Explainable artificial intelligence model to predict mortality in patients with suspected acute coronary syndrome

Edwin Kagereki - B00867154

21 October 2021

Contents

Business Understanding

Introduction

Cardiovascular diseases (CVDs), principally ischemic heart disease and cardiac stroke, are the leading cause of global mortality. In addition to their increasing global prevalence, they are often associated with poor survival(Roth 2020).

Acute Coronary Syndrome (ACS) is a term given to a continuum of CVDs ranging from ST-segment elevation myocardial infarction to non–ST-segment elevation myocardial infarction and unstable angina. Accurate estimation of risk for untoward outcomes after suspected onset of an ACS may help clinicians chose the type and intensity of therapy. Such predictions may be helpful as patients predicted to be at higher risk may receive more aggressive surveillance and/or earlier treatment, while patients predicted to be at lower risk may be managed less aggressively.

Problem statement The establishment of prognostic model for patients with suspected ACS is important in critical care medicine.Numerous risk-prediction models for differing outcomes exist for the different types of ACS.

These models however have some limitations. First, most models have been developed from large randomized clinical trial populations in which the generalizability to risk prediction in the average clinician's experience is questionable(Eagle KA Lim MJ 2004).

Second, these models fail at predicting future behavior while the treatment is underway. An alternative is use predictive modeling, to allow for the to evaluation of the risk or opportunity of a patient to guide a decision.

This project aims at developing a risk-prediction tool for ACS, focusing on clinical end point of all-cause mortality. The motivation is to utilize data collected within the first hour of intervention, to predict the outcome. The complete care pathway is thereafter constructed and comparisons done between the two groups. Adherence to guidelines is thereafter assessed.

Objectives

This project will develop a tool for application in the care pathway of patients suspected to have ACS. This is done in two steps:

1. A binary classification model. This model will use:

- patient demographics
- Interventions within the golden hour (laboratory, medication, procedures and microbiology studies).

The concept of the golden hour refers to the vital period by which a patient with a suspected cardiovascular event should be receiving definitive treatment to prevent death or irreparable damage to the heart.Although not set in stone, the chances to save a patient are usually high if substantive medical attention is given within an hour of the cardiac event(Johnson 2016). In this study the golden hour cut-off was 60 minutes after initial contact with the hospital. This included all interventions that were done prior to admission.

2. Process mining will be used to explain the outcome of the patient based on the care pathway followed. Three aspects will be checked:

- Process discovery - Processes followed by the two classes will be described and compared.
- Conformance checking - The care pathway followed by both patients groups will be checked for conformance with the American Heart Association(AHA) guidelines for CPR and ECC (ACLS 2020)

Success Criteria

The success of this project is to generate forward-looking, predictive insights to improve the management of patients suspected to have ACS by:

- Successfully predicting the ICU mortality outcome of the patient with suspected ACS based on the patient demographics and the interventions given within the first one hour.
- Identify care pathway in which undesirable events will likely be observed.

Situation Assessment

Inventory of resources

Software For this project the following software will be used:

1. SQL database server.
2. R
3. Python

The analysis will be done on a Windows desktop and a Linux server.

Sources of Data and Knowledge

MIMIC III Database MIMIC-III is a patient level database comprising of deidentified health-related data associated with 46,520 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. The database includes information such as demographics, vital sign measurements made at the bedside , laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (including post-hospital discharge).

Although de-identified, the datasets described herein contain detailed information regarding the clinical care of patients, and as such it must be treated with appropriate care and respect.

The following requirements have been met to access the database:

- Become a credentialed user on PhysioNet. This involved completion of a training course in human subjects research.
- Signed the data use agreement.

LOIC tables The Loin(Institute 2021) data tables will be used to enrich the laboratory dataset.

American Heart Association guidelines for CPR and ECC This AHA guideline(ACLS 2020) will be used as the gold standard for ACS care pathway. The pathway for the patients in this project will be assessed for conformity with this pathway.

Requirements, Assumptions, and Constraints

It is also assumed that:

- The patients in this population were only treated in this hospital, therefore mortality is only captured in this hospital.
- All the pre-hospitalization interventions were captured.
- The unit of analysis is the admission.

Risks and Contingencies

- Although the process mining will give a better predictive description of the patient outcomes, alternative surrogate modeling methods like decision tree maybe used.

