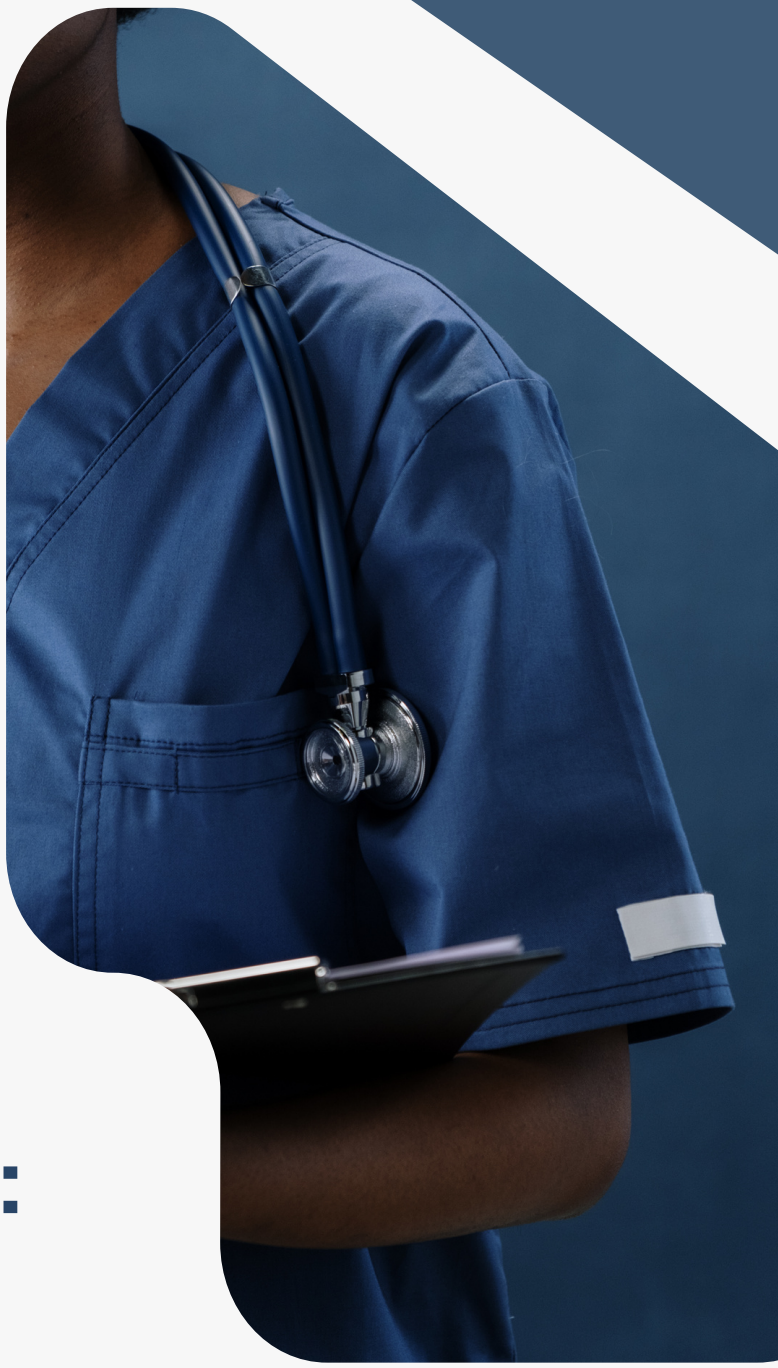
NOVA IMS Information Management School



DECEMBER 2023

READY TO BE DISCHARGED: EXAMINING HOSPITAL READMISSIONS.

Predicting 30-Day Readmissions for Patients with Diabetes

Catarina Silva | 20230368
Mariana Oliveira | 20230429
Pedro Adonis | 20230994
Sebastião Oliveira | 20220558
Tomás Veríssimo | 20230458

Professors:

Roberto Henriques
Ricardo Santos
Rafael Pereira

Index

Abstract

Background and objectives: Predicting hospital readmissions, especially for diabetic patients, is crucial for healthcare management. This machine learning project focuses on developing binary and multiclass classification models to provide nuanced insights into patient risk levels. Binary classification targets readmissions within 30 days, while multiclass classification categorizes them into within 30 days, more than 30 days, or no readmission. Accurate predictions empower timely interventions, improving care quality and reducing costs.

