# Predicting duration of inpatient hospital visits with Machine Learning

Dennis Scheper - 373689

2020-11-16

# 1 Abbreviations

| Full Name | Abbreviation |
|---|---|
| Application Design | AD |
| Area Under the Curve | AUC |
| Exploratory Data Analysis | EDA |
| False Negative | FN |
| False Positive | FP |
| False Positive Rate | FPR |
| US Department of Health and Humans Services | HHS |
| High-Throughput High-Performance Biocomputing | HTHPB |
| NearestNeighbor | IBk |
| Receiver Operator Characteristic Curve | ROC Curve |
| True Negative | TN |
| True Positive | TP |
| True Positive Rate | TPR |

## Table of content

# 2 Introduction

Health care prices have been on a steady rise for decades on end. [1] This phenomenon has been an expensive problem for many Western countries that face an aging population, stirring prices up even more. The US Department of Health and Humans Services (HHS) determined that the average premium for family coverage has increased by 20 percent since 2013 and 55 percent since 2008. Spending on government health programs almost accounts for half of all U.S. health care expenditures, increasing the burden on taxpayers. It predicts that costs continue to increase by another 20 percent over the next five years. [2]

Some costs arise from misdiagnosing patients and adjusting care later on, not giving patients the care they need on their primary diagnosis, which leads to unnecessarily longer hospital stays as patients do not get the care they need. Interestingly, some of these cases have been linked to a second diabetes-related diagnosis. [3] Meaning that if a diabetes test was undergone, the duration of an inpatient visit could potentially be reduced and, ultimately, reducing health care costs.

Machine Learning could be a viable resource in confronting these issues, as recently, machine learning-related processes have been used to develop newer, more sustainable procedures by health institutions. [4] Unraveling patrons between different durations of a diabetes-related inpatient hospital visit is key, which can be done by exploring different algorithms/classifiers with data from patients who suffered from and were hospitalized based on a diabetes-related diagnosis.

## 2.1 Goal

Based on the the information presented above, we like to formulate a research question to work on in this project. Our research question is:

Is it possible, using machine learning techniques to predict the duration of an inpatient hospital visit?

## 3 Materials and Methods

All data used were obtained from a database consisting of 41 tables and 117 features, such as demographics (like gender, race, and age), inpatient or outpatient, and (in-hospital) mortality. [5] Data came from 130 hospitals in the USA for over ten years (1998-2008) and contained around 74 million unique visits by 18 million unique patients. This research used data that needed to accede to the following specifications:

1. The record is a hospital admission;
2. The encounter is diagnosed with "diabetes," any kind will satisfy;
3. The length of admission was at least one day up to eighteen days;
4. Laboratory test results are available; and
5. Medications were administered.

Of all encounters, 101.000 records fulfilled all specifications and were taking into account for further analysis. We took all 55 existing attributes, filtered and adjusted them until the dataset was better structured, had less missing values, and contained only the satisfying information. The final dataset is constructed by a preliminary analysis and preprocessing of the data described in detail in the Results section.

The retrieved data was analyzed in Rstudio (version 4.0.2). The focus was on cleaning the data from missing values, removing attributes that would not contribute to our research goals, adjusting the structure of valuable attributes, and getting a better picture of the data structure. We used many external packages to explore our data. For example, the libraries 'plyr' (version 1.8.6) and 'dplyr' (version 1.0.2) were used extensively to mutate data, remove redundancies, and explore the correlation between attributes. Visualizations were made with the package 'ggplot2' (version 3.3.2). Other external packages used had a smaller function; for example, 'caret' (version 6.0-86) was used to remove near-zero variance attributes. All transformations regarding the data structure are thoroughly explained and choices made are argued in the supplied Exploratory Data Analysis or log paper, which can be found here.[6]

After cleaning the data, our focus turned to explore selecting a machine learning algorithm that could predict the duration of an inpatient hospital visit with high accuracy. We used the open-source machine learning software Weka (version 3.8) to accomplish this. Weka is a work environment, i.e., performing (internal) steps in data mining, preprocessing data, and building a predictive model. It contains many classifiers that include highly adaptable parameter settings to obtain the most accurate, predictive model. Its graphical user interface makes it easy for nonprogrammers such as biologists to work with data mining processes. We explored many classifiers/algorithms and even let them work together to obtain the best predictive model. We ranked classifiers based on their performance by evaluating the metrics True Positive Rate (TPR), False Positive Rate (FPR), accuracy, precision, recall, an F1-measure, and the produced ROC Area. These metrics are a good indication of how well a classifier performs on predicting.

A user-friendly application was created based on the model created in Weka. The application is written in Java (version 11.0) and is accessible from the command-line. It predicts whether new instances are in for a brief visit or a longer visit, which translates to an inpatient visit less or equal to five days or longer than five days, respectively. Instances can be fed through an input file (ARFF or CSV) or from the command-line itself. The app makes use out of the Weka API package (version 3.8) for predicting new instances based on a predefined model, which was developed from our research using the Weka software. Command-line arguments were processed by the package Apache Commons CLI (version 1.4), which allows for a sleek and easy way to use the application. The appropriate link to the repository can be found here.[7]

# 4 Results

## 4.1 Tackling missing values

First, we take a look at how missing values are distributed in our dataset. This is depicted in figure 1, where the color black means a value is missing. The figure shows this per attribute, and we observe that some attributes have many missing values, and others have contained less. To get a better picture of these attributes and their missing values, we zoom in on these attributes in table 1.