Terminology

1. **Acute Cardiac syndrome:** Candidates of acute cardiac syndrome were identified using the *DIAGNOSIS* in the *ADMISSIONS* table which provides a preliminary. This column was is a free text diagnosis for the patient on hospital admission.

“stemi,”“acute coronary syndrome,”“angina,”“tachycardia,”“aortic aneurysm,”“pericardi,”“ortic dissection,”“coronary artery dissection,”“cardiomyopathy,”“heart failure,”“mitral valve disease,”“mitral stenosis,”“coronary artery disease,”“chf,”“congestive heart failure,”“heart failure,”“telemetry,”“myocardial infaction,”“cardiac arrest,”“myocardial infarction,”“aortic stenosis,”“st elevated,”“pericardial effusion,” “cardiomyopathy,”“cath lab,”“tamponade,”“tamponede”

2. Angiotensin-converting enzyme (ACE). The following terms will be used to identify ACE’s from the list of *DRUG* column of the *PRESCRIPTIONS TABLE* table:

“benazepril,” “captopril,” “enalapril,”“enalaprilat,”“fosinopril,” “lisinopril,” “moexipril,” “perindopril,” “quinapril,” “ramipril,”“trandolapril”

3. Beta blockers (beta-adrenergic blocking agents) The following terms will be used to identify Beta blocker from the list of *DRUG* column of the *PRESCRIPTIONS TABLE* table:

“acebutolol,”“atenolol,”“betaxolol,”“bisoprolol,”“carteolol,”“carvedilol,”“labetalol,”“metoprolol,”“nadolol,”“nebivolol,” “penbutolol,”“pindolol,”“propanolol,”“sotalol,”“timolol”

4. Glycoprotein IIb/IIIa inhibitors. The following words will be used to identify Glycoprotein IIb/IIIa inhibitors’s from the list of *DRUG* column of the *PRESCRIPTIONS TABLE* table:

“abciximab,”“eptifibatide,”“tirofiban,”“roxifiban,”“orbofiban”

5. P2Y12 inhibitors. The followeing words were used to identify the P2Y12 inhibitors from the list of *DRUG* column of the *PRESCRIPTIONS TABLE* table:

“clopidogrel,”“prasugrel,”“ticlopidine,”“ticagrelor”

6. HMGCoA The following words were used to identify HMGCoA from the list of *DRUG* column of the *PRESCRIPTIONS TABLE* table: *“altoprev,”“amlodipine,”“atorvastatin,”“caduet,”“crestor,”“ezallor,”“fluvastatin,”“lescol,”“*

Data Mining Goals

1. Build a binary classification machine learning model to predict all-cause mortality of patients based on demographics and interventions within first hour of suspected ACS event.
2. Compare the conformity of care path in patients who died and patients who survived with the with the ACLS care path.
3. Assess and report any work flow variants, and differences between patients who died and those who survived.

Success criteria

After training the binary classifier, evaluation measures will be used to assess the performance of the model. The predictive performance of the classifier will be assessed by calculating the number of correctly identified patients (true positives), the number of correctly recognized patients that are not member of the class (true negatives), the number of the patients that are wrongly recognized (false positives) and the number of the examples that were not identified (false negatives). By using these measures, a confusion matrix will be constituted.

		Ground Truth Values	
		Positive	Negative
Predicted Values	Positive	true positive (tp)	false positive (fp)
	Negative	false negative (fn)	true negative (tn)

The following measures will be calculated from this table:

- Accuracy.
- Precision.
- Recall.
- Specificity.

To benchmark the classification model performance, the results ranging from accuracy of (70% to 90%) as reported in larger, though different models used in the prediction of mortality in CVDs will be used(Sherazi et al. 2020).

