NOVA IMS Information Management School

EXAMINING HOSPITIAL RE-ADMISSIONS

Daniela Erazo - 20230501
Gonçalo Ferreira - 20230492
João Maia - 20230746
Inês Castelhano - 20230478
Pedro Malheiros -20230467

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA IMS Information Management School

Table of Contents

Abstract

A hospital's readmission rate can be both an effective indicator of care quality provided as a benchmark for healthcare systems, and a cornerstone when exploring ways to optimize services and resources.

Machine learning algorithms have vast applicability in diverse fields, and healthcare is no exception; the implementation of these tools, its further analysis and strategy assembling based on them help predicting, preventing, and treating a patient's disease, as well as lessen economic burden of the care unit admitting them.

During this project, we used different machine learning models training on a dataset that, because of the domain of the study, has an inherent class imbalance. We applied different techniques to tackle the imbalanced classification issue, compared model performance with and without some features, as well as tuning and optimizing the models that yielded the most.

Since a hospital readmission within a short timeframe implies that the recently discharged patient was most likely readmitted due to reasons directly correlated to their prior, original admission, we were able to uncover patterns that would allow us to classify a patient as to whether they would be readmitted during a timeframe or not built upon that fact.

The model evaluation metric used during our analysis was training and testing's F-1 scores, as it represents a weighted average of precision and recall, and we dealt with a certainly uneven class distribution.

I. Introduction

Diabetes is a chronic condition which serves as a precursor for other medical conditions and could trigger other health concerns in future, such as blindness, kidney failure, heart attacks, and more[1]. As a common comorbid condition in hospitalized patients, readmissions of diabetic individuals have been noted to contribute significantly to costs and resources' use surge in hospitals. Readmissions not only strain healthcare assets, but also adversely impact patients' well-being and quality of life.

In previous studies, it was noted that thirty-day readmission rates for hospitalized patients with diabetes mellitus (DM) are reported to be between 14.4 and 22.7%, much higher than the rate for all hospitalized patients (8.5–13.5%), and that patients with both a primary and secondary diagnosis of DM have higher readmission rates. Also, those with diabetes as their primary diagnosis have more readmissions related to it, while those with a secondary diagnosis of diabetes have more infection-related readmissions[2]. In another research, it was determined that other key factors that drove readmission are the number of times a patient was formerly admitted both as an inpatient and outpatient, mode of admission, and conditions like heart failure or hypertensive chronic kidney disease and so on[3]. Based on this, we expect variables found in our data, such as admission type, number and types of visits in previous year and variables related to main and additional diagnoses, as possible basis features of our model building.

Our main goal is to identify the factors that contribute to readmissions, which will lead us to accurately predict the probability of a patient being readmitted, and subsequently if said readmitted patient does so in a timeframe of less than 30 days, or not.

In our project we will use Machine Learning techniques to analyze a dataset that contains information about the admission of patients with diabetes and create a model that predicts whether patients with diabetes will be readmitted after being discharged, or not. To do this, we divide our project in two parts: In the first part, we will build a predictive model that predicts if a patient with diabetes will be readmitted to the hospital within 30 days after being discharged; in the second part of the project, we will develop a multiclassification model that will determine the period within which the patient will be readmitted or not, with 3 possible results, "No" for no readmission, "<30 days" for readmitted in less than 30 days and ">30 days" for readmitted after more than 30 days.

As a result, our model and research can serve as a guiding tool for healthcare providers to optimize the use of their resources by focusing on those who need them the most (like patients at risk of readmission) as well as reducing costs by readmission prevention, and finally establishing a long-term impact by making the most out of these models to continuously predict from new data.

II. Data Exploration and Preprocessing

2.1 Exploratory Analysis

The dataset provided consists of data related to the risk of hospital readmission of patients collected from hospitals in the USA, and it covers in detail information related to each event. Said records include 71236 occurrences of patient's encounters for the train data, 30530 for the test

data, with their respective identification number, and 29 other attributes associated with the patient's profile (demographics such as age, sex, and race), type of episode (emergency, outpatient, inpatient), laboratory analyses, medications, data on the patient's diagnoses, among others.

For ease of access, we decided to use the column of unique values, *encounter_id*, as index for both train and test data, and dropped the country column as a univariate feature (dataset contains information from USA's hospitals only).

Initially, we checked the disposition and characteristics of our variables, to detect possible inconsistencies or specific patterns. We were also able to verify that we have 17 251 repeated values in the train dataset and 4128 in the test dataset. This makes sense, as a patient can have multiple encounters. With the function .info() we look in detail the data types of our datasets, dug deeper into the unique values for categorical variables, and came across some peculiar values for features, such as questions marks, literal remarks of "not available", "unknown" or "not mapped" data, etc. We decided to replace such with NaNs instead, and finally verified with .isna().sum() the presence of a very significant amount of missing values in several of the categorical variables in both datasets (refer to tables 1 to 4 in Annexes).

2.2 Metric and Non-Metric Features

We systematically categorized our variables between metric and non-metric features to assess our data in a more efficient way and to simplify following steps, as some techniques used next will depend on whether the variable is numerical or categorical. We also discovered that our output variables are imbalanced, and for this we will later use some techniques to address this problem.

2.3 Outliers

The Interquartile Range (IQR) method was the first option to remove atypical data, but such approach would remove about 34% of the data, most probably resulting in the loss of valuable and relevant information that could be useful to build our prediction model.

We inspected visually for outliers with boxplots (refer to figure 1 in Annexes), analyzing each metric feature separately to decide on limits to perform manual filters. In the case of previous year's outpatient, emergency, and inpatient visits, it was noticeable that the three variables had a somewhat similar behavior, with most of the sample had values of 0 (or just 1) and decided to set a threshold of 20 episodes that would not remove a great number of observations for each one of the three variables. Likewise, we fixed thresholds of 12 days maximum for *length_of_stay_in_hospital*, 110 tests for *number_lab_tests*, 70 different items for *number_of_medications*, and that the *number_diagnoses* kept values between 2 and 13 different pathologies.

When combining both outlier removal methods, we were able to keep almost 97% of our original training data, hence we decided to select this technique to handle outliers.

2.4 Missing Values

Since some of the patients (identified by their unique ID) were noticeably recurrent occurrences, we traced and reassigned the *age* and *race* that were recorded in a different occasion to the same individual, for the ones possible.

The three observations with missing values in *gender* were simply dropped out of the dataframe, as neither of them were readmitted before or after the 30-days' timeframe and therefore can be considered of minimal statistical importance.

Weight had more than 90% of observations amounting to NaNs, it would leave us with barely 10% (even less) of relevant information in both training and testing data; the reason behind it may be the data being collected before the 2009's implementation of the HITECH (Health Information Technology for Economic and Clinical Health Act) legislation of the American Reinvestment and Recovery Act, and so hospital and clinics back then were not required to keep records of this attribute in a structured format[4]. Consequently, we decided this is another variable that will be dropped.

For *payer_code* we put the missing values in a new category "NHI", which means "No Health Insurance".

Lastly, for *glucose_test_result* and *a1c_test_result*, we replaced NaNs with "Not tested", as both contribute on the study of the very critical topic of blood sugar monitoring: the first one is important because it helps patients to meet their glucose targets and avoid long-term diabetes complications[5], and further scrutiny of the A1C test (which is a temporary average of blood sugar levels) helps providers to adjust treatment strategies more accurately[6]. These may prove to be determinative in our model.

After these adjustments, the remaining missing values of categorical variables *race*, *age*, and *discharge_disposition, admission_type, admission_source* and the three diagnoses variables would be reallocated to their respective mode, we can only treat them after the split, to avoid data leakage; thus, they will be dealt with on hereafter.

