Type 2 Diabetes readmissions - causes and predictors by ML approach

Yahel Cohen, Tal Weizman Shapira, Shiri Karagach

Weizmann Institute of Science, 12/05/2024

Section 1 - Introduction

Type 2 diabetes (T2D) is a common disease characterized by either insulin resistance or low insulin activity due to a progressive loss of beta cell insulin secretion in the pancreas[1]. This, in turn results in abnormal blood levels of glucose[2]. This resistance contributes to increased glucose production in the liver and decreased glucose uptake in muscle and adipose tissue[3]. T2D can be diagnosed by blood hemoglobin A1C (HbA1c) levels. The test determines whether the patient is healthy (<5.7%), prediabetic (> 5.7% & < 6.5%), or diabetic (> 6.5%), with high HbA1c levels linked to diabetes complications[4].

Currently, it is estimated that roughly 537M people are diagnosed with diabetes worldwide. This number is predicted to increase even more to 643M by 2030. Additionally, disease prevalence is continent-dependent ranging from one to six people in the Middle East to one in twenty two[5]. Worryingly, In 2021 an estimated 6.7M deaths were caused by diabetes with numbers expected to increase as disease prevalence increases[5].

Past studies have shown that T2D has many risk factors. Older patient's age was found to increase the chance of being diagnosed with T2D[6] however, this is country dependent and several researches reported an increasing number of diagnoses in children and adolescents[7,8]. Obesity was found to be the highest risk factor for T2D, increasing the risk by 90-fold[9] with risk correlating to BMI scores and increasing exponentially at BMI higher than 30[9,10]. Sex also affects the risk for T2D with men being at higher risk than women and tend to develop circulatory disorders (CVD) more than women[11]. Other illnesses also play a role in T2D risk as can be seen from the increased risk of patients with herpes simplex virus type 1 and hepatitis C virus to develop T2D[12,13], though this correlation is yet to be understood mechanistically. Lastly, both smoking and alcohol consumption were associated with the risk of developing T2D. While smoking was found to increase the risk[14], studies have found a U shape relationship between alcohol consumption and risk of having T2D where moderate intake of alcohol actually helps prevent the disease but higher intake adds to its risk[14,15].

Prevention and treatment of T2D can be both behavioral and pharmacological. Studies have shown that strict adherence to a healthy diet, such as mediterranean diet and diets low in refined grains, red or processed meat, and sugar-sweetened beverages, contribute to diabetes prevention[16–18]. Moreover, routine physical exercise, both aerobic exercise and resistance

training, are beneficial to T2DM prevention[19,20]. Though lifestyle modification is preferred, numerous glucose-lowering therapies are available and used when the former treatment is not feasible. Drugs such as Metformin, Sulfonylureas, Meglitinides, Dipeptidyl peptidase 4 inhibitors, Sodium-glucose cotransporter 2 inhibitors, GLP1 receptor agonists, Thiazolidinediones, α-Glucosidase inhibitors and of course Insulin[21].

Quite often, patients suffering T2D experience comorbidity with another illness, which leads to an increased rate of medical complications in comparison to other patients[22]. As a consequence, T2D patients have considerably higher rates of hospital admission[23–25]. On many occasions, these admissions are due to complications of the disease such as CVD[22,24,25],Renal disease[26], Sarcopenia[27], Hypoglycemia [28], and even cognitive impariments[29,30]. Sadly, even after hospital discharge, T2D patients account for up to 20% of readmitted reported cases[31]. Patient readmission can be divided into 3 categories: planned, unplanned and unavoidable with the latter two serving as indicators for proper patient and disease treatment which act as a proxy for hospital functionality.

The higher rates of admissions and their prolonged time, and constant treatment of the disease and its patient care exert an enormous financial toll which is estimated well over US$650 billion[3]. The staggering costs of the disease are increased by the fact that hospital funding is reduced in establishments with high readmission rate and vice versa, hospitals with low readmission dates receive monetary benefits[31,32]. While the discharge decision is based, in the majority of time, on an expert professional, a machine learning (ML) algorithm trained for classification can improve the task. ML can help increase the accuracy of the discharge decision, lower readmission rate and reduce costs by and improve staff efficiency by clearing time for experts to tend to patients. In this work we aim to build and test several ML models to predict diabetic patients' readmission based on clinical data derived from the UC Irvine ML Repository.

Plenty of techniques have been implemented on this data set before including - neural network, Naïve Bayes, SVM,Linear regression, XGboost(https://journals.sagescience.org/index.php/ssraml/article/view/68/65), and random forest (ref). In most articles they found that a random forest give the best overall accuracy (OA) going up to 89.8% (https://link.springer.com/article/10.1007/s10916-020-01686-4) .However, most ML-based algorithms underperform expectations, with OA ranging from 68 to 80 percent(ref).

Preprocessing that was used included GAN to make a more even data set (Synthetic data generation based on medical data: https://medinform.jmir.org/2023/1/e47859 - they used a method called divide and conquer.). k- nearest neighbors for missing data, and PCA.

Section 2 - Description of the dataset (Metadata)

The dataset used in this project is obtained from the Center for Machine Learning and Intelligent Systems at the University of California, Irvine. The data consists of comprehensive clinical records across 130 hospitals in the United States over 10 years (1999–2008), having 101,766 encounters (visits) and 71,518 distinct patients. The diabetes ICD-9-CM code has the “250.xx” pattern. Out of the total number of patients, 5747 patients were primarily diagnosed with diabetes mellitus. The data contains 50 attributes such as encounter ID, patient number, demographics (age, sex, and race), admission type, time in hospital, medical specialty of admitting physician, number of lab tests performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization. Table 1. contains the full list of features and descriptors. Detailed information related to the primary diagnosis appears in Table 2. All the records in the dataset had to meet the following criteria:

1. They had to be inpatient encounters, which means hospital admissions;
2. The patient had to have received a diabetes diagnosis of some kind during the admission;
3. The length of stay had to be between one and 14 days;
4. Laboratory tests had to be performed during the encounter;
5. Medications had to be administered during the encounter.

Every record was labeled as to whether the patient was readmitted within 30 days (<30), readmitted after 30 days (>30), or not readmitted at all (NO).

A plethora of machine learning techniques have been explored in previous studies utilizing the same dataset under scrutiny. These methodologies encompass a diverse range, from the robustness of neural networks to the simplicity of linear regression. Notable approaches include Naïve Bayes, Support Vector Machines (SVM), XGBoost, and the versatile random forest algorithm. The study by (Dixit, R.R., 2021) delves into the efficacy of these methods, highlighting the comprehensive accuracy achieved by random forest, peaking at an impressive 89.8% but this study has no code published and it doesn’t write in detail about the methodology and therefore it cannot be trusted. Despite the prominence of random forest in achieving high accuracy rates, a prevailing trend across the literature reveals a common struggle: the performance of machine learning algorithms often fails to meet expectations. Reported overall

accuracies typically range from 68% to 80%, as observed in various studies (source: referenced but unspecified).[33]

In preprocessing, techniques such as Generative Adversarial Networks (GAN) have been employed to synthesize additional data, addressing class imbalances and enhancing model performance. The study by (Kang et al., 2023) illustrates the efficacy of this approach through "divide and conquer" methodologies. Furthermore, strategies like k-nearest neighbors for imputing missing data and Principal Component Analysis (PCA) for dimensionality reduction have been instrumental in enhancing data quality and facilitating model training.[34]

In summary, while random forest emerges as a formidable contender in achieving high accuracy rates, it's crucial to consider each machine learning algorithm's nuanced strengths and weaknesses. Additionally, preprocessing techniques play a pivotal role in data preparation, addressing challenges such as class imbalances and missing data. Future research endeavors should focus on integrating multiple methodologies to leverage their respective advantages and mitigate inherent limitations, ultimately advancing the efficacy of machine learning applications in medical research and beyond.

Section 3

Your choice of objective (target/s and features)
We set the target from multi-class to binary.
Moving to a binary label enables a simpler model with fewer chances to overfit and generally more successful (excluded “>30”).

• What is your selection for performance measures?
• List your assumptions and explain if you managed to verify them.

Look at the big picture (Quick look at the data structure)

Objective
Our ML problem is supervised because we feed the labels into the learning algorithm (in our case, whether a patient was readmitted within 30 days or not).

Performance measure
Classification model,
A balanced accuracy score was selected because we started with an imbalanced dataset and it was important to consider the minority label and weigh it, otherwise, the score is meaningless. It is a more accurate metric for the entire dataset.

started for imbalance labels and kept the same measure for comparison to be consistent throughout the project, even on the synthesized to equal labels.

- In general, we care more about minimizing FN. We don't want to miss readmission cases, but care less about classifying a patient who will be readmitted but not return.

Assumptions

Two major factors guide us in this project:

1. Financial perspective - hospital costs, insurance.
2. Patient perspective - life-threatening situations, life quality

Section 4 - Cohort allocation

We split our cohort to an 80% training dataset, and 20% testing dataset using the StratifiedGroupKFold function got scikit-learn, grouping by the patients' number ID and stratifying by our target label. Post division, no patients' number ID intersection was observed between the train-test datasets. Moreover, the ratio between the three target labels was kept across the datasets when compared to the original dataset, as seen in the table below.

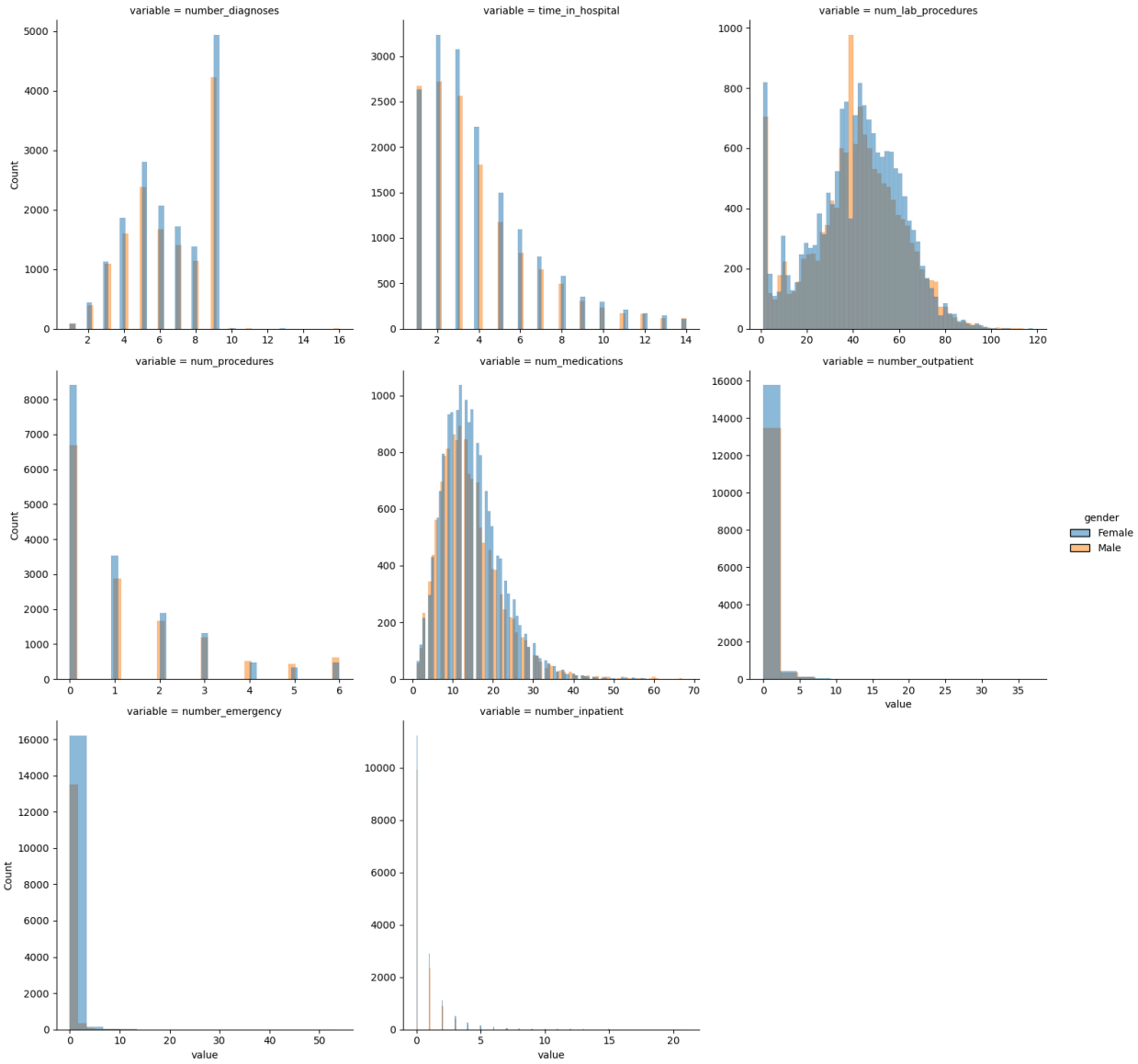
Label	Complete dataset	Train dataset (80%)	Test dataset (20%)
NO	0.547102	0.547107	0.547080
>30	0.343362	0.343360	0.343372
<30	0.109536	0.109533	0.109548