Data Understanding

Data collection

Upon gaining access to the MIMIC-III database, all the data was transferred into a RDMS (relational database management system). This was done with the help of open source scripts(Ganas 2018). Subsequently, SQL queries were used to connect and access the data. Although the database includes 26 tables, only the following tables will be included in the analysis:

- **ADMISSIONS:** Contains information regarding a patients admission to the hospital. Information available includes timing information for admission and discharge, demographic information, the source of the admission, and so on. Record of 58,976 unique admissions.
- **PATIENTS:**Defines each patient in the database, i.e. defines a single patient. There are 46,520 patients recorded.
- **SERVICES:**Lists services that a patient was admitted/transferred under.This table contains 73,343 entries.
- **DIAGNOSIS_ICD**Identify type of data sources (online sources, experts, written documentation, etc.)
- **MICROBIOLOGYEVENTS:**Contains microbiology information, including cultures acquired and associated sensitivities.There are 631,726 rows in this table.
- **PRESCRIPTIONS:**Contains medication related order entries, i.e. prescriptions.This table contains 4,156,450 rows.
- **PROCEDUREEVENTS_MV:**Contains procedures for patients. This table has 258,066 rows.
- **D_ITEMS:** Definition table for all 12,487 items in the ICU databases.
- **D_LABITEMS:** Definition table for 753 laboratory measurements.

Initial Data Collection Report

Due to the complexity of the MIMIC-III database and the massive size of the source data (over 100 gigabytes), an SQL database was created. This was created using SQL scripts to load the MIMIC-III data into a SQL Server 2016 database.(Ganas 2018). Respective tables were queried, tables joined and the some columns dropped. Flat files(csv) were saved for further analysis.

Data quality in flat files

- 6 csv files were saved for further analysis. These will be joined for the modeling.
- Only number of fields to be used were retained.

Rationale for Inclusion/Exclusion

- All the clinical interventions will be included in the analysis.
- Although there were some tables with some data on the interventions, they were excluded because:

1. TEXT is often large and contains many newline characters were excluded because of the complexity of analyzing the data.
2. Echo reports, ECG reports, and radiology reports were excluded because of the complexity of the data.

Data overview

The data was accessed and initial summary analysis was done. The patients details are summarized below.

Biodata

The other tables were:

1. Procedures.
2. Laboratory.
3. Medicine.
4. Microbiology.
5. Services.

Each of these table will be merged with the demographics dataset to form a final csv for further analysis.

Data quality

Data Preparation

Admissions: These are the dataset(s) produced by the data preparation phase,

Procedures: This is the description of the dataset(s)

Microbiology: This is the description of the dataset(s)

Services: This is the description of the dataset(s)

Laboratory: This is the description of the dataset(s)

Prescriptions: This is the description of the dataset(s)

Select Data

Feature Selection Using Filter Methods Example 1 – Using correlation Example 2 – Using hypothesis testing Example 3 – Using information gain for variable selection

Rationale for Inclusion/Exclusion

List the data to be used/excluded and the reasons for these decisions.

- Collect appropriate additional data (from different sources—in-house as well as externally)
- Perform significance and correlation tests to decide if fields should be included
- Reconsider Data Selection Criteria in light of experiences of data quality and data exploration (i.e., may wish include/exclude other sets of data)
- Reconsider Data Selection Criteria in light of experience of modeling (i.e., model assessment may show that other datasets are needed)
- Select different data subsets (e.g., different attributes, only data which meet certain conditions)
- Consider the use of sampling techniques (e.g., A quick solution may involve splitting test and training datasets or reducing the size of the test dataset, if the tool cannot handle the full dataset. It may also be useful to have weighted samples to give different importance to different attributes or different values of the same attribute.)
- Document the rationale for inclusion/exclusion
- Check available techniques for sampling data Based on Data Selection Criteria, decide if one or more attributes are more important than others and weight the attributes accordingly. Decide, based on the context (i.e., application, tool, etc.), how to handle the weighting.

Clean Data

Raise the data quality to the level required by the selected analysis techniques. This may involve the selection of clean subsets of the data, the insertion of suitable defaults, or more ambitious techniques such as the estimation of missing data by modeling.

Data Cleaning Report

Describe the decisions and actions that were taken to address the data quality problems reported during the Verify Data Quality Task. If the data are to be used in the data mining exercise, the report should address outstanding data quality issues and what possible effect this could have on the results.

- Reconsider how to deal with any observed type of noise
- Correct, remove, or ignore noise
- Decide how to deal with special values and their meaning. The area of special values can give rise to many strange results and should be carefully examined. Examples of special values could arise through taking results of a survey where some questions were not asked or not answered. This might result in a

value of 99 for unknown data. For example, 99 for marital status or political affiliation. Special values could also arise when data is truncated—e.g., 00 for 100-year-old people or all cars with 100,000 km on the odometer.
- Reconsider Data Selection Criteria in light of experiences of data cleaning (i.e., you may wish to include/exclude other sets of data).