2.5 Split

We assigned *readmitted_binary* and *readmitted_multiclass* into *target_binary* and *target_multiclass*, respectively. After this, we dropped the *readmitted_binary* and *readmitted_multiclass* in the train dataset. We will be using one notebook for predicting the *readmitted_binary* and another for the *readmitted_multiclass.*

For the split, we used the "train_test_split()", and randomly split the data, in the random state of 0. The test size was 20%. Since we have the test dataset, we only created train and validation variables for X and y, using *target_binary* first as the target variable, since we will be working first on the predictions for the binary.

As previously mentioned, after the split we were able to reassign the remaining missing values with the mode of their respective categories. From this moment onwards, no missing values existed in our training, validating, nor in our test datasets.

2.6 Treat Inconsistencies

One inconsistency that we found was related to *discharge_disposition,* as some people here had the state of "Expired". We kept this data, as it would teach our model that the referred person would not be readmitted again.

Here we also addressed a situation that was the fact of having in *payer_code* only one observation with "FR". In order to prevent bringing unnecessary noise into our model, we decided to drop it, as it is only referred to once. We removed "FR" in X_train,X_val, test dataset and y_train.

In *admission_type,* we verified that we had some imbalanced categories. The 3 most counted ones did not fall below the count of 10 000, while the 2 least counted categories "trauma center" and "newborn" had only 10 and 5, respectively, in X_train . Moreover, in "Newborn" we had 4 people whose age did not correspond to the age interval of 0-10, hence they could not be newborns. "Newborn" and "Trauma center" observations were reassigned to a new category called "Other", because there were some incongruencies and these had low frequencies. These modifications were applied also to the Validation and test datasets.

2.7 Feature Engineering

Variables *gender*, *change_in_meds* and *prescribed_diabetes_meds* were transformed into binary columns for train, validation and test datasets. We also transformed into binary our *y_train and y_val.* These variables were transformed into:"No" to 0 and "Yes" to 1. In our multiclass notebook, the transformation was "No" into 0, "<30 days" into 1 and ">30 days" into 2.

To reduce the number of categories within each categorical variable, we developed functions that would reallocate a new summarized category for both train and test datasets:

In the case of *discharge_disposition* we had a lot of categories with small differences between them, such as "Discharged to home" and "Discharged/transferred to home with home health service", both refer to the circumstance of going home, among many others; therefore, we decided to reduce the number of categories into 5 main groups: "Home_Discharge", "Facility_Discharge", "Hospice", "Other_Facilities", and "Expired" (or deceased)[7], and accordingly, we modified train, validation and test datasets.

For *admission_source* we reordered in the following categories: "Emergency_Services", "Referrals", "Transfers_Hospital", "Transfers_Health_Care", and "Other", to reduce input space.

We created variables called *service_utilization_in_previous_year* (sum of outpatient, emergency, and inpatient visits) and *number_of_visits* (count of the number of encounters for each patient), and both were added to train, validation, and test datasets.

In the variable *medical_specialty*, since we had 68 different value counts, we decided to group them into broader groups. The NaNs remain in the most counted category, "Not Available".

When dealing with variable *age*, we decided to outline new intervals for classification, grouping all patients below 30 years old, the ones between 30 and 50 years, as well as grouping all patients over 80 years old. This arrangement stems from our data's distribution: the fact that age intervals 50-60, 60-70 and 70-80 have the most frequency in occurrences, in contrast to the patients reassigned to their new, bigger range. Thus, instead of having ten different classes, we have almost half the number.

For the variable *medication*, we had for each patient a list containing the names of the medications that they had taken. This was hard to address, as there were immensely different combinations of medications. We created a function that counted the number of times each medication was repeated in the variable, regardless of its combination with other medications. According to the results, insulin was the most repeated value, almost triple as much as the second most

repeated category, which was metformin, for the train dataset. Given this information, we decided to create 2 new variables which were *takes_insulin* ,which is a binary variable with "1" if the person takes insulin and "0" if not, and *medication_category*, stating, according to the patient's medication value, the following possibilities: "takes insulin", "takes other medication", "takes no medication" and "takes insulin and other medicine". After this, we dropped the original variable, since we believe the new category has more relevance.

Regarding all the diagnoses available in three of the dataset's variables, we generated diagnosis groups (table 5 in Annexes) which would encapsule in a more general, summarized manner the conditions of each patient presented in the data. We also created the variable *has_diabetes* which accounted consistently whenever variations of code 250 (which stands for diabetes) were depicted.

2.8 Encoding

We needed to perform encoding to apply models in our data set. We tried with One Hot Encoding; however, since it creates a variable for each category, and there were variables with several subsequent categories, we concluded it was not the best possible approach. As such, we selected Target Encoding to deal with categorical variables; in contrast to OHC, this encoding method can handle high cardinality features without adding to the dimensionality of the dataset.

2.9 Scaling

We implemented the StandardScaler method for both Train and Test dataset. Because of the positive skewness in the distribution of most of the numerical variables (refer to figure 2 in Annexes), we decided to perform a square-root transformation in order to fit the data into a normal, Gaussian distribution which would allow the possibility of trying Machine Learning models that are skewness-sensitive.

III. Binary Classification

3.1 Feature Selection

When developing a predictive model fewer attributes are desirable since it helps reduce the complexity and may even improve its performance. During this phase we will detect irrelevant and redundant features from our data that may not contribute to the final analysis of the patients' attributes assessment and determine the final set of features to be used in our predictive model.

For feature selection we will be using Spearman's correlation, Chi-square test, Recursive Feature Elimination, LASSO Regression, RIDGE Regression, ANOVA and Mutual Information.

First, we applied Spearman's correlation, which only works for numerical features. Because most of our data is not normally distributed, this correlation coefficient is adequate as it evaluates monotonic and not necessarily linear relationships. The only high correlation observed was with *service_utilization* and *inpatient_visits,* with 0.8.

The next model used was the chi-square test. This test is only intended for categorical variables. For an alpha level of 0.05, we were only able to discard *race* and *gender* from the model. We

made attempts with lower alpha levels, as we needed to reduce input space among categorical variables. After several attempts, we settled for an alpha level of 0.0007, which seemed the most reasonable level. Even lower alpha levels resulted in the discard of a lot of information. We were able with this new level to discard *admission_type, admission_source* and *glucose_test_result* as well.

After this, we used Recursive Feature Elimination (RFE) for reducing input space among numerical features. We started by using an np.arange between 1 and 12, to tell us the optimum number of variables, and which score we would obtain with it. We got an optimum number of 4 features, which were *length_of_stay_in_hospital, service_utilization_in_previous_year, number_diagnoses* and n*umber_of_visits.*

LASSO Regression was our next attempt. According to LASSO, we should discard *race, age, payer_code, admission_type ,medical_specialty,admission_source, additional_diagnosis, glucose_test_result, a1c_test_result, has_diabetes* and *medication_category,* as these presented coefficients of 0.

Next, we tried RIDGE Regression. According to Ridge, we should drop a variable if it displays a coefficient of 0. However, all our variables had coefficients different from 0.