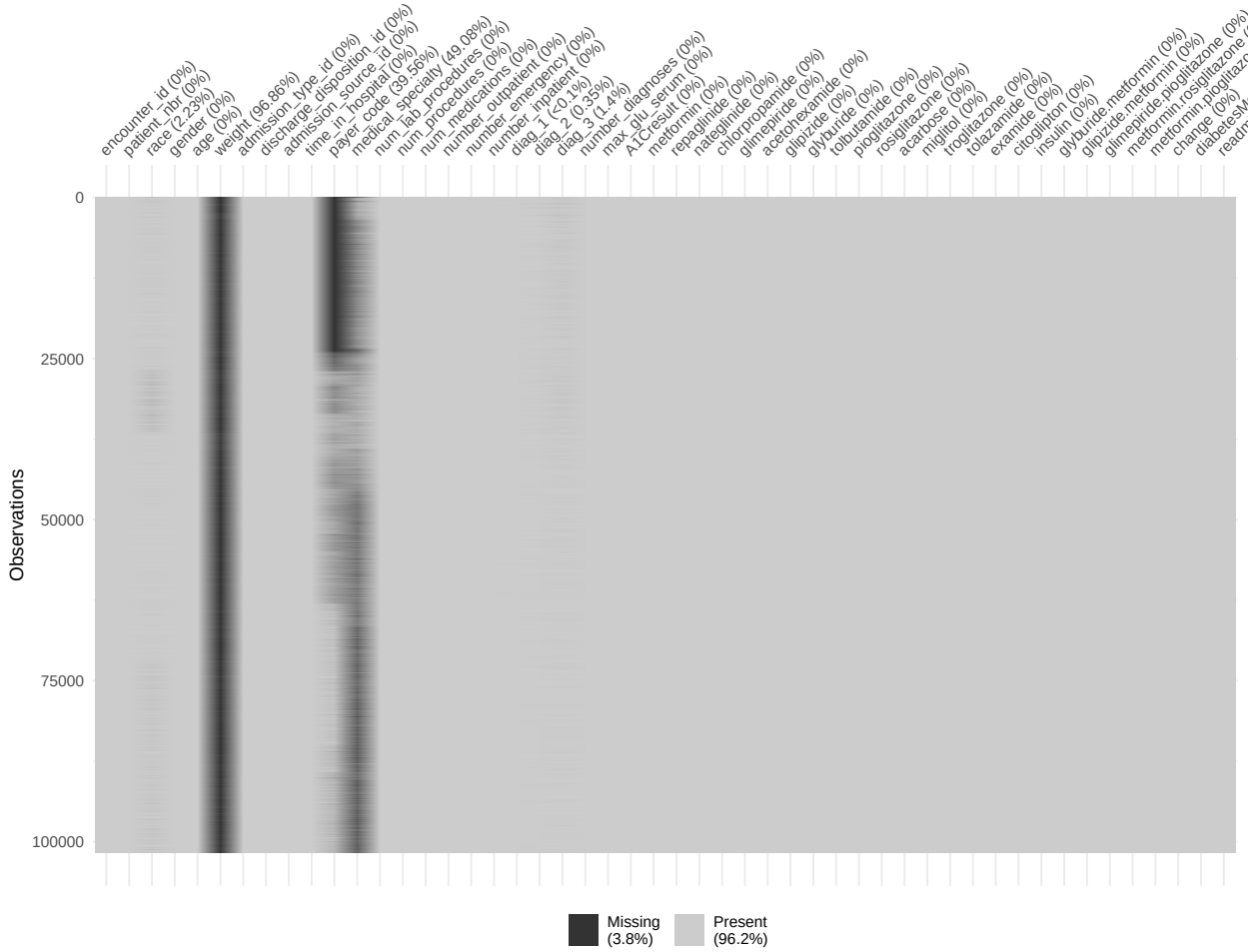


Figure 1: Missing values per attribute (depicted in black).

| Attribute name | Missing values | Percentage missing (%) |
|---|---|---|
| race | 2273 | 2.23% |
| weight | 98569 | 96.86% |
| payer_code | 40256 | 39.56% |
| medical_specialty | 49949 | 49.08% |
| gender | 3 | 0.00% |
| diag_3 | 1423 | 1.40% |

Table 2: All attributes containing missing values, presented in the amount of missing values and its presence expressed in percentage

We observe while looking at table 1 and figure 1 a multitude of different amounts of missing values. For example, the weight attribute is missing a staggering 97% of its valuation and gender, only missing three values. Having such a range of missing values, we decided to look at each attribute's contribution to our research goals and how much data is missing. If that contribution is minimal, we may decide to remove the attribute entirely. An example of such an attribute is payer_code, which is made up of payment; therefore, it is not really interesting to fill up the missing values here, and we removed it from the dataset. Besides removing payer_code, we proposed the same treatment for attribute weight; as mentioned earlier, it almost entirely contains missing values. The original authors stated that it is a result of government protocol. Knowing this, we can remove the attribute safely. We also remove medical_specialty for being too sparse.

For attributes with lesser amounts of missing values, we had the choice of doing two things: remove the instance with the missing values or revalue them on, for example, the most common valuation or label them as 'missing' instead of NA or as a question mark. For attribute race, we relabel the missing values to 'Missing/unknown' as removing has a potentially bigger impact on the outcome of the machine learning process. We handled diag_3 in the same manner. As for attribute gender, it holds three valuations: male, female, and missing or unknown, with only three instances falling into the latter category. For this reason, we decided to remove these records entirely.

### 4.2 Duplicates and removing redundant attributes

The dataset contains multiple observations of inpatients visits. This may cause long-term issues as encounters, which is the unit of our analysis, are not statistically independent. Table 2 presents how many duplicates we have in our dataset and the percentage of the total records.

| Total records | Total duplicates | Percentage of total |
|---|---|---|
| 101.766 | 16.773 | 16.48% |

Table 3: Number of duplicates and its percentage in the total records

The total number of duplicates is 16.773 or 16.48% of total records. To ensure a statistically independent dataset, we remove all of these instances. As a result, we are left with 84.993 unique records, each record corresponding to one patient.

The dataset consisted of a couple of near-zero variances attributes mostly regarding the type of diabetic medicine a patient took. In most cases, none of these medicines were used by encounters and did not contain any useful information. For that reason, we removed all these attributes, which account for an amount of eighteen. The secondary and final diagnosis attributes were also removed as these are not relevant to our research goals. As our research regards inpatient visits only, we took at the attributes of emergency and outpatient visits to harm the results.

### 4.3 Recoding and introducing new attributes

As our dataset contains many categorical attributes, it presents some potential for attributes to be incorporated into one another, thereby creating a more robust and more efficient data structure.

First, we need to introduce a variable that is essential to our research goals - and the authors also adapt that. The variable 'HbA1c test' represents what the original authors describe as a unique opportunity to assess hospital protocol's current efficacy surrounding diabetes testing. So it is vital for our research goals to implement this variable, and has the following four valuations: 1) no HbA1c test performed, 2) HbA1c performed and the result is in the normal range, 3) HbA1c performed and the result is higher than 8%, with no changes in diabetic medications, and 4) HbA1c test performed, with a result greater than 8%, and diabetic medication was changed.