Section 5 - Exploratory data analysis (EDA)

Numerical

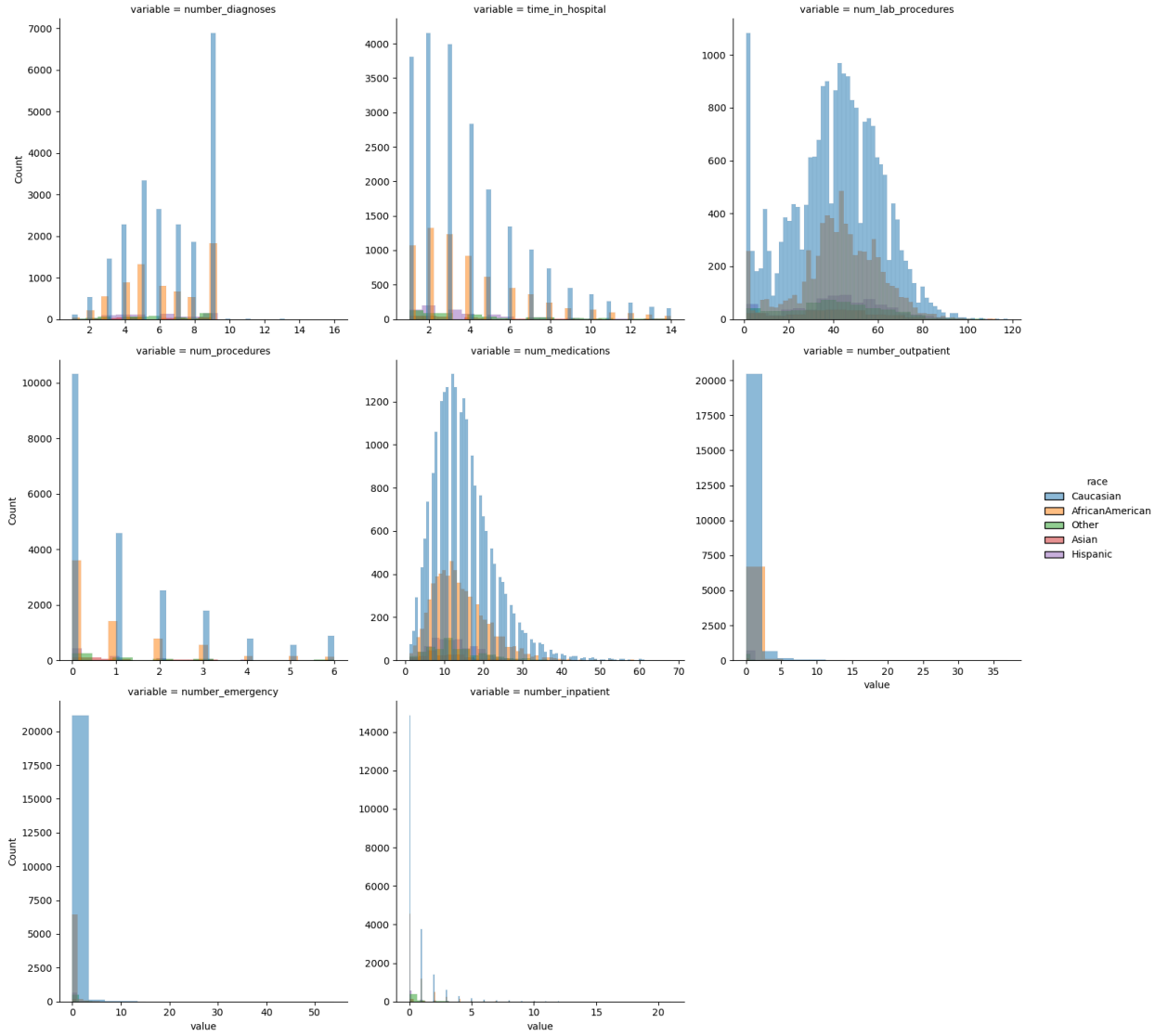
Numerical features colored by gender

There are 8 numerical features in total. The distribution of the numeral features shows a relatively normal distribution of “Number_diagnoses” and “num_lab_procedures”, and “num_medications” which behave similarly between males and felmales. However, one can see very frequent values of 9 and 0 counts of Number_diagnoses” and “num_lab_procedures, respectively. The feature “num_medications” is strongly right-skewed, therefore, we applied Log2 transformation to approximate normal distribution.



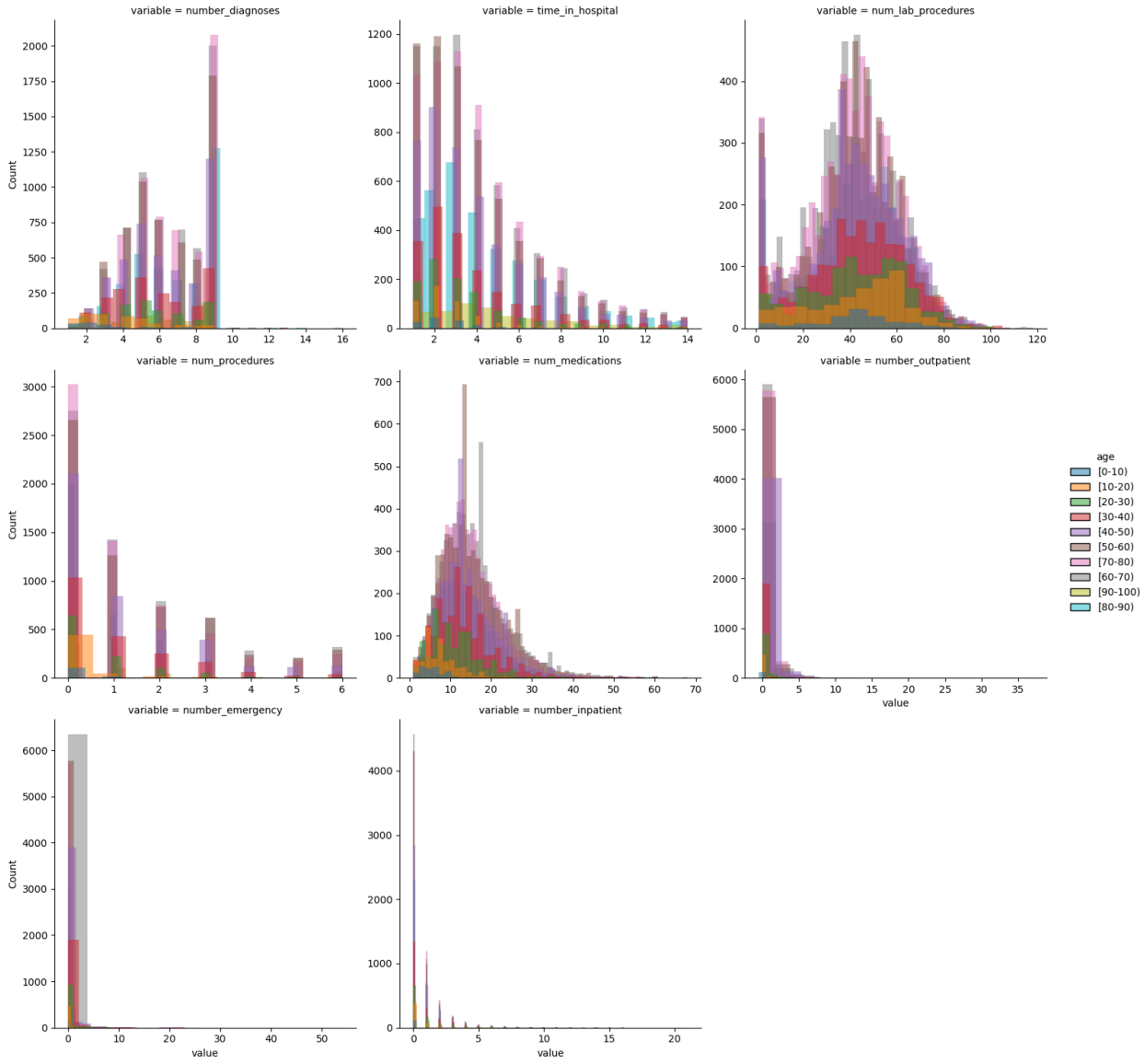
Numerical features colored by race

Looking at the numeric distributions colored by race, it seems that the race doesn't affect the distributions



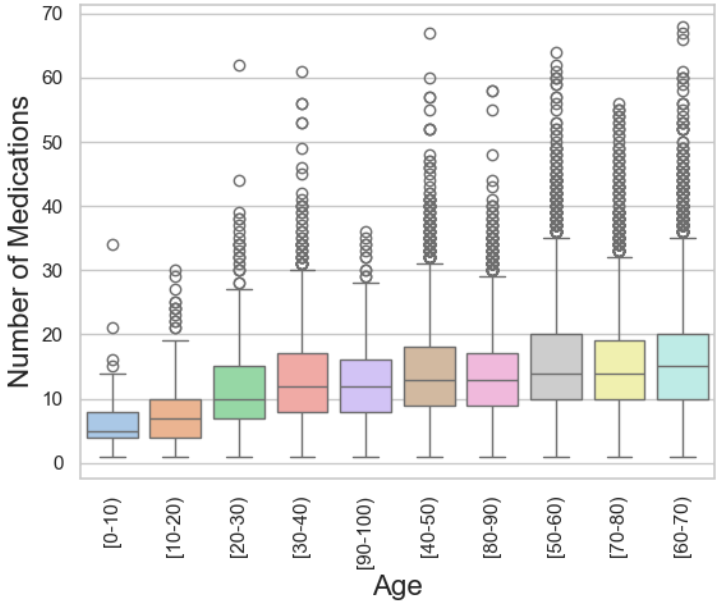
Numerical features colored by age

Looking at the distributions colored by age ranges, one can see that the distributions of feature num_lab_proceedures are affected by age (shape differences). This can be explained biologically, the distributions become more normal as the population ages.



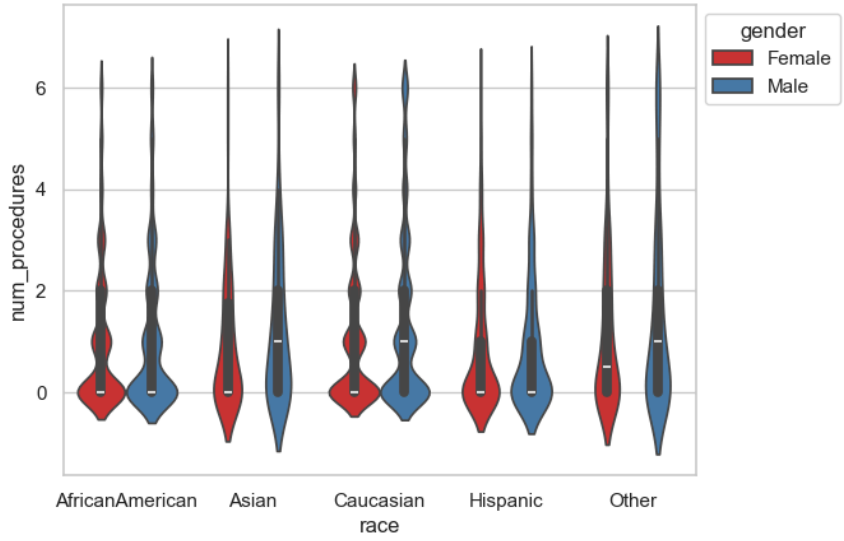
Number of medications across ages

We hypothesized that older people receive more medications. However, according to the following observation, sorted by median values, the number is highest for 60-70 but lower for 90-100 (maybe this is why they live for so long without many medications).



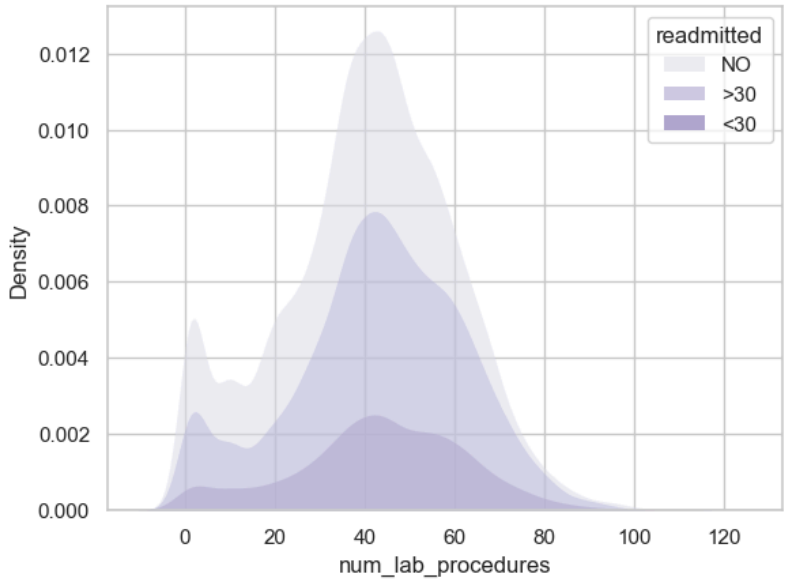
Number of procedures among gender different races

Interestingly, among the races: Asian, Caucasian, and Other, Males have increased “num_procedures” which suggests that males are “tested more” than females.