Even though it was not used in classes, we wanted to support our decision of dropping variables as much as possible, and minding this we used ANOVA. ANOVA is a type of F-Statistic that can be used when we have numerical inputs and categorical outputs, like in this case. ANOVA will tell us which features are independent from the output variable, and therefore, should be removed from the dataset. According to ANOVA, the optimum number of features is 10, and it only told us to drop *average_pulse_bpm.*

Lastly, we used Mutual Information. Mutual Information evaluates the amount of information shared among variables, meaning it works on the entropy of variables. We want to retain the variables that explain the most about the target variable and eliminate the ones that are less informative. As such, using a k=5, the variables that we should keep are *emergency_visits_in_previous_year,inpatient_visits_in_previous_year, service_utilization_in_previous_year, number_diagnoses* and *number_of_visits.*

Regarding numerical features, we decided only to drop *outpatient_visits_in_previous_year* and *average_pulse_bpm.* Even though a lot of models told us to keep *outpatient_visits_in_previous_year,* we decided to discard this one since it refers only to visits in the year prior to the considered encounter. However, according to literature[8], outpatient visits are relevant in the follow up of an inpatient situation. As such, we will be dropping this variable. *Average_pulse_bpm* was the feature with the most discards, so we will not be using it.

In categorical variables, we followed the model's judgement, except for *age, additional_diagnosis* and *has_diabetes,* because we believe these contain relevant information for the target variable. In tables 6 and 7 we can see what our decision regarding the existing features is.

3.2 Techniques for Imbalanced Classification

In machine learning, dealing with imbalanced datasets is crucial for building fair and accurate models. We chose to use two different techniques for our dataset: Synthetic Minority Oversampling Technique (SMOTE) and the Weighted Values Method as our dataset is imbalanced.

SMOTE combats class imbalances by generating synthetic samples for the minority class, ensuring a more equitable representation. On the other hand, the Weighted Values Method adjusts parameters like Class Weight in algorithms such as Logistic Regression, Random Forest, and Support Vector Machines. This modification allows our models to assign greater importance to underrepresented classes —in this case, the patients who were readmitted to the hospital— promoting a balanced learning process and enhancing overall predictive performance.

As such, we will be using both approaches to run the models, and later compare results.

3.3 Model Building and Assessment

During this stage, we created different models in order to predict our target variable. By selecting a list of miscellaneous supervised machine learning algorithms, we were able to introduce different perspectives into our analysis and evaluate their corresponding performance with their default values. Initially, these models were applied: Logistic Regression, Gaussian Naïve Bayes, Artificial Neural Networks, KNN Classifier, Random Forest, Gradient Boosting, Support Vector Machines, Decision Trees, Bagging Classifier, and AdaBoost. As previously stated, for each model, we ran weighted values and SMOTE to address the imbalance.

The results for all the models can be found in Annexes, in table 10, and figures 3 and 4. These contain the F1 scores of the models we ran, both in train and validation datasets, in a table and in graph.

It is important to note that Neural Networks, KNN Classifier, and Gaussian Naive Bayes don't have a "class_weights" parameter, so we cannot use weighted values in those 2 models. As such, we will be using purely default parameters in those 2.

One of the first conclusions that we can take from the table is that on average, SMOTE returns very high train F1 Scores. However, when we look at validation scores, these are a lot lower, meaning that the results with SMOTE are overfitting very much. There is no case where a model with SMOTE doesn't present a high degree of overfitting.

On the other hand, the F1 Scores with weighted values are always lower in the train than the ones verified with SMOTE. However, in contrast to SMOTE, the results with weighted values present lower overfitting, as we have lower differences from the results in train to validation. We can only verify significant overfitting in Random Forest, K-Nearest Neighbors, Decision Trees, Bagging Classifiers and Support Vector Machine. Nevertheless, SMOTE displays the best validation scores in 6 out of the 10 models chosen.

We decided to move on to the next round with: Logistic Regression, Random Forest, Neural Networks, and Gradient Boosting. Even though SVM displayed a good performance, we did not attempt to optimize it, as it is very computationally expensive, and after 20 hours of random search, it was still running. Our selection was not based purely on metrics, we also wanted to focus on models with very specific behaviors. For instance: Decision Trees, Bagging Classifiers and Random Forest present similar theoretical rationales. We attempted to pick the ones that we thought were the most different ones.

In order to optimize the model as best as possible, we used Random Search, since Random Search randomly samples possible combinations instead of running all possible combinations,

like Grid Search. We chose Random Search as it is less computationally expensive, and in the 3rd and Final Round we will use Grid Search on the best models.

In this 2nd round, we ran again the models with weighted values and SMOTE. The F1 scores can be seen in table 11, and figures 5 and 6.

In this case, the results were again higher with SMOTE in training, but once again, presented a higher degree of overfitting, whereas the results with weighted values presented lower overfitting. The F1 Scores recorded higher increases in weighted values, meaning Random Search favored best weighted values.

For the final round, we will be using Grid Search on Logistic Regression, Random Forest, Gradient Boosting and a Stacking Classifier with the best 2 models, Logistic Regression and Random Forest, but only with weighted values.

To our surprise, Stacking showed a poor performance, as combining both models did not return good predictions. It is also important to notice that Gradient Boosting provided the best results with default values. The following attempts of optimization led only to lower scores.

In addition, we used in this round K-Fold, Repeated K-Fold and Stratified K-Fold, as these provide more reliable model performances. These increased the scores.

The best performing model here was Random Forest, which allowed us to obtain an F1 Score of 0.3614 on Kaggle and reach the 8th place, with Grid Search's parameters.

IV. Multiclass Classification

4.1 Feature Selection

For the multiclass requirement, we decided to maintain the same preprocessing, since we believe that the way we dealt with the data was the most proper one, regardless of the target variable. We used a different notebook, intended for predicting the *readmitted_multiclass.* Apart from this, the main differences came to surface during the feature selection for our models since the target variable we are using now is different. You can find a summary of the evaluation methods performed on each one of the variables in tables 8 and 9.

Since the variables *service_utilization_in_previous_year* and *inpatient_visits_in_previous_year* bring in the same information to the model (as demonstrated with their high correlation), we chose to discard the former.

RFE methods suggested to remove a meaningful portion of our metric variables, but it was our final decision to discard only *outpatient_visits_in_previous_year*, *average_pulse_bpm* and *length_of_stay_in_hospital*, as we considered these to contribute with much less information than their counterparts. *Length_of_stay_in_hospital,* is a variable that presents high variability, and for the same problem a person can stay in the hospital 2 or 50 days, based on sources[9]. Since it is very hard to explain, we decided to drop it. *Outpatient visits* refer only to visits in the year prior to the considered encounter; however, outpatient visits are relevant in the follow up of an inpatient situation. As such, we will be dropping this variable.

We tried to follow as much as possible the decisions provided by the models; however, the results were not as straightforward as expected. For instance, the Chi-squared deemed all of the categorical variables as significant, even with alpha levels lower than 0,05. On the other hand, Lasso assessed several variables as irrelevant, nevertheless in the end *race* was the only one we discarded because, even though it is a controversial topic, there is no scientific proof that links this biological factor to the prevalence of certain diseases[10].

4.2 Techniques for Imbalanced Classification

Similar to what was done in the *readmitted_binary*, we used weighted values and SMOTE.

4.3 Model Building and Assessment

First, we started by running all models with default values, for SMOTE and weighted values. It is important to note that with multiclass problems, the F1 Score function requires an additional parameter, called "average", in order to return an output. We went with "weighted", since it calculates the average weight of the classes, being the one that addresses imbalance, like we have in our dataset.

Just like our binary analysis, SMOTE presents higher overfitting. However, in this case the results for validation with weighted values were the best ones for 8 out of 10 models (this information is available in table 12, and figures 7 and 8). Overall, the F1 Scores are higher, and the overfitting is less significant, even though it still persists. This is probably due to the fact that the *readmitted_multiclass* is slightly less imbalanced than the previous target variable.

Correspondingly, we decided to run the same models: Logistic Regression, Random Forest, Neural Networks and Gradient Boosting, with Random Search, and later do a final round with Grid Search for the best ones.