Remember that some fields may be irrelevant to the data mining goals and, therefore, noise in those fields has no significance. However, if noise is ignored for these reasons, it should be fully documented as the circumstances may change later.

Construct Data

This task includes constructive data preparation operations such as the production of derived attributes, complete new records, or transformed values for existing attributes.

- Check available construction mechanisms with the list of tools suggested for the project
- Decide whether it is best to perform the construction inside the tool or outside (i.e., which is more efficient, exact, repeatable)
- Reconsider Data Selection Criteria Data Selection Criteria in light of experiences of data construction (i.e., you may wish include/exclude other sets of data)

Derived Attributes

Derived attributes are new attributes that are constructed from one or more existing attributes in the same record. An example might be: area = length * width.

Why should we need to construct derived attributes during the course of a data mining investigation? It should not be thought that only data from databases or other sources should be used in constructing a model. Derived attributes might be constructed because:

- Background knowledge convinces us that some fact is important and ought to be represented although we have no attribute currently to represent it
- The modeling algorithm in use handles only certain types of data—for example we are using linear regression and we suspect that there are certain non-linearities that will be not be included in the model
- The outcome of the modeling phase suggests that certain facts are not being covered

Derived attributes

- Decide if any attribute should be normalized (e.g., when using a clustering algorithm with age and income, in certain currencies, the income will dominate)
- Consider adding new information on the relevant importance of attributes by adding new attributes (for example, attribute weights, weighted normalization)
- How can missing attributes be constructed or imputed? [Decide type of construction (e.g., aggregate, average, induction).]
- Add new attributes to the accessed data

Before adding Derived Attributes, try to determine if and how they ease the model process or facilitate the modeling algorithm. Perhaps "income per person" is a better/easier attribute to use than "income per household." Do not derive attributes simply to reduce the number of input attributes.

Another type of derived attribute is the single-attribute transformation, usually performed to fit the needs of the modeling tools.

Single-attribute transformations

- Specify necessary transformation steps in terms of available transformation facilities (for example, change a binning of a numeric attribute)
- Perform transformation steps

Transformations may be necessary to change ranges to symbolic fields (e.g., ages to age ranges) or symbolic fields (“definitely yes,” “yes,” “don’t know,” “no”) to numeric values. Modeling tools or algorithms often require them.

Generated Records

Generated records are completely new records, which add new knowledge or represent new data that is not otherwise represented (e.g., having segmented the data, it may be useful to generate a record to represent the prototypical member of each segment for further processing).

Check for available techniques if needed (e.g., mechanisms to construct prototypes for each segment of segmented data).

Integrate Data

These are methods for combining information from multiple tables or other information sources to create new records or values.

Merged Data

Merging tables refers to joining together two or more tables that have different information about the same objects. At this stage, it may also be advisable to generate new records. It may also be recommended to generate aggregate values.

Aggregation refers to operations where new values are computed by summarizing information from multiple records and/or tables.

- Check if integration facilities are able to integrate the input sources as required
- Integrate sources and store results
- Reconsider Data Selection Criteria in light of experiences of data integration (i.e., you may wish to include/exclude other sets of data)

Remember that some knowledge may be contained in non-electronic format.

Format Data

Formatting transformations refers primarily to syntactic modifications made to the data that do not change its meaning, but might be required by the modeling tool.

Reformatted Data

Some tools have requirements on the order of the attributes, such as the first field being a unique identifier for each record or the last field being the outcome field the model is to predict.

Rearranging attributes Some tools have requirements on the order of the attributes, such as the first field being a unique identifier for each record or the last field being the outcome field the model is to predict.

Reordering records It might be important to change the order of the records in the dataset. Perhaps the modeling tool requires that the records be sorted according to the value of the outcome attribute.

Reformatted within-value

- These are purely syntactic changes made to satisfy the requirements of the specific modeling tool
- Reconsider Data Selection Criteria in light of experiences of data cleaning (i.e., you may wish to include/exclude other sets of data)

Dataset

These are the dataset(s) produced by the data preparation phase, used for modeling or for the major analysis work of the project.

