Early predicting the need for aftercare based on patients events from the first hours of stay – A Case Study

Annika L. Dubbeldam[1 [0000−0002−5042−2215]], István Ketykó[1 [0000−0003−4931−4580]], Renata M. de Carvalho[1 [0000−0001−6129−9278]], and Felix Mannhardt[1 [0000−0003−1733−777X]]

Department of Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, The Netherlands
a.l.dubbeldam@student.tue.nl, {i.ketyko, r.carvalho, f.mannhardt}@tue.nl

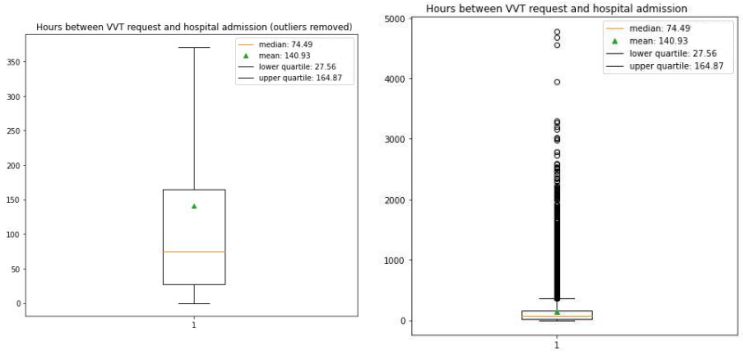
Abstract. Patients, when in a hospital, will go through a personalized treatment scheduled for many different reasons and with various outcomes. Furthermore, some patients and/or treatments require aftercare. Identifying the need for aftercare is crucial for improving the process of the patient and hospital. A late identification results in a patient staying longer than needed, occupying a bed that otherwise could serve another patient. In this paper, we will investigate to what extent events from the first hours of stay can help in predicting the need for aftercare. For that, we explored a dataset from a Dutch hospital. We compared different methods, considering different prediction moments (depending of the amount of initial hours of stay), and we evaluate the gain in earlier predicting the need for aftercare.

Keywords: Early outcome prediction · Healthcare · Patient events · Aftercare demand.

1 Introduction

Many people are admitted into a hospital every day, all of them different, taking their own personalized track, this makes for a lot of variability [10]. However, there is one thing all of these patients have in common during the hospitalization process, someone has to decide if the patient requires aftercare.

Currently, during the patient stay, a nurse might identify the need for aftercare and file an order. This means that some patients can be identified as soon as they enter the hospital, whereas others will only be identified near the end of their stay. As it takes time for the aftercare organizations to make room for a new patient, identifying patients that need this care very late means that they have to remain in hospital (even after their medical discharge date) in order to wait for the next available space. Patients that have to wait in the hospital no longer require any specialized treatment that can only be performed there. Furthermore, as they cannot be moved on, they will remain in their bed occupying a space that could be used by other patients that do require specialized treatment.



(a) without outliers

(b) with outliers

Fig. 1: Time for a patient to be identified as in need of aftercare

In order to establish the importance of this we must derive how long it currently takes the hospital to identify aftercare patients. Fig. 1 depicts two box plots, showing how long (in hours) it took after admission for the patient to be marked for aftercare. Currently, on average after 140 hours (or median of 74 hours) a patient first gets noted. This leaves quite a lot of room for possible earlier detection, and consequently earlier arrangements with the aftercare organizations to make sure a place is available as soon as the patient gets discharged.

By noting these patients early on in their stay gives the hospital employees more time to inform the aftercare locations causing the patient's possible in-hospital wait time to be reduced. In this paper we aim to explore the possibility to identify those patients who need aftercare and to evaluate how early on during the process this can be done. For that we plan to make use of various decision tree related models which we will give different inputs (patient data with events from admission until a certain moment in time) to determine simultaneously if it is possible to predict aftercare and how soon in the process we can do this.

Within this paper we will first mention other similar studies in Section 2. After which in Section 3, the preliminaries will be explained as well as the importance of this study. Section 4 explains what the data looks like and how it was formatted accordingly. In Section 5 you can read about how we used the formatted data with the various models. Section 6 gives the results from the methods described in Section 5. Lastly Section 7 states the final conclusion.