For this 2nd round, we can check the results in table 13, and figures 9 and 10. Some models recorded lower performance after random searches, but most cases recorded improvements.

For the final round, we used Logistic Regression, Random Forest, Neural Networks and a Stacking Classifier, combining the best 2 models for multiclass, Neural Networks and Random Forest. Once again, we applied K-Fold, Repeated K-Fold and Stratified K-Fold, for more reliable results. The best result was obtained with Random Forest. However, it exhibits a significantly high level of overfitting. Therefore, we can consider the Neural Networks model as having the best result without overfitting.

V. Conclusion

In this project, we aimed to create the best predictive models for *readmitted_binary* and *readmitted_multiclass,* using all types of Machine Learning's techniques. This was a challenging task, as the dataset itself presented several difficulties, such as high percentage of outliers, high input space, variables with high percentage of missing values, some incongruencies, imbalanced target variables and variables with a lot of different categories.

Along the process, we tried to reduce the number of categories into more meaningful ones, to avoid bringing noise into the model. Also, in the cases we saw fit, we created new variables. We

used 2 notebooks, one for each target variable, both identical up until the moment of the split, as the target variable needed to be different.

We found ourselves with different struggles while running feature selection models, as the results provided from the models were sometimes hard to interpret, recommending dropping a significant number of variables or keeping most of the variables. We tried to follow the models' decision to avoid bias when selecting features and, in some cases, support our decision with different scientific and medical sources.

To address the imbalance problem, we used SMOTE and Weighted Values, and compared the models' results along the report, for both target variables. Firstly, we ran a round of models with default values, and recorded the results for all models with SMOTE and Weighted Values, for *readmitted_binary* and *readmitted_multiclass*. The F1 Score was our main measurement for assessment as it balances the precision and recall metrics and since the observations are unevenly distributed (large number of actual negatives, in this case patients that were not readmitted).

In the second round of models, we used fewer models, but optimized ones, with parameters obtained from random search.

For the last round, we used Grid Search on the best performing models in Random Search. Regarding the *readmitted_binary*, the best performing model was Random Forest, with weighted values, which provided an F1 Score of 0.3614 and 8th place on the competition in Kaggle. Nevertheless, it showed overfitting, which may suggest that the best model is Neural Networks, as it exhibits a good F1 score and a low level of overfit. In reference to the *readmitted_multiclass,* Random Forest was also the best performing model.

One of our main findings is that using more sophisticated or optimized models provides slightly higher F1 Scores. Nevertheless, throughout our project, we could verify that the preprocessing steps, like designing new features or reducing high cardinality categorical variables into more generalized factions, had the most effective impact in models' performances; for instance, *number_of_visits* played a vital role after incorporating it in our models. Additionally, this particular variable corroborates that our initial premise of a patient's previous visits being a decisive aspect when trying to predict readmission, remains valid.

Our work suggests that sequential information from patients, the succession of recent incidents during and before admission, is a valuable asset for healthcare providers to gauge the risk of their readmission. Consolidating records in a proficient and comprehensive manner are undoubtedly areas to focus on in order to design ways of ensuring patients' well-being, lessen workload and pressure on medical staff, and reallocate and reduce costs.

We consider that one of the main limitations was that the dataset itself, as some critical features, that could prove to be significant, were documented in an ambiguous way; it is the case of variables such as age, weight, laboratory tests and procedures without actual values or specificity, and others. Another obstacle was the computation time that some models took to run when we wanted to make decisions or make comparisons about their performance.

VI. Annexes

Tables

```
<class 'pandas.core.frame.DataFrame'>
Index: 71236 entries, 533253 to 459757
Data columns (total 30 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   country                               71236 non-null  object
 1   patient_id                            71236 non-null  int64
 2   race                                  67682 non-null  object
 3   gender                                71236 non-null  object
 4   age                                   67679 non-null  object
 5   weight                                71236 non-null  object
 6   payer_code                            71236 non-null  object
 7   outpatient_visits_in_previous_year    71236 non-null  int64
 8   emergency_visits_in_previous_year     71236 non-null  int64
 9   inpatient_visits_in_previous_year     71236 non-null  int64
 10  admission_type                        67530 non-null  object
 11  medical_specialty                     71236 non-null  object
 12  average_pulse_bpm                     71236 non-null  int64
 13  discharge_disposition                 68646 non-null  object
 14  admission_source                      66518 non-null  object
 15  length_of_stay_in_hospital            71236 non-null  int64
 16  number_lab_tests                      71236 non-null  int64
 17  non_lab_procedures                    71236 non-null  int64
 18  number_of_medications                 71236 non-null  int64
 19  primary_diagnosis                     71236 non-null  object
 20  secondary_diagnosis                   71236 non-null  object
 21  additional_diagnosis                  71236 non-null  object
 22  number_diagnoses                      71236 non-null  int64
 23  glucose_test_result                   3688 non-null   object
 24  a1c_test_result                       11916 non-null  object
 25  change_in_meds_during_hospitalization 71236 non-null  object
 26  prescribed_diabetes_meds              71236 non-null  object
 27  medication                            71236 non-null  object
 28  readmitted_binary                     71236 non-null  object
 29  readmitted_multiclass                 71236 non-null  object
dtypes: int64(10), object(20)
memory usage: 16.8+ MB
```

Table 1 - Variables data type (Train)

```
patient_id                                  0
race                                     5070
gender                                      3
age                                      3557
weight                                  68990
payer_code                              28201
outpatient_visits_in_previous_year          0
emergency_visits_in_previous_year           0
inpatient_visits_in_previous_year           0
admission_type                           7240
medical_specialty                       34922
average_pulse_bpm                           0
discharge_disposition                    3269
admission_source                         4825
length_of_stay_in_hospital                  0
number_lab_tests                            0
non_lab_procedures                          0
number_of_medications                       0
primary_diagnosis                           0
secondary_diagnosis                         0
additional_diagnosis                        0
number_diagnoses                            0
glucose_test_result                     67548
a1c_test_result                         59320
change_in_meds_during_hospitalization       0
prescribed_diabetes_meds                    0
medication                                  0
readmitted_binary                           0
readmitted_multiclass                       0
dtype: int64
```

Table 2 - Missing values (Train)

```
<class 'pandas.core.frame.DataFrame'>
Index: 30530 entries, 499502 to 914270
Data columns (total 28 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   country                               30530 non-null  object
 1   patient_id                            30530 non-null  int64
 2   race                                  28996 non-null  object
 3   gender                                30530 non-null  object
 4   age                                   28999 non-null  object
 5   weight                                30530 non-null  object
 6   payer_code                            30530 non-null  object
 7   outpatient_visits_in_previous_year    30530 non-null  int64
 8   emergency_visits_in_previous_year     30530 non-null  int64
 9   inpatient_visits_in_previous_year     30530 non-null  int64
 10  admission_type                        28945 non-null  object
 11  medical_specialty                     30530 non-null  object
 12  average_pulse_bpm                     30530 non-null  int64
 13  discharge_disposition                 29429 non-null  object
 14  admission_source                      28467 non-null  object
 15  length_of_stay_in_hospital            30530 non-null  int64
 16  number_lab_tests                      30530 non-null  int64
 17  non_lab_procedures                    30530 non-null  int64
 18  number_of_medications                 30530 non-null  int64
 19  primary_diagnosis                     30530 non-null  object
 20  secondary_diagnosis                   30530 non-null  object
 21  additional_diagnosis                  30530 non-null  object
 22  number_diagnoses                      30530 non-null  int64
 23  glucose_test_result                   1658 non-null   object
 24  a1c_test_result                       5102 non-null   object
 25  change_in_meds_during_hospitalization 30530 non-null  object
 26  prescribed_diabetes_meds              30530 non-null  object
 27  medication                            30530 non-null  object
dtypes: int64(10), object(18)
memory usage: 6.8+ MB
```

