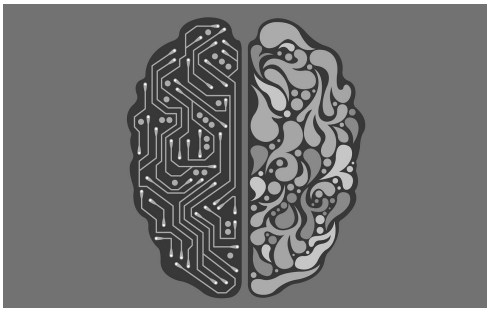
Research Internship (PRE)

Field of Study: STIC/IAC
Scholar Year: 2018-2019

Acute kidney injury prediction in intensive care patients

Building machine learning model suitable for real-time analysis



Confidentiality Notice

Author:
BASSET Aymeric

Promotion:
2020

ENSTA ParisTech Tutor:
CHAPOUTOT Alexandre

Host Organism Tutor:
BEAN Daniel

Internship from 20/05/19 to 09/08/19

Name of the host organism: King's College London
Address: 16 De Crespigny Park
London SE5 8AF
United Kingdom

Abstract

The prediction of Acute Kidney Injury (AKI) in Intensive Care Unit (ICU) is a flagship example of how new machine learning techniques can be used to deal with a common yet dangerous medical issue. Indeed, the occurrence rate of AKI during hospitalization is around 5% and can result in up to 70% mortality.[1] Thanks to the development of electronic health records, real-time updates and medical databases, it is now possible to built and implement within the hospital machine learning models and digitally-enabled care [2]. Our goal was to develop a model with a similar approach as those presented in state of the art papers for this disease[1][3], and to enhance it using Natural Language Processing (NLP). The dataset used was based on the Medical Information Mart for Intensive Care (MIMIC) database and composed of 20742 adult patients. Our best model is based on XGBoost and has an auroc score of 86.5% 24 hours before the event, able to beat Stanford's XGBoost model by more than 10% [1], but under 4% of a very recent deep learning model proposed by DeepMind.[3] In addition to the good performance of the model, the SHapley Additive exPlanations (SHAP) [4] technique was used to analyze it.This allowed us to provide comprehensive insights into the main causes of the disease in general or at an individual patient scale, and to compare the evolution during real-time analysis. Notwithstanding the performance limits of our and other current models, this approach might be useful for a real-time alerting application and to give relevant explanations to health practitioners within a reasonable time window.

Keywords: Acute Kidney Injury; Machine learning; Structured and unstructured data; Real-time; Natural Language Processing; XGBoost

Acknowledgment

I would like first to thank Professor Richard Dobson, head of the Biostatistics and Health Informatics department at King's College London, for giving me the opportunity to work inside his department. I also, want to thank very much Doctor Daniel Bean, my supervisor. It was a pleasure to work for and with him. He was there when I needed him and would always listen to my ideas while giving me new insights in this fantastic field. Finally, I would like to thank all the PhD students I met during my time at King's College, who helped me work in the friendliest conditions possible.

Contents

Introduction

Acute Kidney Injury (AKI), formerly known as Acute Renal Failure (ARF), is characterized by a dangerous decline in renal function. The decline of the renal functions goes from mild impairment to complete failure. It may happen within a few hours or a few days. AKI is common in patients who are hospitalized and especially in older adults[5].

Different symptoms of acute kidney injury include:

- Decreasing Urine
- Legs, ankles an eyes Swelling Fatigue
- Shortness of breath
- Confusion
- Nausea
- Seizure or coma
- Chest pain

To label the different patients between case (AKI) or control (no AKI), the National Health System (NHS) algorithm called Gold STANDARD Algorithm [1], is going to be used. Basically, the algorithm is based on creatinine's ratio evolution: A sudden increasing or too high level of creatinine in the blood classify the patient as ill (see the appendix for the precise rules). The rules are based on creatinine since creatinine is a muscle waste that the kidneys are supposed to filter, and a high concentration of it in the blood indicates that the kidneys don't work properly.

In order to predict AKI among ICU patients, a machine learning model will be used. The idea is that with enough samples of cases and controls, those models can learn criteria or pattern among hundreds of features to classify accurately the illness. This is called supervised machine learning. The general mathematical idea behind is that for each sample, a wrong classification is associated with a cost, and the algorithm tries to reduce this loss function [6] by giving more importance to certain features for example.

Many studies have tried to predict AKI in ICU using different techniques and datasets. The most relevant and recent ones are from Stanford(2018) and DeepMind(2019), where they respectively developed an XGBoost[1] model on the MIMIC database and a recursive neural network with US Department of Veterans Affairs's dataset[1][3].

[1]XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. It is based on a tree ensemble technique, which means that to do one prediction with a strong model, the algorithm

is going to use a lot of trained decision trees, which are weak learners. The idea is to train sequentially the trees to reduce the error on the samples. In this case, compared to other boosting techniques, the weight of a new tree is calculated in order to reduce a loss function, based on gradient descent method[7].

Part I

Context

I.1 The Institute of Psychiatry, Psychology and Neuroscience

The Institute of Psychiatry, Psychology and Neuroscience (IoPPN) is a world-leading centre for research on the brain and the mind. It is part of both the NIHR Maudsley BRC (a major national research centre for mental health) and Health Data Research UK (the UK's new national health data science research institute). It aims at improving the medical care provided to mentally ill patients. It was founded in 1948 but his origin goes back to 1896. Situated in the south of London, it's a faculty of King's College London, working closely with King's College Hospital since they are both inside King's Denmark Hill Campus.

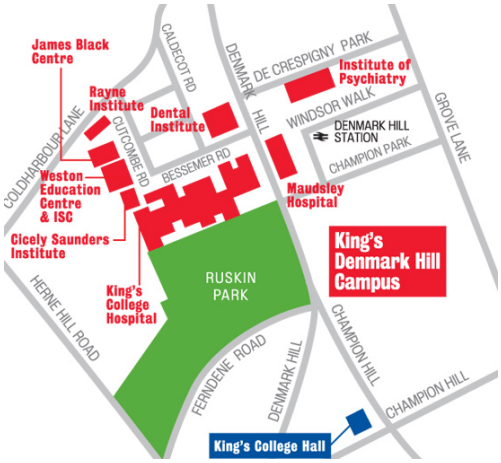


Figure I.1: Plan of King's College Hospital and the IoPPN

I was part of the Health Informatics and Biostatistics department, based inside the IoPPN, and lead by Professor Richard Dobson, which is also the head of the Precision Health Informatics Data Lab (PHIDL). I worked under the supervision of Doctor Daniel Bean. My work is part of a bigger two-years-long collaboration between Health Informatics experts and physicians at University College London hospital and Bristol hospital.

I.2 Projects within the department

There is more than 30 doctors and PhD students working in the Health Informatics department on various subjects.

I.2.1 Cogstack

CogStack is a platform built to improve text manipulation in a medical context. It's a pipeline allowing simple or complex queries to retrieve or extract information inside medical documents, reports, diagnostics etc. This data mining is possible thanks to the NHS Trusts. Millions of documents have been processed withing CogStack and it is very useful to health professionals since this process would be time-consuming or sometimes impossible without this platform[8].

I.2.2 Natural language processing

There is a lot of NLP research ongoing inside the department, and two self-developed were used inside this project: and MedCat.
SemEHR was released in 2018 and is an open-source tool to do semantic search in electronic health records, within structured and unstructured data[9]. MedCat[**?**] is an alternative approach using more advanced word embedding techniques compared to the rule-based approach of SemEHR to give greater speed and flexibility.

I.2.3 Radar-CNS

Remote Assessment of Disease and Relapse – Central Nervous System (RADAR-CNS) is a public program focused on wearable devices, enabling the monitoring of patients suffering from depression, sclerosis and epilepsy.The goal is to improve the ways of collecting data on this deceases, to widen knowledge on this deceases out of clinical rating, and eventually predict a relapse.[10]

I.3 Goal and methodology

My work is related to health data science and analytics. The process of collecting the data and creating a first usable dataset was already done by another Ensta's student last year. A first model using XGBoost has been tested and yielded promising results.
Therefore, my goal is to enhance it, trough the process of data engineering and feature selection, then to use different parameters or models such as neural networks to compete with the one already existing, and finally to improve our understanding of the model, either as a time-point predictor or when running real-time diagnostics.
During the whole process, it will be paramount to handle carefully the data provided. From a statistical point of view, good practises such as cross-validation is to be implemented. From an ethical point of view, it is mandatory to avoid any data leak and protect the identity of the patients, following human research laws.

Part II

Data Engineering

II.1 The dataset

II.1.1 Origin of the data

The information used was extracted from the MIMIC (Medical Information Mart for Intensive Care) database [11]. It is a large, freely-available medical database developed by MIT. It contains health data to over forty thousand anonymized patients who stayed in the Bet Israel Deaconess Medical Center between 2001 and 2012. It is composed of structured data (health signals such as heart rate, temperature etc) and unstructured data (text from medical reports most of the time).