Moving on to the three attributes that describe admission type (admission_type_id), disposition type (disposition_type_id), and admission source (admission_source_id), present us with the potential to move these attributes into a single one as we are interested in only the difference between ICU and non-ICU protocols. These attributes are made up of a digit reference to, for example, what the source of admission was. These are described in a separate ID codebook, shown in tables 3, 4 and 5. Concerning the number of references to an ICU or non-ICU related instance, the admission type has three values representing ICU related instances, the disposition type five, and the admission source also has five valuations regarding ICU; all of these are labeled as 'yes,' the rest of the values will be labeled as 'no.' All of this will be contained by a new attribute called 'icu_related' After this implementation, the three original attributes are up for removal. Additionally, we removed all instances of patients dying, which is depicted as yellow in table 4.

| ID | Description |
|---|---|
| 1 | Emergency |
| 2 | Urgent |
| 3 | Elective |
| 4 | Newborn |
| 5 | Not Available |
| 6 | NULL |
| 7 | Trauma Center |
| 8 | Not Mapped |

Table 4: Admission type ID and their description. Depicted in red are ICU related; no color means non-ICU related.

| ID | Description |
|---|---|
| 1 | Discharged to home |
| 2 | Discharged/transferred to another short term hospital |
| 3 | Discharged/transferred to SNF |
| 4 | Discharged/transferred to ICF |
| 5 | Discharged/transferred to another type of inpatient care institution |
| 6 | Discharged/transferred to home with home health service |
| 7 | Left AMA |
| 8 | Discharged/transferred to home under care of Home IV provider |
| 9 | Admitted as an inpatient to this hospital |
| 10 | Neonate discharged to another hospital for neonatal aftercare |
| 11 | Expired |
| 12 | Still patient or expected to return for outpatient services |
| 13 | Hospice / home |
| 14 | Hospice / medical facility |
| 15 | Discharged/transferred within this institution to Medicare approved swing bed |
| 16 | Discharged/transferred/referred another institution for outpatient services |
| 17 | Discharged/transferred/referred to this institution for outpatient services |
| 18 | NULL |
| 19 | "Expired at home. Medicaid only, hospice." |
| 20 | "Expired in a medical facility. Medicaid only, hospice." |
| 21 | "Expired, place unknown. Medicaid only, hospice." |
| 22 | Discharged/transferred to another rehab fac including rehab units of a hospital. |
| 23 | Discharged/transferred to a long term care hospital. |
| 24 | Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare. |
| 25 | Not Mapped |
| 26 | Unknown/Invalid |
| 27 | Discharged/transferred to a federal health care facility. |
| 28 | Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital |
| 29 | Discharged/transferred to a Critical Access Hospital (CAH). |
| 30 | Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere |

Table 5: Disposition type ID and their description. Depicted in red are ICU related; no color means non-ICU related.

Attribute depicting the primary (diag_1) consists of a three-digit referring to an ICD code. All the different codes can be categorized into a smaller range of valuations. Table 6 shows what categories are used. For example, ICD codes 390 to 459 are circulatory diseases, according to [8]. This attribute now contains eight valuations instead of a much wider range. This allows analyzing more correlations between, for example, time spent in hospital and disease categories.

| ID | Description |
|---|---|
| 1 | Physician Referral |
| 2 | Clinic Referral |
| 3 | HMO Referral |
| 4 | Transfer from a hospital |
| 5 | Transfer from a Skilled Nursing Facility (SNF) |
| 6 | Transfer from another health care facility |
| 7 | Emergency Room |
| 8 | Court/Law Enforcement |
| 9 | Not Available |
| 10 | Transfer from critical access hospital |
| 11 | Normal Delivery |
| 12 | Premature Delivery |
| 13 | Sick Baby |
| 14 | Extramural Birth |
| 15 | Not Available |
| 17 | NULL |
| 18 | Transfer From Another Home Health Agency |
| 19 | Readmission to Same Home Health Agency |
| 20 | Not Mapped |
| 21 | Unknown/Invalid |
| 22 | Transfer from hospital inpt/same fac reslt in a sep claim |
| 23 | Born inside this hospital |
| 24 | Born outside this hospital |
| 25 | Transfer from Ambulatory Surgery Center |
| 26 | Transfer from Hospice |

Table 6: Admission source ID and their description. Depicted in red are ICU related; no color means non-ICU related.

| ICD-9 codes | Categorical valuation |
|---|---|
| Code 240-279: endocrine, nutritional and metabolic diseases | Diabetes |
| Code 390-459: disease of the circulatory system | Circulatory |
| Code 460-519: disease of the respiratory system | Respiratory |
| Code 520-579: disease of the digestive system | Digestive |
| Code 580-629: disease of the genitourinary system | Genitourinary |
| Code 710-739: disease of the musculoskeletal system and connective tissue | Musculoskeletal |
| Code 800-999: Injuries | Injury |
| Other codes | Others |

Table 7: ICD-9 codes and their new corresponding categorical valuation.

The variance of the class attribute caused some difficulty predicting the exact duration of an inpatient hospital visit. The attribute consisted of fourteen different valuations (1-14 days). This wide range caused every classifier to have poor results regarding predicting; the top-performer was the classifier Simple Logistics with a mere 23.64% correctly predicted instances. Even after the optimization of parameter settings on multiple classifiers, we concluded that the class variable's existing structure was not going to bring any progression. For that reason, we decided to relabel it into two classes: a 'brief visit' and a 'long visit', which categorizes for an inpatient visit less or equal to five days and more than five days, respectively. This resulted in far better classification and makes it still viable to determine differences between both new valuations to, ultimately, reduce the duration of inpatient hospital visits by responding to those differences. The new and old distribution is depicted in figure 2.