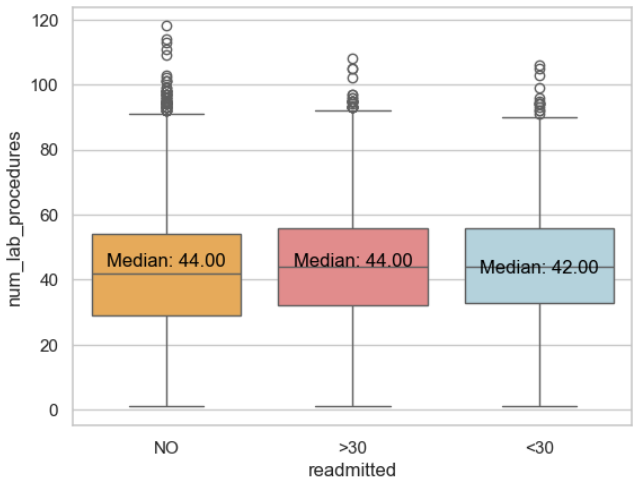
Readmission and number of lab procedures

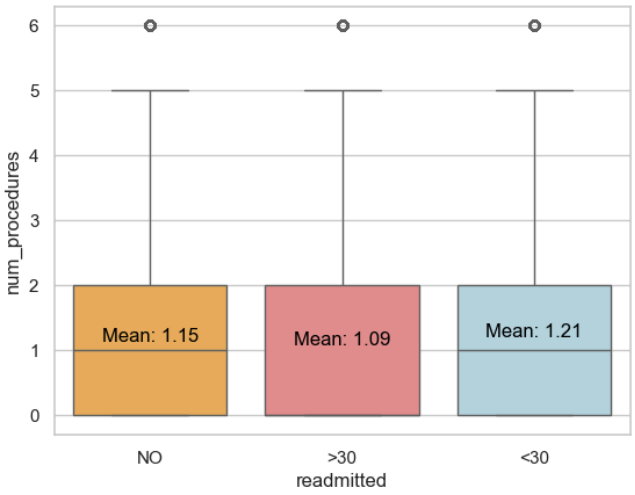
We hypothesized that a higher number of lab procedures might be associated with a reduced readmission rate. However, the density distribution of the labels looks similar.



Number of lab procedures and number of procedures

Do a higher number of lab procedures lead to a lower readmission rate? (lower chance of missing a problem). It seems that overall the median of num_lab_procedures is similar with a slight decrease in readmission<30 (42 compared to 44). Also, there are many outliers in the number of lab procedures, with more than 90 lab procedures. The mean of num_proceedures is comparable in all labels.



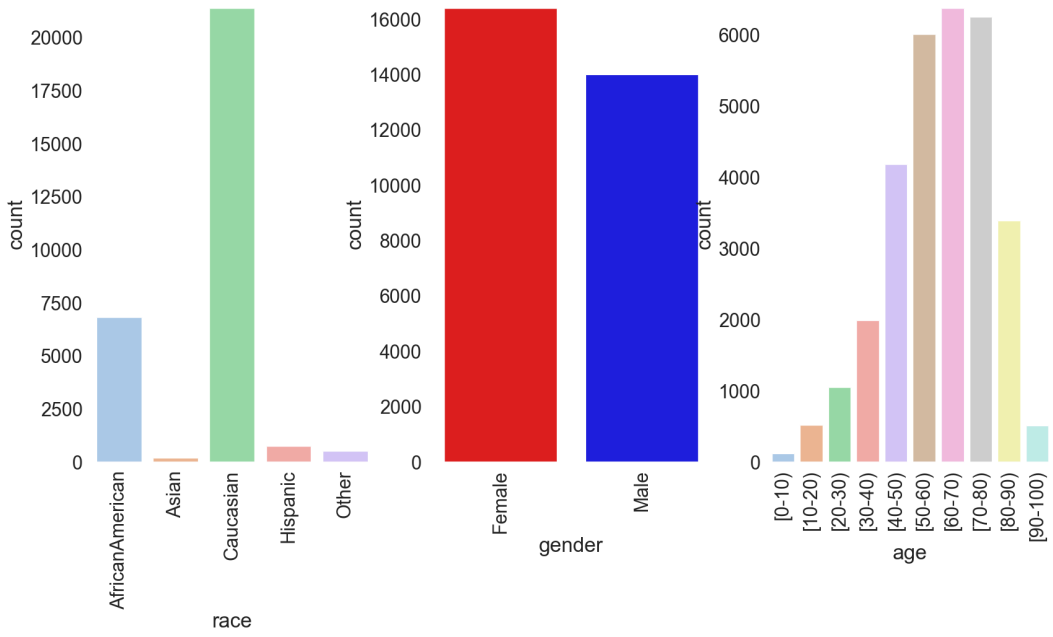


Categorial

There are 32 categorical features in total. First, we looked at the demographic features: race, gender, and age.

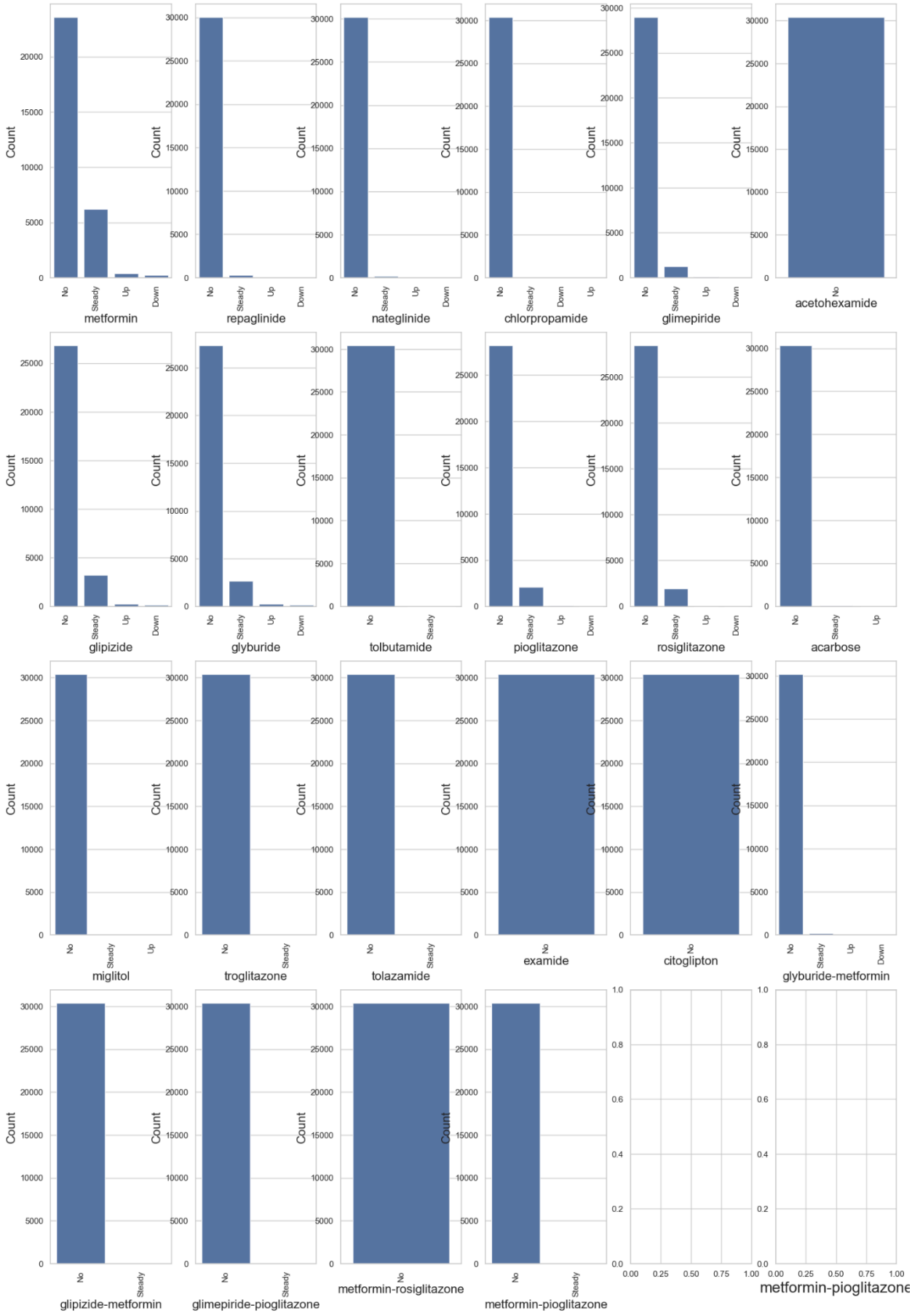
Insights:

1. The most abundant race is Caucasian, followed by African American.
2. There are slightly more females in the data compared to males, 53.96 and 49.06%, respectively.
3. The mode range of ages is 60-70 followed by 70-80 and 50-60 age ranges. One can immediately see the low count of patients among the ages 0-10.

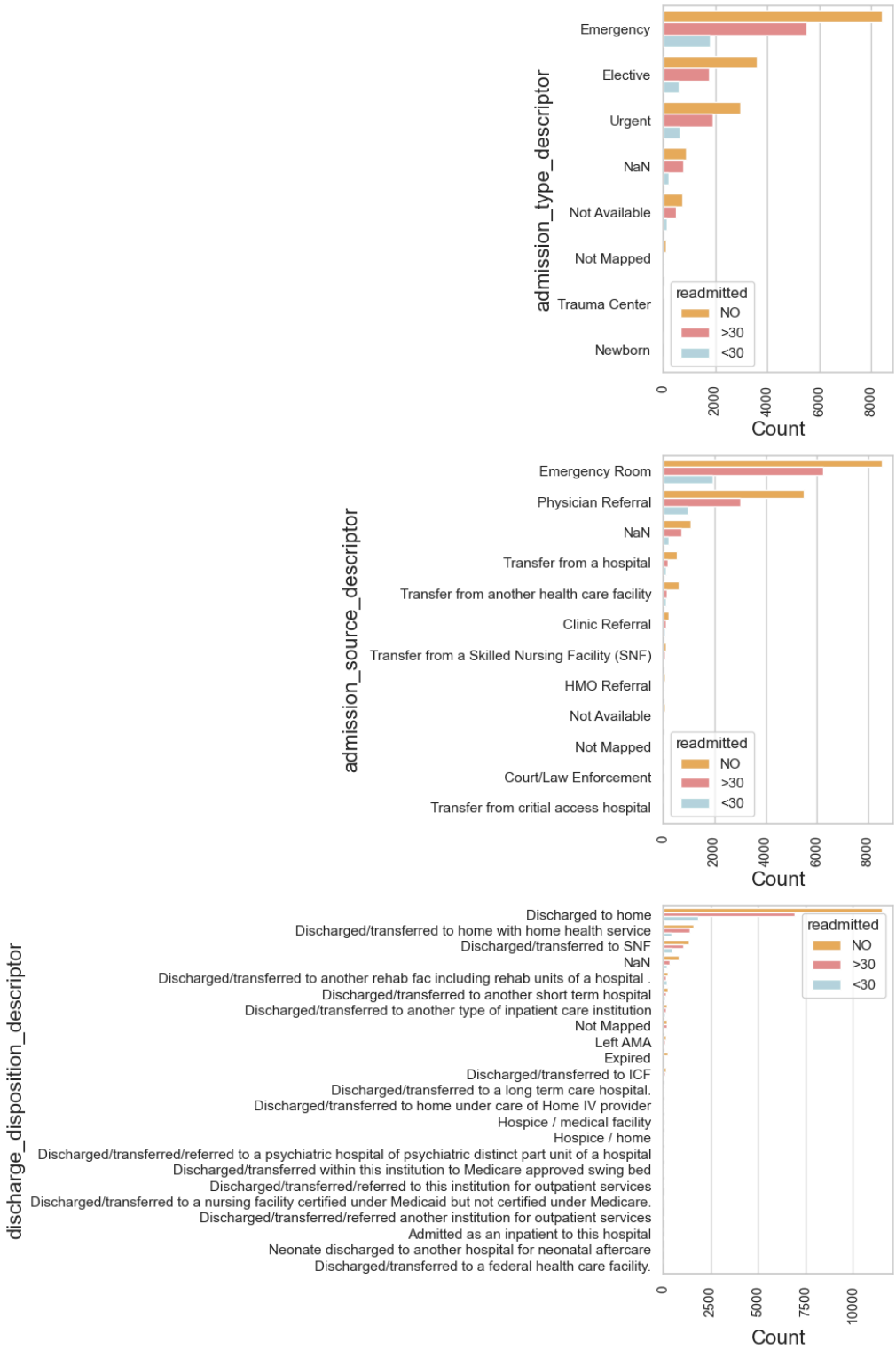


Medications

Next, we wanted to see the medication status and their combinations (n = 22). Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed. A first glance at the count plots data shows that The most frequent value in all medications is “no”. Also, The medications ‘acetohexamide’, ‘examide’, ‘citoglipton’ as well as the combo of metformin-rosiglitasone have only 'No' in the training set. Therefore, for reasons of zero variance, we excluded them from the downstream analysis.

