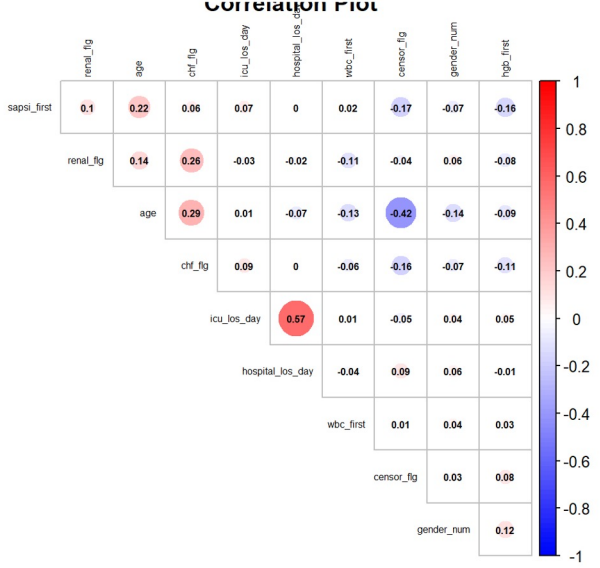
- Elevated White Blood Cell (WBC) counts, associated with immunity, are linked to shorter ICU stays.
- Notably, patients with chronic diseases, those with an initial WBC count between 0 and 30 on the first day of ICU admission tended to have an extended stay in the ICU compared to individuals with an initial WBC count exceeding 30 on the first day of admission.



Approach:
Using ggplot function on prepared dataset to show trend between ICU Length of Stay and WBC count.

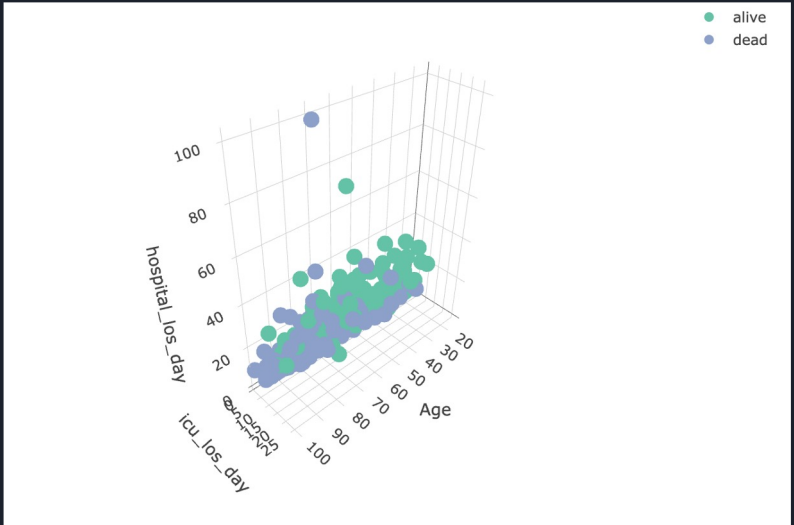
Correlational Plot

- The results revealed a significant correlation between the length of ICU stays and the duration of hospitalization, indicating a strong association between the two variables.



Approach:
Used the combination of 10 physiological factors to find the correlation between them.
Code:
corr_conf <- cor(without_conf[1:10])

3D Scatter Plot



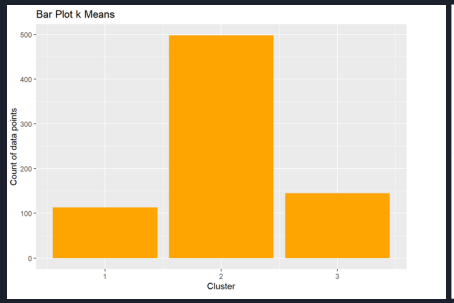
Utilized the ‘plotly’ library to create a 3D scatter plot visualizing age, ICU length of stay, and hospital length of stay, with marker colors representing mortality status.