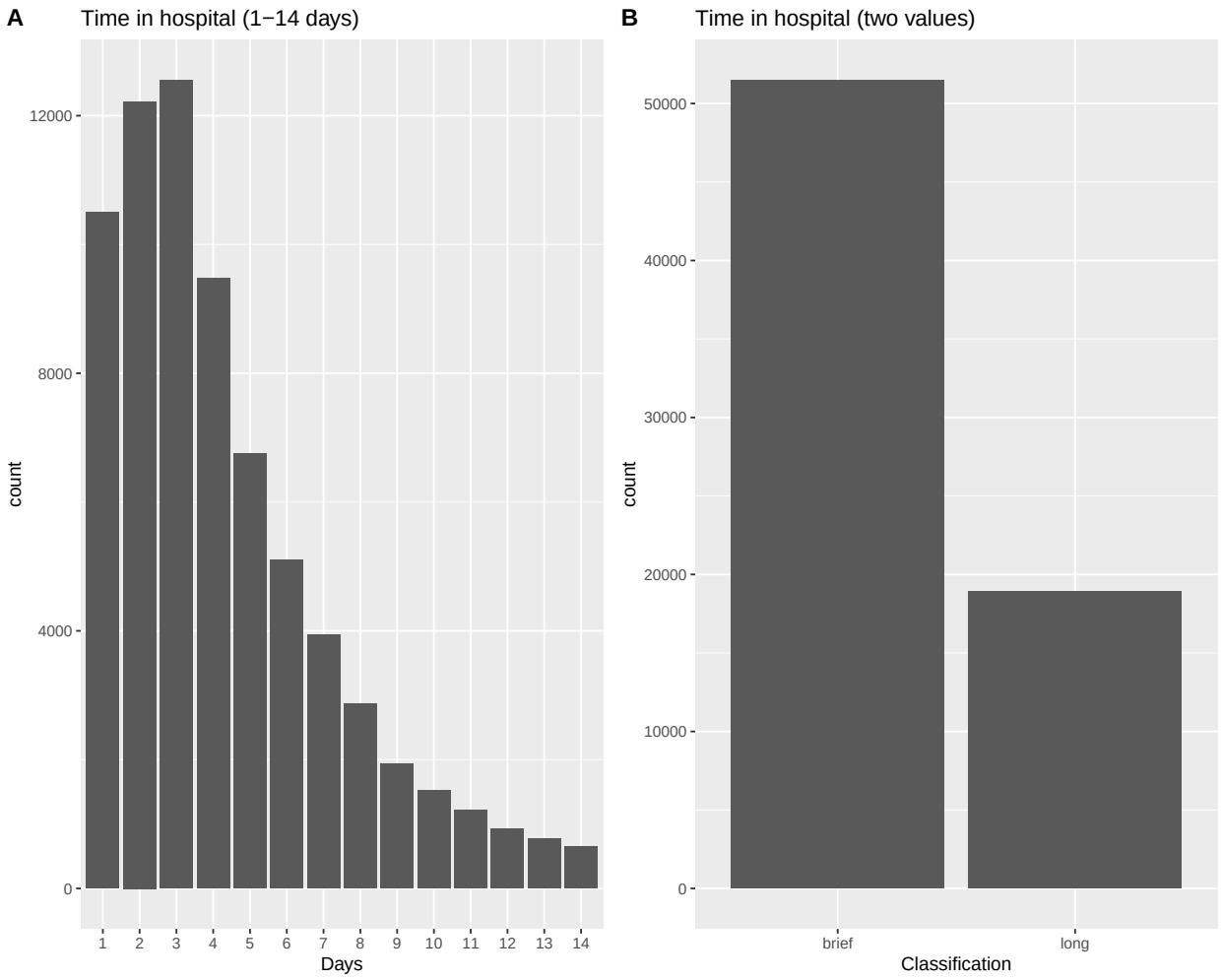


Figure 2: Changes in distribution from relabeling the class variable, reducing variance from fourteen to two valuations

### 4.4 Normalization

Since our dataset does not contain many numeric attributes, it makes normalization up for debate. In figure 3, we observe all numeric data without normalization represented in histograms. The distribution is very diverse across all plots; for example, plot A ranges between 0 and 15, whereas B's distribution is much wider (between 0 and >100 for some outliers). We notice across almost all histograms that counts of a valuation increases in the beginning and decrease slowly over higher valuations. This creates, in some plots, many outliers. Examples of this are plots E, F, and G, where there seem to be many instances regarding lower values and almost no higher valuations (in all of the examples, this is after the value of ten). For the reasons mentioned, we introduce a log-2 scale to reduce the range of valuation between numeric attributes; the results are shown in figure 3.

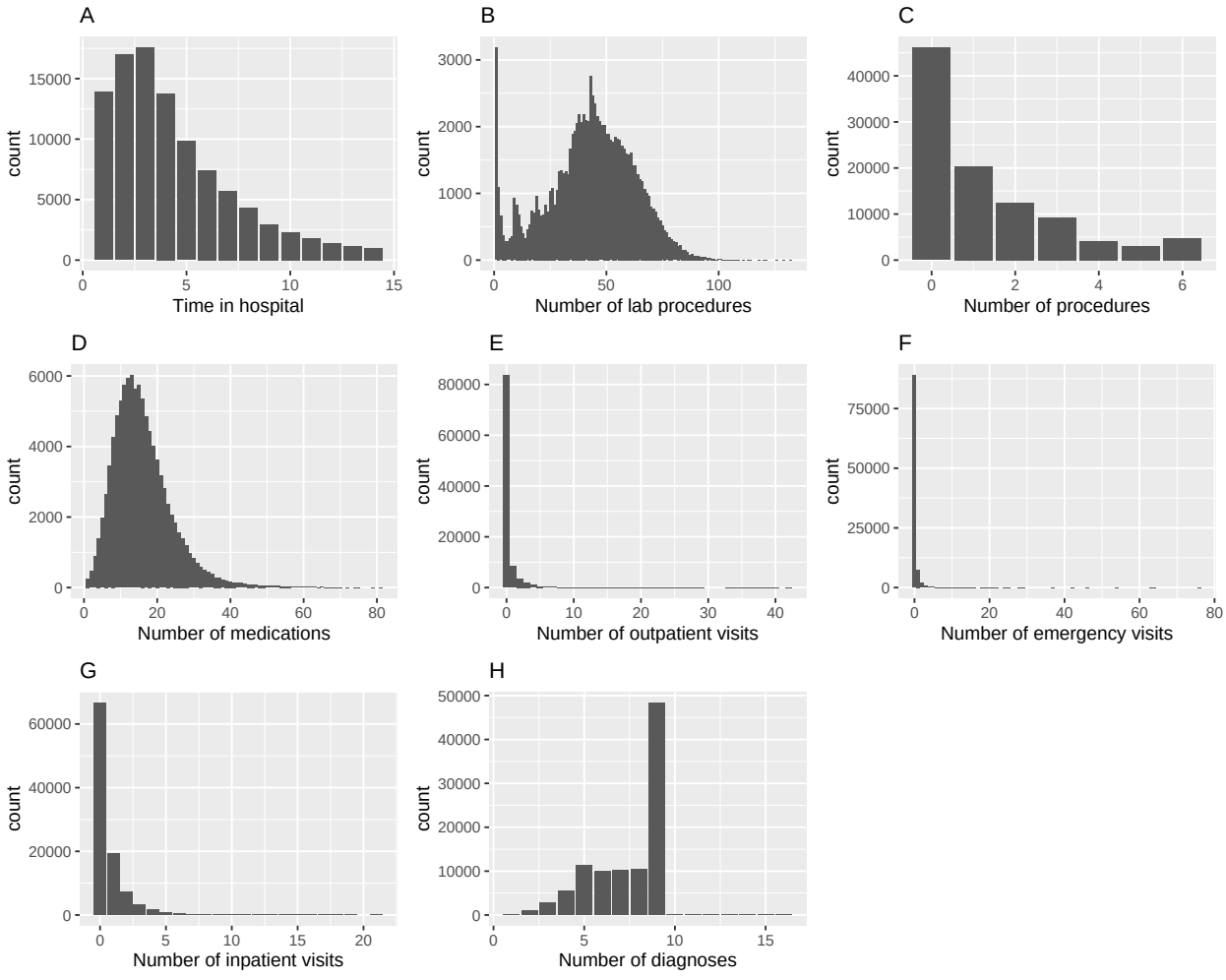


Figure 3: Numeric data presented in histograms without any normalization.