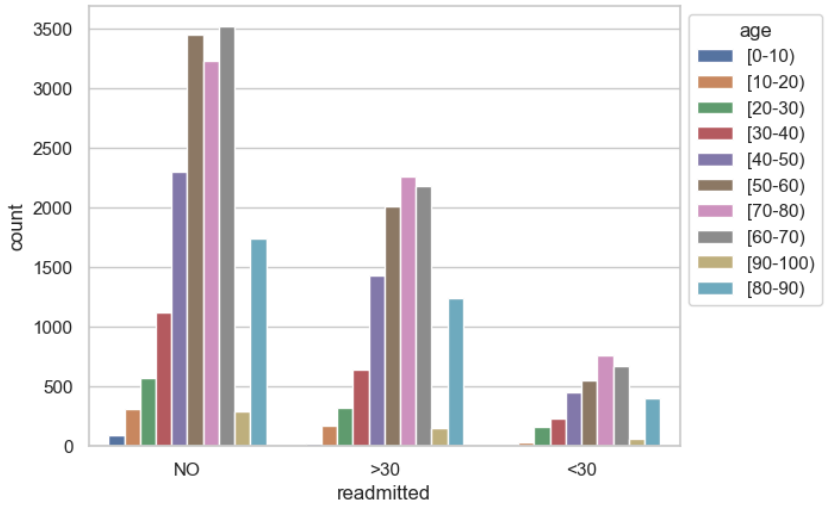

Count
metformin
repaglinide
nateglinide
chlorpropamide
glimepiride
acetohexamide
glipizide
glyburide
tolbutamide
pioglitazone
rosiglitazone
acarbose
miglitol
troglitazone
tolazamide
examide
citoglipton
glyburide-metformin
glipizide-metformin
glimepiride-pioglitazone
metformin-rosiglitazone
metformin-pioglitazone
No
Steady
Up
Down

To have meaningful insights on: admission_type, admission_source, and discharge_disposition for EDA purposes, we used the mapping_IDs table to merge and convert numerical IDs into informative descriptions. The most common admission type is emergency, with the expected most frequent admission source Emergency room.



Count plot of readmission colored by age range

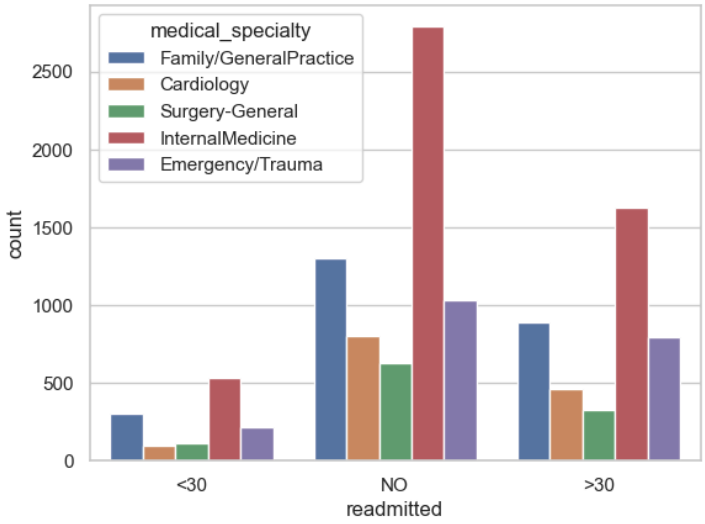
The age ranges are distributed similarly among the readmission labels according to their initial count.

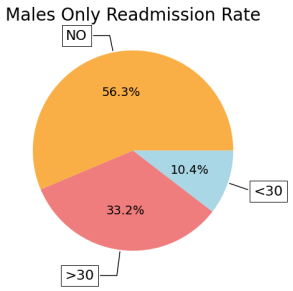


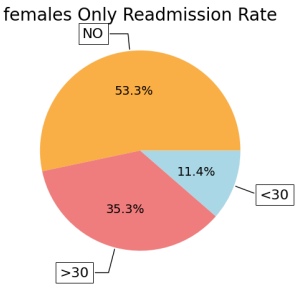
Top 5 medical specialties and their labels

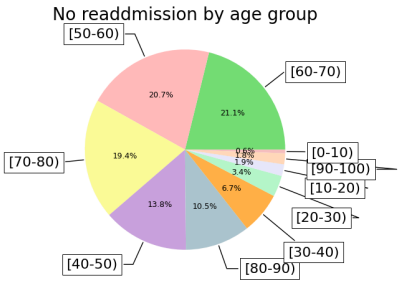
We were interested in the leading medical specialties and the association to readmission. Do specific specialty better or worse in “missing” valuable information that leads to readmission? Overall, the most frequent specialty is internal medicine.

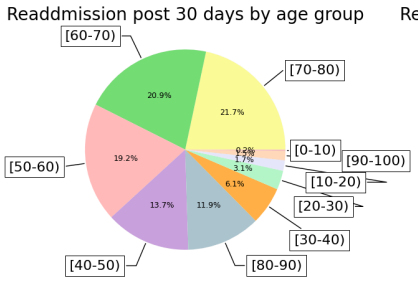
InternalMedicine	4946
Family/GeneralPractice	2491
Emergency/Trauma	2045
Cardiology	1363
Surgery-General	1071

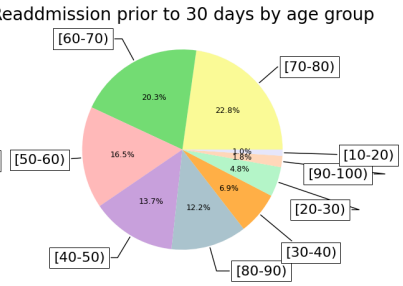


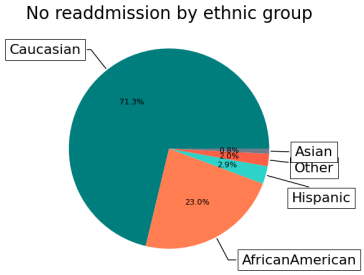

Males Only Readmission Rate
NO
56.3%
10.4%
<30
33.2%
>30

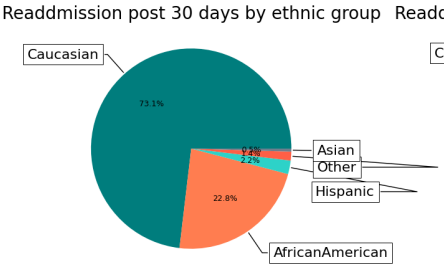

females Only Readmission Rate
NO
53.3%
11.4%
<30
35.3%
>30

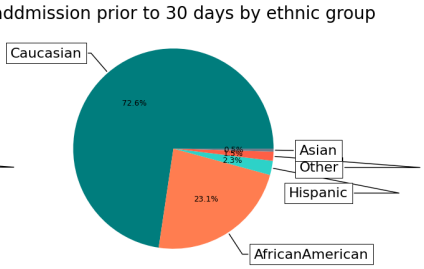

No readdmission by age group
[50-60)
[60-70)
20.7%
21.1%
[70-80)
19.4%
[0-10)
[90-100)
[10-20)
[20-30)
[30-40)
[40-50)
[80-90)
13.8%
10.5%
6.7%
3.4%

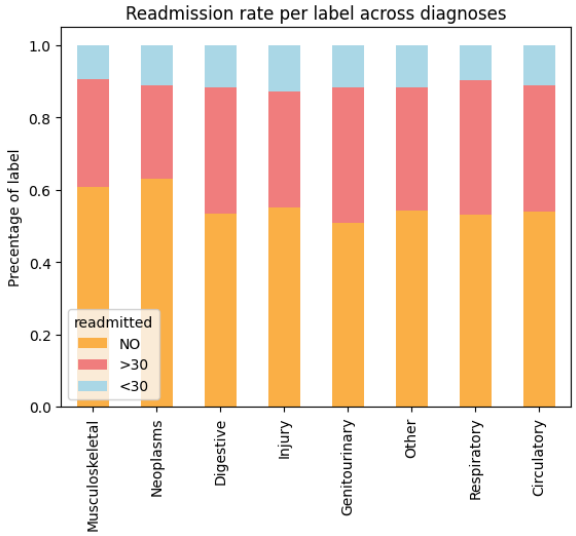

Readdmission post 30 days by age group
[60-70)
[70-80)
20.9%
21.7%
[50-60)
19.2%
[0-10)
[90-100)
[10-20)
[20-30)
[30-40)
[40-50)
[80-90)
13.7%
11.9%
6.1%
3.1%


Readdmission prior to 30 days by age group
[60-70)
[70-80)
20.3%
22.8%
[50-60)
16.5%
[10-20)
[90-100)
[20-30)
[30-40)
[40-50)
[80-90)
13.7%
12.2%
6.9%
4.8%


No readdmission by ethnic group
Caucasian
71.3%
Asian
Other
Hispanic
AfricanAmerican
23.0%
2.9%


Readdmission post 30 days by ethnic group
Caucasian
73.1%
Asian
Other
Hispanic
AfricanAmerican
22.8%
2.2%


Readdmission prior to 30 days by ethnic group
Caucasian
72.6%
Asian
Other
Hispanic
AfricanAmerican
23.1%
2.3%


Readmission rate per label across diagnoses
Precentage of label
1.0
0.8
0.6
0.4
0.2
0.0
readmitted
NO
>30
<30
Musculoskeletal
Neoplasms
Digestive
Injury
Genitourinary
Other
Respiratory
Circulatory

Section 6 - Data Preparation (preprocessing)

Here, we detail the data engineering steps we have performed. We checked that the dataset complies with the Tidy data requirements:

1. Each variable (Feature) forms a column
2. Each observation (Target) forms a row
3. Each type of observational unit is stored in its own table

Make existing features more informative
Using the 'IDS_mapping.csv' table, we convert diagnosis values from numbers to valuable string categories and removed original integer columns.

Samples removal
Observations in the age range 0-10 were excluded due to low counts and no readmission labels. Also, observations with ‘Trauma Center’ and ‘Newborn’ in the admission_type were excluded as they have only five records.

The following steps were introduced in scikit-learn pipeline format.

Feature removal
We had four criteria to remove features from the dataset:

1. **Irrelevant** features based on domain/common knowledge.
2. **Sparseness**
    a. features with more than 50% missing values prior one hot encoding
    b. One hot encoded features with <1% positive value post one hot encoding
3. **Low variance** in the feature cannot contribute to the model.
4. **Redundancy,** for example, 'admission_source_descriptor’ was very similar to the admission type. Also, it is evident that if the ‘description_admission_type’ is ‘Emergency’, then the admission_source_descriptor would be “Emergency room”.

The following features were removed from the dataset based on criteria 1,3 & 4:
IRRELEVANT_FEATURES = [
'nateglinide','chlorpropamide','tolbutamide','acarbose','miglitol','troglitazone', 'tolazamide','glyburide-metformin','glipizide-metformin','glimepiride-pioglitazone','metformin-pioglitazone','admission_source_descriptor','repaglinide', ‘payer_code’, ’patient_nbr’ (after splitting to train and test), 'diag_1','diag_2','diag_3' (these features had special encoding, see below), ,'admission_type_id','discharge_disposition_id','admission_source_id']

New features
Encoding
One Hot Encoding was performed on all categorical features except for the medications.
A unique encoding method was created for medications (OHE_4_to_2_cols). Many medications were removed due to low variance reason (see feature removal explained above).

Therefore, only medications with ‘Up’, ‘Down’, ‘Steady’ and ‘No’ were kept and the 4 categories were merged into 2 to decrease the curse of dimensionality. ‘Up’, ‘Down’ and ‘Steady’ transformed into 1 (change in medication), and ‘Steady’ and ‘No’ transformed into 0 (no change in medication). This special encoding allowed us to reduce dramatically the number of features Thus, avoiding the curse of dimensionality while also retaining important knowledge based on domain expertise that changing the dosage of a medication may indicate a medicinal state regardless of the change’s direction.
Based on this feature, the feature “change” was modified accordingly, to indicate whether the patient had a change in the medication type or not

We thought that although ‘age’ is a categorical feature in this dataset, the age ranges are identical (ten years gap) and there is a meaning to the directionality (e.g 40-50 is higher than 30-40). Therefore, we applied the LabelEncoder method on ‘age’. (0-8)

Imputations
In our pipeline, we imputed the following features: race and medical_specialty. Missing values in race were imputed with “Other” because, in any case, “other” can represent many unknown races. However, medical_specially was imputed with the mode (most frequent value) which is “internal_medicine” (see top 5 specialties in EDA section). This is based on domain knowledge as well as a consultant with MD, who confirmed that when a patient is admitted in the ER, the chances to be accessed by an internal doctor are the highest.