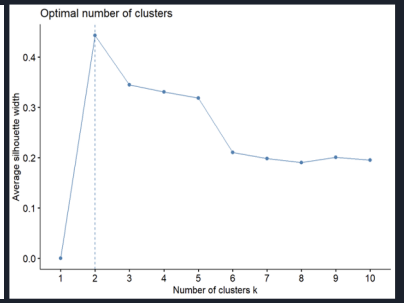
- Findings from the plot indicate that individuals aged over 80 with chronic diseases tend to have shorter stays in both the hospital and ICU, coupled with a higher likelihood of mortality. On the other hand, patients in the 20 to 50 age group admitted to the ICU exhibit varying lengths of stay, influencing their chances of survival.

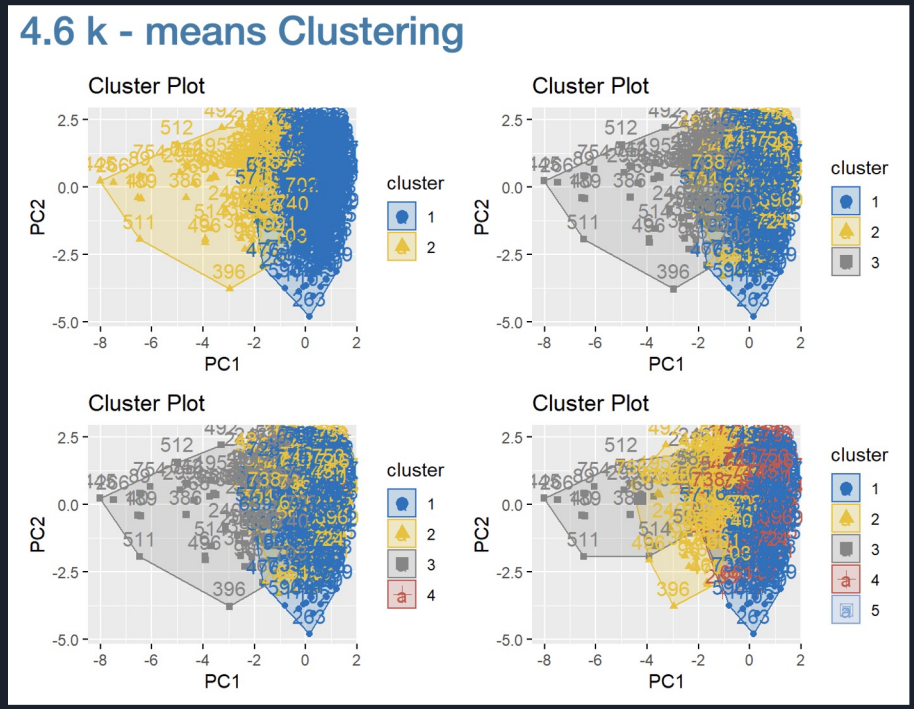
K-Means Clustering

Notably, it suggests that 2 or 3 clusters effectively capture the representation of the data.

Further insights from the silhouette plot indicate that the optimal number of clusters representing the data is 2, reinforcing the conclusion drawn from the bar plot.