Methods: Strict adherence to the scikit-learn framework guided algorithm implementation. A detailed analysis of 71,236 medical records from 53,985 diabetic patients involves strategic data exploration, preprocessing, and feature engineering. Various encoding methods, including count encoding for categorical variables, were explored. For normalization, the binary classification employed square root transformation followed by standard scaling, while the multiclass classification relied on MinMaxScaler. Feature selection in both binary and multiclass classification involved RFE, SFS, and correlation analysis. Hyperparameter tuning, Randomized Search, and Grid Search were employed to optimize the models. Class imbalance was addressed through Random Under Sampling for the binary classification, while the multiclass classification explored different sampling strategies for various algorithms. Stacking was tested for both binary and multiclass problems.

Results: In binary classification, Gradient Boosting emerges as the top-performing model with a Kaggle score of 0.3653, using 13 impactful features, including discharge disposition and inpatient visits. Class imbalance is effectively addressed through Random Under Sampling, enhancing F1 scores. In multiclass classification, HistGradient Boosting was selected with a F1 score of 0.6534, demonstrating a balance between training and validation sets, utilizing a carefully chosen set of 28 features.

Conclusion: This project optimizes a large dataset for predicting hospital readmissions among diabetic patients. Despite the difficulties we found throughout the work, we were able to predict with some accuracy both classification problems. Nevertheless, to ensure the models yield accurate predictions, it would be essential to evaluate them in a clinical context to confirm the validity of our results.

Introduction

Hospital readmissions present a significant challenge in healthcare as they reflect care quality and increase costs. When patients are readmitted shortly after being discharged, it exposes gaps in care and increases the financial burden on the healthcare system. This is particularly true for readmissions among diabetic patients, which significantly impact these costs. Predicting such readmissions has the potential to enhance patient care and reduce costs.

This project focuses on developing machine learning models for binary and multiclass classification, aiming to predict if a patient will be readmitted to the hospital within 30 days of discharge. The binary classification model provides a yes/no prediction, while the multiclass model further categorizes readmissions into "<30 days", ">30 days", and "No". These predictive models serve as tools to empower healthcare providers to implement preventive measures, intervene promptly, and customize post-discharge care. This reduces readmission rates and healthcare spending.

While the field of hospital readmission prediction has been extensively studied, the particular emphasis on diabetic patients and the use of a multiclass framework offers a more detailed perspective. Existing research provides a foundation, but our goal is to contribute unique insights and tailored solutions to this critical healthcare scenario.

Based on the findings from previous studies, we anticipate that the results will be influenced by a variety of factors like number of inpatient admissions, age, sex, race, diagnosis, admission type, insulin therapy, and length of stay. Machine learning models, more specifically Random Forest, have been shown to be effective in predicting readmissions. Therefore, our machine learning approach is expected to yield insightful predictions.

In the following sections, we discuss data exploration, preprocessing, binary and multiclass classification models, and conclude with our findings and future research opportunities.

Data Exploration and Preprocessing

Before analyzing readmissions, a comprehensive exploration and preprocessing of the dataset were conducted. The analyzed data consists of 71,236 hospitalized medical records of 53,985 diabetes patients. It includes detailed attributes related to the patient encounters, such as demographic information, medical records, and target variables for binary and multiclass classification.

Regarding the dataset, nearly half of the patients are aged between 60 and 80 years (Fig. 1A). The distribution is almost balanced between male (around 46%) and female (about 54%) patients (Fig. 2A) and the vast majority of the patients are Caucasian (Fig. 3A).

The preprocessing phase involved identifying duplicates (none were found) and dropping the 'country' column due to its singular value, 'USA'. Addressing missing values across various features (Fig. 4A), strategic measures were taken. The 'weight' attribute, with 97% missing values, was removed. Missing values in 'race' and 'age' were imputed based on observations with matching 'patient_id,' while the three missing 'gender' values were filled with the mode.