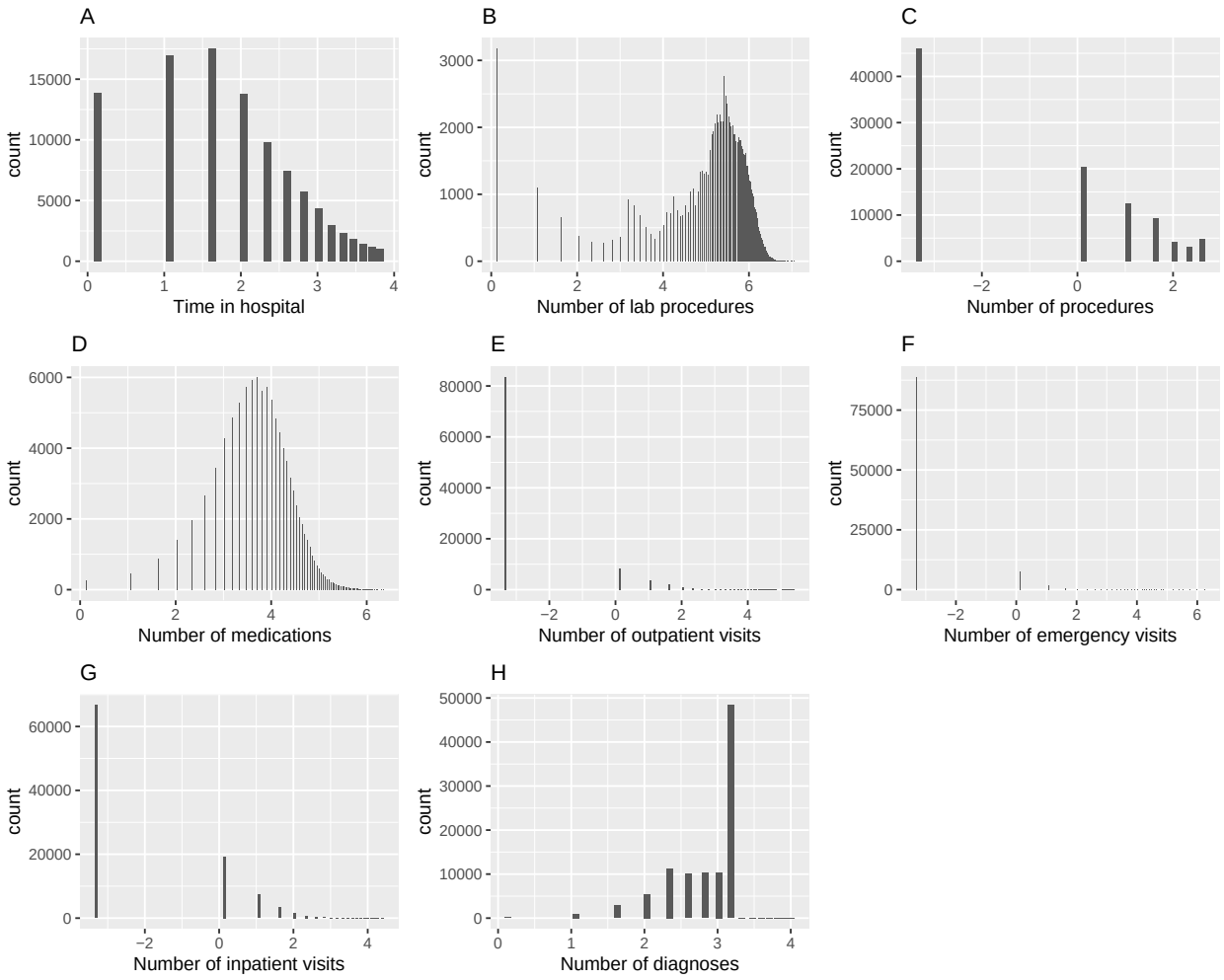


Figure 4: Numeric data presented in histograms on a log-2 scale.

Many instances have a valuation of zero, and with the introduction of a log-2 scale, we are about to transform these values into '-Inf' if nothing is done. To combat this, we added 0.1 to every value so that the valuation is still representable to its original valuation, and we do not have to remove all of these values. Looking at figure 4, we observe that the range of valuation between attributes is much smaller.

### 4.5 Correlation

Correlation between attributes is important to determine. Highly correlated attributes may cause problems with machine learning speed, and it can reduce the precision of estimated coefficients. It is, therefore, necessary to look at the correlation between attributes. We do this for every numeric attribute.