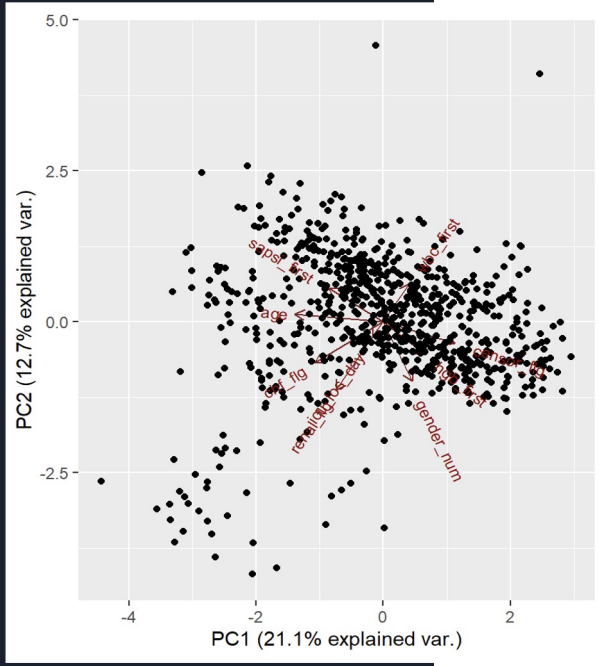


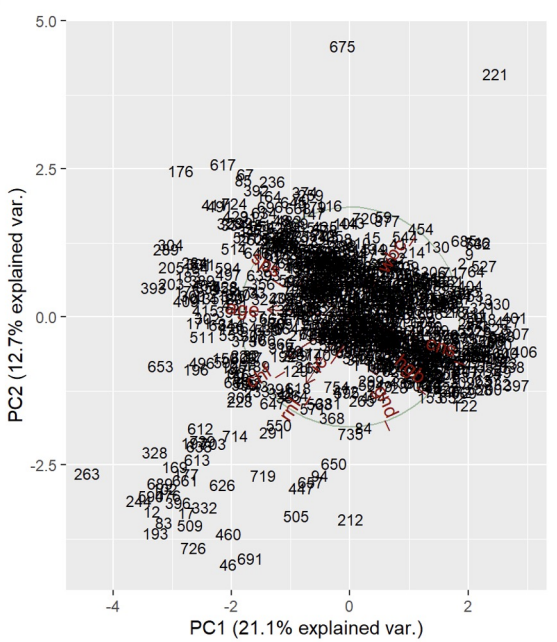
Here clustering data has used centers 2,3,4,5

```
k2 <- kmeans(Clustering_Data1, centers = 2, nstart = 25)
k3 <- kmeans(Clustering_Data1, centers = 3, nstart = 25)
k4 <- kmeans(Clustering_Data1, centers = 4, nstart = 25)
k5 <- kmeans(Clustering_Data1, centers = 5, nstart = 25)
```

PCA

A majority of data points exhibit a first principal component (PC) score of 0, while a few data points deviate with a first PC score of -2.