2 Related Work

As can be read in the introduction, the main goal of this study is to determine if it is possible to detect early on which patients need aftercare and which do not. Not many other works can be found that deal with this specific topic of patient predictions. However, there are many that are related to developing a so called *early warning system* for hospital patients.

In [8], authors try to predict circulatory failure in the intensive care unit as early on as possible. Using three different machine learning techniques they tried to predict in a binary manner every 5 minutes after admission if a patient needed extra care or not. Similar to this, in [5] authors try to identify patients early on if they are at risk for Sepsis. Here they used gradient boosting at various timestamps within the first 24 hours after admission to identify possible at risk patients. In [13], an architecture combining process mining and deep learning was proposed to improve the severity score measure for diabetes patients.

As the aforementioned works, our aim is also to make a reliable prediction as soon as possible. Contrary to [8], our research cannot focus on predictions every 5 minutes, as patients should be analyzed and confirmed by a hospital employee. In this context, we need to decide for a certain moment in time where the prediction can be done. Besides, while [5] compares different early moments to find the best time for an early prediction, they do not consider the events that happen during such period. Moreover, [13] combines both event and patient data, they also consider data only from the first hours, but their focus is to provide a severity score rather than a prediction with a high imbalanced positive class.

3 Background

3.1 Predictive process mining

Within process mining, predictions are usually made on incomplete traces regarding future events and/or outcome and related attributes [6]. A *trace* is a timely-ordered sequence of events related to the same context (in this research, such context is a patient admission). Commonly, the prediction is done (the *prediction moment*) based on all previous events known (denoted as *prefix*) and the prediction target is some event or outcome in the future. So, the prefix trace is used as input for the prediction model. Making predictions within process mining might be valuable for many organizations, as having an idea of the future might lead to early actions that can improve the remaining of the process.

3.2 Preliminaries

As the problem statement can be seen as a binary one (do/do not), it allows us to use decision trees, random forest, and XGBoost solutions within this study. A decision tree represents a series of sequential steps that can be taken in order to answer a question and provide probabilities, costs, or other consequences with it [9]. There is no way of knowing what the best tree depth is for a decision tree, which means that tests should be performed in order to reach a conclusion. One option is to use cross validation [11],which is a procedure that resamples the data it is been given to evaluate a machine learning model in many ways on a limited data set [7].Random forest is a collection of decision trees producing a single aggregated result [9]. For random forests there is also the question of how many decision trees is best to use. Similarly to a single decision tree, this question

cannot be answered very easily and requires for example cross validation as well over various combinations to find the best option [9].Lastly, XGBoost, which is alike random forest but uses a different algorithm to build the needed trees. Random forest builds each tree independently whereas XGBoost builds them one at a time [9]. Also for XGBoost the problem of deciding on the amount of trees to use exists, and also here a possible solution is the usage of cross validation [9].

While a decision tree is a white box solution [3]and therefore preferred by the hospital due to it being explainable [10], random forest and XGBoost are also experimented with to allow for comparisons in the end. In order to be able to compare the results from the various models we keep track of the recall and precision scores [1]. With the hospital it was discussed that the recall scores weights more heavily than those of precision as it was deemed more important to be able to identify all of the aftercare patients (even though this might give many more false positives) than to miss them. Another way of comparing the models is by using a Receiver Operating Characteristic (ROC) curve or the Precision-Recall (PR) curve [4], which are also created. The preference in this paper goes to the usage of the PR curve, this due to the large class imbalance that we are dealing with. By computing the Area Under the Curve (AUC) for both ROC and PR allows for easy comparison between different models.

We will also make use of feature importance such that we can determine which datapoints we should keep and which can be removed. Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node in a decision tree [12].

4 Data

4.1 Data Introduction and Processing

As a data set we received the patient records from 2018. We filtered this set to traces that are at least 24 hours long but at most 2 months. Each patient used in this data set had given their permission to their data being used for analysis purposes. This resulted in a set containing 35380 unique hospital stays of which only 4627 required a type of aftercare, which is only 13% and could thus be classified as an infrequent behaviour [10].