II.1.2 Features extracted

To create the dataframe, relevant features are extracted from MIMIC:

- Heart rate
- Temperature
- Creatinine serum level
- Urine output
- Glasgow Coma Scale (GCS)[1]
- Respiratory rate
- Medical concepts extracted from reports via NLP pipeline

Starting from a time point where we want a prediction, we go back 24 hours (since we want to predict 24 hours in advance). The last 50 hours of unstructured data are collected, and 5 measures for each structured feature (3 for creatinine) from the last hours (one every hour) are collected. Those features are untitled Temperature-3 for example for the temperature of the patient 3 hours before the asked prediction (so 27 hours before the event).

It is important to note that if a value is missing, the last known measure is used to fill the gap. Also, the last structured value (so -5 or -3 for creatinine) is the mean value of this

[1] GCS is a way to communicate about the level of consciousness of patients, based on motor, verbal and eye response. The scale goes from 0 to 15, where 15 is a fully responding patient[12].

feature during the whole stay of this patient, acting as a reference for a patient. In the end, the database contains 55870 features.

II.1.3 Selecting relevant patients

The original cohort of patients is composed of more than forty thousand patients, and to assure that the results are relevant and to avoid bias, some selection criteria were applied to filter some patients:

- Patient must be between 18 and 89 years old
- No AKI code in the report or already high level of creatinine
- At least 3 creatinine values and at least one during ICU
- Only AKI stage 2 and 3 were retained as cases to be predicted, others patients are controls.

After those manipulations, the database contained 20742 patients that were judged suitable for the study.

II.1.4 Statistics

The following table sums up all the relevant information about the population studied:

	Characteristic	Proportion(%)	AKI(%)
Gender	Male	42.5	46.7
	Female	57.5	53.3
Age	18-29	4.4	3.3
	30-39	5.4	5.4
	40-49	11.1	11.8
	50-59	18.6	19.5
	60-69	22.4	24.1
	70+	38.1	35.9
Severe AKI based on the gold standart algorithm	Yes	11.5	-
	No	88.5	-

Table II.1: Statistics within the cohort used for the dataset

It is important to note that the dataset is unbalanced towards the controls. It is usually the case when handling medical data, and this matter will be dealt with later.

II.2 Feature selection

To enhance the model,the first possible thing to do is to manage the dataset. To improve the model from a memory and time point of view, but also to avoid overfitting, the first objective is to reduce the number of features used.

II.2.1 Cleaning the data

Since the features generated from the NLP pipeline are specific words picked through the reports, it happens that a specific concept only corresponds to very few patients. To deal with this, a first filter is applied on the whole dataframe, where only the features appearing in it at least one percent of the time are kept (this mean that there is a number different than zero at least one percent of the time). Using this technique, the number of features drops below 3000, depending on the NLP technique used (2676 using SemEHR and 1676 using MedCat).

II.2.2 Top-Down selection

Even though the number of features was divided by thirty, there is still a lot of features, and this may lead the model to associate certain features to a very specific scenario, and then not to be general enough. To avoid this, a second feature selection technique is used, based on the built-in function SelectFromModel from the python Scikit-Learn package. Basically, this is a Top-Down selection technique, where the model is trained using all the features, and then tested using only the top n features. The top features are extracted from the feature_importances_ attribute of the model (here XGBoost).

Since the number of features is quite high, techniques such as correlation matrix or contingency table were avoided, even though it is possible to use them. After cross-validating[2] the results and testing the features retained through Permutation Importance,also known as Mean Decrease Accuracy (MDA), 200 features were selected. It is still a high number of features, but some further selection is going to be applied through L1-regularization built-in the XGBoost model.

Feature	Feature's MDA Weight(std)
Creatinine -1	0.070(0.005)
Creatinine -3	0.053(0.001)
GCS -5	0.013(0.002)
Extubate	0.011(0.001)
Bulging	0.007(0.000)
Resp rate -4	0.007(0.002)
Heart Rate -1	0.005(0.000)
.....	

Table II.2: MDA's head table results

[2]10-folds cross-validation with 80% training set, 10% validation set and 10% test set use later for generalization error.

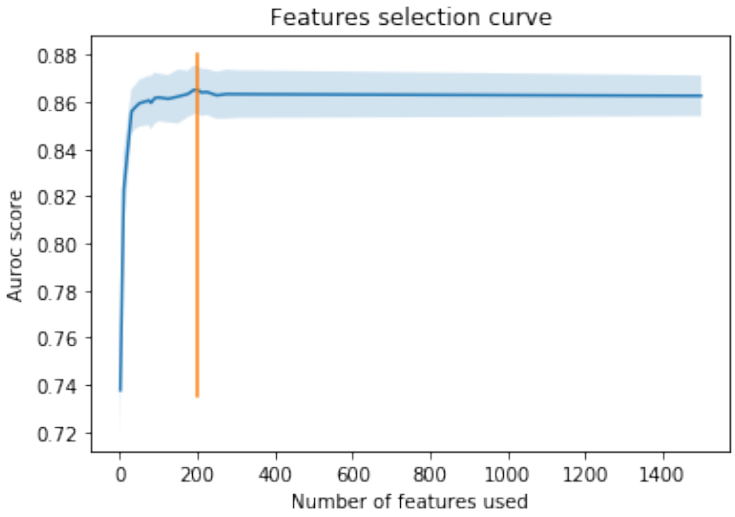


Figure II.1: Feature selection using sklearn built-in function(10-folds-cross-validation)

II.3 First results

II.3.1 Minimalist models

A striking result is that the model is already able to have a surprisingly high score only using one feature, in this case GCS-5. Moreover, according to the MDA, two other features are very important for the prediction, here creatinine-3 and creatinine-1. For example, we can see on Table II.2 that if we replace all the values of the feature Creatinine-1 by random noises following the same distribution as the original values, the model would worsen his score by 7%.Yet, it's not surprising since the creatinine level is strongly linked to the definition of AKI. However, there is no sign in recent papers of such a big impact of the Glasgow Coma Scale to predict AKI. The model using only GCS-5 is almost able to compete with other released models, achieving an Auroc[3] score of 0.74 ± 0.06.

To understand this, density plots and box plots were realized, available in figure II.2, II.3 and II.4.

The first thing to say is that there is some aberrant values (very high or negative creatinine level for example). Those values were kept and are probably a malfunction of the monitoring device.
Creatinine level seems to have more or less the same density between controls and cases, yet it's strange that at low level the density for AKI is higher. However there is a clear difference between the two categories when looking at GCS. It looks like being fully "awake" is a strong indicator for the model that there is no risk of AKI, since the density of controls at score 15 is very high compared to the rest.It is also important to remember that GCS-5 and Creatinine-3 are the mean value of all the previous measures.The density plot for the temperature is displayed figure II.4 to compare with the two others.

[3]Area Under The Curve Receiver Operating Characteristics is a metric used to evaluate the performance of a machine learning model. It is based on the ROC curve, comparing true positive rates (true labels classified as true) and false positive rates (false labels classified as true).