Categorical features underwent thorough preprocessing. The generic medications prescribed to the patient during the encounter were organized into distinct types, serving the purpose of identifying the specific category of generic medications the patient received for treatment in the hospital. Age ranges were transformed into the midpoint ages of these intervals. Admission types, admission sources,

medical specialties, and discharge dispositions were aggregated into more general types, not only for simplification but also to address the low representation of some categories, as they had very few observations and constituted a minimal percentage of the dataset. The diagnoses, coded as the first three digits of ICD9, were assigned to broader diagnostic categories. For the attribute 'average_pulse_bpm', a study for each age normal values were made to get a binary value where whether the patient had average pulse in a normal range value. Additionally, other attributes were converted into binary format. Also, duplicated patient IDs were addressed, by introducing a new feature, 'number_encounters_total', to uniformly represent the number of encounters per patient. The 'patient_id' attribute was then eliminated.

In terms of numerical variables, two new features were created: 'Total_visits', calculated as the sum of all patient visits in the previous year, and 'Serious_condition_visits,' derived from the total of inpatient and emergency visits within the same timeframe. (Fig. 5A)

Correlation analysis, conducted using the Spearman method (Fig. 6A), revealed high correlation among variables related to visit parameters, as expected, given their representation of similar information from different perspectives. However, these were the only variables that showed noticeable correlation, as the other variables did not exhibit any discernible patterns to address their correlation.

Statistical analysis of the data revealed unbiased data distribution, evident in histograms (Fig. 7A). However, certain features still have a significant number of outliers (Fig. 8A). Despite their prevalence, no removal was done, since it was noticed since the early stages that outliers are important. Especially on the most impactful features, such as 'inpatient_visits' and 'number_encounters_total'. Traditional outliers on these features are significant values, representing a large percentage of the total values. Discarding them would neglect patients with more serious problems, as these outliers indicate critical cases.

Categorical variables were subjected to Count Encoding, which involves replacing each category in a categorical column with the count of how frequently that category appears in the dataset, enhancing representation. In our exploration, we considered alternative encoding methodologies, such as Target Encoding and One-Hot Encoding. However, One-Hot Encoding led to too many features, complicating the subsequent feature selection process. Frequency encoding emerged as a more suitable choice, especially for categorical variables with high cardinality and because of the need for a more streamlined representation that facilitated subsequent analysis and modeling. To further improve encoding strategies, future experiments could explore a hybrid approach, combining one-hot encoding for variables with few categories and frequency encoding for those with a higher number of categories, tailoring the mix based on each variable's specific characteristics.

Following the separation of the target variables from the dataset, we used a Random Forest Regressor to estimate missing values in 'Age' based on other features. This ensemble learning model is known for accurately predicting continuous values and handling complex data relationships.

Instead of the conventional train-test split for training and validation, a more robust data-splitting strategy was employed during the subsequent model training and evaluation phases: k-fold cross-validation. The advantage lies in providing a more comprehensive assessment of the model's effectiveness across diverse data subsets, mitigating the risk of overfitting associated with relying on a single random split. Cross-validation enhances the reliability of model evaluation and contributes to a more accurate estimation of its generalization capabilities.

Binary Classification

Feature Selection

The binary classification analysis commenced with a feature selection phase.

Numerical variables underwent Univariate Methods, Spearman (Fig. 6A), and Point Biserial Correlation to eliminate univariate variables, assess correlation, and eliminate variables based on their correlation with the target, respectively.

For categorical variables, Chi-square and the Phik Correlation Matrix (Fig. 9A) were applied to assess their correlation with the target variable. The latter calculates correlations between various variable types, aiding in the selection of features with significant relationships.

Subsequently, Recursive Feature Elimination (RFE) (Fig. 10A & Fig. 11A) and Sequential Feature Selection (SFS) methods were executed, utilizing Logistic Regression and Random Forest as machine learning algorithms for both categorical and numerical variables. These approaches play a pivotal role in determining the features that significantly impact model performance.