For each hospital stay we collected the following patient information: aftercare required, aftercare type, age, gender, activities, timestamps, and additional information. Activities can be one of the following: hospital admission, hospital discharge, admit medication, poli appointment, start operation, end operation, start lab, and end lab. The additional information is related to the activity admit medication and specifies how it was admitted. Each stay was also assigned a random unique case id this to make sure patient information is anonymous [10].

We are currently not taking patient departments or any data regarding why they were admitted to the hospital into account. This is due to the fact that within this research we only considered the data of 2018. However, keeping in mind that a lot newer data exists and that this follow up data covers the Covid-19

years we had to create a dataset using features that stayed consistent throughout these years. After discussing with the hospital about what changed the most for patients during these years, and what would thus be an unreliable feature, we excluded the patient departments. The admission reason was excluded as this field is filled in manually within their systems, this creates a lot of possible different descriptions for the same issue.

In order to guarantee a certain quality event log, traces in which events happened before admission or after discharge were removed. Traces where end operation was before or at the same time as start operation or if either one of the two was missing (similar for start and end lab activities) were also removed.

4.2 Extending the Feature Set

We supplemented each trace with some manually created features of which the possible importance was questioned by the hospital. For each trace it was calculated how many times the same patient had been admitted previously, how many of those stays required aftercare, the average hospital duration of previous stays, the standard deviation of previous stays, the average duration in between admissions, and the standard deviation between admissions.

4.3 Formatting and Predicted Values

As decision trees do not take event logs as input data, we had to encode the datapoints in such a manner that all relevant points can be inputted at once. An initial dataset consisting of the manually created features combined with the patients age and gender was created. For a second dataset we had to make a distinction between *amount* and *occurrence*. With *amount* we count how often a certain activity took place (*how often did a patient receive medication? etc.*), whereas with *occurrence* we take note in a yes/no (1/0) manner if a certain activity took place at least once (*did the patient receive any medication? etc.*). This created two similar looking datasets, both containing the data from the first dataset one augmented with the *amount* and the other with the *occurrence* encoding. Later on, as will be described in 6, we also perform a count on specific medication and operation groups.

As prediction value, the dataset also contains a column indicating if that patient does need aftercare (denoted by 1) or not (denoted by 0).

5 Methods

The hospital records many different data points. It is of course not possible to just use everything and hope for the best. Therefore, we will approach the problem in the following manner. We start by using only data available at admission. This would give first insights for each patient at arrival moment if aftercare might be needed. Earlier than arrival time we cannot predict. This also provides us with a benchmark to which we can compare the models. Secondly, we will use a

selection of features, after discussing with the hospital on possible relevancy, if based on the full trace data the same or an increase in prediction accuracy can be seen compared to the benchmark. Using feature importance we can eliminate features that are not as beneficial as expected and rerun the models.

While in this paper we do combine prediction making and process mining on incomplete traces, we do so in a different manner. We chose two different prediction moments to be evaluated: 24 and 48 hours. As part of the research question is to determine how early we can predict for each patient, we will evaluate whether postponing the prediction moment contributes to an increase in prediction accuracy. For that, we will create two different models: one that considers only events happened within 24 hours as a prefix trace, and 48 hours for the other. They will be compared to the two benchmark models described.

The data used is split in two different ways. First in a 5-fold manner to perform cross validation for all three of the methods allowing us to find the best possible tree depth or number of trees used. Secondly, there is a 8:2 split for the final train/test set based on the results from the cross validation.