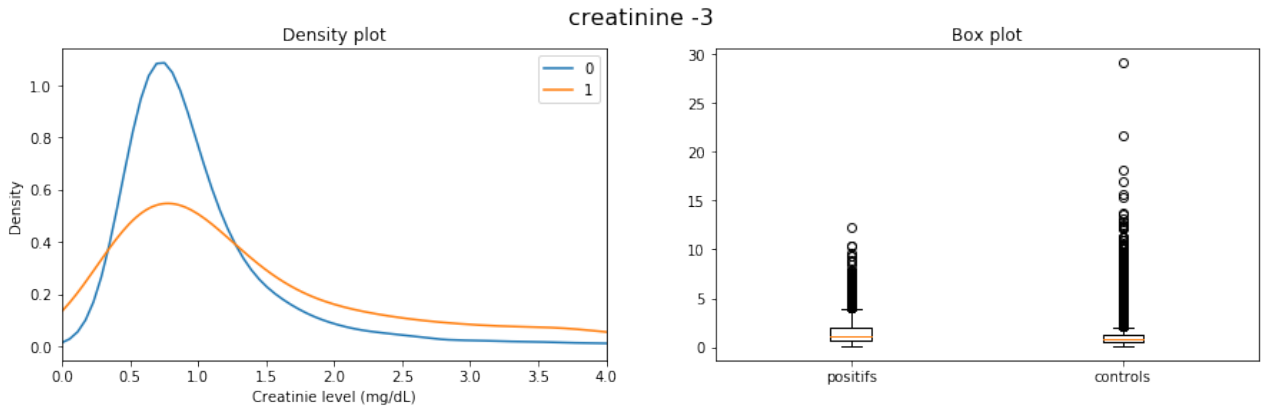


Figure II.2: Creatinine-3 density and box plot

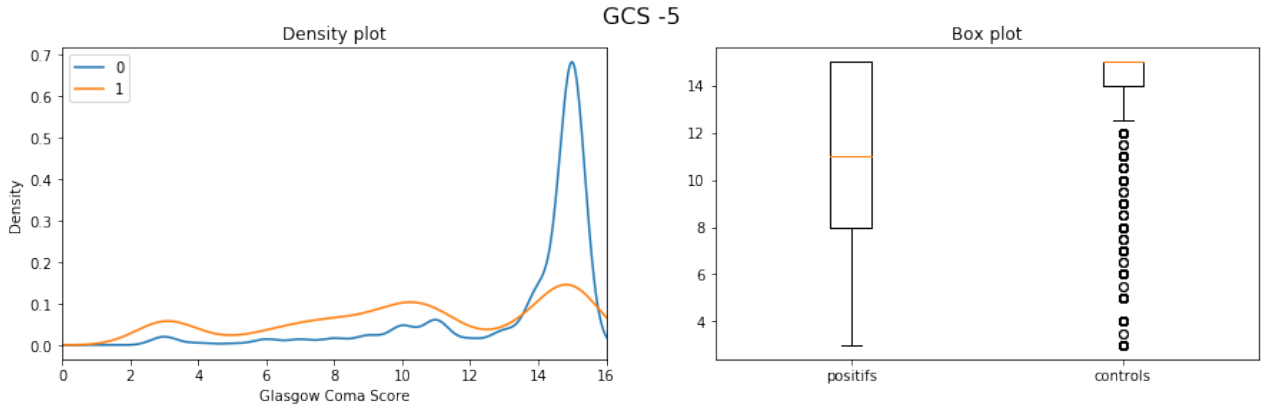


Figure II.3: GCS-5 density and box plot

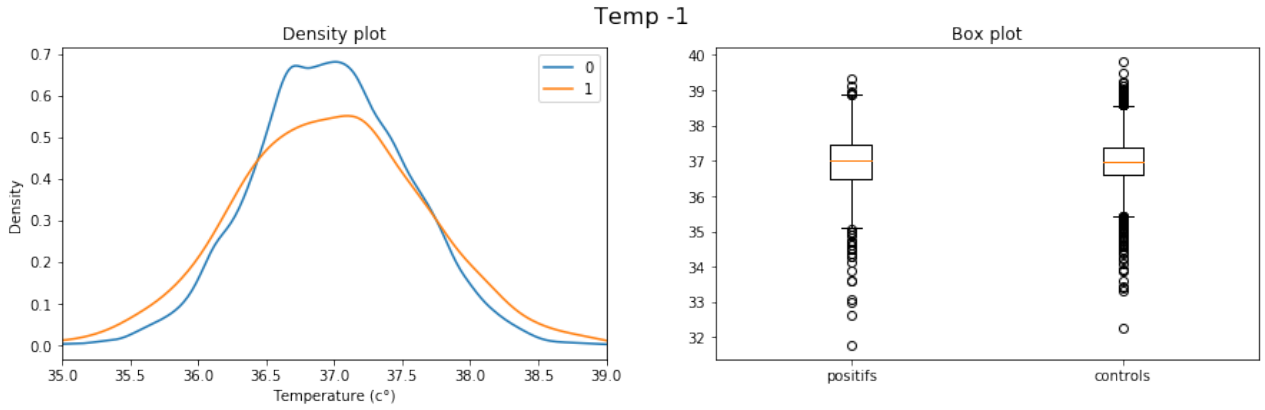


Figure II.4: Temperature-1 density and box plot

II.3.2 Models without hyper-parameter tuning

The following table sums up the results obtain with XGBoost default model to predict 24 hours in advance. The only parameter tuned is to deal with the unbalanced dataset by overweighting the minority class (different sampling techniques were also tried but yielded similar results).

It appears that the two NLP techniques yield the same results. In the rest of this report, we will stick to SemEHR. The table II.3 shows that features selection was able to keep the

Model	SemEHR + all features	SemEHR + top 200 features	MedCat + top 200 features	Stanford
Score(95% CI)	86.4(1.5)	86.5(0.3)	86.4(0.4)	75.8(0.4)

Table II.3: Comparaison of the scores for feature selection

Auroc score as high as before, but reduced by a lot the standard deviation. This is logical since the model avoid overfitting and therefore the results are more consistent across the different folds during cross-validation.

Part III

Building and understanding the model

III.1 XGBoost hyper-parameters

To further enhance the performances, some hyper-parameters tuning was done. The goal is to fit some parameters of the algorithm to the training data, to evaluate multiple times the score on validation sets and finally to assess the performance on an unknown test set to verify that the model is not over or underfitting. If the model is over-fitting, it means that the parameters chosen were too specific to the validation set, and therefore a drop of performance in the test set is recorded. To proceed, a grid search was realized using the GridSearchCV function implemented in Scikit-learn package. This function allows us to test multiple values for multiple hyper-parameters and find the best combination through cross-validation sets. The resulting parameters are presented in Table III.3.

Once this part finished, a final evaluation was realized on the test set, never seen by the model yet.

	Precision	Recall	F1-Score	Support
Controls	0.94	0.96	0.95	1833
Cases	0.65	0.57	0.61	242
Accuracy	-	-	0.91	2075
Macro average	0.80	0.77	0.78	2075
weighted average	0.91	0.91	0.91	2075

Table III.1: Sklearn evaluation report on the test set

XGBoost model	All features	Top 200 features	Top 200 + Hyper-parameters	Test set
Auroc Score(95% CI)	86.4(1.5)	86.5(0.3)	87.5(0.3)	87.4

Table III.2: Model score's evolution through all the steps

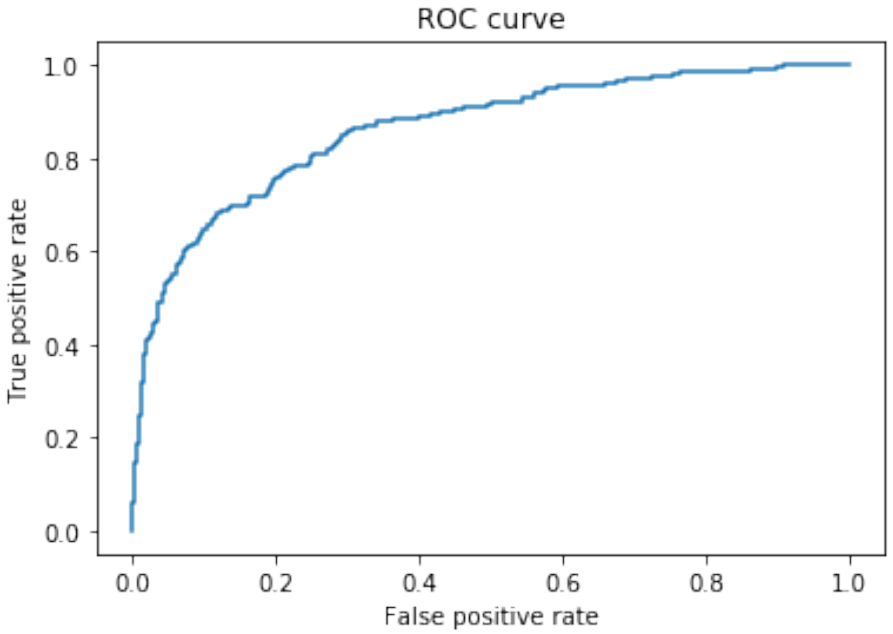


Figure III.1: ROC curve

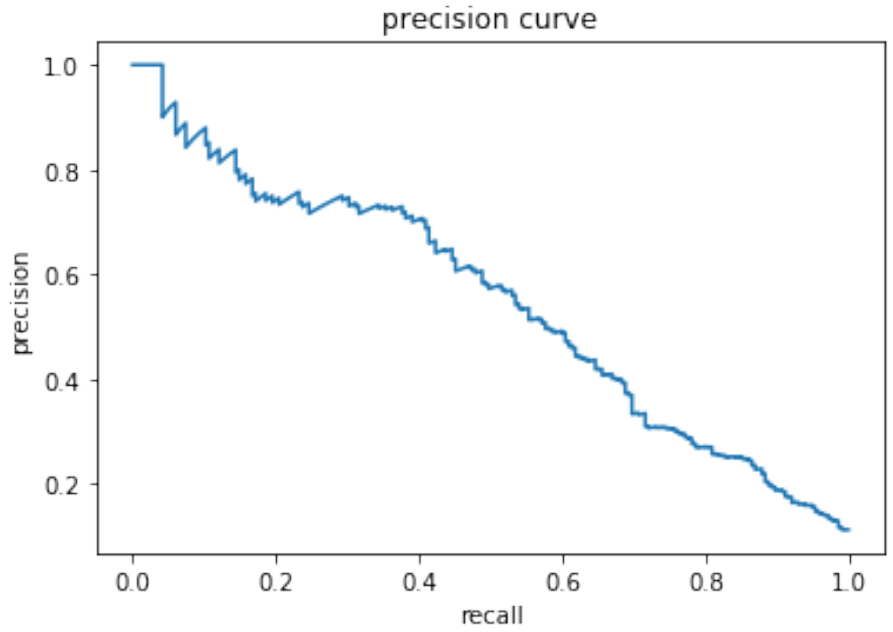


Figure III.2: Precision Curve

III.2 Analysis of the features

III.2.1 SHapley Additive exPlanations

Most of the time, analysis of a model is limited to feature importance. Firstly, this can be tricky, for example a tree-based model can judge the importance of a feature based on gain, coverage or frequency of a feature among the different trees. This is not consistent. Moreover, this only indicates the relative impact of a feature toward the performance score, not the way the feature is actually used to do the prediction. That's why a new technique is being used