Additional methods, including Mutual Information, ANOVA, Lasso Regression (Fig. 12A) and Ridge Regression (Fig. 13A), were also employed, providing diverse perspectives and valuable insights that contributed to the selection of features for the models.

The final set of features used in the binary classification models comprised 13 variables: 'admission_type', 'number_encounters_total', 'inpatient_visits', 'a1c_test_result', 'medical_specialty', 'primary_diagnosis_types', 'presc_diabetes_meds_binary', 'secondary_diagnosis_types', 'discharge_disposition', 'glucose_test_result', 'race_caucasian', 'Has_Insurance' and 'metformin' (Table 1A & Table 2A).

Normalization

Following, two normalization steps were implemented to optimize the data and enhance algorithm efficiency. A comparative analysis tested variations with and without the square root, along with standard scaler and min-max scaler, to determine the most effective approach. In conclusion, after combining several approaches, the aggregation of square root transformation followed by standard scaling emerged as the most effective method for normalizing the data.

Class Imbalance Handling

Usamos o standard scaler e o square root na data inteira

With feature selection and data normalization completed, an additional challenge emerged due to class imbalance. The "no readmission" class constituted around 89% of the data, while the "readmitted before 30 days" class comprised only 11%.

Problema de desbalancemanto desbalanceando os algoritmos, dando preferência para a classes majoritária, afetando os scores

To address this imbalance, further investigation and research were undertaken to explore solutions for imbalanced datasets. Various methods were identified, with two primary approaches standing out: employing imbalanced learning methods and leveraging the 'class_weight' parameter in certain algorithms utilizing the 'balanced' option.

An investigation into imbalanced learning methods covered a range of re-sampling techniques, including Random Under Sampling, SMOTE, Random Over Sampling, among others, covering both

undersampling and oversampling approaches. The undersampling technique involves reducing the number of instances in the majority class, while oversampling aims to increase instances in the minority class. The study carefully evaluated trade-offs between model accuracy and computational demands, providing valuable insights for selecting effective machine learning strategies in scenarios with imbalanced data.

The research identified Random Under Sampling as the most consistently effective imbalanced method across various machine learning algorithms, demonstrating favorable performance and efficiency. This method, known for its speed, utilizes real data without introducing synthetic data, offering advantages over certain oversampling techniques.

Problema de imbalancing

Following the selection of the undersampling technique, further enhancements were pursued by experimenting with the 'sampling_stategy' parameter, which determines how the dataset is sampled or divided when resampling, that is, how many values are left on the majority class after using the technique. Some experimentation on this was done to find the optimal percentage of undersampling, which is the desired ratio of the number of samples in the minority class over the number of samples in the majority class after resampling. The experimentation was done by using different machine learning algorithms and checking the best sampling strategy of each one. Notably, different algorithms exhibited distinct preferences for sampling strategies; Gaussian Naive Bayes and Logistic Regression performed better with higher percentages (0.9 or 1), while Decision Trees, Multi-Layer Perceptron (MLP), and Support Vector Classifier (SVC) achieved superior results with slightly lower percentages (0.6 or 0.7). Based on this experimentation, a sampling strategy of 0.7 was adopted for all algorithms, striking a balance that yielded high performance across multiple models (Fig. 14A & Fig. 15A).

Os valores que vai se diminuir da classe majoritária

Based on this experimentation, a sampling strategy of 0.7 was adopted for all algorithms, because it had many algorithms with the highest performance and also the models that tend to perform better were included on the ones that got 0.7 as the best percentage. This process highlights the significance of addressing dataset imbalances, with Random Under Sampling (RUS) proving crucial in boosting the models' F1 scores.

Additionally, an alternative technique involving the use of the 'class_weight' parameter set to 'balanced' in scikit-learn algorithms was explored, among them are Decision Trees, Random Forest, Support Vector Classifier and Logistic Regression. This approach, tested in parallel with the previous technique, demonstrated notable performance improvement only in Logistic Regression.