Outliers
Based on the EDA we saw some outliers in the features: ‘num_medications’, ‘num_proceedures’, and ‘‘num_lab_proceedures’’. Nevertheless, since most of our features are categorical, we decided to keep the observations without any outlier exclusion. Also, maybe these features with so-called IQR outliers, are important to predict the labels more accurately.

Numerical Features which were encoded as NUMERICAL = ['time_in_hospital', 'num_lab_procedures', 'num_procedures', 'num_medications', 'number_diagnoses', 'number_outpatient', 'number_emergency', 'number_inpatient'] were all scaled and centered by scikit-learn StandardScaler function with the exception of ‘num_medications’ which was $\log_2$ transformed prior to scaling after examination if its distribution in the EDA part to approximate it to a normal distribution.

In addition, prior to every model’s fitting we used LabelEncoder to our target to encode values to 0s and 1s.

Finally, to avoid a multiclass labeling problem requiring more data to predict accurately, and based on domain expertise after consulting doctors, we decided to focus only on predicting patients who were readmitted back to the hospitals below 30 days from their discharge. As the doctors explained, these patients tend to have more severe symptoms and are more likely to have been misdiagnosed or overlooked in their first admission. Finding features that explain and detect this rapid readmission could benefit these particular patients healthcare and wellness

Due to the imbalance nature of the data, we sought to understand whether oversampling could help improve our models performance. Therefore we tested two different techniques based on python off-the-shelf modules; CTGAN, CapulaGAN and Synthetic Minority Over-Sampling Technique for Numerical and Categorical data (SMOTE -NC).

Oversampling

Given that our minority class is underrepresented (as in many cases in ML projects), we tested three different strategies of oversampling the minority class, "<30" in our case. Both CapulaGAN and CTGAN are specialized GAN architectures for generating synthetic tabular data, but they employ different techniques to capture the underlying data distributions and dependencies. Because we don't have any apriori intuition which GAN model will work better, we tried both.

Conditional Tabular GAN (CTGAN)
To better represent our original train-test and avoid overfitting bias in our model training we split our training data to five folds. At each fold of the GAN, we split into a 20-80 train-validation cohorts.

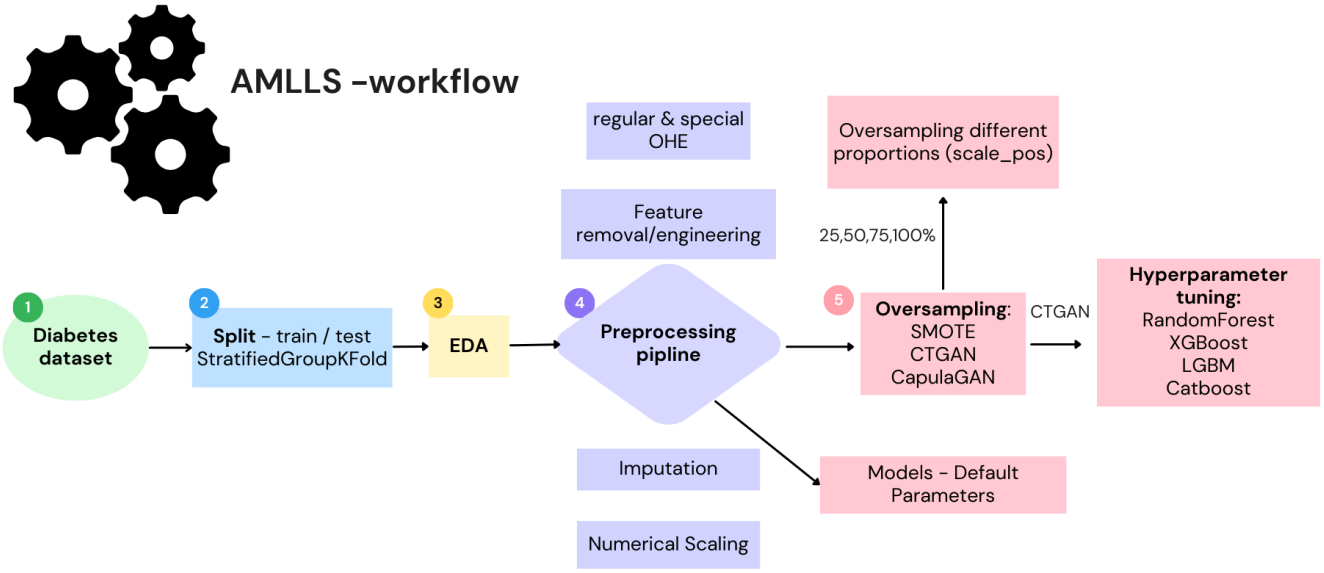
CapulaGAN
The same approach was applied in CapulaGAN imported from the SVD module.

Notes related to GAN: In both GAN models, the synthesized data was associated with three IDs that were used later in the hyperparameter tuning step cross-validation in the 'folds' argument. Thus, having the synthesized information of "encounter_ID", fold_num (1-5), and whether it is the train or validation sample, we could make sure that the same validation cohort is used for the specific training fold with the correct "encounter IDs". For more information, see "fold_inx.csv" in GitHub.

SMOTE-NC
The five folds used in the GAN methods were used in the SMOTE-NC methods as well to oversample the training subset and then evaluate models' performances on the held out validation subset. Since the method requires feature scaling and centering but does not require imputation and one hot encoding we divided our pipeline into two parts. The first removed features and scale and center the numerical features. Then we oversample using SMOTE-NC following the second part of our pipeline which includes imputation and categorical columns encoding and target label encoding. In each fold, columns the were found in the training subset after OHE but not in the validation subset were added to the latter as all 0s columns. Vice-versa columns that were found in the validation subset data but not in the training subset were removed. In general only columns that were found in the complete dataset after our initial pipeline were considered in our analyses.

Graphical Abstract of our workflow

May-2024, Weizmann Institute of Science



Tal Weizmann, Yahel Cohen, Shiri Karagach

Section 7 - Models training

We first wished to evaluate the following questions, oversampled dataset or not, with which methods of oversampling, and what models we wished to focus on as best performing ones. We, therefore, compared a list of seven different models with their default parameters across four datasets, the original without synthesized data or oversampled data with synthesized records balancing between labels completely with SMOTE-NC, CTGAN, or CapulaGAN. The models tested were: Logistic regression, Tree classifier, Support vector machine (SVM), Extreme gradient boosting classifier (XGBoost), Category boosting classifier (CatBoost), Light gradient boosting method (LGBM) and additionally Random forest classifier. For the original data, where label imbalance was prominent, the BalancedRandomForest classifier (BRF) from imblearn module was applied rather than Scikit-learn Random forest classifier. All are models suit for the classification problem.

All models were trained and evaluated in a five-fold cross-validation. In the original data, we used the StratifiedGroupKFold for fold assignment. In the oversampled data we used the same folds for validation used when training the GAN method. At the same time, training was done on the same training subset with additional synthesized data to balance the target labels completely.

We chose the balanced accuracy which is equal to the arithmetic mean of sensitivity and specificity, as a measure to evaluate each model's performance. In the case of balanced target labels it is equal to the accuracy, while in case of imbalance it corrects for it by dividing the accuracy by half in our case.

Figure 1 and Table 1 summarize the results of all default models. In most models oversampling increased the model's performance. While the best performance of an oversampled dataset on average was found on the oversampled data by SMOTE-NC (average imbalanced accuracy = 0.595) we feared it may be due to uneven noise introduced by the system that may lead to overfitting. Therefore we chose to continue with oversampled data by CTGAN, which had the second best performance on average (average imbalanced accuracy = 0.568). Of all 6 models we chose to optimize the ensemble ones (XGBoost, LGBM, Catboost & RF) though RF and LGBM had lower scores than Tree and Logistic regression. This is because we believed that more options for hyperparameter tuning for these models will increase their performance better. Noteworthy, BRF on the original data performed best across all models in all datasets, therefore we chose to include him in the list of models with tuned in a later stage.

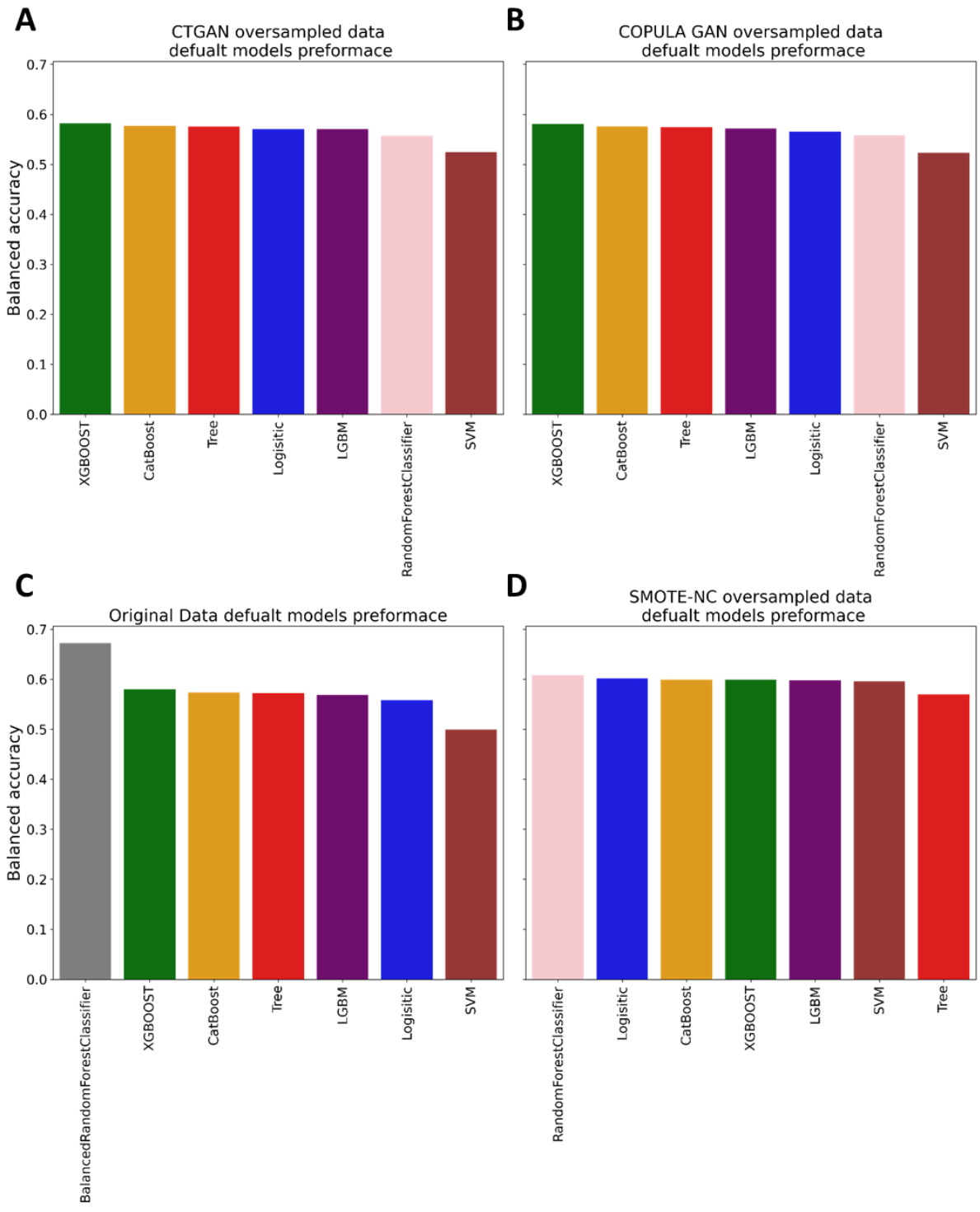


Figure 1 - Oversampling improves performance in most models:
Bar plots of model performances by balanced accuracy in 5-fold cross-validation across all datasets. A) Oversampled data by CTGAN, B) Oversampled data by CapulaGAN, C) original i, balanced dataset. D) Oversampled data by SMOTE-NC. Models are color-coded as well Green - XGBoost, Yellow - Catboost, Red- Tree, Logistic - Blue, Purple - LGBM, Pink - Random Forest, Grey = Balanced Random Forest, Brown - SVM.

Dataset	XBoost	Catboost	LGBM	RF	BRF	Tree	SVM	Logistic regression
original	0.580	0.574	0.569	N/A	0.672	0.572	0.5	0.558
CTGAN	0.583	0.577	0.57	0.525	N/A	0.576	0.578	0.571
Copula GAN	0.58	0.576	0.571	0.558	N/A	0.575	0.523	0.566
SMOT	0.599	0.599	0.598	0.608	N/A	0.57	0.596	0.602

Table 1 - Summary of balanced accuracy across four datasets and models

Section 8 - Hyperparameter tuning

To find the best set of hyperparameters, we used the automated framework Optuna, followed by manual fine-tuning using our intuition and documentation Hyperparameters table, to avoid overfitting without compromising the accuracy. In other words, reaching the Bias-Variance minimum. Each optimization was conducted in five folds similar to the ones used to train the GAN itself, with the exception of the training dataset was now balanced to maximum

Prior to optimizing all other hyperparameters, we wished to check the effect of balancing the data to different degrees while adjusting for the weight given to each label in the prediction in parallel. We sampled synthesized samples to balance our dataset using CTGAN to different target label ratios of 0 0.25, 0.5, 0.75, and balanced. We then fitted a default XGBoost model and used Optuna to search for the best value of the 'scale_pos_weight' hyperparameter across 100 iterations, each time computing the mean logistic loss of the model and comparing it to the lowest score achieved and to the default model's score. For the completely balanced dataset we just calculated the default model's performance as no imbalance weight is required.