Hyper-parameter	Value retained	Explanation
max_depth	5	The maximum depth of a single tree. A higher value controls over-fitting by preventing the model to learn too specific relations. 5 is a mid-range value.
min_child_weight	6	The minimum sum of weights in all samples required in a "child", ie a new leaf of a tree. Same effect as max_depth. A high value might lead to under-fitting. 6 is above average, but this can be explained due to the unbalanced dataset and over-weighting the cases.
learning_rate	0.05	it controls the weighting of new trees added to the model to correct residual errors. A low value prevents over-fitting. 0.05 is a mid-range value.
n_estimators	600	The number of decision trees in the model. Should be as high as possible as long as there is no sign of over-fitting. Range values depend on the complexity of the problem, but the higher the number is the longer it takes to fit the data.
gamma	0.2	The minimum loss reduction required to make a split. Also known as the Lagrangian multiplier. It depends on the loss function used. The higher it is the more conservative the algorithm is, which means that it is a regularization parameter.
colsample_bytree	0.75	The random sample of features randomly used for a new tree. This is useful to have a wider spectrum of predictors (trees) and try random features' combination. 0.75 is a mid-range value.
subsample	0.95	The fraction of samples used for each new tree. This is used to have a more conservative model. 0.95 is in the typical value range.
reg_lambda	0	Same as Ridge or L2 regularization. Used to avoid over-fitting and therefore have a more conservative model. It adds the squared error coefficient as a penalty term to the loss function. 0 means that it is not used.
reg_alpha	5	Same as Lasso or L1 regularization. Used to avoid over-fitting and therefore have a more conservative model. It adds the absolute value error coefficient as a penalty term to the loss function. This means that it can actually select features by shrinking to zero the impact of unsuitable ones, compared to L2 which only tends towards zero. This is very useful here since we have a lot of features. 5 is a high value.

Table III.3: XGBoost hyper-parameters after optimization via GridSearch

nowadays, resolving both those issues. SHAP (SHapley Additive exPlanations) is " a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods and representing the only possible consistent and locally accurate additive feature attribution method based on expectations."[4] Figure III.3 shows the plot of SHAP values. As an example, let's look at GCS-5:

- GCS is the most important feature of the model (top of the plot).
- A low GCS (blue) increases the probability to have AKI, since those values are in the right part of the plot.
- The opposite is also true, a high GCS (red) decrease the output probability to have AKI, better those values are in the left part of the plot.
- There is a high concentration of the same value in the red part of GCS. If we go back to figure II.3, we can link this to the high density of scores equal to 15. Therefore the reason of the wider part.

From this plot, we can then gain some understanding of the model.Among the top 5 features, we found 3 structured features (GCS-5, Creatinine -3 and Creatinine -1) and 2 concepts (Lasix and Extubate). A high mean creatinine value is strongly linked to AKI, which is concordant with the medical definition. However a strange thing is that a recent low Creatinine level is also linked to AKI. We didn't find a good explanation, and this matter needs to be discussed further. A high occurrence of the concept Lasix (the drug is furosemide) is linked to AKI. There is not a consensus about how Lasix impact AKI patients in different papers [13]. The drug is used sometimes to treat AKI, but different results are available, from positive to detrimental. Still, there is strong evidence for the model that the more the concept is detected, the higher the probability to have AKI. On the other hand, many other drugs were found in the top 50 features (not visible on the figure III.3). For example Captopril, which is known to triggers acute kidney injury (AKI), mainly via aggravating hypoxia, oxidative stress, inflammation and renin-angiotensin system activation[14]. In this case, we could indeed see that a higher occurrence of this concept was linked to higher risk of AKI. For Extubate, an explanation might be that this is a sign that the patient is recovering, so the occurrence of the word tends towards no AKI. Also, the third highest concept "chest" might look irrelevant. However according to the symptoms of AKI stated in the introduction, AKI can lead to chest pain, and that might be the meaning of it.

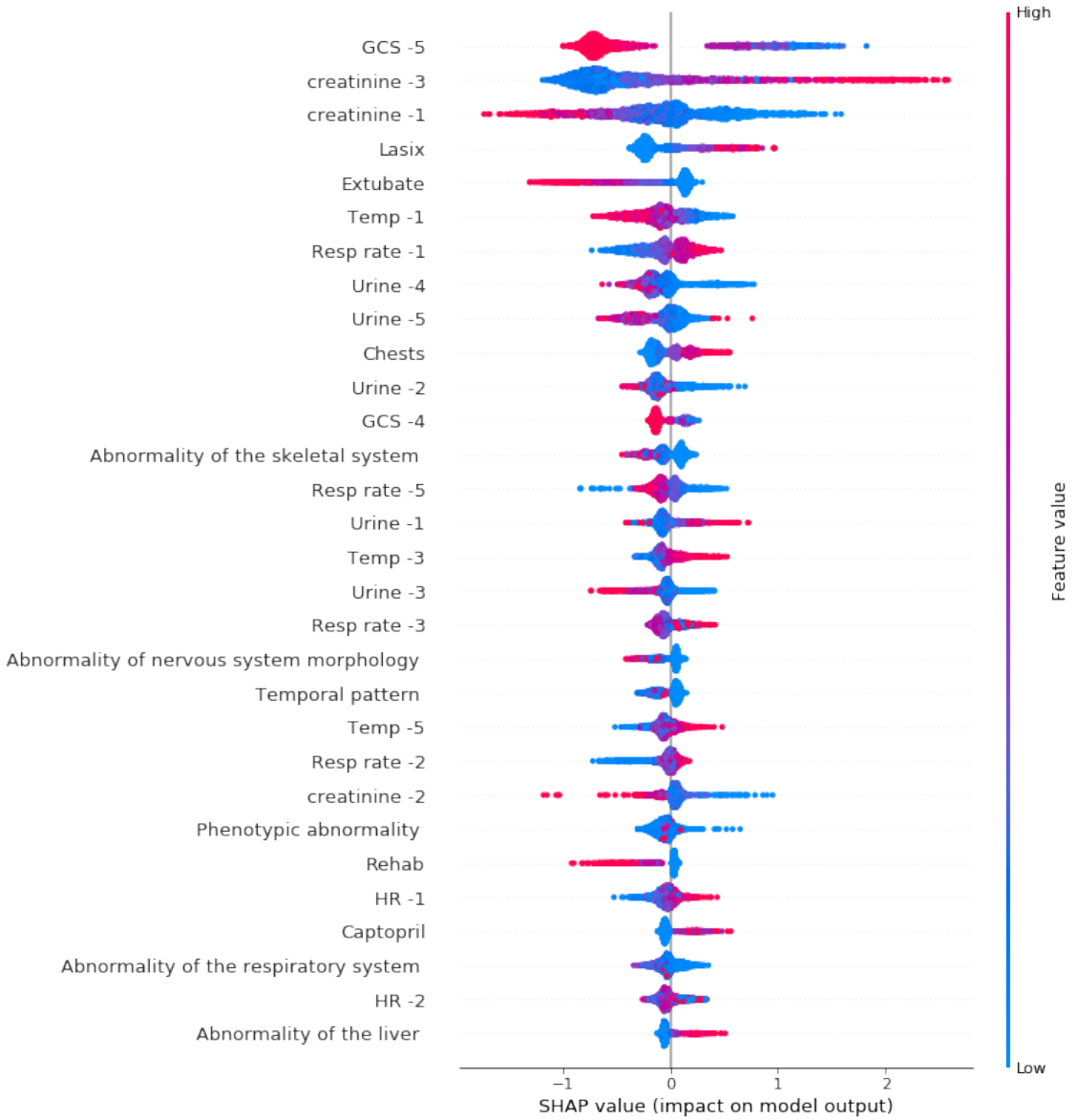


Figure III.3: SHAP analysis output

III.2.2 Clustering top predictions

Another idea to better understand the model was to use unsupervised learning focused on top probabilities, in this case with an output probability of 90% or more. The goal is to determine if there are different "ways" according to the model to be diagnosed with AKI. To cluster this subsample of the dataset, the k-mean algorithm was used. K-mean requires a specified k number of clusters to found to work properly. The elbow technique was used to determine the optimal number of clusters, as shown in figure III.4, and k was set to 6. The resulting clusters were then analyzed using SHAP, to determine which features were the more salient for each of them. Three groups can be extrapolated from those results. Firstly, five of them have GCS-5 as the dominant feature used for prediction, they could be merged as one bigger cluster. Then, among those five clusters, one stands out having the concept "Lasix" with higher importance than all the others (behind GCS-5). Finally, one cluster has a different top feature, being the creatine-3. A possible way to see it is that it is possible to differentiate high probability profiles based on a global illness of the body(GCS-5), one more specific to

kidney illness (creatinine-3), and finally one where the drug used is either interpreted as a worsening, or maybe an adverse effect of the drug overloading the kidneys.