Table 3 - Variables data type (Test)

```
patient_id                                  0
race                                     2191
gender                                      0
age                                      1531
weight                                  29579
payer_code                              12055
outpatient_visits_in_previous_year          0
emergency_visits_in_previous_year           0
inpatient_visits_in_previous_year           0
admission_type                           3156
medical_specialty                       15027
average_pulse_bpm                           0
discharge_disposition                    1411
admission_source                         2117
length_of_stay_in_hospital                  0
number_lab_tests                            0
non_lab_procedures                          0
number_of_medications                       0
primary_diagnosis                           0
secondary_diagnosis                         0
additional_diagnosis                        0
number_diagnoses                            0
glucose_test_result                     28872
a1c_test_result                         25428
change_in_meds_during_hospitalization       0
prescribed_diabetes_meds                    0
medication                                  0
dtype: int64
```

Table 4 - Missing values (Test)

Scope of the disease/condition	ICD-9 Codes
Diabetes	250.xx
Infectious and parasitic diseases	001-139
Neoplasms	140-239
Endocrine (nutritional, metabolic, immunity disorders)	240-279 (except 250)
Blood and blood forming organs	280-289
Mental disorders	290-319
Nervous system and sense organs	320-389
Circulatory	390-459
Respiratory	460-519
Digestive	520-579
Genitourinary	580-629
Pregnancy, childbirth, and postpartum complications	630-679
Skin and subcutaneous tissue	680-709
Musculoskeletal	710-739
Congenital anomalies	740-759
Other symptoms, signs, or ill-defined conditions	780-799
Injury and poisoning	800-999
External causes of injury	E
Supplemental classification	V

Table 5 - Values for diagnosis in our final datasets

Predictor	Spearman	RFE	ANOVA	Mutual Info	Lasso	Ridge	Final Decision
outpatient_visits_in_previous_year	Keep	Discard	Keep	Discard	Keep	Keep	Discard
emergency_visits_in_previous_year	Keep	Discard	Keep	Keep	Keep	Keep	Keep
inpatient_visits_in_previous_year	Keep	Discard	Keep	Keep	Keep	Keep	Keep
average_pulse_bpm	Keep	Discard	Discard	Discard	Keep	Keep	Discard
length_of_stay_in_hospital	Keep	Keep	Keep	Discard	Keep	Keep	Keep
number_lab_tests	Keep	Discard	Keep	Discard	Keep	Keep	Keep
non_lab_procedures	Keep	Discard	Keep	Discard	Keep	Keep	Keep
number_of_medications	Keep	Discard	Keep	Discard	Keep	Keep	Keep
number_diagnoses	Keep	Keep	Keep	Keep	Keep	Keep	Keep
service_utilization_in_previous_year	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
number_of_visits	Keep	Keep	Keep	Keep	Keep	Keep	Keep

Table 6 – Summary of feature selection methods´ results for numerical variables (Binary)

Predictor	Chi-Square	Lasso	Final Decision
race	Discard	Discard	Discard
gender	Discard	Keep	Discard
age	Keep	Discard	Keep
payer_code	Keep	Discard	Discard
admission_type	Discard	Discard	Discard
discharge_disposition	Keep	Keep	Keep
admission_source	Discard	Discard	Discard
primary_diagnosis	Keep	Keep	Keep
secondary_diagnosis	Keep	Keep	Keep
additional_diagnosis	Keep	Discard	Keep
change_in_meds_during_hospitalization	Keep	Keep	Keep
prescribed_diabetes_meds	Keep	Keep	Keep
medication_category	Keep	Discard	Discard
has_diabetes	Keep	Discard	Keep
glucose_test_result	Discard	Discard	Discard
a1c_test_result	Keep	Discard	Discard
medical_specialty	Keep	Discard	Discard

Table 7 – Summary of feature selection methods´ results for categorical variables (Binary)

Predictor	Spearman	RFE	ANOVA	Mutual Info	Lasso	Ridge	Final Decision
outpatient_visits_in_previous_year	Keep	Discard	Keep	Discard	Keep	Keep	Discard
emergency_visits_in_previous_year	Keep	Discard	Keep	Keep	Keep	Keep	Keep
inpatient_visits_in_previous_year	Keep	Keep	Keep	Keep	Keep	Keep	Keep
average_pulse_bpm	Keep	Discard	Discard	Discard	Keep	Keep	Discard
length_of_stay_in_hospital	Keep	Discard	Keep	Discard	Keep	Keep	Discard
number_lab_tests	Keep	Discard	Keep	Discard	Keep	Keep	Keep
non_lab_procedures	Keep	Discard	Keep	Discard	Keep	Keep	Keep
number_of_medications	Keep	Discard	Keep	Discard	Discard	Keep	Keep
number_diagnoses	Keep	Discard	Keep	Keep	Keep	Keep	Keep
service_utilization_in_previous_year	Discard	Keep	Keep	Keep	Keep	Keep	Discard
number_of_visits	Keep	Keep	Keep	Keep	Keep	Keep	Keep

Table 8 – Summary of feature selection methods´ results for numerical variables (Multiclass)

Predictor	Chi-Square	Lasso	Final Decision
race	Keep	Discard	Discard
gender	Keep	Keep	Keep
age	Keep	Keep	Keep
payer_code	Keep	Keep	Keep
admission_type	Keep	Discard	Keep
discharge_disposition	Keep	Keep	Keep
admission_source	Keep	Keep	Keep
primary_diagnosis	Keep	Keep	Keep
secondary_diagnosis	Keep	Keep	Keep
additional_diagnosis	Keep	Keep	Keep
change_in_meds_during_hospitalization	Keep	Keep	Keep
prescribed_diabetes_meds	Keep	Keep	Keep
medication_category	Keep	Discard	Keep
has_diabetes	Keep	Keep	Keep
glucose_test_result	Keep	Discard	Keep
a1c_test_result	Keep	Discard	Keep
medical_specialty	Keep	Keep	Keep

Table 9 – Summary of feature selection methods´ results for categorical variables (Multiclass)

	Train: F1-Score SMOTE	Validation: F1-Score SMOTE	Train: F1-Score Weighted Values	Validation: F1-Score Weighted Values
Logistic Regression	0.72230	0.28957	0.31999	0.31453
Gaussian Naïve Bayes	0.69205	0.26171	0.24573	0.24980
Random Forest	0.99999	0.19655	0.99976	0.02662
Neural Networks	0.86483	0.24000	0.10903	0.07038
K-Nearest Neighbor	0.88265	0.26842	0.26728	0.11185
Support Vector Machine	0.80714	0.29763	0.32563	0.32292
Decision Trees	1.00000	0.19637	1.00000	0.19411
Bagging Classifier	0.99155	0.19681	0.91794	0.06156
AdaBoost Classifier	0.89899	0.20036	0.32071	0.32296
Gradient Boosting	0.92601	0.20068	0.33249	0.33045

Table 10 – Performance summary of models with default parameters (Binary)

	Train: F1-Score SMOTE	Validation: F1-Score SMOTE	Train: F1-Score Weighted Values	Validation: F1-Score Weighted Values
Logistic Regression	0.72177	0.28926	0.32104	0.31853
Random Forest	0.96367	0.19819	0.42866	0.34664
Neural Networks	0.88132	0.20713	0.34704	0.13606
Gradient Boosting	0.94301	0.20045	0.44997	0.24636