Vimos o mlhor método de imbalancing, que foi o random undersampling que usamos para todas as variáveis e chegamos na melor percentagem que é 0.7;

Algorithms: Main algorithms chosen

The determination of the imbalanced learning method has been concluded, prompting a transition in focus towards the individual algorithms utilized within the project. Various algorithms, including Gaussian Naive Bayes (GaussianNB), Bernoulli Naive Bayes (BernoulliNB), Logistic Regression, Decision Tree, Random Forest, AdaBoost, Bagging, Gradient Boosting, Support Vector Machine (SVC), Multi-Layer Perceptron (MLP), and K-Nearest Neighbors (KNN), were utilized. Hypertuning was required for all algorithms except GaussianNB to identify optimal parameters and, consequently, the best model for each algorithm.

Having this in mind, RandomizedSearch and GridSearch were systematically applied throughout the testing phase. A designated pipeline, featuring Random UnderSampling as a preventative measure against data leakage and overfitting, was established for all algorithms, excluding Logistic Regression. In the most recent evaluations, RandomizedSearch was executed with a 10-fold cross-validation to

Usamos o Randomized search e o Grid serch foram usados várias vezes os dois, e até certo ponto vimos que Randomized search otimiza o tempo e é mais eficiente pra conseguir testar mais. Pq o gridsearch testa todas as combinações possíveis.

augment robustness, employing the f1 score as the scoring metric and adjusting the number of iterations.

Upon completion of this extensive experimentation, as it is shown in table1, it is evident that ensemble methods, including Random Forest, Gradient Boosting, AdaBoost and Bagging, performed exceptionally well. Additionally, Decision Trees, MLP and SVC, were close in performance. As expected KNN was not close to the best ones due to its reliance on distance, which is less meaningful in the context of categorical variables. GaussianNB, BernoulliNB and Logistic Regression did not deliver high performance, as expected because of their relative simplicity compared to the more sophisticated ensemble and neural algorithms commonly recognized as top performers in Kaggle competitions.

At each iteration of the project, after the hyperparameter tuning part, the optimal models from each algorithm underwent evaluation in the kaggle competition, in order to identify which algorithms performed better, what to improve, and to check issues related to underfitting and overfitting.

Consistent with prior research findings, it was affirmed early in the project that algorithms such as Gradient Boosting, Random Forest, MLP, and SVC exhibit superior performance in Kaggle competitions. This observation persisted through the latest iteration of the project, where the best-performing models remained consistent, even as overall scores improved.

The highest achieved score came from Gradient Boosting (0.3714), consistent with prior research indicating that Gradient Bosting often outperforms Random Forest but requires more hyperparameter tunning – an observation validated in our study.

Although the best Kaggle score in the latest iteration was attained by MLP (0.3675), the final selected model was Gradient Boosting with a score of 0.3653. This choice was guided by the model's greater robustness compared to MLP, as evidenced by its higher training data score (0.356 vs. 0.3479), making it less susceptible to underfitting.

Even though MLP had the best score on kaggle, the best algorithm was Gradient boosting due to its robustness and a better score on the train

Models	Scores	Standard Deviation	Kaggle Score
mlp	0.3479	0.0107	0.3675
histgradient boosting	0.3549	0.0112	0.3655
adaboosting	0.3528	0.0098	0.3655
gradient boosting	0.356	0.0109	0.3653
bagging	0.3508	0.0095	0.3646
random forest	0.3514	0.0115	0.364
decision tree	0.3371	0.0089	0.3625
svc	0.337	0.0033	0.3554
logistic regression	0.3175	0.0047	0.3296
bernoullinb	0.3138	0.0057	0.3225
knn	0.2983	0.0114	0.3216
gaussiannb	0.2898	0.0082	0.2903

Hist gradient boosting e o bernoulli NÃO VIMOS EM AULA, REVISAR!!!