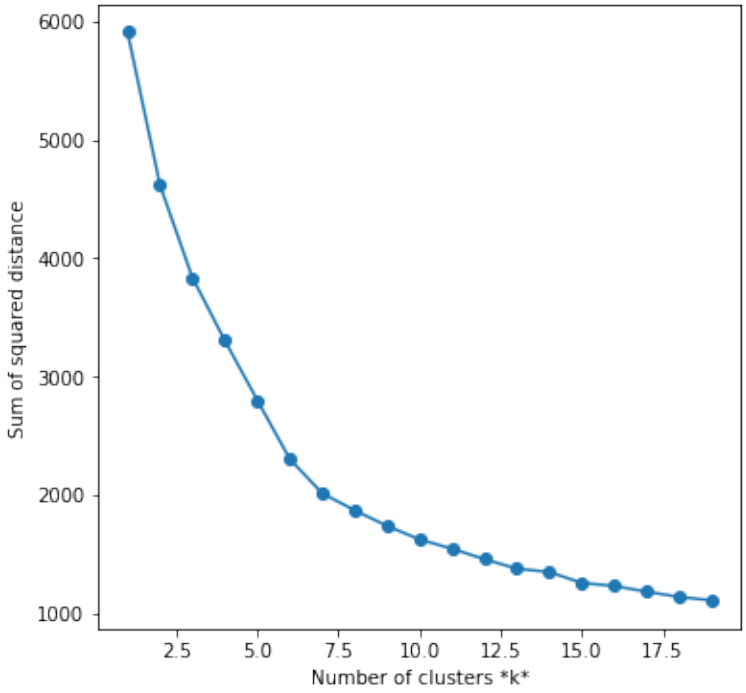


Figure III.4: Elbow curve to find the optimum number of clusters

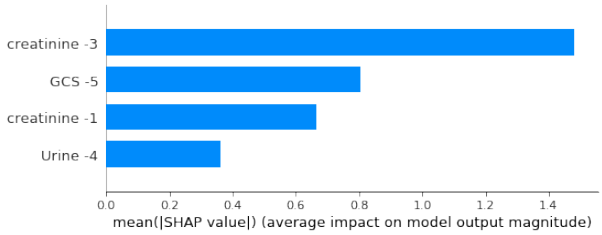


Figure III.5: Features' importance within the cluster 1

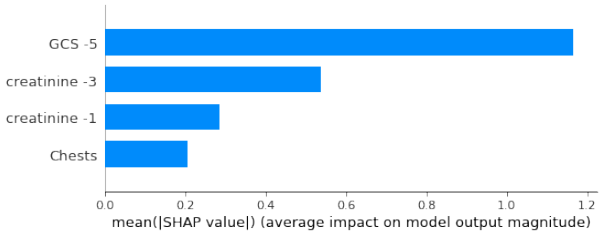


Figure III.6: Features' importance within the cluster 2

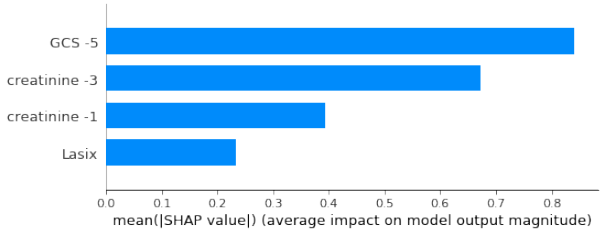


Figure III.7: Features' importance within the cluster 3

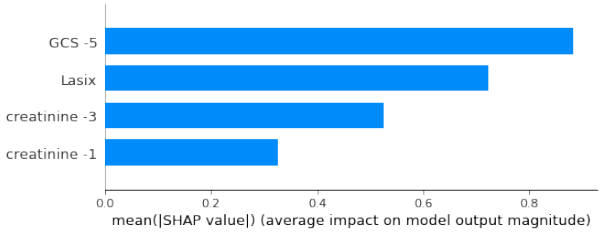


Figure III.8: Features' importance within the cluster 4

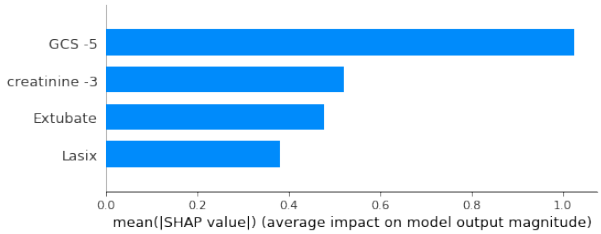


Figure III.9: Features' importance within the cluster 5

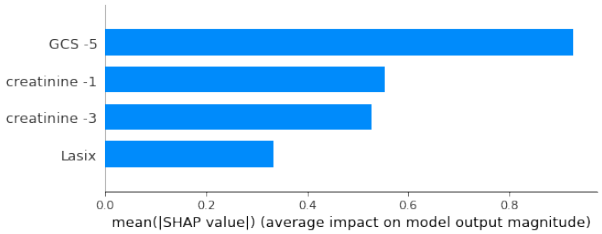


Figure III.10: Features' importance within the cluster 6

III.3 Neural Network

To challenge the previous model, an attempt was made to built a model based on a different technique. A neural network was realized using Pytorch's framework. The main issue with this technique is that the model would almost always overfit, or if the number of epochs was reduced it would underperform the current XGBoost model. It achieved a peak Auroc score of 0.80 on the validation set. I reached Doctor Zeljko Kraljevic, a permanent member of the department specialized in NLP and Neural Networks to discuss the different possibilities. We concluded that overfitting the data is a good sign, it means that at least the model is able the learn the patterns to predict AKI. Moreover, it is theoretically possible to aim in between the overfitted score and the actual score with some Neural Networks tools and maybe using a slightly different dataset.
The first architecture I built was a Multi-layer Artificial Neural Network, with 3 linear hidden layers of 50, 100, and 10 neurons respectively, using rectified linear units as the activation function. The output layer had one neuron to generate the probability, using a sigmoid activation function. The network had a BCELoss criterion with a Stochastic Gradient Descent optimizer.
Here is a list of the different proposition made to improve the model and avoid overfitting:

- Use a different optimizer, for example Adam.
- Add dropout.
- Use batch normalization to avoid updating the network after each sample, but rather after a certain number of them.
- Add some regularization.
- Use less features, and create categorical one rather than continuous values. For example, rather than to say to the model that we had 1,5,6,15,0 occurrences for a certain word, we create a category saying if there is between 0 and 10 occurrences, then 10-20 and so on.

I was able to try dropout, changing the optimizer and batching. but the first results impacting directly the model (dropout and new optimizer) were not conclusive. it is possible that modifying the dataset would have yielded better results.