Dataset Description

This is the description of the dataset(s) used for the modeling or for the major analysis work of the project.

Appendix

```
con <- DBI::dbConnect(RPostgreSQL::PostgreSQL(),
  host = "AWS end point",
  user = "eKagereki",
  password = rstudioapi::askForPassword("Database password")
)

Admissions <- tbl(con, "ADMISSIONS")
Patients <- tbl(con, "PATIENTS")

## The following code:
## 1. Converts the dates into the right format.
## 2. Calculates the age of the individuals.
## 3. Created a variable - Period that
## 4. Calculates the exact end time


admin<-merge(x = admin, y = pt, by = "SUBJECT_ID", all.x = TRUE) %>%
  mutate(ADMITTIME = ymd_hms(ADMITTIME)) %>%
  mutate(DISCHTIME = ymd_hms(DISCHTIME)) %>%
  mutate(DEATHTIME = ymd_hms(DEATHTIME)) %>%
  mutate(DOB = ymd_hms(DOB)) %>%
  mutate(AGE = round(as.numeric(difftime(ADMITTIME,DOB, units = "days")/365),0)) %>%
  mutate(LOS2 = as.numeric(difftime(DISCHTIME,ADMITTIME, units = "hours"))) %>%
  mutate(AGE = ifelse(AGE<300, AGE, AGE-211)) %>%
  mutate(Period = ntile(as.numeric(ADMITTIME),12)) %>%
  mutate(endGoldenHour = ADMITTIME + minutes(60)) %>%
  group_by(SUBJECT_ID) %>%
  mutate(admissionCycle = 1:n()) %>% ## The admission cycle
  group_by(SUBJECT_ID) %>%
  arrange(DISCHTIME) %>%
  mutate(nAdmissions = n_distinct(HADM_ID)) %>%
  mutate(deadBefore = as.numeric(DEATHTIME-endGoldenHour)) %>%
  mutate(deadBefore = ifelse(HOSPITAL_EXPIRE_FLAG==0, 0, deadBefore)) %>%  ## Remove all the patients w
  filter(deadBefore>=0) %>%
  mutate(DIAGNOSIS2 = tolower(DIAGNOSIS))

##
## Prepare the Lab data:

## The lab data is 1.72GB, manipulations were limited, therefore it was run in the server

lab<- tbl(con, "LABEVENTS") %>%
  select(HADM_ID,SUBJECT_ID,FLUID,LABEL,FLAG,CHARTTIME)
```

```r
data2<-data[data$SUBJECT_ID %in% cardiac$SUBJECT_ID, ]
  lab %>%
  filter(gene_ID %in% accessions40$V1)
  labiTem<-read.csv("D_LABITEMS.csv")
  lab2<-merge(x=lab, y=labiTem, by="ITEMID", all.x = TRUE) %>%
  filter(ROW_ID>1) %>% ## The first row contains the test lable and not the actual test
  unite(combined, FLUID, LABEL, sep = "-", remove = FALSE) %>%
  select(HADM_ID,CHARTTIME,FLAG,combined) %>%
  mutate_if(is.character, list(~na_if(.,""))) %>%   ## Replace the blanks
  mutate(CHARTTIME = ymd_hms(CHARTTIME))
  lab3<-merge(x=cardiacSyndromes2,y=lab2,by='HADM_ID',x.all=TRUE) %>%
  mutate(Checktime = ifelse(endGoldenHour>=CHARTTIME, "After", "Before")) %>%
  filter(Checktime=="Before",FLAG=="abnormal") %>%
  select(-Checktime) %>%
  select(HADM_ID,combined,FLAG) %>%
  unite(combined, combined, FLAG, sep = "_", remove = FALSE) %>%
  count(HADM_ID, combined, sort = TRUE)

## The labs in those missing the HADM were done as outpatients
missingHADM<-labNew %>%
  filter(is.na(HADM_ID))

## We check teh admission of the patient!
missingHADM2<- merge(x=missingHADM,y=admin,by="SUBJECT_ID", x.all=TRUE) %>%
  select(SUBJECT_ID,HADM_ID.x,HADM_ID.y,CHARTTIME,ADMITTIME,DISCHTIME,combined,LOINC_CODE) %>%
  mutate(Checktime = ifelse(CHARTTIME>=ADMITTIME & CHARTTIME<=DISCHTIME, "After", "Before")) %>%
  filter(Checktime=="Before") %>%
  mutate(HADM_ID = HADM_ID.y) %>%
  select(SUBJECT_ID,HADM_ID,CHARTTIME,combined,LOINC_CODE)

## These had admission numbers - Labs were done in patient
presentHADM<-labNew %>%
  drop_na(HADM_ID)

## The dataset below will be merged with the admission data for the
## the classification model and also used to prepare the log file for the analysis
allLAbs<-bind_rows(presentHADM, missingHADM2)

## Convert this into the wide formart
labWide<-merge(x=cardiacSyndromes2,y=allLAbs,by='HADM_ID',x.all=TRUE) %>%
  mutate(Checktime = ifelse(endGoldenHour>=CHARTTIME, "After", "Before")) %>%
  filter(Checktime=="Before") %>%
  select(-Checktime) %>%
  select(HADM_ID,combined) %>%
  count(HADM_ID, combined, sort = TRUE) %>%
  spread(combined, n) %>%
  replace(is.na(.), 0)