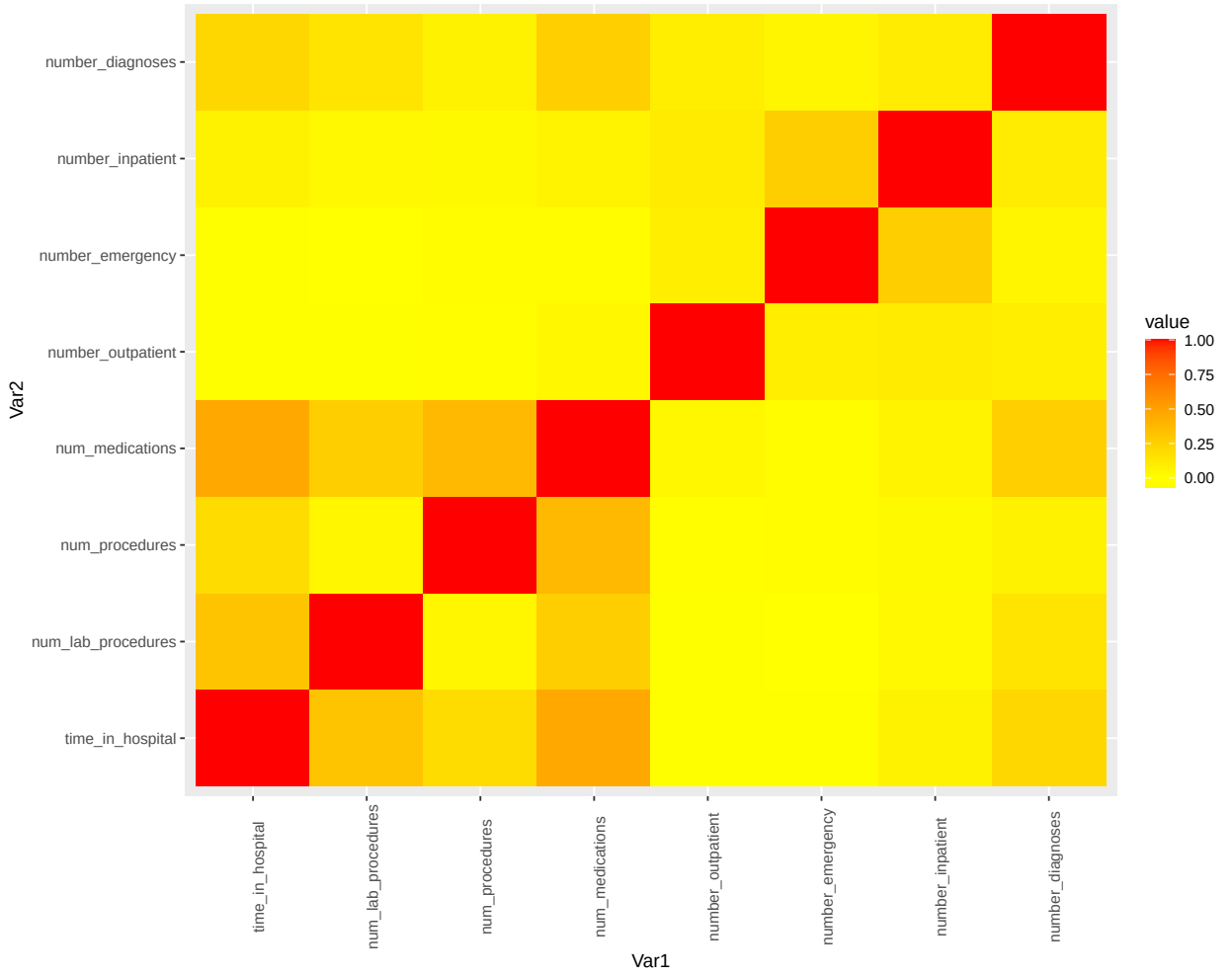


Figure 5: Heatmap of numeric attributes, with red depicted as a strong correlation, orange as a (small) correlation, and yellow as no correlation.

Our heatmap (shown in figure 5) shows a strong correlation as red, a small correlation as orange, and no correlation as yellow. The combination of attributes that stand out is, for example, num_medications-time_in_hospital and num_procedures-num_medications. As we can see, it is mostly the num_medication attribute that has some correlation. Others do not stand out that much. We see that most attributes do not correlate much with each other, at least for the numeric attributes.

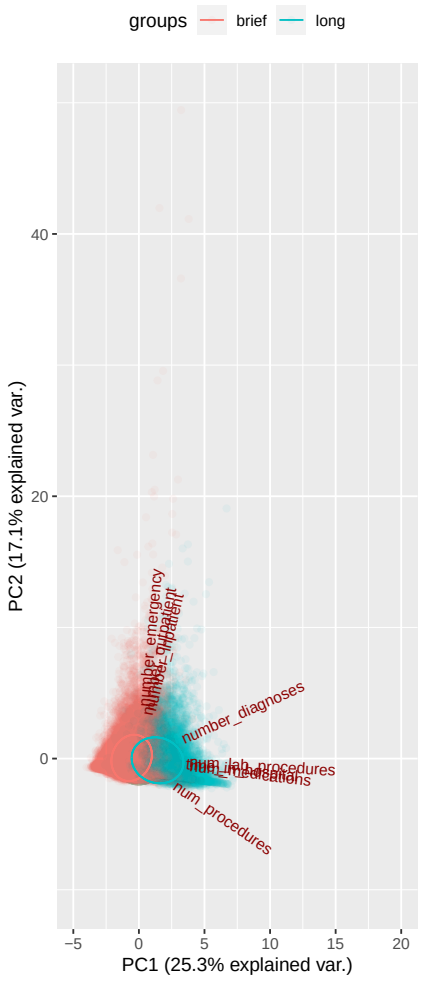


Figure 6: PCA plot of every numeric attribute, grouped around the relabeled class attribute time_in_hospital

While observing figure 6, which is a PCA plot with every numeric attribute and grouped around class attribute time_in_hospital with its two valuations (not fourteen), there is not much we can conclude about the correlation between attributes. The data seems all over the place, and clustering does not give a clear picture of correlation. We can see that attributes are mostly independent, which is essential information for classifying new instances.

### 4.6 Final dataset

After removing whole attributes and some instances and recoding and revaluing others, our final dataset contains 70.438 records and 22 attributes, including the class variable. Which is the result of parting with a total of 23 attributes and 31.328, which is a net loss of 30.78%.

### 4.7 Machine Learning

After cleaning the data thoroughly, testing to find the best-predicting classifier could begin. As already stated, this was achieved by using the open-source data mining platform Weka. Weka offers a large set of classifiers to choose from. Not fully understanding which single or group of classifiers would work the best, testing with at least one classifier per group of similar function was decided upon. Based on this reasoning, the classifiers NearestNeighbor (IBk), Random Forest, Naïve Bayes, Simple Logistics, and J48/C4.5 were used. ZeroR and OneR are good classifiers to measure baseline performance and were included as well. We measure performance in the already discussed metrics. The results of all involved classifiers are depicted in table 7.

Table 8: Results regarding each chosen classifiers with base settings

| Algorithm | Accuracy | TP.Rate | FP.Rate | Precision | Recall | F.Measure | ROC.Area |
|---|---|---|---|---|---|---|---|
| **ZeroR** | 73.12% | 0.731 | 0.731 | 0.000 | 0.731 | 0.000 | 0.500 |
| **OneR** | 76.31% v | 0.762 | 0.540 | 0.740 | 0.762 | 0.792 | 0.611 |
| **J48/C4.5** | 77.79% v | 0.778 | 0.454 | 0.761 | 0.778 | 0.761 | 0.718 |
| **Naïve Bayes** | 72.80% | 0.731 | 0.365 | 0.745 | 0.731 | 0.373 | 0.753 |
| **IBk** | 66.71% * | 0.667 | 0.522 | 0.664 | 0.667 | 0.666 | 0.573 |
| **Simple Logistics** | 78.55% v | 0.786 | 0.467 | 0.771 | 0.786 | 0.764 | 0.790 |
| **Random Forest** | 78.72% v | 0.789 | 0.454 | 0.775 | 0.789 | 0.770 | 0.802 |

We observe table 7, where we can see that the baseline-performance regarding our data is decent and sits at 73.12% and 76.31% accuracy for ZeroR and OneR, respectively. The front-runners are Simple Logistics and Random Forest (78.55% and 78.72% accuracy), with around even-scoring numbers in each metric. Based on these results, both classifiers had their parameters optimized for even better classification.

Only continuing with the two best-performing classifiers gives the change to explore so-called meta-learners. Meta-learners learn upon the performances of baseline classifiers or better the performance of a classifier and, if goes right, do a better job at predicting in general. The meta-learners of AdaBoost (Boosting), Bagging, Vote, and Stacking were utilized. AdaBoost and Bagging were employed to better the performance of Simple Logistics only, as Random Forest is a meta-learner itself and uses a form of Bagging already. AdaBoost does not seem to better performance of Random Forest from a general perspective, so it was excluded. Vote was performed on every classifier no matter their previous results; Simple Logistics and Random Forest were utilized with optimal parameter settings. Stacking included the two best-performing classifiers to build even a better model potentially. The results of all meta-learners' performances are depicted in table 8.