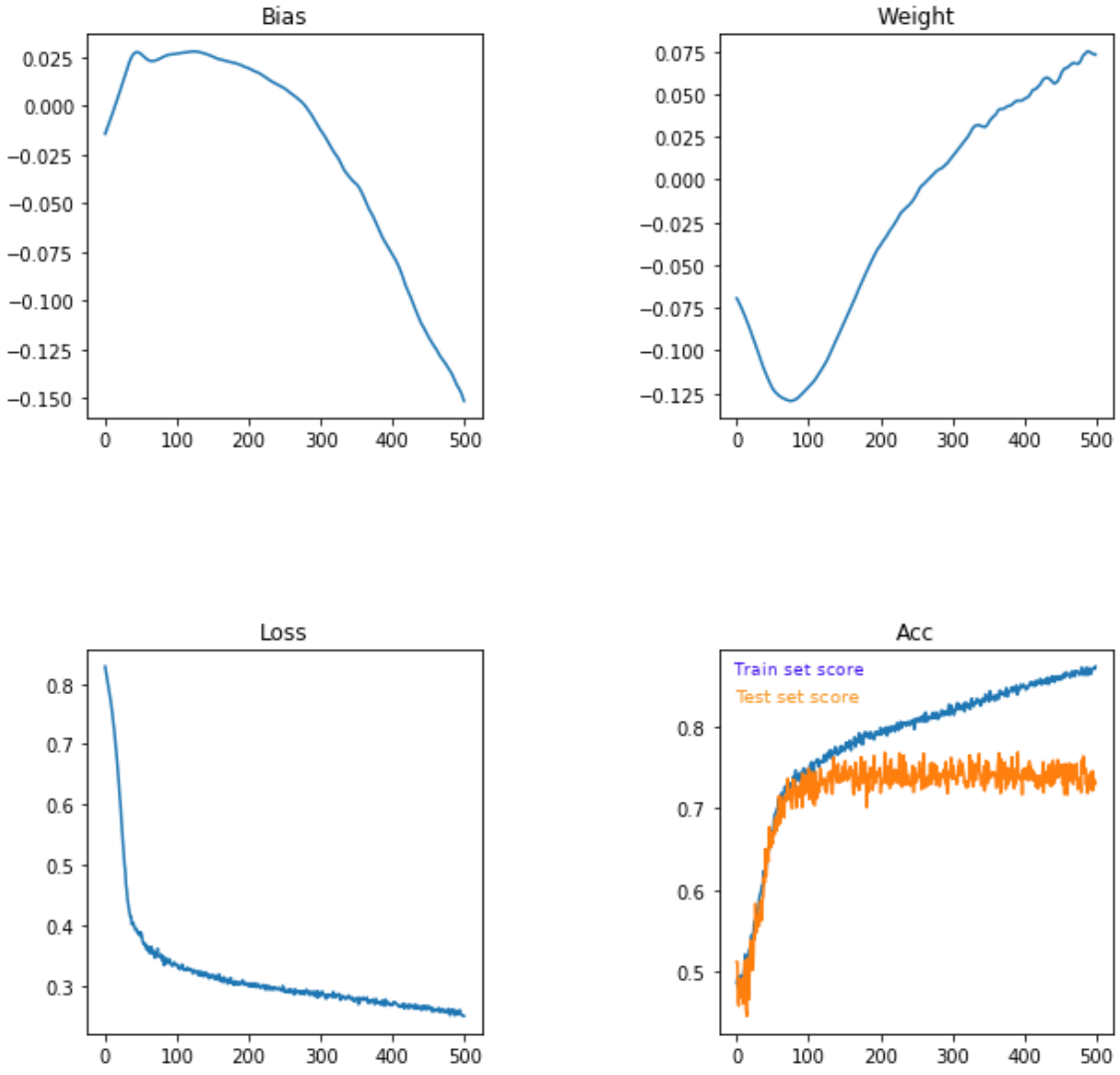


Figure III.11: Network's evolution during the epochs

Part IV

Real-time analysis of the model

During the whole process of building the machine learning model, we always had in mind that the model should be able to run continuously and output the probability to have AKI in real-time for a patient.To simulate a live implementation inside the hospital, we used the datasets built in order to predict 12,24,36,48 and 72 hours in advance, following the same patients if possible. We then recorded the probabilities to have AKI in the next 24 hours using our model.

IV.1 Real-time probability

The figure IV.1 shows that even if cases and controls probabilities overlap each other during the process, the model seems to get more and more confident when predicting if someone is not going to have AKI. The same effect if also present following the mean probability for cases, but to a lesser extent. A possible explanation is that due to the unbalanced dataset, the model has more example of controls, and is performs better on the control class (see figure II.3 for the precision and recall score for controls compared to cases). Also, the closer to the 24h threshold, the better our 24-hours model is, as one would expect. Notwithstanding the standard deviation, it is still an encouraging result to see that controls' probability goes down and cases' probability goes up on average.

IV.2 Further investigation of patient trajectories

Among all the patients, we wanted to focus especially on those where the model prediction changed during a long stay, and try to understand why this change was happening.Those groups were detected by running a first prediction among all the patients with the model at 24h and 72h time point, then applying a filter in the dataframe where the probability at 72h was under/above the 50% threshold, and the other way around for 24h time point.

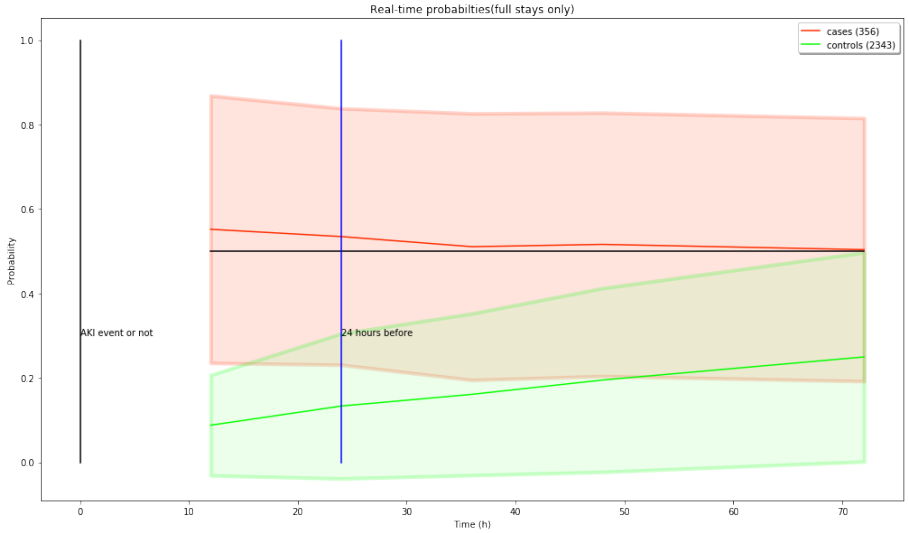


Figure IV.1: Evolution of the probability during 72-hours-stays(*mean in full-line and standard deviation in shade*)

IV.2.1 Decreasing probability patients

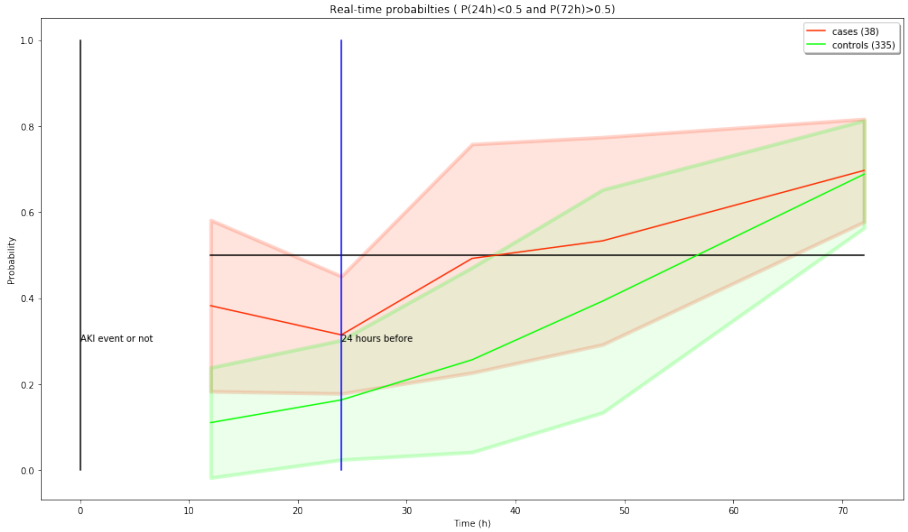


Figure IV.2: Evolution of the probability for decreasing probability patients(*mean in full-line and standard deviation in shade*)

For the patients where the probability to have AKI is decreasing, it looks like the main reason is an increasing GCS on average, and the occurrence of the concept "Extubate". Those are signals mainly saying that the overall state of the patient is improving.

Figure IV.3: Shap results at 24h time point(AKI high probability)

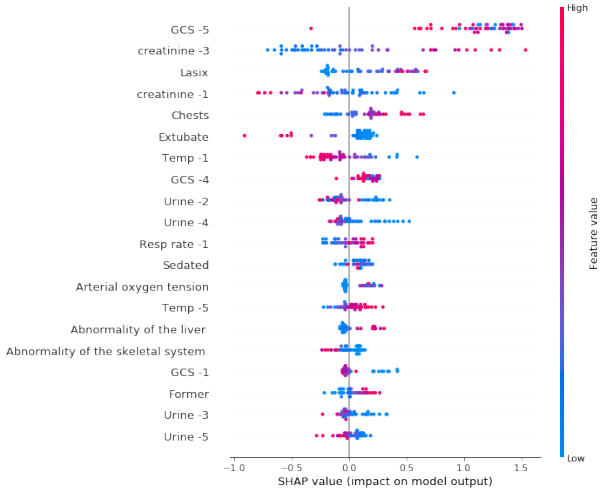


Figure IV.4: Shap results at 72h time point(AKI low probability)