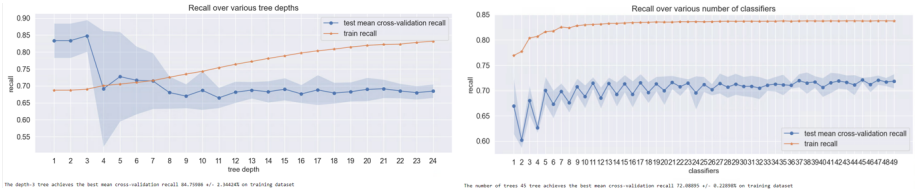
For the decision tree we will take cross validation over various tree depths (1 to 25), and for random forest and XGBoost we compare different amount of tree usages (1 to 50). For each step we calculate the accuracy and recall score, as can be seen in Fig. 2. The best possible tree depth/amount of trees is derived from where the recall score is highest along the plot. The recall score is taken here as we mentioned that this is the score the hospital is the most interested in.

6 Results

Admission data only. Starting with just the datapoints known upon arrival we get the recall plots over various depths/amount of trees using cross validation as can be seen in Fig. 2. From each plot we note the highest possible value with parameter and use those to create a final model. The final model results can be viewed in Table 1. Based on these results we can draw an intermediate conclusion that it is indeed possible to predict if a patient needs aftercare to a certain degree the moment they enter the hospital without knowing all too much about them. All three models appear very similar in overall results (F1).

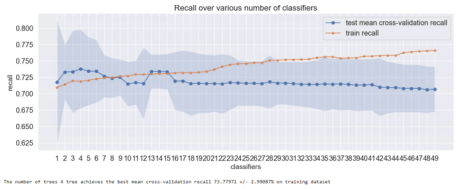
Full dataset. Next up was using the full dataset to determine feature importance. The full dataset was constructed in two manners (amount and occurrence). Similar to the previous section, here we also first tested using cross validation what the best depth/amount of trees is (Fig. 4). With this, we get the results in Table 1. Here we can clearly see an increase in scores for both the random forest and XGBoost compared to just the admission data. The decision tree results are similar with regards to just the recall score, however, the F1 scores did improve.

We did not find much difference in scoring between using either the amount or the occurrence datasets (hence we only show one of them). Given that the difference is minimal between the two means we can decide on one of them to use from this point forward. We decided on using the amount encoding.



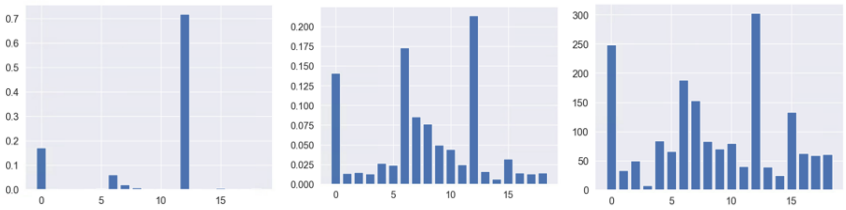
(a) Recall scores over various tree depths for a single decision tree

(b) Recall scores over various amount of trees for random forest



(c) Recall scores over various amount of trees for XGBoost

Fig. 2: Recall scores for various tree models using the admission data only dataset