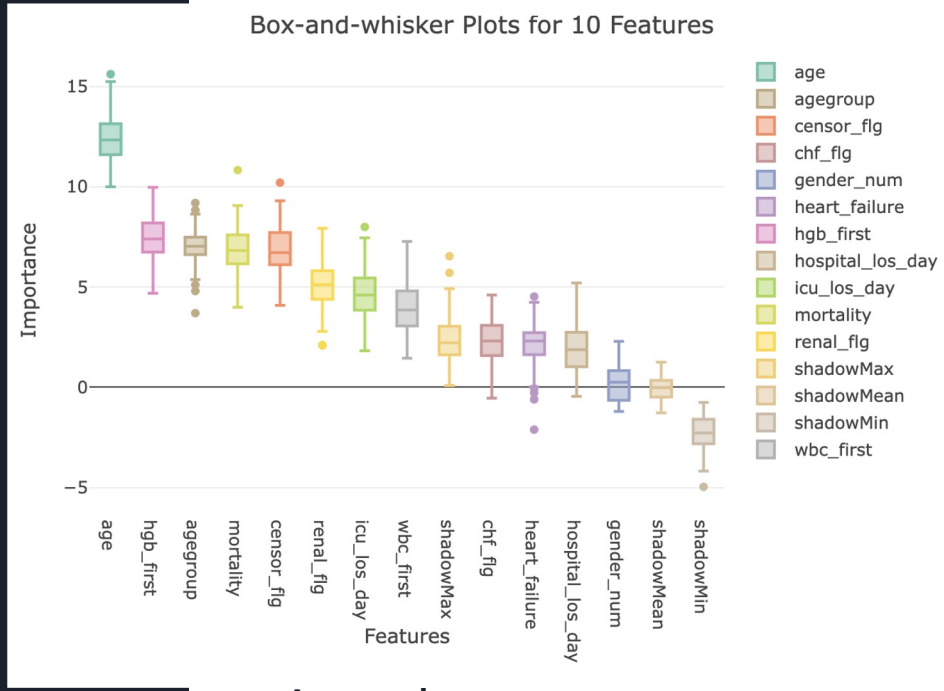


Approach

The code performs Principal Component Analysis (PCA) on the subset dataset and generates a customized biplot using the 'ggbiplot' library in R. The biplot visually represents the relationships between data points and variables in reduced-dimensional space, with additional features such as ellipses indicating confidence intervals, variable axes, and customizable point shapes, labels, and colors based on grouping variables.

Feature Selection

The whiskerplot enables a clear view of the importance score distribution for each variable, aiding in the identification of tentative and crucial variables. This visualization assists in assessing the range of importance scores within a single variable and may guide decisions on whether to retain or discard tentative features.