IV.2.2 Increasing probability patients

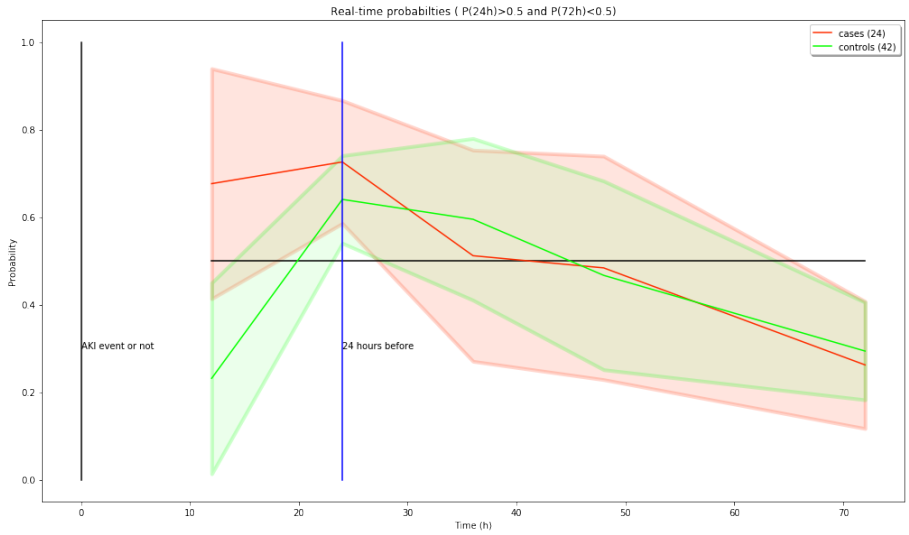


Figure IV.5: Evolution of the probability for Increasing probability patients(*mean in full-line and standard deviation in shade*)

For the patients where the probability to have AKI is increasing, the reasons are less obvious. There is still a shift in the GCS, with lower values at 24 hours than 72 hours. The features that are interesting to highlight are creatinine-3, Chest, and Urine-4. We can see higher creatinine outputs and lower urine outputs, signals that a physician could link to AKI easily. The concept "Chests" is more recurrent and important according to the SHAP value. We can suppose that this is linked to "chest pain", one of the symptom of AKI.

IV.3 Real-time Diagnosis

Something important for the physician is to understand why the model thinks that a specific patient is predicted a certain way or not. The SHAP package provides such tools, where it's

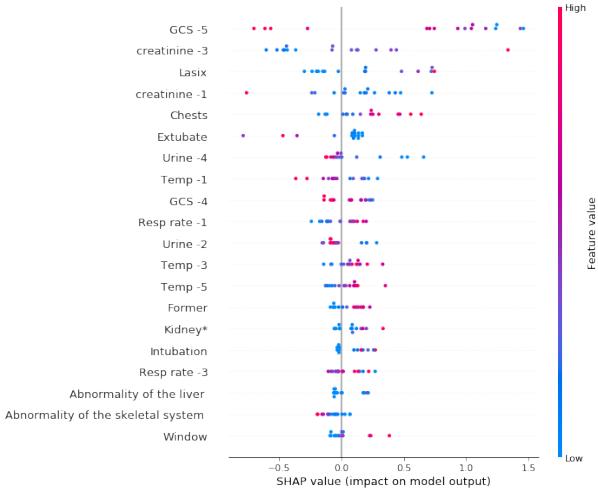


Figure IV.6: Shap results at 24h time point(AKI low probability)

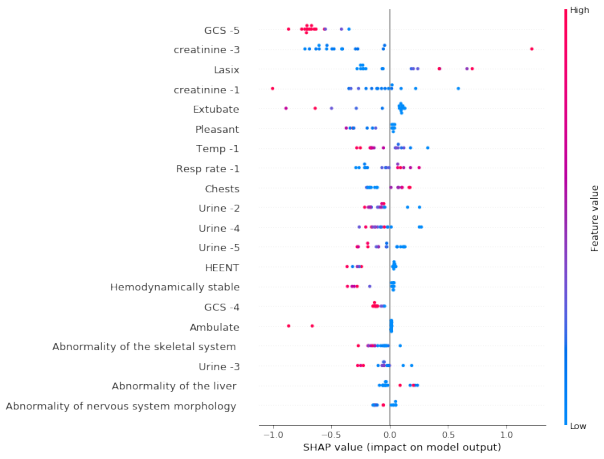


Figure IV.7: Shap results at 72h time point(AKI high probability)

possible to have insights on an individual prediction. If we look at Figure IV.9, the patient here "is going to have AKI in the next 24h", meaning that he is classified in the dataset as a case. Our model ouputs a probability of 85% to have AKI in the next 24h, in this case he is then right. Then we have the SHap tool, that explains to us that for this patient the main reason to expect AKI is a low GCS-5 (6 in this case) and the main reason not to expect AKI is the mean creatinine level (creatinine-3). In the end, the pros factors out weight the cons factors, and the output value is higher than the base value, so the patient is to be classified positively to AKI.
This could be very helpful since it allows the physician to have a quick overview of the situation. For example, in figure IV.8, even though the model is wrong if we only look at the output probability, the individual plot gives three important information: The patient had Lasix and Atrovent (two drugs), and has an abnormality of the liver. This might be enough for a physician to realize that there is a higher risk than predicted.

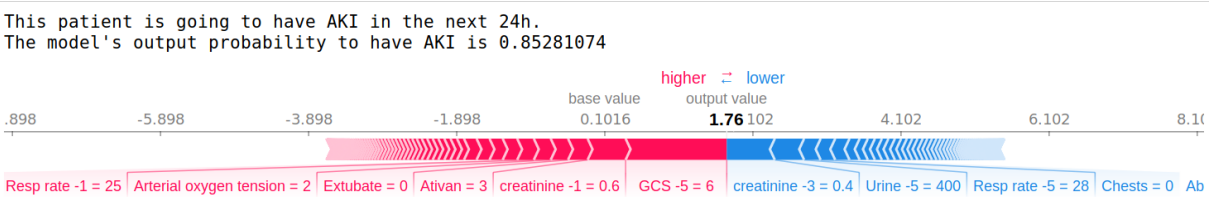


Figure IV.8: Individual diagnostic 1:Model predicts AKI

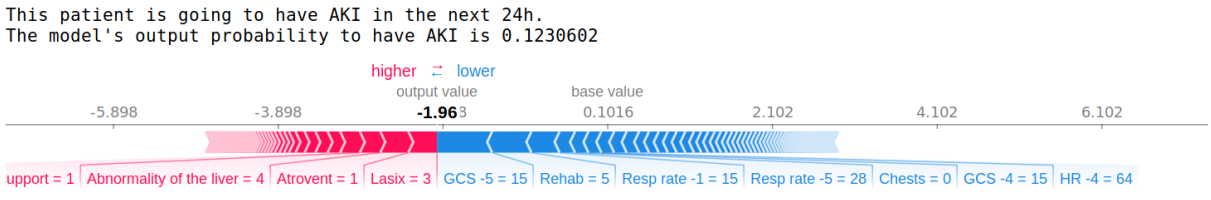


Figure IV.9: Individual diagnostic 2:Model fails to predict AKI

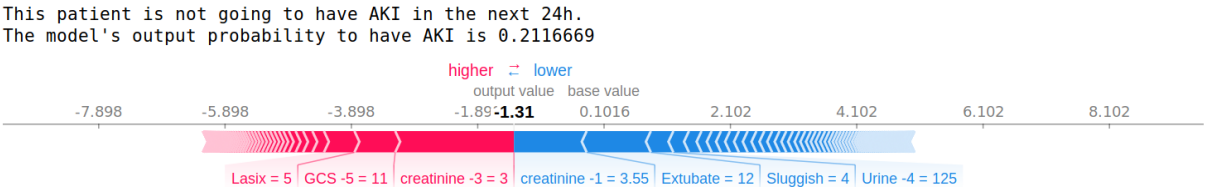


Figure IV.10: Individual diagnostic 3:Model predicts no AKI

Conclusion

During this project, we were able to enhance an already existing model by focusing on feature selection and hyper-parameter tuning. The resulting XGBoost model used 200 features to reach an 87.6% Auroc score 24 hours in advance. We then enhance our understanding of the model using mainly SHAP analysis. Those results mixed with more focused analysis based on clustering top predictions shown that the features with the highest impact to predict AKI were GCS-5, Creatinine-3 and Lasix. We finally studied the evolution of the model's outputs through simulated real-time implementation. SHAP's tool allowing individualized understanding of a prediction also shown interesting possibilities if the model were to be used alongside a physician's diagnostic.

	Project	Courses	Meetings
Week 1	Bibliography	Ethics of Data	
Week 2	Dataset manipultation		Team meeting
Week 3	Real-time probabilities		
Week 4	Feature selection	Introduction to 6* NLP in medical context	Department meeting
Week 5	Hyper-parameters		
Week 6	SHAP analysis		Team meeting
Week 7	Clustering		
Week 8	2*Neural Netwok		
Week 9			Team meeting (Presentation of my work)
Week 10	Code formating		
Week 11	2*Report		
Week 12			Department meeting

Table IV.1: Planning of the project

Of course, some improvements or further analysis are possible. The following items are the next possible steps or discussions to be made:

- The neural network should be able to outperform the XGBoost model if built properly. This is even more true since a very recent paper from DeepMind shown that recursive neural networks could achieve Auroc score above 90% when predicting AKI (but on a different dataset)[3].
- It is also possible to follow the path of STACKING models or ensemble models, where more models are combined in order to gain performance. Such techniques are most of the time a must-have, but take time to be built[15].