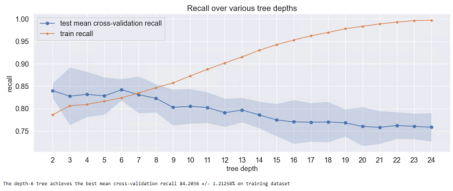
(a) Decision tree

(b) Random forest

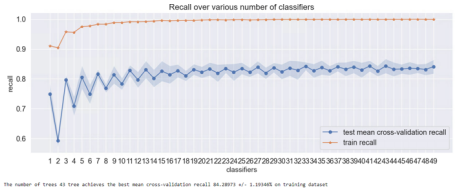
(c) XGBoost

Fig. 3: Feature importance. 0 - age, 6 - medication, 12 - total time stay

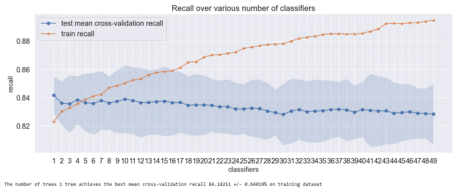
Before we can do additional testing based on a shorter timeframe it is important to derive which of the features actually contributed to the results. Calculating the feature importance for each model we obtain the plots in Fig. 3. The three highest bars correspond to: the admission data, *total time stay*; the *age*; and the *medications*. From the first two features we cannot create a dataset based on time as they are constant values. Medication however is a value that might change throughout a patient stay. Therefore, we created a new dataset based on medications a patient received within 24 hours. Within the hospital, more than 1000 different medications are used and creating a dataset that differentiates between them would result in very sparse dataset that takes a long time to train. Luckily, each medication comes with an ATC code [2]. An ATC code consists of 7 characters where the first four represent a medication group. There are only 268 medication groups. Grouping the medications quickly downsizes the dataset. Now, instead of counting each individual medication, we count how often a patient got admitted a medication from which medication group.



(a) Recall scores over various tree depths for a single decision tree



(b) Recall scores over various amount of trees for random forest



(c) Recall scores over various amount of trees for XGBoost

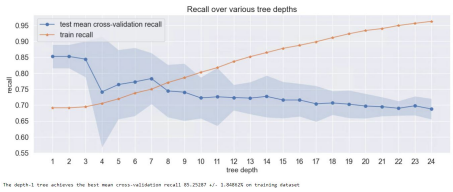
Fig. 4: Recall scores for various tree models using the full dataset with amount

Admission and first 24 hours medication data. The combination of admission data and the first 24 hours of medication group counts gives the third dataset. We again use cross validation to find the best parameters (Fig. 5) after which we obtain the final test scores (Table 1). Comparing these results to *admission data only* we can immediately tell that the recall score increased. Although the recall increased for all, the AUC PR for the decision tree was lower.

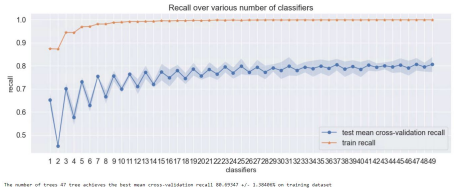
According to hospital domain experts, the way in which medication was admitted and the operation type might indicate a need for aftercare. Therefore a fourth and last test set was constructed based on the feature importance results and the domain knowledge of the hospital.

Admission, medication, admittance way, and operation data (24h & 48h). The last dataset also took all the same steps as the previous datasets (Fig. 6, 7). For this dataset we did create an extra test for the first 48 hours of events in order to see to what extent waiting for more information would provide better results, which can be found in Table 1. We can compare these results to the ones from the previous section. Here, one model had a slightly decrease in the recall score whereas others increased. Similarly for the AUC PR scores. Also, comparing the scores in Table 1, we do not see a major increase in using a dataset that considers 48 hours.

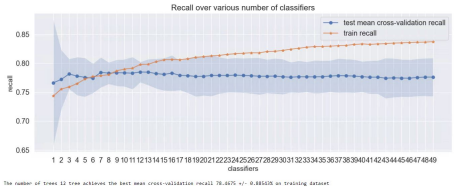
There is one major downside with how the results are now portrayed. Our testset consists out of 1/5 from the total dataset, which was the entire year of 2018. However, this is not the amount of patients that the hospital will work with on a day to day basis. Therefore it is important to look at the effect that the final and best model would have each day for a certain timeframe (Fig. 8). Based on this figure we can see that on average the hospital will have to



(a) Recall scores over various tree depths for a single decision tree

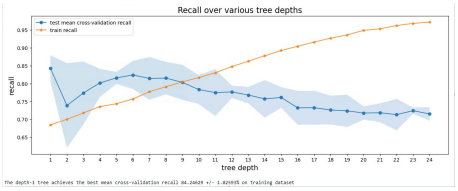


(b) Recall scores over various amount of trees for random forest

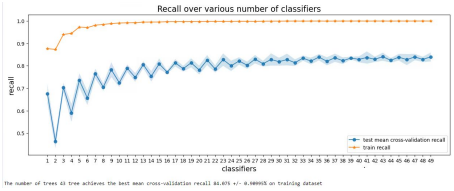


(c) Recall scores over various amount of trees for XGBoost

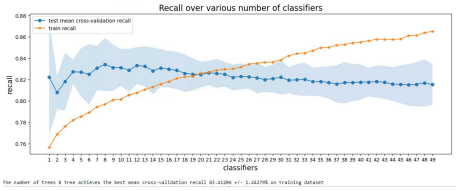
Fig. 5: Recall scores for various tree models using the admission with medication first 24 hours dataset