As can be seen in Table 2 adjusting of imbalance weights only decreased the performance of the model in each oversampling ratio. In addition, while the best performance was on the original dataset, it had only a minor improvement on the completely balanced dataset. Considering the results from section 7, we decided to continue with the completely balanced dataset.

Ratio	Default logistic loss	Post tuning logistic loss
No oversampling	0.396378	0.490557
25% balanced	0.401599	0.487940
50% balanced	0.425823	0.488971
75% balanced	0.407658	0.485966
Balanced	0.401137	N/A

Table 2 - mean logistic loss across 5 folds cross-validation of different oversampled ratio datasets

XGBoost:

Hyperparameter tuning for the XGBoost model was done in two steps using GPU. First the hyperparameters, "num_estimators" ,"max_depth", "learning_rate", "subsample", "gamma" And "min_child_weight" were optimized than fixed (Highlighted in green) followed by optimization of the rest of the hyperparameters mentioned in table 3 below (highlighted in yellow).

Each optimization step had 200 iterations across five folds. In each iteration the mean logistic loss of the model was calculated and compared to the lowest score achieved by now and that of a model with a parameter known to yield reasonable prediction i.e. 'max_depth' = 5, 'subsample' = 0.8, 'gamma' = 0, 'colsample_bytree' = 0.8, ‘seed’ = 42. We run 5 steps each time changing the search space under model parameters converged to the best outcome.

Optimization improved the model’s prediction on the held out test dataset as measured by AUC score increase of 0.01 from 0.72 to 0.73 (Figure 3A) and an increase of the balanced accuracy of 1.3 points from 0.634 to 0.648 (Figure 3F).

Type	Value
num_estimators	803
max_depth	4
subsample	0.8017681217877746
gamma	2.430402439508061
colsample_bytree	0.8301598634480943
learning_rate	0.1629114048891201
min_child_weight	7
seed	42
lambda	3.394448397873383
alpha	2.583061480631797
colsample_bynode	0.9631059074619839
max_delta_step	4
grow_policy	depthwise
sampling_method	uniform
colsample_bylevel'	0.7183281335547681
'max_leaves'	104

Table 3 - XGBoost best hyperparameter table

RandomForest (RF)

In contrast to the tuning of XGBoost, the tunings of RF BRF were done using CPU and with only one step of tuning in each. Other than that optimization was similar to that of XGBoost with 3 steps of optimization of 200 interactions.

For RF, we used 5 fold cross validation to evaluate the model and the folds were identical to those used in the GAN training and XGBoost optimization. Initially, we tried optimizing based on the mean logistic loss across all folds but we could not improve our model performance. Therefore, we changed our optimization to our measured score of balanced accuracy.

Table 4 displays the tuned hyperparameters which led to an increase of model performance on our held out test dataset, measured by both balanced accuracy (default = 0.532, tuned = 0.545, Figure 4D) improvement of 1.3 points and AUC scores (default = 0.74, tuned = 0.75, Figure 4B). Though the AUC scores were higher than our tuned XGBoost performance, the accuracy, which is more important to us, was much lower.

Type	Value
max_depth	35
n_estimators	297
min_samples_split	6
min_samples_leaf	3
random_state	42
max_features	'sqrt'
criterion	'log_loss'
max_leaf_nodes	952

Table 4 -Random Forest best hyperparameter

LightGBM

In this model, hyperparameters were tuned in one step using Optuna. Based on the importance of the hyperparameter (Figure 2 upper panel) and documentation, the optimization ranges were adjusted to improve performance and manual fixation of some of the values to follow the general trend. The performance was evaluated based on the cross-validation (CV) with the folds coupled to the same training subset explained above. After optimizing the parameters using CV, the best parameters were used in the lgb.LGBMClassifier() model and compared to base parameters that were given during the course as the base level: base_params = {"objective":"binary",'max_depth':5,'subsample':0.8, 'colsample_bytree':0.8, "seed":42} The tuned model showed improvement of 0.023 in balanced accuracy score. (Figure 2 bottom right). However, in the ROC curve, we see a worse AUC in the tuned compared to the untuned. This could be explained by the fact that we optimized the model based on balanced accuracy and not sensitivity, which not always correlate.

Cross-validation results

Balanced_accuracy with default parameters (random_state = 42): 0.4275003499901908
Best balance accuracy: 0.4345479071618842

Type	Value
num_estimators	1000
learning_rate	0.31
num_leaves	230
lambda_l1	10
lambda_l2	80
min_gain_to_split	0.2380984754136244
bagging_fraction	0.7
bagging_freq	1
feature_fraction	0.7999999999999999
random_state	42

Table 5 -LGBM best hyperparameter table

Test results

Balanced accuracy for base model: 0.577
Balanced accuracy for tuned model: 0.6
An improvement of: 0.023

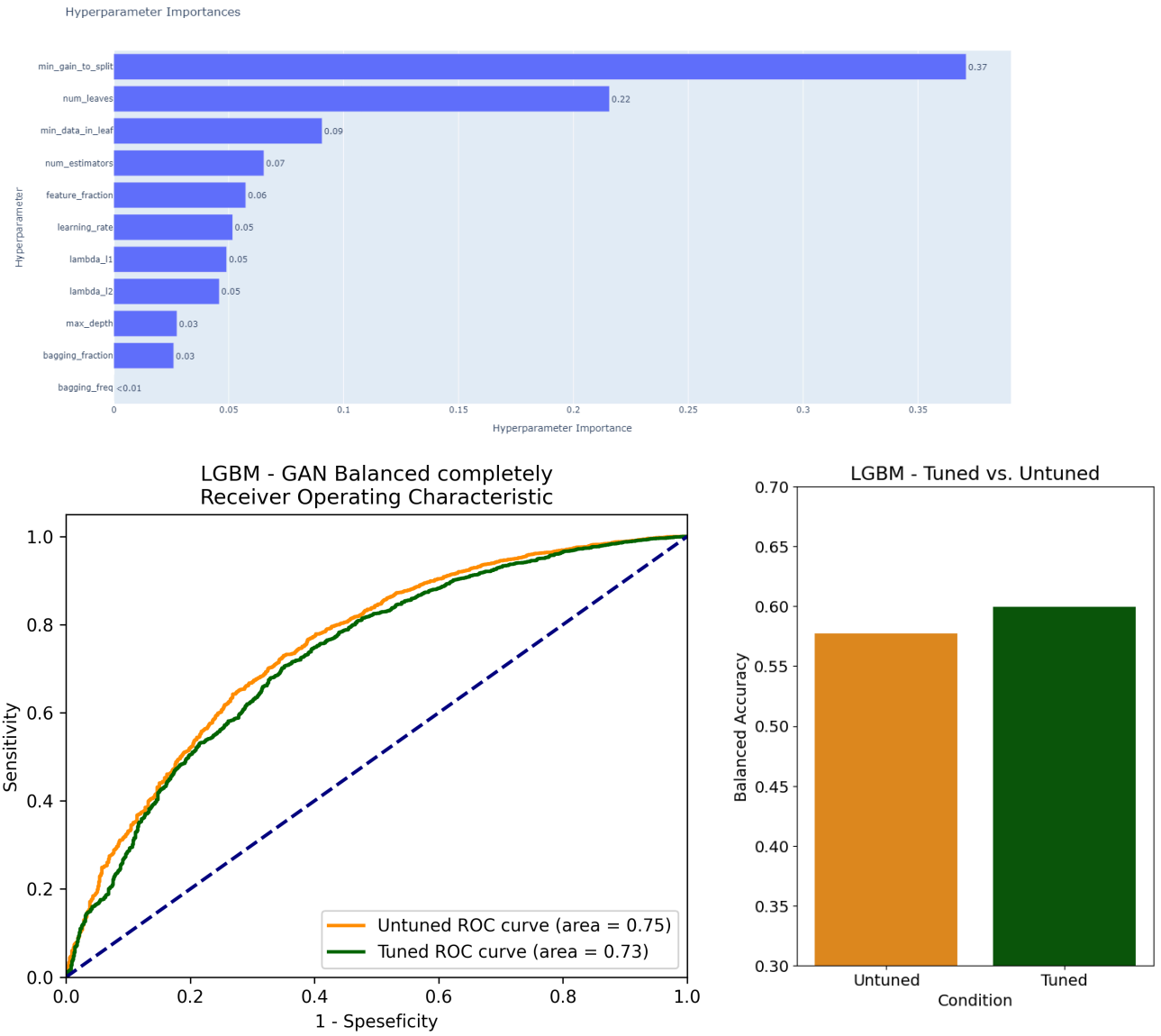


Figure 2 - LGBM hyperparameter tuning and results. Hyperparameter importances in LGBM based on Optuna optimization (upper panel). ROC AUC of LGBM. Y axis is sensitivity, X axis is 1-specificity, the orange line denotes untuned model performance, the green line denotes the tuned model's performance, and the dashed line denotes prediction by chance (bottom left panel). Bar plot of models' performance as measured by balanced accuracy. orange default untuned model, green tuned model (bottom right panel).

CatBoost:

Cross-validation results

Balanced_accuracy with default parameters (random_state = 42): 0.42809261015828276

Best balance accuracy: 0.42948118252674516

Type	Value
learning_rate	0.13948861596733592
l2_leaf_reg	88.18802352033121
bagging_temperature	1.0210260597520087
random_strength	1.2300217338005526
depth	6
min_data_in_leaf	299
n_estimators	857

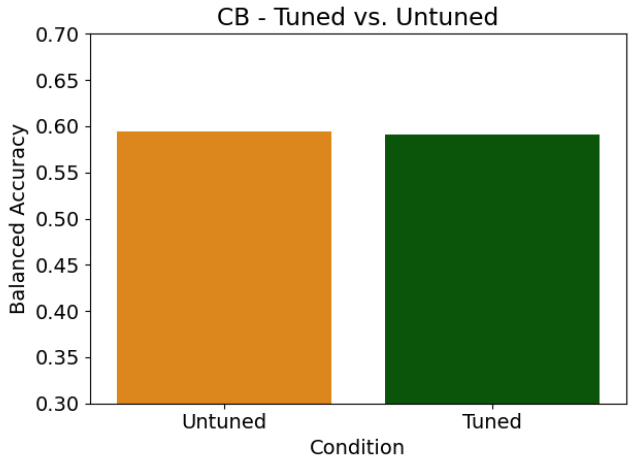
Table 6 -CatBoost best hyperparameter table

Test results

Balanced accuracy for tuned model: 0.591

Balanced accuracy for base model: 0.594

An improvement of -0.003



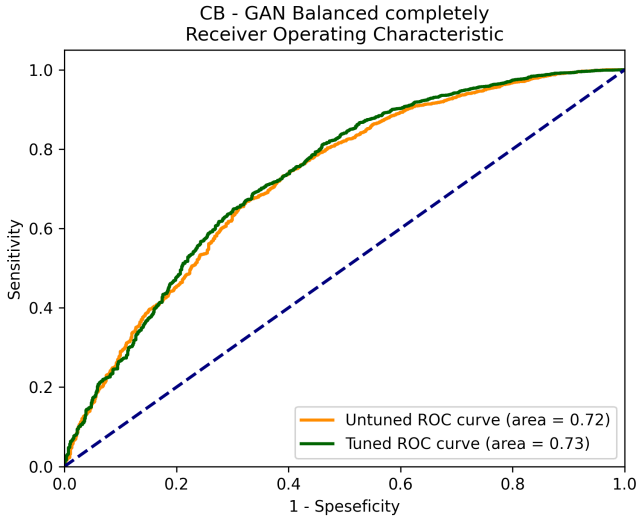


Figure 3 - Catboost hyperparameter tuning and results:
Hyperparameter-tuned Catboost based on Optuna optimization. ROC AUC of catboost. Y axis is sensitivity, X axis is 1-specificity, the orange line denotes untuned model performance, the green line denotes the tuned model's performance, and the dashed line denotes prediction by chance (right panel). Bar plot of models' performance as measured by balanced accuracy. orange default untuned model, green tuned model (left panel) .

BalancedRandomForest (BRF):

Since the default BRF model performed well on the original data we decided to tune it as well and compare its results with the rest of our models.