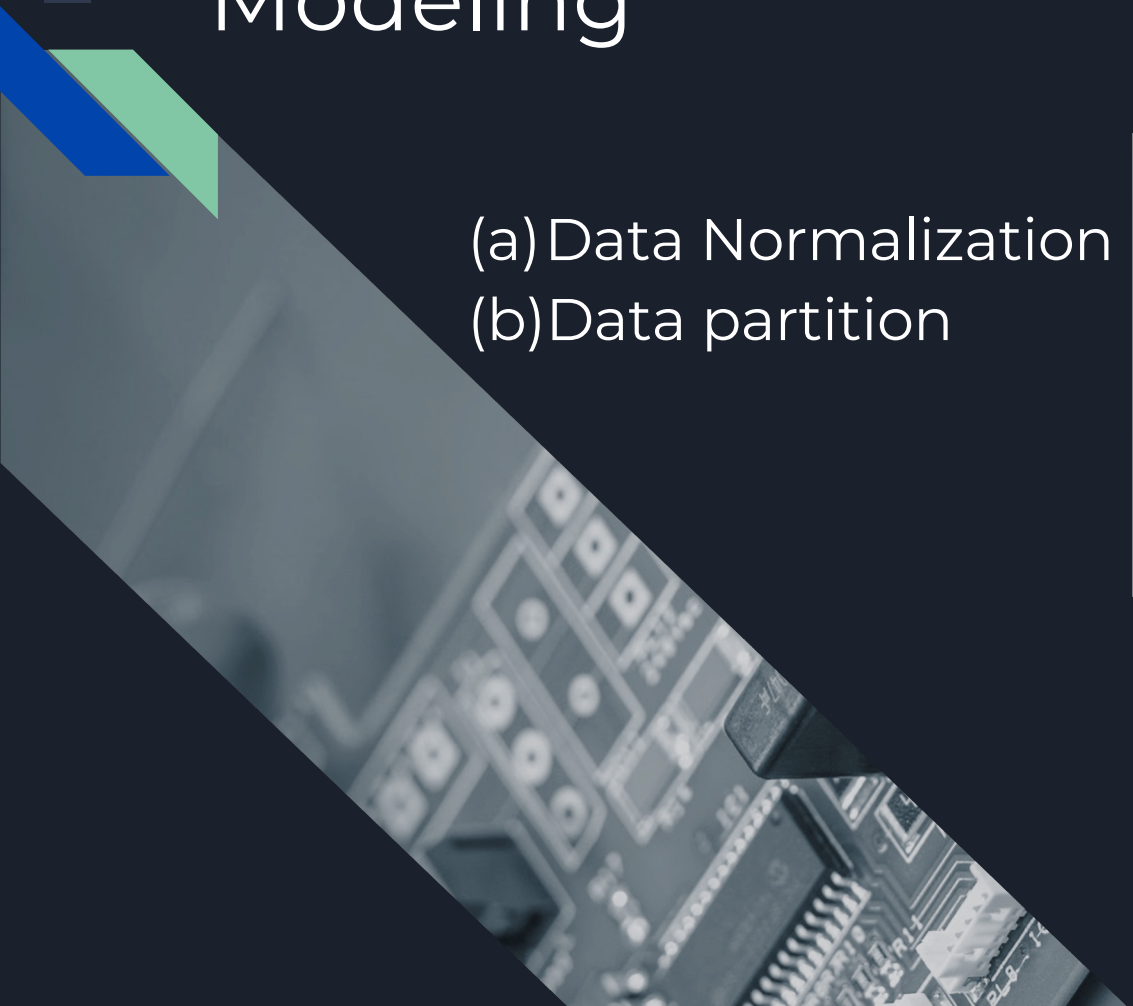


Approach

After removing features with infinite importance, I organized the remaining significant features by their median importance. Using plotly, I illustrated these features through boxplots, showcasing their median, quartiles, and minimum and maximum values.

Modeling

(a) Data Normalization
(b) Data partition



```
### 4.9.1 Normalizing Data
```{r, echo=FALSE}
normalize<-function(x){
# be careful, the denominator may be trivial!
return((x-min(x))/(max(x)-min(x)))
}

without_conf_n<-as.data.frame(lapply(without_conf[,-c(11:13)], normalize))

```

 - Checking if normalization worked
```{r}
summary(without_conf_n$sapsi_first)
```

4.9.2 Partition data

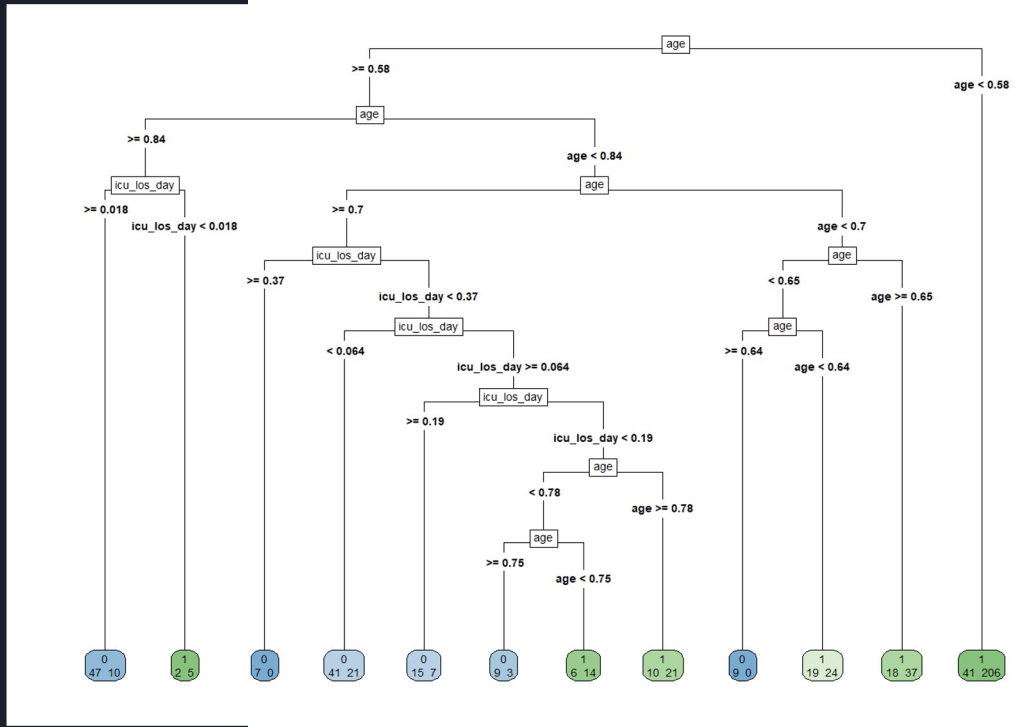
```{r echo=TRUE}
set.seed(111)
ind <- sample(2, nrow(without_conf_n),
              replace = TRUE,
              prob = c(0.75, 0.25))
training <- without_conf_n[ind==1,]
testing <- without_conf_n[ind==2,]
```
```