We now observe the table 8, visualizing the optimal parameter settings of Simple Logistics (78.62%) and Random Forest (78.77%). Together with the performance of every meta-learner, which shows a clear best-performing classifier. That classifier is Stacking with an accuracy of 79.04% and scores best in every other metric than any other. Stacking utilizes the combining of Random Forest and Simple Logistics to develop an even better model.

Table 9: Results regarding meta-learners on the two best-performing classifiers with optimal parameter settings

| Classifier | Accuracy | TP.Rate | FP.Rate | Precision | Recall | F.measure | ROC.Area |
|---|---|---|---|---|---|---|---|
| **Simple Logistics** | 78.62% | 0.786 | 0.467 | 0.772 | 0.786 | 0.765 | 0.789 |
| **Random Forest** | 78.77% | 0.788 | 0.455 | 0.773 | 0.788 | 0.768 | 0.796 |
| **Bagging** | 78.62% | 0.786 | 0.467 | 0.772 | 0.786 | 0.765 | 0.789 |
| **Boosting** | 78.60% | 0.786 | 0.469 | 0.771 | 0.786 | 0.764 | 0.772 |
| **Vote** | 78.66% | 0.787 | 0.498 | 0.776 | 0.787 | 0.757 | 0.644 |
| **Stacking** | 79.04% | 0.790 | 0.442 | 0.777 | 0.790 | 0.773 | 0.803 |

Bagging and Boosting did not result in better performance for Simple Logistics than the optimal settings did. For Bagging, it stayed the same at 78.62% and worsened slightly regarding Boosting, set at 78.60% accuracy. To get a better overview of the best-performing meta-learner Stacking, the Receiver Operator Characteristic (ROC) curve can be seen in figure 6.

Looking at figure 6, we see the ROC curve of stacking depicted. The red line is the classifier's performance, and the blue line divides the ROC space. We notice that our classifier performs well above the diagonal line, which means the classification is good. The optimum point seems to be around a TPR of 0.65 and an FPR of 0.25. This means that the positive cases that are predicted are actually positive ones, indicating high recall. A high recall/sensitivity is something we strive for. The Area under the ROC, which also one of our chosen metrics, is set at 0.803 and can be used as an accurate indicator. A high ROC Area means good accuracy. Our final model scores descent enough values to be a good model.

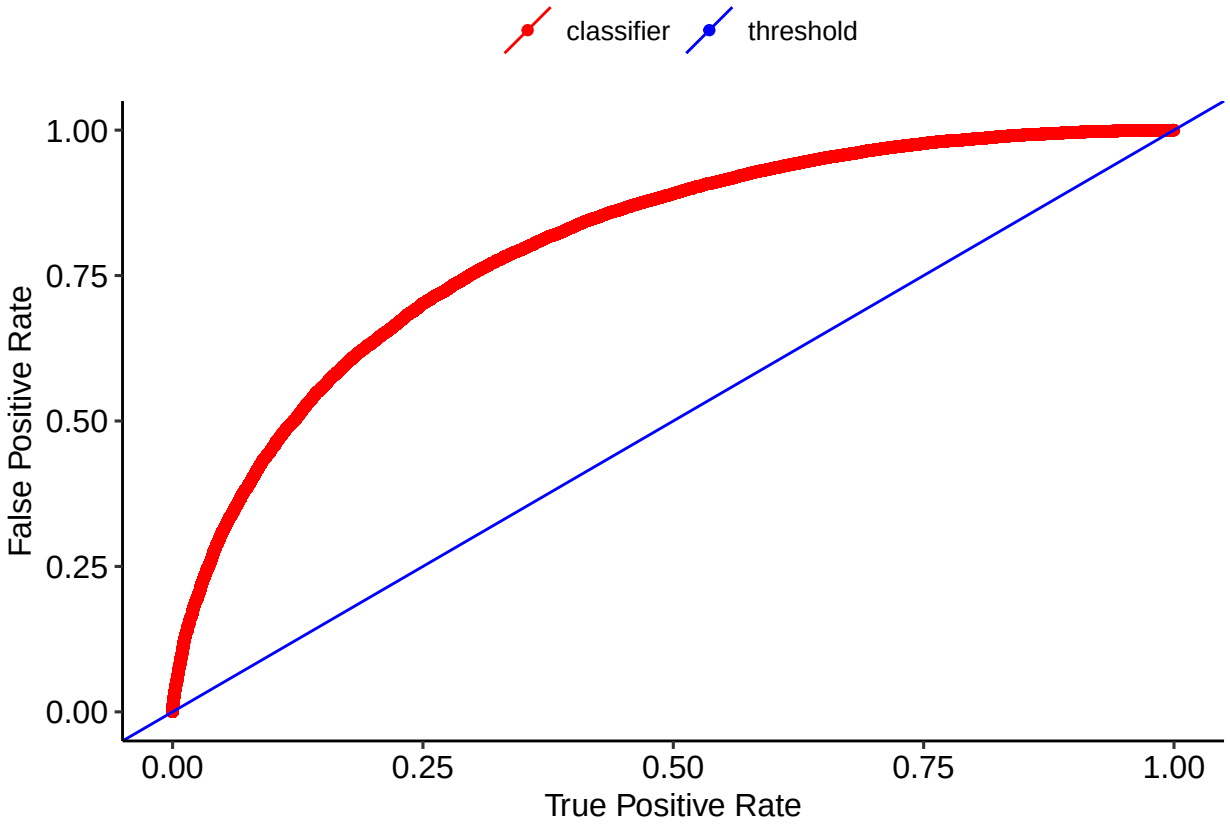


Figure 7: ROC-Curve for the Stacking meta-learner using both Simple Logistics and Random Forest, shown in red. Compared to a No-Skill line represented as the diagonal blue line

# 5 Discussion and conclusion

## 5.1 Discussion

There have been a couple of recoding of attributes, for example, the introduction of the icu_related attribute and, as a consequence, the removal of three redundant attributes. Some other attributes still can be recoded, thereby creating a more efficient data structure. An example of such an occurrence is the attributes for primary diagnosis; this attribute is now categorized according to their ICD code. As we are mostly interested in the diabetes category, constructing a new structure with the categories 'Diabetes-related' and 'Others' would create a binary valuation, which is very efficient for machine learning algorithms. The danger of removing so many instances is a great reduction in available information, which could be useful in further analysis down the road.

The attribute medical_specialty is one that was retired due to its large percentage of missing values. However, it has the potential to give more insight into where exactly diabetes protocols have been lackluster. Some may state that the attribute for admission source could be used for the same intentions. However, medical_specialty gave a lot more valuations and was a great candidate for a more in-depth analysis to improve hospital protocol regarding diabetes testing. Its large part of missing values could have been relabeled to 'Missing' and its information been used. Since the original authors did not continue with the attribute, it was a strong belief to part ways with it.