BRF tuning was similar to that of the RF model however, Since the BRF was on the fitted to the original data without oversampling, the folds we tested on were different, and to divide to dataset into folds we used the scikit-learn StratifiedGroupKFold function with patient number ID as a group indicator. The tuned hyperparameters are listed in Table 7. With them, The model performance improved on our held out test dataset, by 1 point in the balanced accuracy (default = 0.673, tuned = 0.683, Figure 4D) and a 1 point improvement in AUC score (default = 0.74, tuned = 0.75, Figure 3C)

Type	Value
n_estimators	618
max_depth	12
min_samples_split	10
min_samples_leaf	2
max_leaf_nodes	1171

Table 7 -Balanced Random Forest best hyperparameter

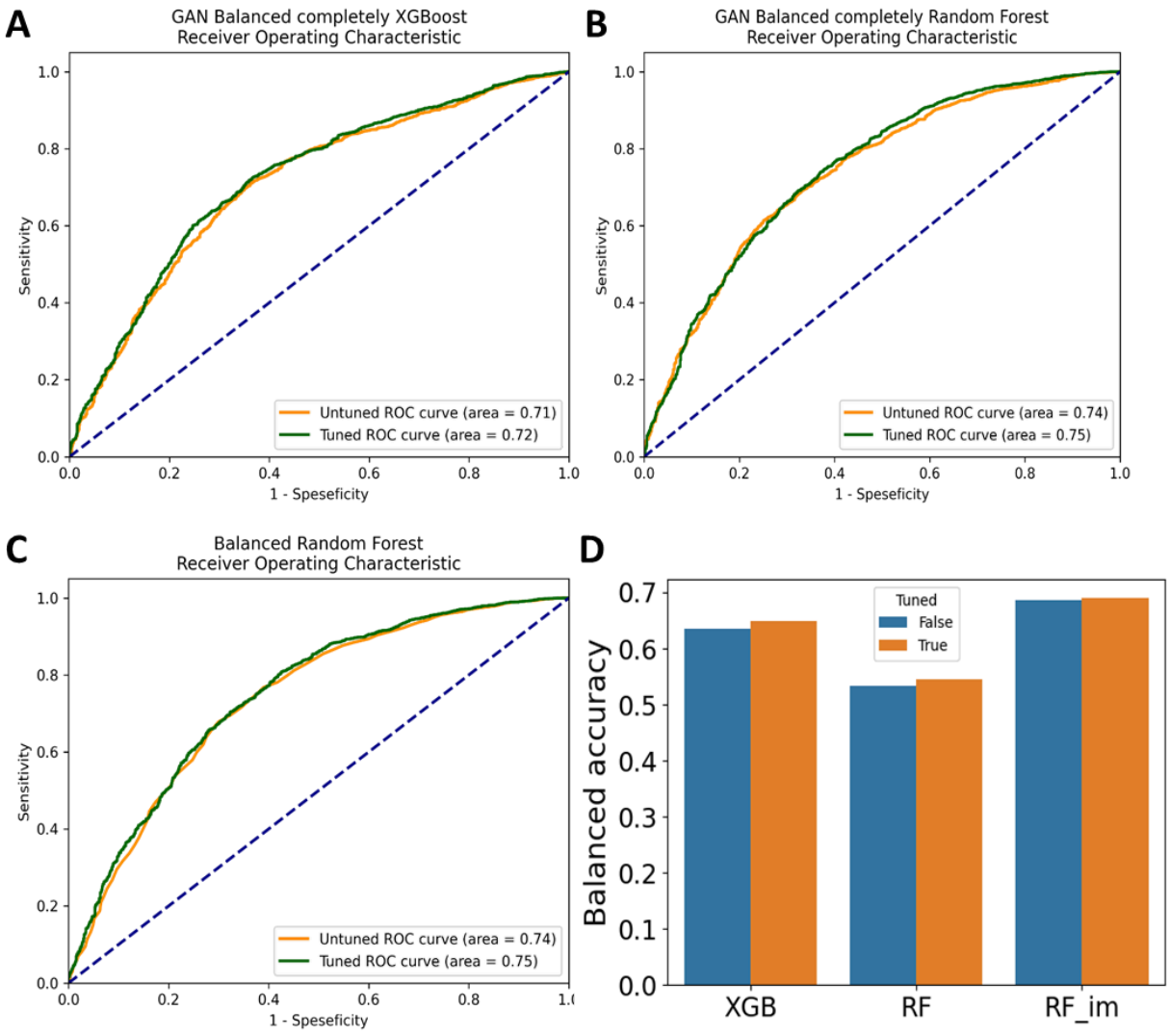


Figure 3 - Chosen models performance prior and post hyperparameter tuning:
A-C) ROC AUC of XGBoost, RF & BRF accordingly. Y axis is sensitivity, X axis is 1-specificity, orange line denotes untuned models performance, the green line denotes tuned model's performance, and the dashed line denotes prediction by chance.D) Bar plot of models' performance as measured by balanced accuracy. Blue default untuned model, orange tuned model.

These results demonstrate the power of the BRF model. As a result, we chose it as our preferred model and went on to calculate its feature importance using SHAP[35] for explainability. Furthermore, we tested its stability across different pseudo-random seeds.

When looking at the top 15 most important features as calculated by SHAP we see that the most important one is the number of inpatient visits, i.e. the number of times a patient has been hospitalized in the past year with higher occurrences related to readmission below 30 days. This is followed by whether the patients were discharged home after treatment, if he did not than his chances to readmit are higher. Other important features are number of diagnosis the patient had , the time they spent in the hospital , the number of emergency they had in the past year, number of medication they got the number of times the patient was admitted but not hospitalized (in all, higher number correlates to readmission), change in Insulin dosage (reduction leads to readmission), age at admission, the number of procedures the patient underwent (few procedures are correlated to higher admission),whether or not the admission was elective and whether the patient was moved to another ward at the end of the treatment ,but not home, if moved had higher chance to get readmitted (Figure 5A).
In addition a closer look at the effect Age has on the model revealed an age dependent effect where only group ages 6 (70-80) and 4 (50-60) seem to have an effect on the readmission, the former predicting readmission while the latter correlates to non-readmission (Figure 5B).

The number of inpatient visits was so correlated to readmission that by itself it had more impact than all other 43 features not mentioned here together (Figure 6A). While number of diagnosis and discharge home were equally important following time spent in hospitals and number of emergency events in the past year with the rest of top 15 features contributing equally.

To test the credibility of our model's prediction, we iterated across 15 different pseudo-random seeds, each time fitting our tuned model with a different random seed. After each fit we calculated the feature importance based on a decrease in impurity. The order of importance of features across different random seeds did not vary as seen by the low value of STD of feature importance (Figure 6B). Moreover, the order of feature importance based on impurity decrease was similar in many cases to the order of feature importance measured by SHAP, another testimony of the robustness of our model's predictive value. Lastly, Our model's prediction was also stable across different random seeds (Figure 6C), further emphasizing its robustness and stability.

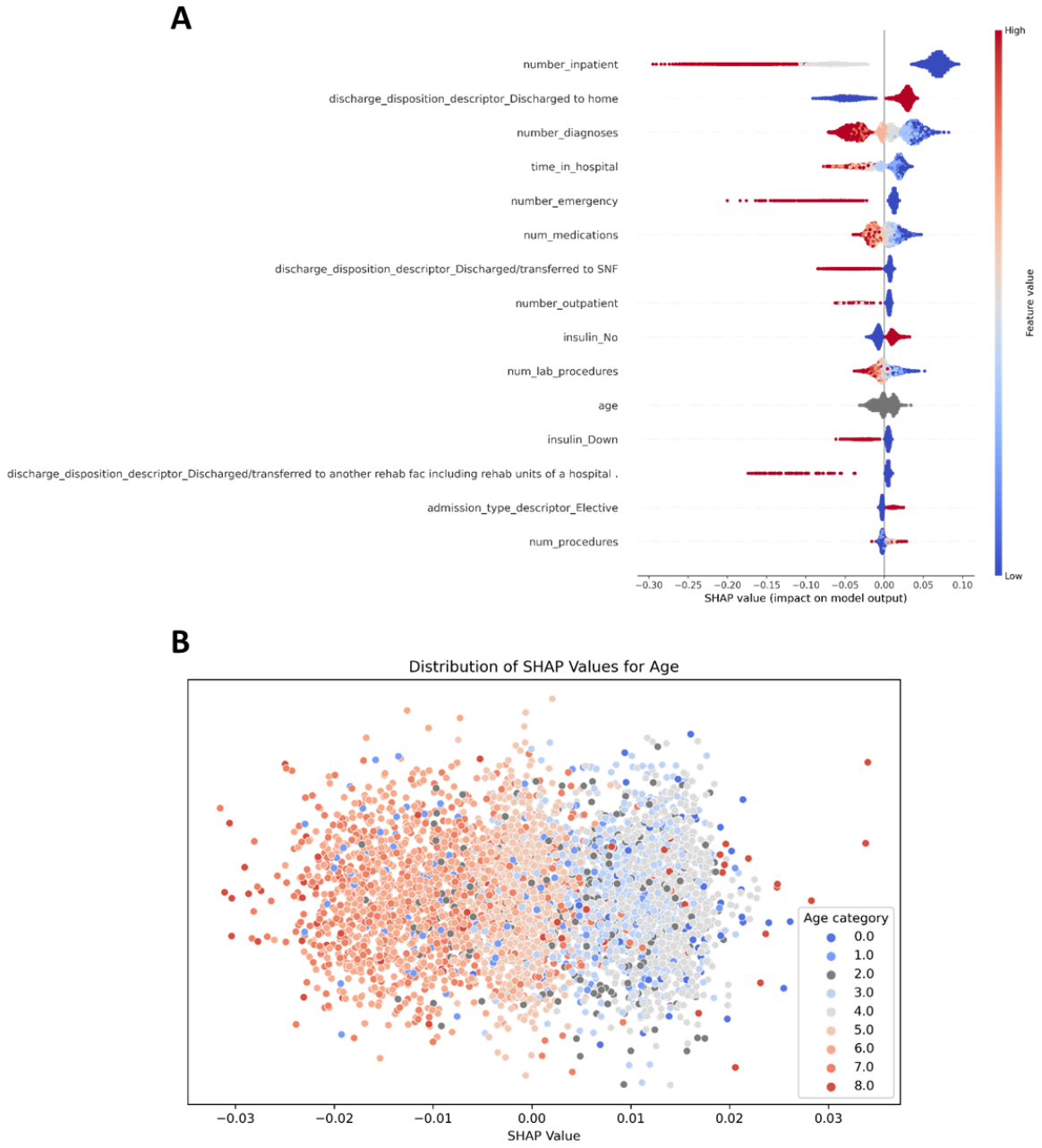


Figure 5 - BRF top 15 feature importance exhibit critical role for patient condition and readmission.
A) SHAP feature importance summary of the top 15 most important features. Features are ordered from most (top) to least (bottom) important. Feature value is color-coded from low values (blue) to high values (red). X-axis denotes SHAP value (feature contribution to the model's prediction). Negative values represent the feature's contribution to predicting readmission. B) SHAP values for age feature. The colors and X-axis are similar to that of Panel A.

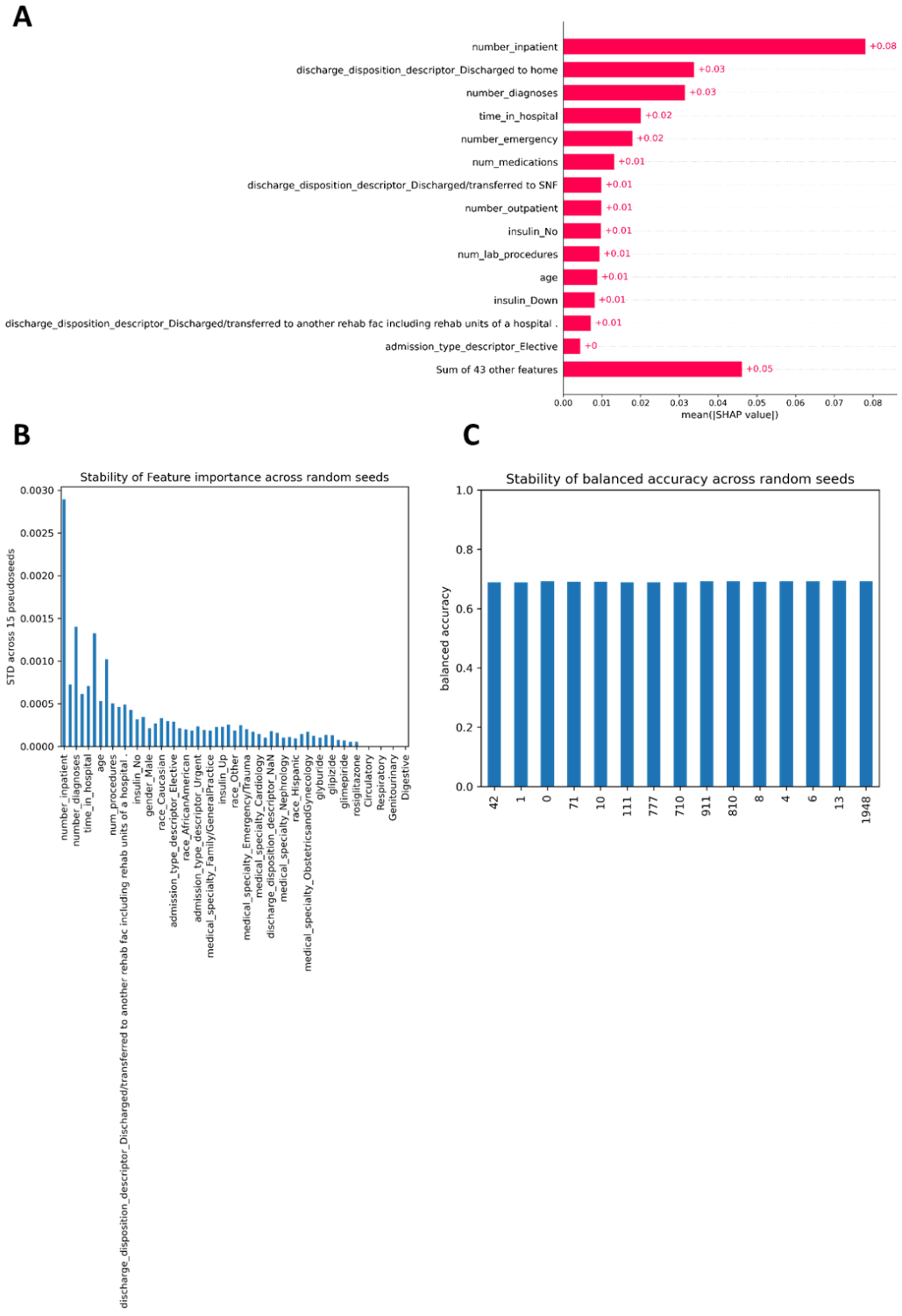


Figure 6 - BRF model prediction and feature importance are stable across pseudo-random seeds.
A) Bar plot of the absolute value of SHAP feature importance of top 15 most important features. Features are ordered from most (top) to least (bottom) important, with their effect mentioned right on the bar. X axis denotes the absolute SHAP value. B) barplot of STD of feature importance as calculated by impurity decreases across 15 random seeds. C) barplot of model's performance as measured by balanced accuracy across 15 random seeds.