Table 1 – Final model scores for binary classification

O gradient boosting foi escolhido ao invés MLP pq a diferença dos scores entre o train e o score no kaggle, causando um menor overfitting

Upon obtaining the final models, comprehensive classification reports were generated, encompassing metrics such as recall, precision, and accuracy. Based on these reports, the GaussianNB and KNN models demonstrated the most elevated accuracy, registering 0.79 and 0.81, respectively. In the realm of recall, BernoulliNB emerged as the preeminent model, achieving a score of 0.71. Notably, KNN displayed the highest precision at 0.25.

Further analysis involved the calculation of the Area Under the ROC curve (AUC) for all algorithms, measuring the trade-off between sensitivity and specificity (True positive rates vs False positive rates). The optimal value has the largest area under the ROC curve. Knowing this, the best algorithms were

Gradient Boosting, Bagging and AdaBoost, as you can see in Figure 1, all achieving an AUC of 0.76. Other algorithms, including MLP and Random Forest, closely followed with an AUC of 0.75.

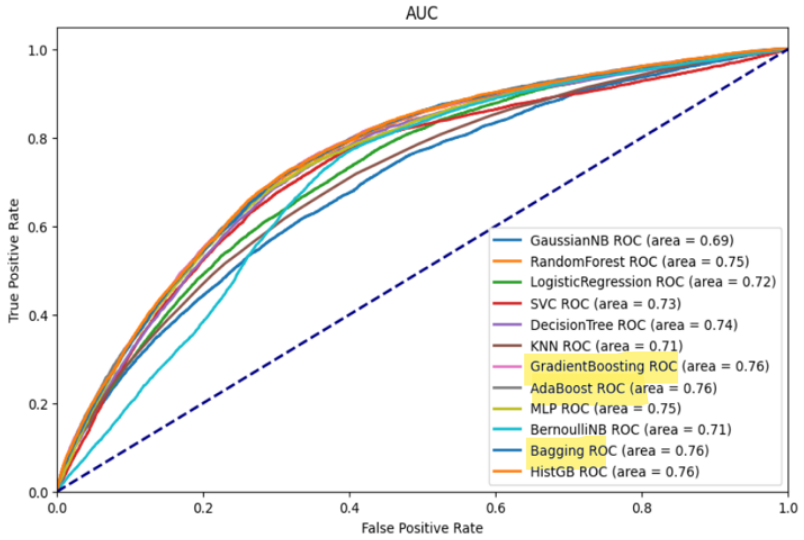


Figure 1 – "AUC curve based on different machine learning algorithms "

Exploration of ensemble methods, specifically the Voting and Stacking methods, was undertaken, but without the success that was expected. These methods combine various algorithms together to try to enhance performance. In Voting, all methods are simultaneously combined, and changes can be made on the weights of each model if needed. Stacking, on the other hand, involves combining some models as base estimators first and using their outputs as inputs in the final estimator that computes the ultimate prediction. Several tests were done with both the Voting and the Stacking methods, exploring combinations of 2 or 3 different models, with a focus on those with different characteristics, such as combining an ensemble method with MLP or with SVC. Surprisingly, the results were always lower than the highest individual model score (Gradient Boosting with 35.6). Notably, the highest score achieved with a Voting method was combining MLP and Gradient Boosting (0.3511); and with a Stacking method was combining Random Forest and Gradient Boosting as base estimators and Logistic Regression as final estimator (0.3473). This runs counter to the anticipated trend, as these ensemble methods generally can surpass basic models and have demonstrated strong performances in Kaggle competitions.

As mentioned earlier, Gradient Boosting outperformed other algorithms in various tests, showing minimal underfitting compared to MLP, which excelled in the last Kaggle iteration. Its reputation as one of the top-performing algorithms in Kaggle competitions influenced our model selection. Lastly, Gradient Boosting, along with two other ensemble methods, demonstrated the largest area under the curve (AUC) in the test. With all of this considered, Gradient Boosting was the model chosen for binary classification.

Multiclass Classification

Ver a diferença pro binary e o f1 weighted

Feature Selection

Firstly, we've started by an exploration of numerical variables, attempting to find univariate significance. However, no variables displayed significance univariate impact. We then utilized

Spearman Correlation and a Heat Map visualization (Fig. 16A), revealing notable correlations among certain variables, such as inpatient_visits vs Total_visits (0.80), inpatient_visits vs Serious_condition_visits (0.92) and Total_visits vs Serious_condition_visits (0.87).