## Servive points that the patient has been seen
services<-tbl(con, "SERVICES")
services2<-merge(x=cardiacSyndromes2,y=services,by='HADM_ID',x.all=TRUE) %>%
  mutate(TRANSFERTIME2 = ymd_hms(TRANSFERTIME)) %>%
  mutate(Checktime = ifelse(endGoldenHour>=TRANSFERTIME2, "After", "Before")) %>%
  filter(Checktime=="Before") %>%
  select(HADM_ID,PREV_SERVICE,CURR_SERVICE) %>%
  unite(cService, PREV_SERVICE, CURR_SERVICE, sep = "_", remove = FALSE) %>%
  select(HADM_ID,cService)
services3<-services2 %>% count(HADM_ID, cService, sort = TRUE)
servicesWide <- services3 %>%
  spread(cService, n) %>%
  replace(is.na(.), 0)

servicesWide<-merge(x=cardiacSyndromes2, y=servicesWide, by="HADM_ID", all.x = TRUE) %>%
  replace(is.na(.), 0)
checkServicesBeforeAdmin<-servicesWide %>% filter(HADM_ID==0)
nrow(checkServicesBeforeAdmin) ## To check if there was any service before the patient was admitted.



## ETL for the procedures:
procedures<-tbl(con, "PROCEDUREEVENTS_MV")
D_ITEM<-tbl(con, "D_ITEMS")
procedures<-merge(x=procedures,y=D_ITEM, by = "ITEMID",x.all=TRUE) %>%
  select(SUBJECT_ID,HADM_ID,STARTTIME,ENDTIME,LABEL)

## This will  be used for the log files
logProdecures<-procedures
procWide<-merge(x=procedures,y=cardiacSyndromes2, by = "HADM_ID",x.all=TRUE) %>%
  mutate(STARTTIME2 = ymd_hms(STARTTIME)) %>%
  mutate(Checktime = ifelse(endGoldenHour>=STARTTIME2, "After", "Before")) %>%
  filter(Checktime=="Before") %>%
  select(HADM_ID,LABEL) %>% count(HADM_ID, LABEL, sort = TRUE)%>%
  spread(LABEL, n) %>%
  replace(is.na(.), 0)

procWide<-merge(x=cardiacSyndromes2, y=procWide, by="HADM_ID", all.x = TRUE) %>%
  replace(is.na(.), 0)

## MicroB
microb<- tbl(con, "MICROBIOLOGYEVENTS") %>%
  select(-ROW_ID,-SUBJECT_ID)
microb<-merge(x=cardiacSyndromes2,y=microb,by='HADM_ID',x.all=TRUE) %>%
  mutate(CHARTTIME = ymd_hms(CHARTTIME)) %>%
  unite(combined, SPEC_TYPE_DESC, ORG_NAME,AB_NAME,INTERPRETATION, sep = "_", remove = FALSE)
Logmicrob<-microb
microbWide<-microb%>%
  mutate(Checktime = ifelse(endGoldenHour>=CHARTTIME, "After", "Before")) %>%
  filter(Checktime=="Before") %>%
  select(HADM_ID,combined) %>%
  count(HADM_ID, combined, sort = TRUE) %>%
  spread(combined, n) %>%
  replace(is.na(.), 0)

## MEDS:

medication<-tbl(con, "PRESCRIPTIONS")  %>%
  mutate(drug2 = tolower(DRUG)) %>%
  filter( grepl(paste(subset, collapse="|"),drug2)) %>%
  mutate(DRUG = case_when(grepl("aspirin", drug2) ~ "Aspirin",
                           grepl("morphine", drug2) ~ "Morphine",
                          grepl((paste(HMGCoA, collapse="|")), drug2) ~ "HMGCoA",
                          grepl((paste(ACE, collapse="|")), drug2) ~ "ACE Inhibitors",
                          grepl((paste(betaBlockers, collapse="|")), drug2) ~ "Beta blockers",
                          grepl((paste(glycoproteinInhibitors, collapse="|")), drug2) ~ "GpIIb/IIIa inhi
                           grepl("nitroglycerin", drug2, ignore.case = TRUE) ~"Nitroglycerine")) %>%
  na.omit() %>%
  mutate(STARTDATE=ymd_hms(STARTDATE,tz="Europe/London")) %>%
  mutate(ENDDATE=ymd_hms(ENDDATE,tz="Europe/London")) %>%
  unite(case_id, SUBJECT_ID,HADM_ID,DRUG, sep = "-", remove = FALSE) %>%
  group_by(case_id) %>%
  mutate(start = min(STARTDATE),complete = max(ENDDATE)) %>%
  select(-X.1,-X,-STARTDATE,-ENDDATE,-drug2) %>%
  distinct(case_id, .keep_all= TRUE) %>%
  unite(case_id, SUBJECT_ID,HADM_ID, sep = "-", remove = FALSE) %>%
  group_by(case_id) %>%
  arrange(start) %>%
  mutate(activity_instance = 1:n())



## Write the flat files:
##

# Classification:
write.csv(cardiacSyndromes,"cardiacSyndromes.csv")
write.csv(servicesWide,"servicesWide.csv")
write.csv(labWide,"labWide.csv")
write.csv(procWide,"procWide.csv")
write.csv(microbWide,"microbWide.csv")

# Process mining:
write.csv(logProdecures,"logProdecures.csv")
write.csv(microbWide,"microbWide.csv")
write.csv(allLAbs, "logLab.csv")

###
###
##EXPLORATORY DATA ANALYSIS
###
```