Section 9 - Discussion:

Generally, throughout this project, we learned how to manage a ML project practically from end to end. Technically, we learned how to work with pipelines, write classes, synthesize data, train the leading models to date in the field, and get an intuition about the “dark art” of hyperparameter tuning. In addition, for some of us, working with GitHub, and with GPU were completely new skills.

Conclusion

We suggest tracking severe patients' phenotypes since they tend to readmit more (also a sanity check).

This can be seen with the following feature importance:

- number of inpatient visits
- Number of medications, number of emergencies, time in hospital,
- Number of lab procedures
- Time in hospital
- Elective readmission

- Susceptible age group for readmission (70-80) and resilient age group of (50-60). Older patients are just too old and are probably being readmitted due to other reasons. Younger ones are also spread uniformly.
- Moving to other wards is related to readmission - perhaps due to changes in the caregiving team and treatment, no decent tracing/tracking of treatment.
- Reduction of insulin treatment amount is correlated to readmission, perhaps there are other symptoms that are not captured by tests and reducing the Insulin treatment too early may increase the risk of being readmitted.

Assumptions and limitations

1. Our model excludes the age category 0-10, so it cannot be applied to this age range both because there were no readmittions <30 and because small sample size.
2. We have removed sparse features in order to reduce the curve of dimensionality. However, there might be features that were excluded that could increase the model performance
3. In this project we used balance accuracy score as our measure for performance (due to the reasons explained above). Optimizing other metrics with a different focus on the target could lead to other results.
4. We excluded the class of readmission of >30 to simplify the model from multi-class classification to a binary problem. This limits our observation since a new case that will be tested might be readmitted >30 days, but our model will not identify it.
5. We did not check feature importance or the contribution of our new features before training the models. Therefore, we eliminated irrelevant features that we thought might not contribute to the models but might actually be relevant.
6. Imputations - race: there were missing values in the race column, so we imputed them as other, under the assumption that the doctor filling the chart could not determine the specific race while filling

7. **From the model results and feature importance, it is possible that the model predicts readmission based on severe patients in general and not due to diabetes disease, so there might be a predicting confounder.** The solution to this limitation is to check the comorbidity of diseases, we could test each of the deceases (not only diabetes) and see how much it correlates with readmission.

What we would have changed for future model improvement? (if we would start from scratch)

1. Check feature importance after EDA and take out each time one feature to test its contribution to the model at the cross-validation level.
2. Running and fitting our model again with only top 14 features found to be important in the SHAP analysis.
3. We could also check other metric scores, but this would have changed our target question (for example, look at precision or f1 score).
4. We could also test the other hyperparameter optimization techniques, such as GridSearchCV or RandomSearchCV.
5. We didn't have the time to see the effect of different scaling techniques, such as MinMaxScalar or RobustScalar, which is less affected by outliers.
6. Our models should deal with missing values, so in principle, we could test it without imputing.

Section 10 - Team Members Individual Contribution:

All decision-making and brainstorming were done equally; we met in person and virtually throughout the project. During the project, we used VScode, PyCharm, JupterNotebook, and a GitHub account. We used a “requirement.txt” file to avoid environment dependencies (in GitHub). More specific contributions are specified below.

- Data split - Yahel
- EDA all students, mainly Shiri.
- Feature engineering in the pipeline - equal
    - IDS mapping -Shiri & Tal
    - Disease mapping - Yahel
    - Feature remover - Shiri
    - Imputation - Shiri
    - Numerical columns scaling and transformation - Yahel
    - One Hot encoding (special and regular) -Yahel
- Default models across all datasets - Yahel
- GAN - all students, Mainly Tal
- SMOTENC - Yahel
- Workflow visualization - Shiri
- Sample size and imbalance weights optimization - Yahel
- XGboost, RF, BRF - Yahel
- CatBoost - Tal
- LGBM - Shiri
- Feature importance and model stability - Yahel
- Best code practice and debugging - Tal & Yahel
    - And, of course, ChatGPT for debugging and OOP assistance.

Personal Comment:

We would like to thank you, Ortal, for the opportunity to experience an end-to-end, hands-on project at such a high level. We appreciate the time you dedicated to us as students and your willingness to assist at the expense of your personal after-work time. It is not obvious at all. We learned SO MUCH during the course, and we will definitely apply this knowledge throughout our scientific career, either in the academy or in the industry. Thanks a lot, Shiri, Tal, and Yahel.

Section 11 - References

1. American Diabetes Association Professional Practice Committee. 2. Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes-2022. *Diabetes Care* **45**, S17–S38 (2022).
2. Stumvoll, M., Goldstein, B. J. & van Haeften, T. W. Type 2 diabetes: principles of pathogenesis and therapy. *Lancet* **365**, 1333–1346 (2005).
3. Zheng, Y., Ley, S. H. & Hu, F. B. Global aetiology and epidemiology of type 2 diabetes mellitus and its complications. *Nat. Rev. Endocrinol.* **14**, 88–98 (2017).
4. CDC. All About Your A1C. *Centers for Disease Control and Prevention* https://bit.ly/2Nc2IA0 (2023).
5. IDF Diabetes Atlas. https://diabetesatlas.org/.
6. Shaw, J. E., Sicree, R. A. & Zimmet, P. Z. Global estimates of the prevalence of diabetes for 2010 and 2030. *Diabetes Res. Clin. Pract.* **87**, 4–14 (2010).
7. Reinehr, T. Type 2 diabetes mellitus in children and adolescents. *World J. Diabetes* **4**, 270–281 (2013).
8. D'Adamo, E. & Caprio, S. Type 2 diabetes in youth: epidemiology and pathophysiology. *Diabetes Care* **34 Suppl 2**, S161–5 (2011).
9. Golay, A. & Ybarra, J. Link between obesity and type 2 diabetes. *Best Pract. Res. Clin. Endocrinol. Metab.* **19**, 649–663 (2005).
10. Kyrou, I., Randeva, H. S., Tsigos, C., Kaltsas, G. & Weickert, M. O. Clinical Problems Caused by Obesity. in *Endotext [Internet]* (MDText.com, Inc., 2018).
11. Sattar, N. Gender aspects in type 2 diabetes mellitus and cardiometabolic risk. *Best Pract. Res. Clin. Endocrinol. Metab.* **27**, 501–507 (2013).
12. Sun, Y., Pei, W., Wu, Y. & Yang, Y. An association of herpes simplex virus type 1 infection with type 2 diabetes. *Diabetes Care* **28**, 435–436 (2005).

13. Wang, C.-S., Wang, S.-T., Yao, W.-J., Chang, T.-T. & Chou, P. Hepatitis C Virus Infection and the Development of Type 2 Diabetes in a Community-based Longitudinal Study. *Am. J. Epidemiol.* **166**, 196–203 (2007).
14. Willi, C., Bodenmann, P., Ghali, W. A., Faris, P. D. & Cornuz, J. Active Smoking and the Risk of Type 2 Diabetes: A Systematic Review and Meta-analysis. *JAMA* **298**, 2654–2664 (2007).
15. Baliunas, D. O. *et al.* Alcohol as a Risk Factor for Type 2 Diabetes: A systematic review and meta-analysis. *Diabetes Care* **32**, 2123–2132 (2009).
16. Pan, X.-R. *et al.* Effects of Diet and Exercise in Preventing NIDDM in People With Impaired Glucose Tolerance: The Da Qing IGT and Diabetes Study. *Diabetes Care* **20**, 537–544 (1997).
17. Ley, S. H., Hamdy, O., Mohan, V. & Hu, F. B. Prevention and management of type 2 diabetes: dietary components and nutritional strategies. *Lancet* **383**, 1999–2007 (2014).
18. Salas-Salvadó, J. *et al.* Prevention of diabetes with Mediterranean diets: a subgroup analysis of a randomized trial. *Ann. Intern. Med.* **160**, 1–10 (2014).
19. Grøntved, A., Rimm, E. B., Willett, W. C., Andersen, L. B. & Hu, F. B. A Prospective Study of Weight Training and Risk of Type 2 Diabetes Mellitus in Men. *Arch. Intern. Med.* **172**, 1306–1312 (2012).
20. Ekelund, U., Brage, S., Griffin, S. J. & Wareham, N. J. Objectively Measured Moderate- and Vigorous-Intensity Physical Activity but Not Sedentary Time Predicts Insulin Resistance in High-Risk Individuals. *Diabetes Care* **32**, 1081–1086 (2009).
21. Tahrani, A. A., Barnett, A. H. & Bailey, C. J. Pharmacology and therapeutic implications of current drugs for type 2 diabetes mellitus. *Nat. Rev. Endocrinol.* **12**, 566–592 (2016).
22. Fraze, T., Jiang, H. J. & Burgess, J. Hospital Stays for Patients with Diabetes, 2008. (2011).
23. Comino, E. J. *et al.* Impact of diabetes on hospital admission and length of stay among a general population aged 45 year or more: a record linkage study. *BMC Health Serv. Res.*

15, 12 (2015).

24. Reasons for admission of individual with diabetes to the Tripoli Medical Center in 2015. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews* **13**, 2571–2578 (2019).
25. Hospital admissions in diabetic and non-diabetic patients: A case-control study. *Diabetes Res. Clin. Pract.* **73**, 260–267 (2006).
26. van Dieren, S., Beulens, J. W. J., van der Schouw, Y. T., Grobbee, D. E. & Neal, B. The global burden of diabetes and its complications: an emerging pandemic. *Eur. J. Cardiovasc. Prev. Rehabil.* **17 Suppl 1**, S3–8 (2010).
27. Park, S. W. *et al.* Excessive Loss of Skeletal Muscle Mass in Older Adults With Type 2 Diabetes. *Diabetes Care* **32**, 1993–1997 (2009).
28. Graveling, A. J. & Frier, B. M. Hypoglycaemia: An overview. *Prim. Care Diabetes* **3**, 131–139 (2009).
29. Munshi, M. *et al.* Cognitive Dysfunction Is Associated With Poor Diabetes Control in Older Adults. *Diabetes Care* **29**, 1794–1799 (2006).
30. Cholerton, B., Baker, L. D., Montine, T. J. & Craft, S. Type 2 Diabetes, Cognition, and Dementia in Older Adults: Toward a Precision Health Approach. *Diabetes Spectr.* **29**, 210–219 (2016).
31. Website. https://www.researchgate.net/publication/331249588_Predicting_30-day_Hospital_Readmission_for_Diabetes_Patients_Using_Multilayer_Perceptron.
32. Graham, E., Saxena, A. & Kirby, H. Identifying High Risk Patients for Hospital Readmission. *SMU Data Science Review* **2**, 22 (2019).
33. View of Risk Assessment for Hospital Readmissions: Insights from Machine Learning Algorithms. https://journals.sagescience.org/index.php/ssraml/article/view/68/65.
34. Kang, H. Y. J. *et al.* Synthetic Tabular Data Based on Generative Adversarial Networks in Health Care: Generation and Validation Using the Divide-and-Conquer Strategy. *JMIR*

Medical Informatics **11**, e47859 (2023).

35. Lundberg, S. M. *et al.* From Local Explanations to Global Understanding with Explainable AI for Trees. *Nat Mach Intell* **2**, 56–67 (2020).