The normalization process conducted had some minor implications; we used a log-2 scaling normalization technique where many '-Inf' values were introduced due to zeroes' presence. A temporary solution was to add 0.1 to every instance in numeric attributes. This choice may cause issues with that; not every log-transformed value is representable to its original valuation. Although the added value is relatively low compared to some instance values, it can cause a difference in machine learning outcomes.

After normalization, we still notice some outliers in numeric attributes. After introducing the log-2 scale, all valuations have an improved range compared to one another, but the amount outliers could still be harmful when coming to the machine learning process. Outliers can give poorer outcome results in a worst-case scenario. In contrast, having some outliers can have no serious implications. The question is if the normalization process we defined was effective enough to get rid of most outliers.

Our biggest problem may be that the class variable is positively skewed, resulting in a bias against lower durations of inpatient hospital visits. We combatted this by relabeling the class variable from fourteen to just two valuations (brief and longer visits), but the skewness is still there; $>$50.000 (brief) and $<$20.000 (long) instances, respectively. This skewness can still affect our model's bias to brief visits, resulting in overfitting, having a model that is not comparable to real hospitalizations. More instances regarding longer visits can improve the model to predict a more real duration of a visit.

Another issue is the cut-off regarding the relabeling; when we decided to restructure the class variable, it was challenging to develop a new distribution. What is a brief and a long visit? It

was a very arbitrary choice, as no universal definition of a brief and long visit exists. Sources give other definitions. For example, the NHS defines a long stay as over ten days [9], another study as seven days [10] for the USA, and according to the European Union, a definition of a 'long' visit is different per country, ranging from a low 4 days in The Netherlands to a high of ten in Russia. [11] All-in-all, it is complicated to precisely determine a measurement for the number of days considered a 'long' and a 'brief' visit. We might want to increase the number of days seen as brief, as most sources give a higher definition of a longer visit. If we decided to do that, we would increase the effects of the skewness of our dataset, making the potential over-fitting even more apparent. What could be done is relabeling to include more classifications like a 'short' and an 'average' visit. Determining the exact days for each class is difficult to determine and might require expertise.

## 5.2 Conclusion and futher work

The goal of this project was to create an algorithm that can predict whether a new patient is in for a brief (<= 5 days) or long (> 5 days) visit. To achieve this, retrieved data has been cleaned, normalized, and potential relationships visualized. A final model with an accuracy of 79.04% based on meta-learner Stacking that combines performances from both Simple Logistics and Random Forest—both with optimal parameter settings—was achieved. However, whether the model can be a framework for future procedures regarding reducing diabetes-related inpatient hospital visits is debatable. A brief and long visit's classification varies among health institutions and countries, making our model less believable and impactful. Due to the used classification, the model's performance can be a result of over-fitting. In this state, we would not recommend adjusting existing healthcare procedures based on the model's outcomes.

Future work can focus on spelling out a (better) definition of what a 'brief' and a 'long' visit are or exploring more ways to classify new labels such as an 'average' visit so that the model is closer to predicting a real hospitalization length. Another aspect to focus on could be obtaining more instances that endured a long visit, according to the definition used in this research. This would result in a less potential situation where the model is over-fitted.

# 6 Project proposal for minor

The project proposal presented here would be a better match for High-Throughput High-Performance Biocomputing (HTHPB) than for Application Design (AD), as we talk about processing data from potentially millions of patients.

The goal is to create an application using a suitable classifier/algorithm that can determine or predict whether a recently admitted patient is potentially in for a "brief" ($<= 5$ days) or a "long" ($> 5$ days) stay. Based on this assessment, the application could indicate why such a classification is given to that particular patient. The indication gives an estimation of how likely a patient is for a particular duration. A patient's care could be based or reassessed on the application's findings, as the application could give factors that contributed the most to the prediction. Those factors could then be assessed and tackled. An example could be a change in medicine. The number of days and the number of labels can be adjusted to someone's liking or expertise.

The implementation does not have to be universal as only hospitals, healthcare institutions, and other research initiatives would be interested in a tool like this. Implementation could be a simple but easy-to-use Desktop app.

The application should be developed for health professionals to determine what care is suitable for diabetic patients. This would not be necessary to have for your everyday Joe. Outcomes need to be analyzed carefully and by someone knowledgeable.

In this case, input would be patient data that is diabetic-related and output estimations of the duration of stay and what factors contributed to that particular assessment.

# 7 References

## References


[1] The Organisation for Economic Co-operation and Development: Health expenditure and financing, https://stats.oecd.org/Index.aspx?DataSetCode=SHA, 15 November, 2020.

[2] U.S. Department of Health and Human Services, U.S. Department of the Treasury, U.S. Department of Labor: Reforming America's Healthcare System Through Choice and Competition, https://www.hhs.gov/sites/default/files/Reforming-Americas-Healthcare-System-Through-Choice-and-Competition.pdf, October 12, 2017.

[3] Comino, E.J.; Harris, M.F.; Islam M.D.F.; et all.: Impact of diabetes on hospital admission and length of stay among a general population aged 45 year or more: a record linkage study, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4310177/#:~:text=A%20number%20of%20studies%20hhttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4310177/#:~:text=A%20number%20of%20studies%20have,5%2C9%2C11%5D.ave,5%2C9%2C11%5D., 22 January 2015.

[4] Ashrafian, H; Darzi, A: Transforming health policy through machine learning, https://www.researchgate.net/publication/328926097_Transforming_health_policy_through_machine_learning, November 2018.

[5] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014. DOI: https://doi.org/10.1155/2014/781670

[6] Dennis Scheper: Theme 9: Introduction to Data Mining, https://bitbucket.org/djscheper/thema09/src/master/, November 2020.

[7] Dennis Scheper: Theme 9: Javawrapper, https://bitbucket.org/djscheper/javawrapper/src/master/, November 2020.

[8] Centers for Disease Control and Prevention, National Center for Health Statistics, ICD-9, https://www.cdc.gov/nchs/icd/icd9.htm, November 6, 2015.

[9] National Health Service: Guide to reducing long hospital stays, https://improvement.nhs.uk/documents/2898/Guide_to_reducing_long_hospital_stays_FINAL_v2.pdf, June 2018.

[10] Silber, J.H; Rosenbaum, P.R; Rosen, A.K; et all.: Prolonged Hospital Stay and the Resident Duty Hour Rules of 2003, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3279179/, December 2009.

[11] EuroStat: Hospital discharges and length of stay statistics, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Hospital_discharges_and_length_of_

stay_statistics&stable=0&redirect=no#Average_length_of_hospital_stay_for_inpatients, August 2020.