Approach

This R code defines a normalization function and applies it to all columns (excluding the last three) of the dataset without_conf, creating a new normalized dataset without_conf_n. The success of normalization is checked by summarizing a specific variable (sapsi_first).

R Part

Notably, Age and icu_los_day stood out as the most important variables for the recursive partitioning process. The plot indicates that for individuals aged above 58, a predominant outcome is "0," implying a higher incidence of patient deaths in the ICU for those aged 58 or older.



Approach

The R code uses the rpart package to build a classification model (without_conf_model_rpart) predicting the censor_flg variable based on selected features. The decision tree is visualized using rpart.plot to interpret the model's decision-making structure.

K-Nearest Neighbor

Experimenting with different values of k to train the model revealed that the highest accuracy achieved was 38% for k=28, indicating a relatively low accuracy level. In response, we proceeded to train the dataset using the Logistic Regression algorithm.

```r
```{r, echo=FALSE}
# Model Evaluation - Choosing K
# Calculate out of Sample error
misClassError <- mean(knn_model != test_class$sapsi_first)
print(paste('Accuracy =', 1-misClassError))
```
```

4.9.4 K Nearest Neighbours

```
## [1] "Accuracy = 0.374338624338624"
```

```r
```{r, echo=FALSE}
# Splitting data into train and test data
split_data <- sample.split(without_conf_n, SplitRatio = 0.8)
train_class <- subset(without_conf_n, split = "TRUE")
test_class <- subset(without_conf_n, split = "FALSE")

# Feature Scaling
scale_train <- scale(train_class[, 1:10])
scale_test <- scale(test_class[, 1:10])

# Fitting KNN Model
# to training dataset
knn_model <- knn(train = scale_train,
                 test = scale_test,
                 cl = train_class$sapsi_first,
                 k = 28)
```
```

```r
```{r, echo=FALSE}
cm <- table(test_class$sapsi_first, knn_model)
#cm
```
```

The code employs the e1071 and class packages to implement the K Nearest Neighbors (KNN) algorithm on the subset dataset.. It involves splitting the data into training and testing sets, scaling features, and fitting the KNN model using a specified value of 'k' (28 in this case). The confusion matrix is then generated to evaluate the model's performance on the test set.

Linear Regression

```
### 4.9.5 Linear Models (Logistic Regression)

```{r, echo=FALSE}
#install.packages('aod')
library(aod)
library(ggplot2)
```

```{r, echo=FALSE}
sapply(without_conf_n, sd)
```

```{r,include=FALSE, echo=FALSE}
xtabs(censor_flg~age+chf_flg+sapsi_first+icu_los_day, data = without_conf_n)
```

```{r, echo=FALSE}
lr <- glm(censor_flg~age+chf_flg+sapsi_first+icu_los_day, data = training, family = "binomial")
summary(lr)
```

```{r}
#calculate probability of default for each individual in test dataset
predicted <- predict(lr, testing, type="response")

#calculate AUC
library(pROC)
auc(testing$censor_flg, predicted)
```
```

4.9.5 Linear Models (Logistic Regression)

```
##             age       gender_num      sapsi_first          chf_flg
##       0.23401548       0.49976343       0.15878730       0.37410439
##       censor_flg        renal_flg         wbc_first         hgb_first
##       0.49167711       0.22400212       0.06767688       0.13801630
##      icu_los_day hospital_los_day
##       0.13859862       0.08316255
```