Table 11 – Performance summary of models with optimized parameters (Binary)

	Train: F1-Score SMOTE	Validation: F1-Score SMOTE	Train: F1-Score Weighted Values	Validation: F1-Score Weighted Values
Logistic Regression	0.55791	0.60950	0.58743	0.58419
Gaussian Naïve Bayes	0.49074	0.49204	0.52967	0.52715
Random Forest	0.99998	0.23136	0.99995	0.64101
Neural Networks	0.63735	0.62414	0.64295	0.63992
K-Nearest Neighbor	0.72396	0.57056	0.70525	0.59914
Support Vector Machine	0.60651	0.62797	0.60812	0.59574
Decision Trees	0.99999	0.16920	0.99998	0.55209
Bagging Classifier	0.98164	0.14964	0.97977	0.63232
AdaBoost Classifier	0.64796	0.21042	0.60628	0.60615
Gradient Boosting	0.67680	0.18998	0.61696	0.61096

Table 12 – Performance summary of models with default parameters (Multiclass)

	Train: F1-Score SMOTE	Validation: F1-Score SMOTE	Train: F1-Score Weighted Values	Validation: F1-Score Weighted Values
Logistic Regression	0.55742	0.60436	0.62658	0.62912
Random Forest	0.90853	0.21955	0.84287	0.65376
Neural Networks	0.67180	0.63382	0.63949	0.63637
Gradient Boosting	0.92639	0.23693	0.67508	0.62700

Table 13 – Performance summary of models with optimized parameters (Multiclass)