References

ACLS. 2020. "Acute Coronary Syndromes Algorithm - ACLS Version Control: This Document Follows 2020 American Heart Association Guidelines for CPR and ECC. American Heart Association Guidelines Are Updated Every Five Years." https://www.acls.net/images/algo-acs.pdf.

Eagle KA Lim MJ, et al., Dabbous OH. 2004. "A Validated Prediction Model for All Forms of Acute Coronary Syndrome: Estimating the Risk of 6-Month Postdischarge Death in an International Registry." *Jama*. https://doi.org/doi:10.1001/jama.291.22.2727.

Ganas, Spiro. 2018. "MIMIC-on-SQL-Server." *GitHub Repository*. GitHub. https://github.com/SpiroGanas/MIMIC-on-SQL-Server.

Institute, Regenstrief. 2021. "LOINC." *LOINC*. https://loinc.org/downloads/.

Johnson, et al, A. 2016. "MIMIC-III Clinical Database." https://doi.org/https://doi.org/10.13026/C2XW26.

Roth, et al, Gregory A. 2020. "Global Burden of Cardiovascular Diseases and Risk Factors, 1990-2019: Update from the GBD 2019 Study." *Journal of the American College of Cardiology* 76 (25): 2982–3021. https://doi.org/10.1016/j.jacc.2020.11.010.

Sherazi, Syed Waseem Abbas, Yu Jun Jeong, Moon Hyun Jae, Jang-Whan Bae, and Jong Yun Lee. 2020. "A Machine Learning Based 1-Year Mortality Prediction Model After Hospital Discharge for Clinical Patients with Acute Coronary Syndrome." *Health Informatics Journal* 26 (2): 1289–1304. https://doi.org/10.1177/1460458219871780.