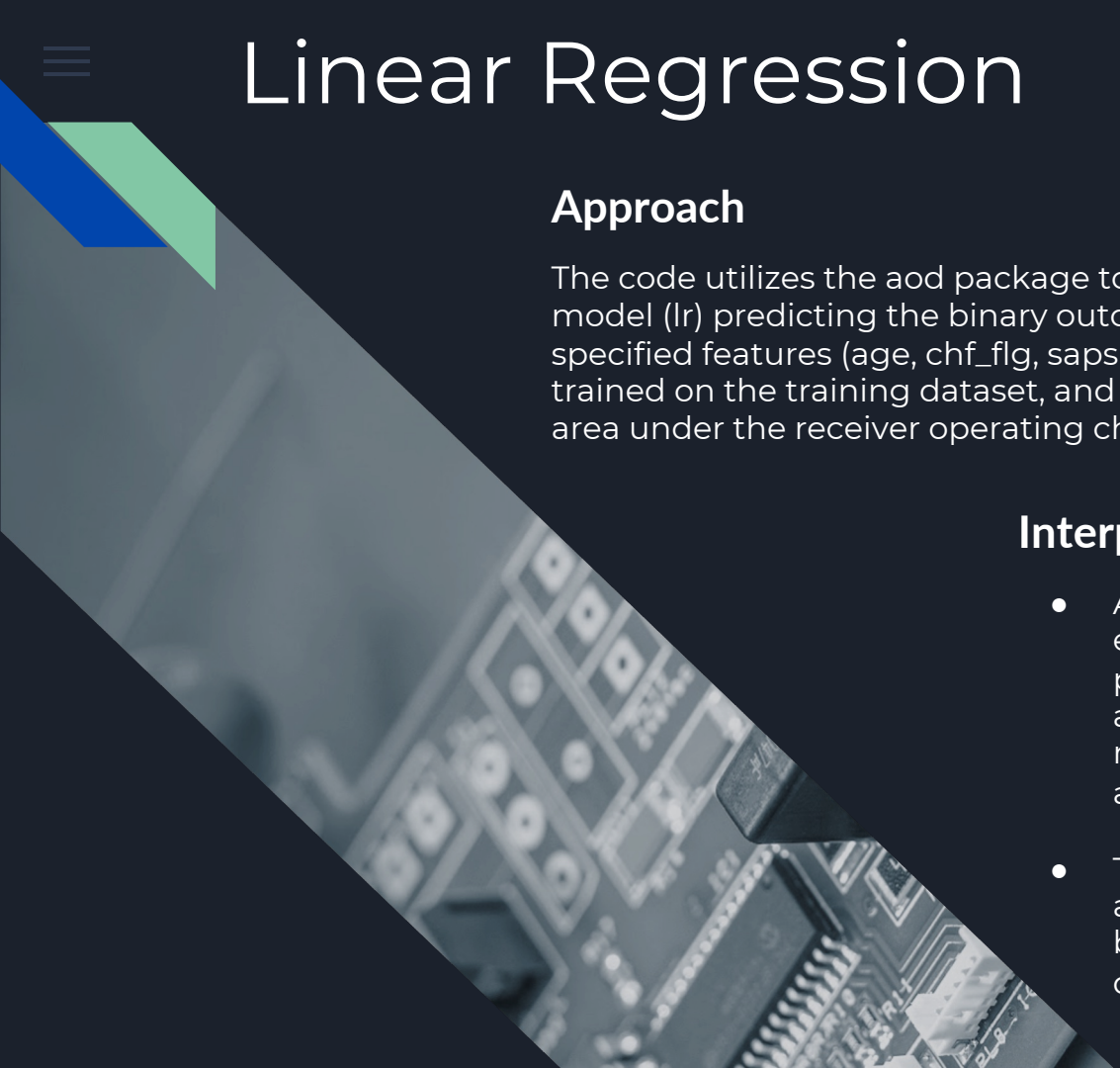
```
##
## Call:
## glm(formula = censor_flg ~ age + chf_flg + sapsi_first + icu_los_day,
##     family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4671  -0.9968   0.4035   0.9178   1.8318
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.80406    0.39175   9.710  < 2e-16 ***
## age         -4.73250    0.55722  -8.493  < 2e-16 ***
## chf_flg     -0.06498    0.25421  -0.256  0.79826
## sapsi_first -1.64142    0.60815  -2.699  0.00695 **
## icu_los_day -1.11898    0.73715  -1.518  0.12902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 765.86  on 571  degrees of freedom
## Residual deviance: 634.26  on 567  degrees of freedom
## AIC: 644.26
##
## Number of Fisher Scoring iterations: 4
```

```
#calculate probability of default for each individual in test dataset
predicted <- predict(lr, testing, type="response")

#calculate AUC
library(pROC)
auc(testing$censor_flg, predicted)
```

```
## Area under the curve: 0.7146
```