Recursive Feature Elimination (RFE) was then employed with Decision Trees (Fig. 19A), Random Forest (Fig. 17A) and Logistic Regression (Fig. 18A), all configured with Cross Validation. Decision Trees and Logistic Regression led to substantial feature discards, while Random Forest retained all.

Next, was applied Sequential Feature Selection (SFS), a technique in where features are selected for a model in a stepwise manner, with 'k_features' set to 'best' and using Random Forest and Logistic Regression aimed to identify the best feature combination yielding the highest F1 score.

Lasso (Fig. 20A) and Ridge Regression (Fig. 21A) were explored, with Lasso discarding two variables and Ridge retaining all. Lastly, for numerical variables, we tested Mutual Information, a method that measures the dependency between variables, selecting features that have the most information in common with the target variable, and ANOVA, used to identify statistically significant differences between the means of numerical variables, helping to identify those that have a strong impact on the response variable. Trade-offs between the different approaches resulted in the exclusion of 'average_pulse_bpm,' 'total_visits,' and 'Serious_condition_visits'.

For categorical data, we applied Chi-Squared test, ANOVA, Mutual Information, Lasso, Ridge, and Phik Correlation Matrix (Fig. 22A). The latter calculates the correlation between various variables types, enabling the selection of features with significant relationships with the target variable. Following careful consideration, 'Payer_code', 'Glucose_test_result', 'Insulin', 'Combination_medications', 'Race_caucasian', 'is_normal_pulse' and 'Presc_diabetes_meds_binary' were discarded.

By synthesizing these methods for both numerical and categorical variables, we've arrived at a final selection of 28 features (Table 3A, 4A) for subsequent model implementation, aiming for optimal performance.

Normalization

Before initiating model testing, it was imperative to ensure that the features were on a consistent scale. Inconsistent feature scales can adversely impact the performance of certain machine learning algorithms, particularly those sensitive to the magnitude of input features.

Our selection of the MinMaxScaler was predicated on the outcomes derived from various feature selection methods and their corresponding performance scores. In this matter, the MinMaxScaler emerged as the method yielding superior scores in our evaluations.

Model Parameter Exploration for Randomized Search

In the initial stages of our model development, we recognized the potential value in conducting a comprehensive exploration of hyperparameter space for three ensemble models that we considered essential for our testing — Random Forest, AdaBoosting, and Gradient Boosting. Our primary goal was to identify promising parameter ranges which could then form the foundational basis for subsequent Randomized Search optimization, aiming to enhance efficiency by pinpointing values within hyperparameter space likely to yield improved model performance.

To achieve these results, some key functions were defined to assess the performance of the chosen ensemble models. Therefore, the exploration of hyperparameters involved systematic variation in essential variables, such as 'max_depth', 'n_estimators', and 'learning_rate'. The results were visually represented using boxplots, offering insights into the distribution of F1 scores across diverse parameter settings. The inclusion of mean F1 scores provided a central tendency perspective on model performance. In this manner, these foundational steps establish the groundwork for subsequent model optimization, providing comprehensive insights into the behavior of the chosen ensemble models. These findings guided our subsequent exploration of hyperparameter space using Randomized Search and Grid Search to further refine model configurations.

Model Testing

In our pursuit of identifying the most effective model for the multiclass problem, we initiated a random search across various models to identify potential candidates for further exploration. The table below (table 2) showcases the F1 weighted scores obtained through random search:

Models	Scores RS
Random Forests	0.6537
Histgradient boosting	0.6534
Gradient boosting	0.6508
Decision Trees	0.6445
AdabBoosting	0.6385
MPL	0.6384
Bernoulli NB	0.6235
Softmax	0.6050
Gaussiannb	0.5287
KNN	0.4712

Table 2 – Model Scores for multiclassification under Randomized Search

Analyzing the top three models, it becomes evident that they share commonalities, potentially explaining their superior performances compared to other models. It's noticeable that these models belong to the ensemble learning category, utilizing multiple base models to enhance overall performance, fostering robustness, and mitigating the risk of overfitting. Additionally, all three models leverage decision trees as foundational elements, seamlessly aligning with the complexity of healthcare datasets and their ability to capture intricate relationships within the data. Ensemble tree models excel in capturing non-linear patterns, a crucial aspect in healthcare scenarios where patient readmission prediction involves intricate, non-linear interactions.