- More research could be done in order to understand the behaviour of the model regarding creatinine. Some feedbacks we received from members of the department encouraged us to compare the creatinine level and the drugs used. The idea behind it is that in some cases, patients could already be treated for AKI, and the creatinine dropping is only a result of this. In this case, the model interprets this as "this patient is going to have AKI".

- It could be interesting to know if our pre-built model could be directly used in another hospital with the same results. This question is partially answered since another student working on the project tried the former model on the UCLH database and the performance were almost similar to the one we had. This is very encouraging.

- Finally, we tried to outperform the Stanford paper using the same dataset and algorithm, only adding medical concepts. Therefore, we used the same structured data. Yet, it looks like a very important feature not used here is blood pressure, and adding this could have a huge impact since this is highly correlated to AKI[16].

Bibliography

[1] Christopher Barton Hamid Mohamadlou, Anna Lynn-Palevsky. Prediction of acute kidney injury with a machine learning algorithm using electronic health record data. *Canadian Journal of Kidney Health and Disease*, 5, 2018.

[2] Peter Martin Alistair Connell, Hugh Montgomery. Evaluation of a digitally-enabled care pathway for acutekidney injury management in hospital emergency admissions. *Nature*, 67, 2019.

[3] Jack W. rae Michal Zielinski Harry Askham Andre Saraiva Anne Mottram Clemens Meyer Suman ravuri Ivan Protsyuk Alistair Connell Cían . Hughes Alan Karthikesalingam Julien Cornebise Nenad tomašev, Xavier Glorot. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 117:116–125, 2019.

[4] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[5] National Kidney Foundation. Acute Kidney Injury (AKI). https://www.kidney.org/atoz/content/AcuteKidneyInjury. Accessed: 2019.

[6] Ravindra Parmar. Common Loss functions in machine learning. https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23. Accessed: 2019.

[7] xgboost developers. Introduction to Boosted Trees. https://xgboost.readthedocs.io/en/latest/tutorials/model.html. Accessed: 2019.

[8] Stringer Clive Dobson Richard Jackson Richard, Kartoglu Ismail. Cogstack - experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital.

[9] Katherine I Morley Zina Ibrahim Amos Folarin Ismail Kartoglu Richard Jackson Asha Agrawal Clive Stringer Darren Gale Genevieve M Gorrell Angus Roberts Matthew Broadbent Robert Stewart Richard J B Dobson Honghan Wu, Giulia Toti. Semehr: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment and clinical research.

[10] M. Radaelli G. Locafaro S. Simblett C. Barattieri di San Pietro V. Bulgari P. Burke J. Devonshire J. Weyer T. Wykes G. Comi M. Hotopf I. Myin-Germeys A. Rintala, F. Matcham. Emotional outcomes in clinically isolated syndrome and early phase multiple sclerosis: a systematic review and meta-analysis. *Journal of Psychosomatic Research*, 124, 2019.

[11] Power BM Ho KM. Mimic-iii, a freely accessible critical care database. *US National Library of Medicine*, 2010.

[12] Royal College of Physicians and Surgeons of Glasgow. Glasgow Coma Scale. https://www.glasgowcomascale.org. Accessed: 2019.

[13] M. Radaelli G. Locafaro S. Simblett C. Barattieri di San Pietro V. Bulgari P. Burke J. Devonshire J. Weyer T. Wykes G. Comi M. Hotopf I. Myin-Germeys A. Rintala, F. Matcham. Benefits and risks of furosemide in acute kidney injury. *Scientific Data*, 2016.

[14] Shai Efrati, Sylvia Berman, Ramzia Abu Hamad, Yariv Siman-Tov, Eduard Ilgiyaev, Ilia Maslyakov, and Joshua Weissgarten. Effect of captopril treatment on recuperation from ischemia/reperfusion-induced acute renal injury. *Nephrology Dialysis Transplantation*, 27(1):136–145, 2011.

[15] Venko Bernard Deroski Saso. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273, 2004.

[16] Nasu M Sato R, Luthe SK. Blood pressure and acute kidney injury. *Crit Care*, 21, 2017.

Glossary

AKI Acute Kidney Injury

GCS Glasgow Coma Scale

ICU Intensive Care Unit

IoPPN Institute of Psychiatry, Psychology and Neuroscience

MDA Mean Decrease Accuracy

MIMIC Medical Information Mart for Intensive Care

NHS National Health System

NLP Natural Language Processing

RADAR-CNS Remote Assessment of Disease and Relapse – Central Nervous System

SHAP SHapley Additive exPlanations

List of Tables

List of Figures

Appendix

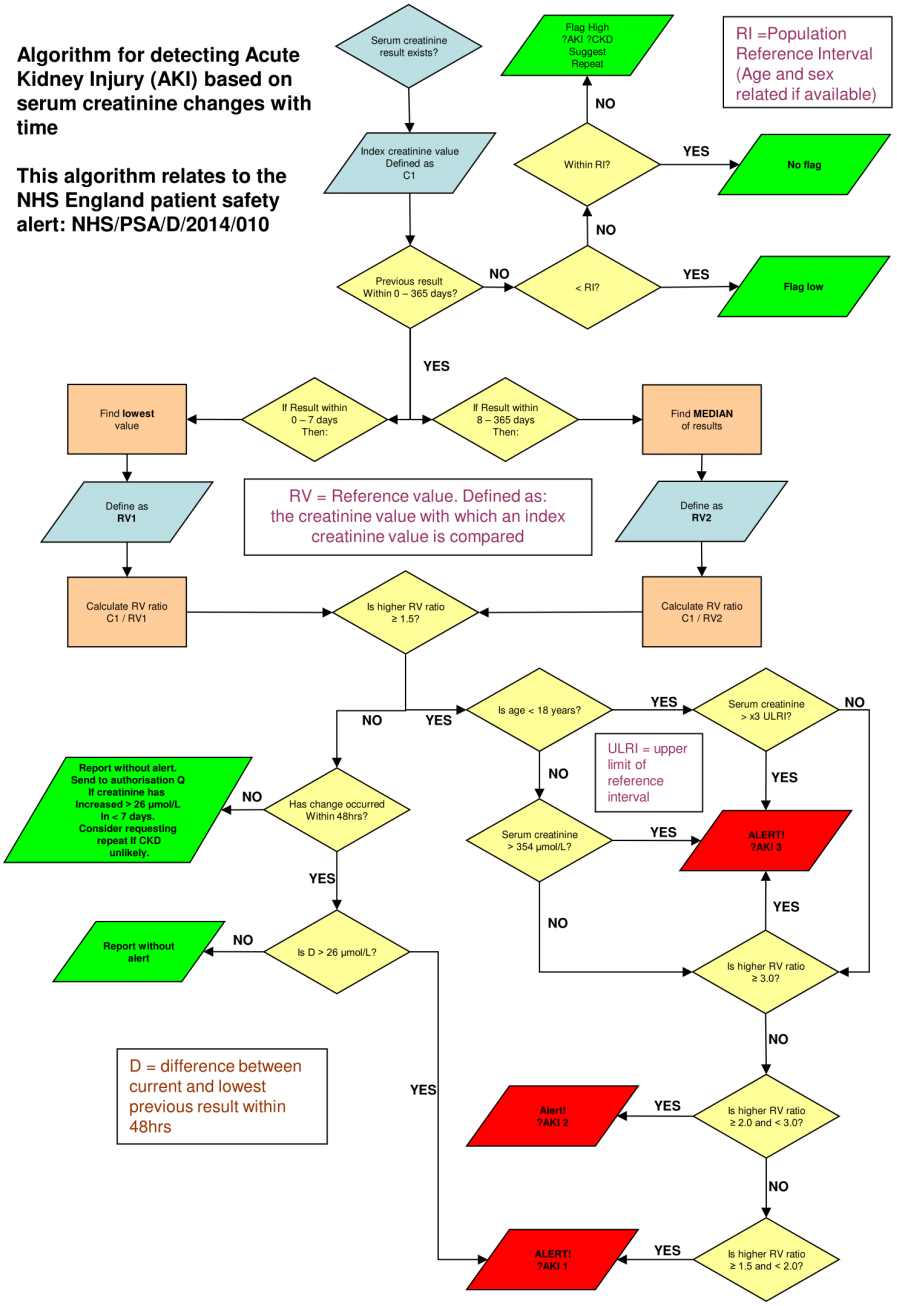


Figure IV.11: Gold Standart Algorithm