Figures

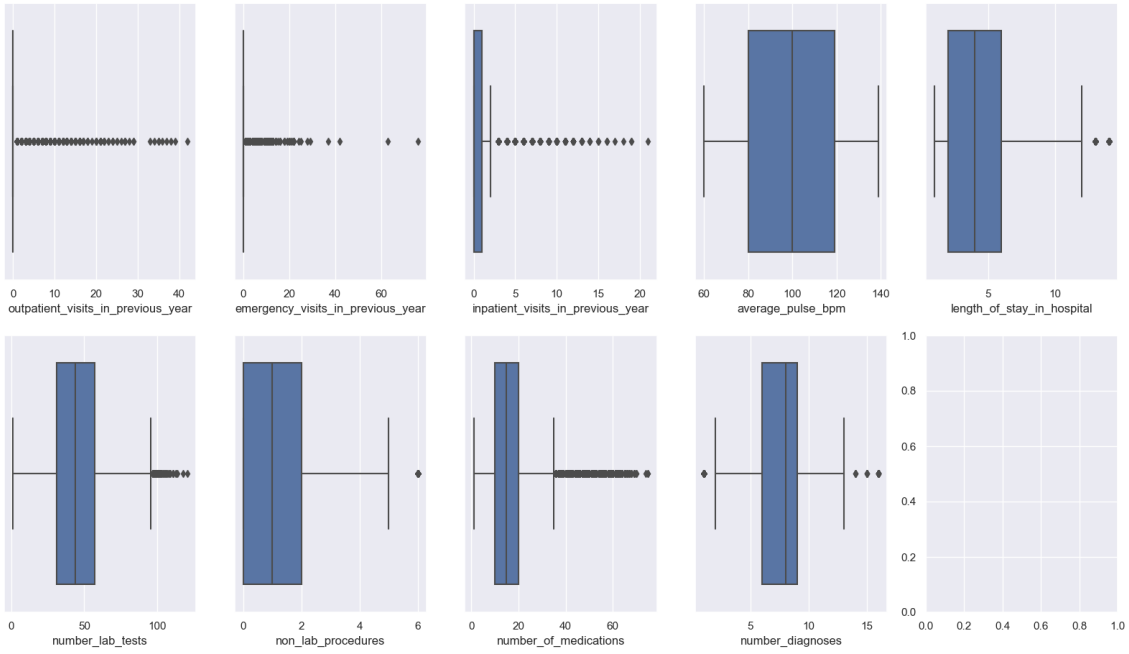

Figure 1 – All Numeric Variables' Box Plots

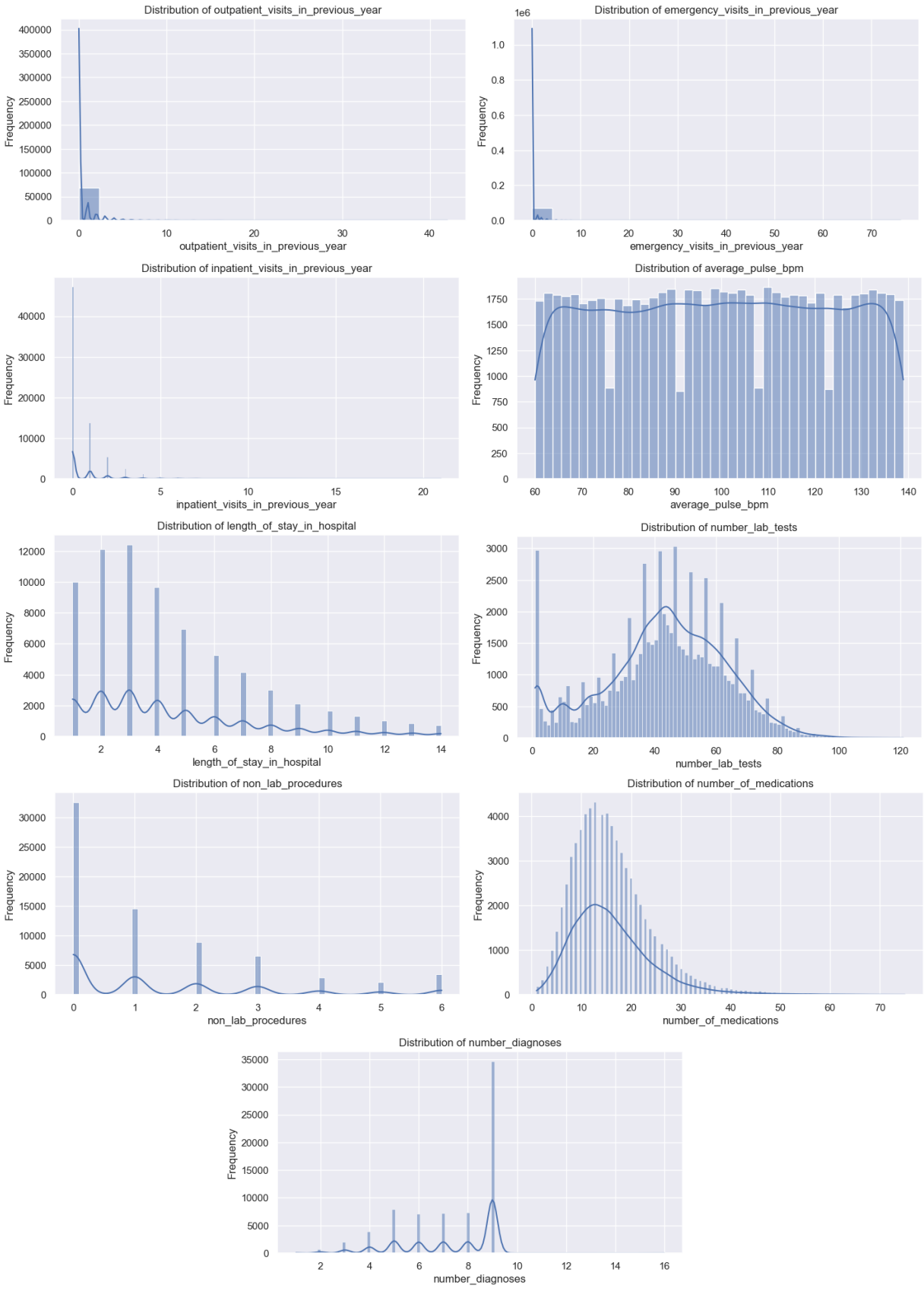

Figure 2 – All Numeric Variables distribution

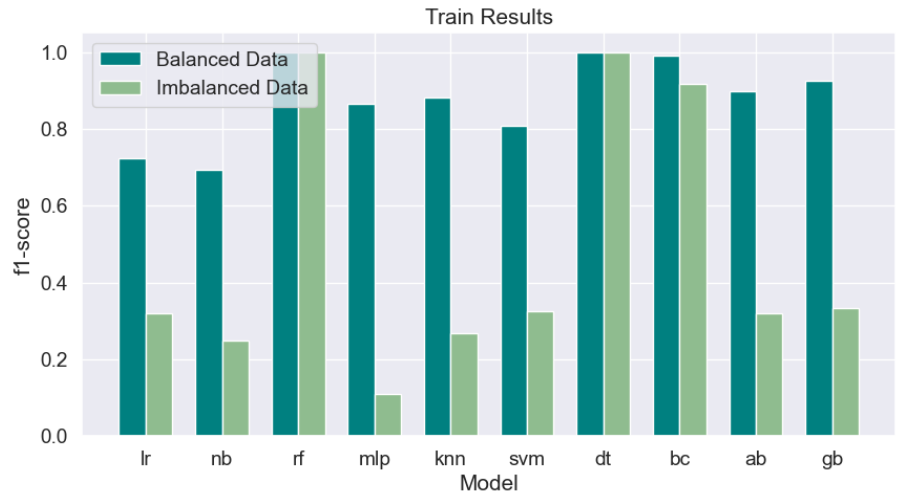

Figure 3 – Performance summary of models with default parameters in train dataset (Binary)

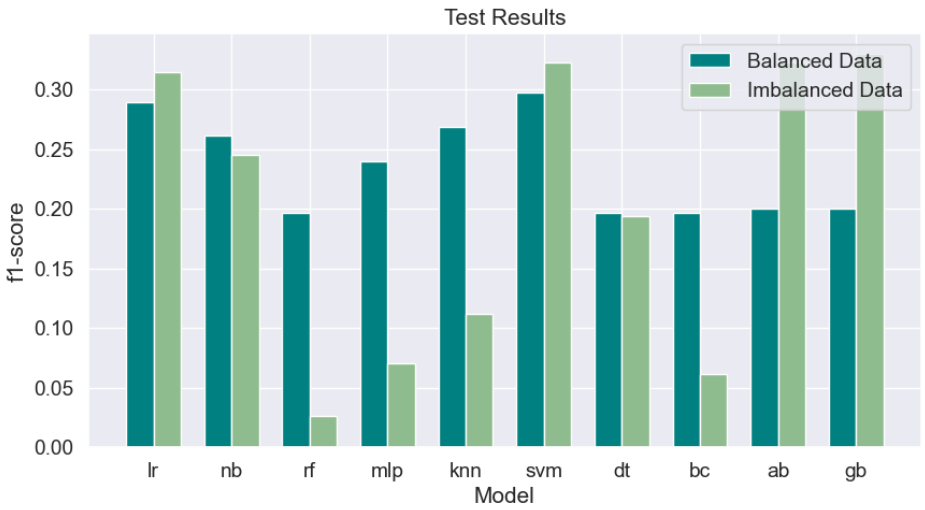

Figure 4 – Performance summary of models with default parameters in test dataset (Binary)

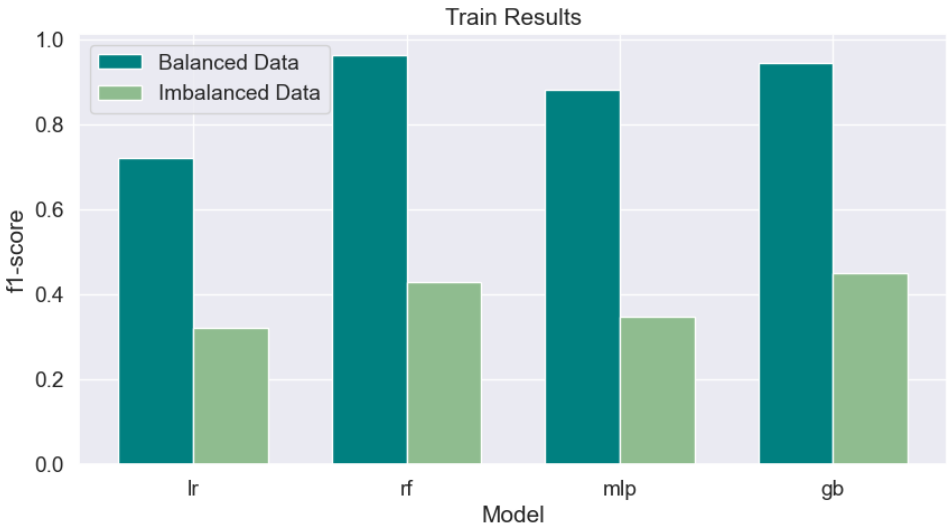

Figure 5 – Performance summary of models with optimized parameters in train dataset (Binary)

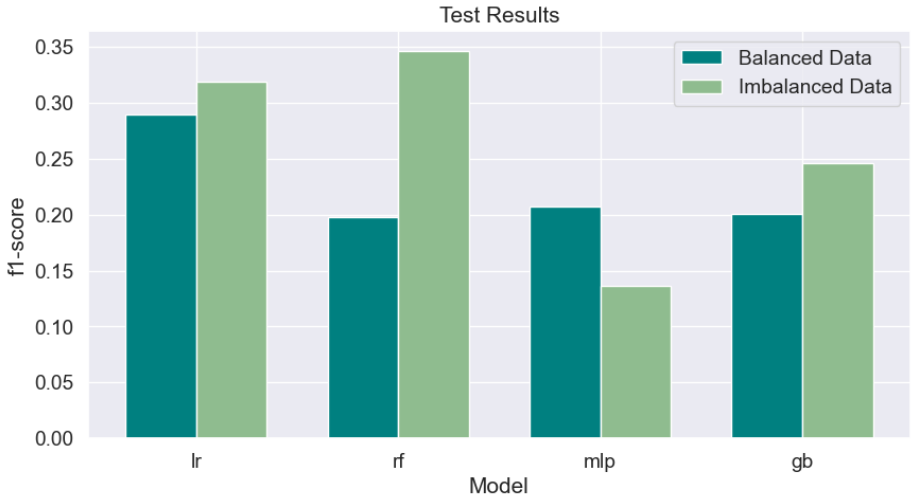

Figure 6 – Performance summary of models with optimized parameters in test dataset (Binary)

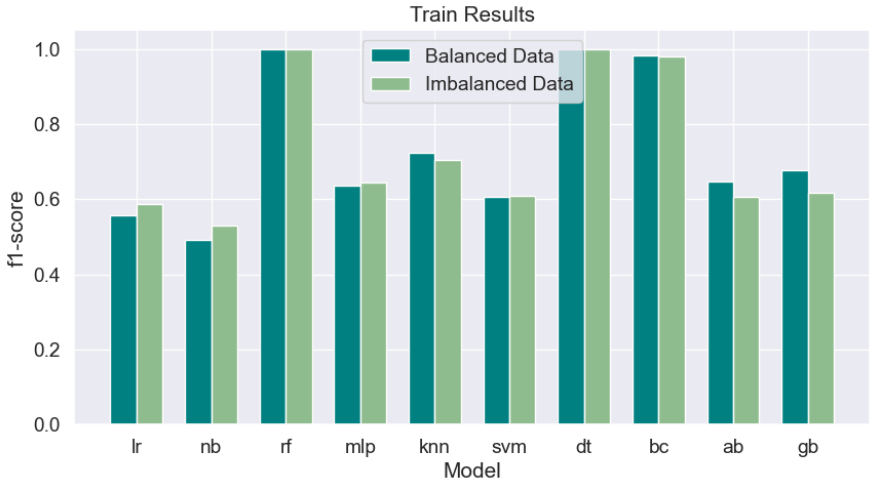

Figure 7 – Performance summary of models with default parameters in train dataset (Multiclass)