(a) Recall scores over various tree depths for a single decision tree



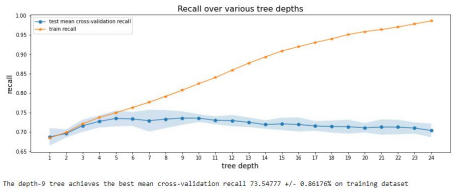
(b) Recall scores over various amount of trees for random forest



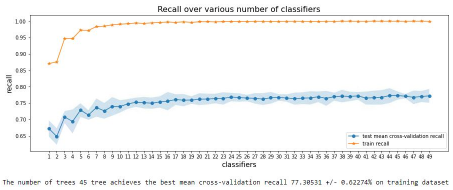
(c) Recall scores over various amount of trees for XGBoost

Fig. 6: Recall scores for various tree models using the admission with medication, admittance way and operation first 24 hours dataset

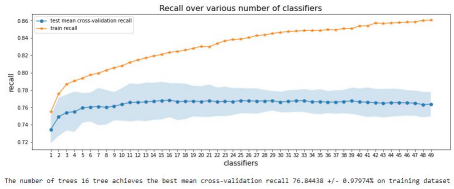
verify between 40 and 60 patients per day of which 10 to 20 will indeed require aftercare. Generating these results daily only takes a matter of seconds and is thus very doable. After discussion with the hospital it was concluded that these numbers were reasonable which means we can draw a final conclusion.



(a) Recall scores over various tree depths for a single decision tree

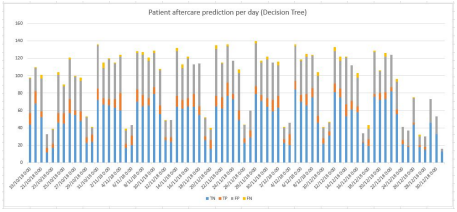


(b) Recall scores over various amount of trees for random forest



(c) Recall scores over various amount of trees for XGBoost

Fig. 7: Recall scores for various tree models using the admission with medication, admittance way and operation first 48 hours dataset



(a) Decision tree

Fig. 8: Model prediction per day based on the 24 hours model of admission, medication, medication admittance way, and operations dataset

7 Conclusions and Recommendations

Looking back at the research question stated in the introduction, we can definitely conclude that it is possible to predict which patients need aftercare and which do not to a certain degree.

We also wanted to analyze if patient could be identified earlier on in their trajectories compared to what is happening now. We had already derived that currently it takes about 74 hours before a patient is marked for aftercare. Within this paper we provided models at three different timestamps that are of relevancy (0, 24, 48 hours). How ever large the time benefit will be depends on what model the user chooses based on the first part of the research question.

Given that the hospital cares about the model being explainable only leaves one viable usable option for them, the decision tree. Comparing the third, fourth

Dataset	Model	Accurracy	Recall	Precision	F1	AUC ROC	AUC PR	FP	TP
Admission data only	Decision Tree	0.60	0.82	0.23	0.35	0.70	0.52	2635	767
	Random Forest	0.68	0.66	0.24	0.35	0.67	0.45	1941	619
	XGBoost	0.64	0.74	0.23	0.35	0.68	0.49	2289	692
Full	Decision Tree	0.78	0.80	0.36	0.49	0.79	0.58	1351	750
	Random Forest	0.79	0.81	0.37	0.51	0.80	0.59	1319	760
	XGBoost	0.78	0.81	0.35	0.49	0.79	0.58	1379	754
Admission with Medication (24 hours)	Decision Tree	0.57	0.83	0.21	0.34	0.68	0.52	2892	776
	Random Forest	0.71	0.80	0.29	0.42	0.75	0.54	1879	747
	XGBoost	0.71	0.78	0.28	0.42	0.74	0.53	1859	731
Admission with Medication, Admittance way, Operation (24 hours)	Decision Tree	0.66	0.80	0.24	0.37	0.72	0.52	2175	696
	Random Forest	0.70	0.81	0.27	0.40	0.75	0.54	1892	698
	XGBoost	0.72	0.75	0.27	0.40	0.73	0.51	1738	648
Admission with Medication, Admittance way, Operation (48 hours)	Decision Tree	0.70	0.78	0.24	0.37	0.72	0.51	2100	677
	Random Forest	0.71	0.84	0.28	0.43	0.77	0.56	1847	732
	XGBoost	0.73	0.79	0.29	0.42	0.75	0.54	1719	686