Linear Regression



Approach

The code utilizes the aod package to implement a logistic regression model (lr) predicting the binary outcome variable censor_flg based on specified features (age, chf_flg, sapsi_first, icu_los_day). The model is trained on the training dataset, and its performance is evaluated using the area under the receiver operating characteristic curve (AUC) on the test set.

Interpretation

- A greater Area Under the Curve (AUC) signifies enhanced model performance, indicating its proficiency in distinguishing between positive and negative classes. With an AUC of 0.71, this model's predictions demonstrate moderate accuracy and leave room for improvement.

- The Akaike Information Criterion (AIC) stands at 644.26. A smaller AIC value indicates a better fit for the model, emphasizing the quality of its overall performance.

CONCLUSION

- Overall Inference
- Discussion
- Implication: Triple Aim
- Challenges faced and strategies to overcome

RESULTS

I experimented with two models – KNN and logistic regression. The logistic regression model emerged as the most suitable fit, with an Area Under the Curve (AUC) of 0.71. This suggests that the model achieved a 71% accuracy rate in predicting mortality, taking into account factors such as age, SAPS score at ICU admission, and the presence or absence of congestive heart failure (chf_flg=1 or chf_flg=0).

Discussion & Future Scope

For associations and correlations to be scientifically valid, it's crucial to ensure their reliability. In future analyses of this inquiry, we can enhance the model's robustness by collecting additional data and devising strategies to select features that truly represent the sample, making the results more meaningful. Following internal validation of the model, it's advisable to conduct pilot tests in different geographic areas for external validation, addressing any discrepancies before implementing it in real-world scenarios. It's imperative to disclose regulations preventing the model's misuse by for-profit agencies, particularly in adjusting insurance premiums based on health conditions, to prevent potential disparities.

HEALTHCARE IMPACT (TRIPLE AIM)

Improving the experience of care

Healthcare organizations might consider utilizing a greater portion of the facilities for patients with worsen physiological conditions.

Improving the health of population

Analysis of physiological and anatomical factors leading to chronic diseases can help diminish the chances of deteriorating conditions

Reducing per capita costs of healthcare

As a preventive measure, providing medical attention and care earlier to a vulnerable population will lead to less cost injection in the later deteriorating stages.

Challenges Faced & Strategies to Overcome

Dataset selection

Make sure in advance that the hypothesis is answerable and of value to the community

Selection of Visualization

Understand the datatype to figure out the best fit visualization, and match it with the ease of inference from that visual.

Understanding new terminologies

Intensive use of web and experiment with MDCalc to dig deeper into new terminologies.

Analysis strategy

Do not hesitate to ask questions from the teaching team/instructor. Prof. Dinov, thanks for the guidance.

Motivation

Self-Motivation to keep at it even when graphs fail to reveal at times, what was expected.

Thank you!