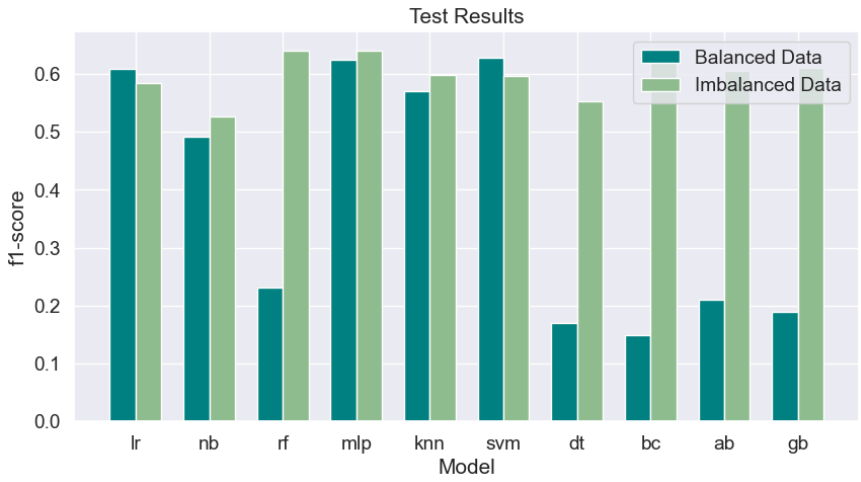

Figure 8 – Performance summary of models with default parameters in test dataset (Multiclass)

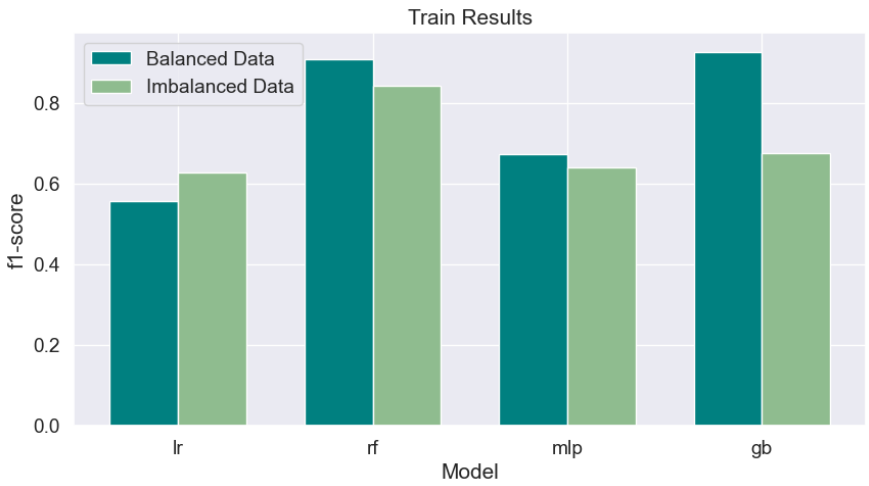

Figure 9 – Performance summary of models with optimized parameters in train dataset (Multiclass)

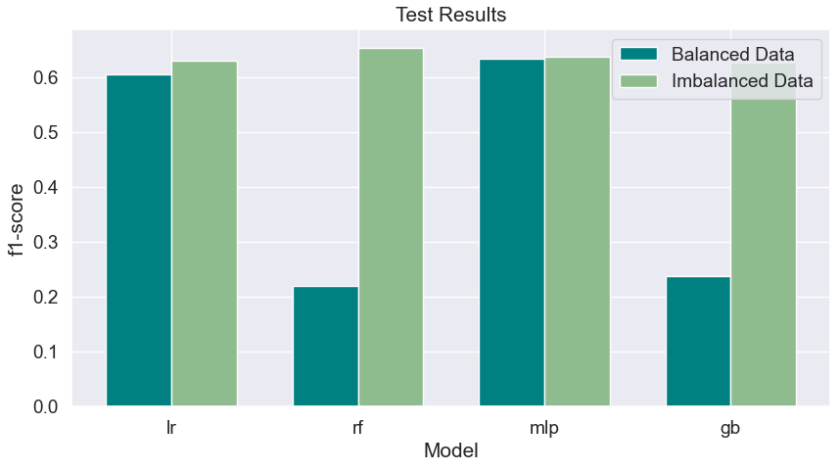

Figure 10 – Performance summary of models with optimized parameters in test dataset (Multiclass)

VII. References

1. Loke A, World Health Organization. *Diabetes*. 2023. Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes
2. Ostling S, Wyckoff J, Ciarkowski SL, Pai CW, Choe HM, Bahl V, Gianchandani R. *The relationship between diabetes mellitus and 30-day readmission rates. Clin Diabetes Endocrinol*. 2017. Available at: https://pubmed.ncbi.nlm.nih.gov/28702257/
3. Munnangi H, Chakraborty G. *Predicting Readmission of Diabetic Patients using the high-performance Support Vector Machine algorithm of SAS® Enterprise Miner™*. 2015. Available at: https://shorturl.at/fruF6
4. Burke T. (2010). *The Health Information Technology Provisions In The American Recovery And Reinvestment Act Of 2009: Implications for Public Health Policy and Practice*. Public Health Reports. (141–144) Available at: https://shorturl.at/sDFS4
5. Garber AJ, Handelsman Y, Grunberger G, Einhorn D, Abrahamson MJ, Barzilay JI, Blonde L, Bush MA, DeFronzo RA, Garber JR, Garvey WT, Hirsch IB, Jellinger PS, McGill JB, Mechanick JI, Perreault L, Rosenblit PD, Samson S, Umpierrez GE. (2020) *Consensus Statement by the American Association of Clinical Endocrinologists and American College of Endocrinology on the Comprehensive Type 2 Diabetes Management Algorithm-2020 Executive Summary*. Endocrine Practice. Vol. 26. No.1 January
6. Cleveland Clinic. *A1C*. Available at: https://my.clevelandclinic.org/health/diagnostics/9731-a1c
7. New York State Department of Health. *Patient Discharge Status FAQs*. Available at: https://shorturl.at/tADEU. University of Minnesota School Of Public Health. *Health Policy & Management Destination upon discharge from facility code*. Available at: https://shorturl.at/cnpyK
8. Karunakaran A, Zhao H, Rubin DJ. *Predischarge and Postdischarge Risk Factors for Hospital Readmission Among Patients With Diabetes*. Med Care. 2018. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6082658/
9. Stone K, Zwiggelaar R, Jones P, Mac Parthaláin N. *A systematic review of the prediction of hospital length of stay: Towards a unified Framework*. PLOS Digit Health. 2022. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9931263/
10. Tong, M. and Artiga, S. *Use of Race in Clinical Diagnosis and Decision Making: Overview and Implications*. 2021. Available at: https://www.kff.org/racial-equity-and-health-policy/issue-brief/use-of-race-in-clinical-diagnosis-and-decision-making-overview-and-implications/