Table 1: All model scores for the various datasets

and fifth datasets from Table 1, we can say that there is no need to wait 48 hours before making a prediction. The decision on which model to then use is more up to them. Both the 24 hours results are very similar, one results in a slightly higher recall and the other in a slightly higher precision. Our recommendation would go to the model that uses admission, medication, medication admittance way, and operations given that the this one appears to be more of an increase compared to the first dataset.

If we were to give a final conclusion without being limited by the explainability rule, then we would recommend the random forest model at both the 24 hours and the 48 hours mark. This model has the highest AUC PR and overall higher F1 score.

In both conclusions we make use of the 24 hours model, which compared to the current time median from Fig. 1, is a major speed up. Using these models to aid the nurses currently working on this to help identify patients as early as 24 hours after admission, would give the employees who talk to the aftercare organization a lot more time to organise.

During the discissions with the hospital many more mentions were made about other datasets that they have in their possession. These can be used to possibly enhance the current models.

References

1. Chapter 5 - diagnosing of disease using machine learning. In: Singh, K.K., Elhoseny, M., Singh, A., Elngar, A.A. (eds.) Machine Learning and the Internet of Medical Things in Healthcare, pp. 89–111. Academic Press (2021)
2. Medicijntab: Geneesmiddelen op ATC-code (4) (2022)
3. Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine learning interpretability: A survey on methods and metrics. Electronics **8**(8) (2019)
4. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. Proceedings of the 23rd international conference on Machine learning - ICML '06 (2006)
5. Delahanty, R.J., Alvarez, J., Flynn, L.M., Sherwin, R.L., Jones, S.S.: Development and evaluation of a machine learning model for the early identification of patients at risk for sepsis. Annals of Emergency Medicine **73**(4), 334–344 (2019)
6. Di Francescomarino, C., Ghidini, C.: Predictive Process Monitoring, pp. 320–346. Springer International Publishing, Cham (2022)
7. Fushiki, T.: Estimation of prediction error by using k-fold cross-validation **21**(2) (2011)
8. Hyland, S.L., Faltys, M., Hüser, M., Lyu, X., Gumbsch, T., Esteban, C., Bock, C., Horn, M., Moor, M., Rieck, B., et al.: Early prediction of circulatory failure in the intensive care unit using machine learning. Nature Medicine **26**(3), 364–373 (2020)
9. Larose, C.D., Larose, D.T.: Data Science Using Python and R. Wiley (April 2019)
10. Munoz-Gama, J., Martin, N., Fernandez-Llatas, C., Johnson, O.A., Sepúlveda, M., Helm, E., Galvez-Yanjari, V., Rojas, E., Martinez-Millana, A., Aloini, D., et al.: Process mining for healthcare: Characteristics and challenges. Journal of Biomedical Informatics **127**, 103994 (2022)
11. Painsky, A., Rosset, S.: Cross-validated variable selection in tree-based methods improves predictive performance. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(11), 2142–2153 (2017)
12. Ronaghan, S.: The mathematics of decision trees, random forest and feature importance in scikit-learn and spark (Nov 2019), `https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3`
13. Theis, J., Galanter, W.L., Boyd, A.D., Darabi, H.: Improving the in-hospital mortality prediction of diabetes icu patients using a process mining/deep learning architecture. IEEE Journal of Biomedical and Health Informatics **26**(1), 388–399 (2022)