On the other hand, the suboptimal performance of Gaussian Naive Bayes (GaussianNB) and K-Nearest Neighbors (KNN) models in this multiclassification problem may stem from their inherent limitations in capturing intricate relationships within the data. GaussianNB assumes independence between features, potentially struggling with complex dependencies present in healthcare data. Furthermore, KNN, relying heavily on distance metrics, may encounter difficulties in high-dimensional spaces or datasets with imbalanced class distributions—common characteristics in medical datasets.

Subsequently, a grid search was conducted to assess potential score improvements in the top three models. The model's performance exhibited the following improvements: Gradient Boosting increased from 0.6508 to 0.6525, HistGradient Boosting remained unchanged at 0.6534, indicating no improvement, and Random Forests surged from 0.6537 to 0.6581, demonstrating the most significant

enhancement. Comparing the scores obtained, the top-performing models demonstrated consistent and competitive scores across both random search and grid search.

As an attempt to enhance classification performance, we explored the one-vs-rest (OvR) classification strategy. However, inconsistent patterns in scores emerged, with some models showing marginal improvement while others exhibited worsened performance. Given these inconsistencies and the limited significance of improvements, we opted not to incorporate the one-vs-rest approach into our final model. We prioritize reliability and robustness in the classification methodology. The tabulated data below (table 3) illustrates the observed disparities in the one-vs-rest (OvR) strategy and the minimal improvements in the models, contrasting with the outcomes from the final grid searches conducted without the utilization of this method:

Models	Scores	One-vs-Rest
Random Forests	0.6581	0.6599
Histgradient boosting	0.6534	0.6498
Gradient boosting	0.6525	0.6533
Decision Trees	0.6445	0.6460
AdabBoosting	0.6385	0.6324
MPL	0.6384	0.6374

Table 3 – "Comparative Analysis of Final Model Scores for multiclassification Using the One-vs-Rest Approach "

In this manner, as we decided not to include the One-vs-Rest strategy, the top-performing models based on F1 scores in GS were Random Forest (0.6581), HistGradient Boosting (0.6534), and Gradient Boosting (0.6525), showcasing their potential for our multiclass problem.

To leverage the strengths of these models, we implemented a stacking algorithm (StackingClassifier). Initially, we stacked the two top-performing models, namely Random Forest and HistGradient Boosting. Subsequently, we expanded the stack to incorporate the three leading models (Random Forest, HistGradient Boosting, and Gradient Boosting), resulting in a score improvement from 0.6541 to 0.6551.

The ultimate assessment of our models, aimed at selecting the optimal one, involved a train-test split to evaluate their performance on previously unseen data.

The results indicated that Random Forest achieved a high training F1 score (0.8685) but showed a performance drop on the validation set (0.6515). HistGradient Boosting exhibited competitive scores on both sets (Training F1: 0.6652, Validation F1: 0.6490), while Gradient Boosting displayed slightly lower performance (Training F1: 0.6651, Validation F1: 0.6463). The Stacking Classifier, combining all three models, demonstrated promising results (Training F1: 0.6784, Test F1: 0.6485).

After careful consideration, we selected HistGradient Boosting as our final model. Its consistent performance across varied evaluation scenarios, demonstrating strategic balance between training and validation, and the reduced risk of overfitting compared to other models, reaffirmed its suitability for our multiclass classification task.

Conclusion

In conclusion, this machine learning project analyzed the chances of predicting hospital readmissions for diabetic patients. Through data exploration and preprocessing, it was managed to refine a large dataset into meaningful features. For the binary classification, we worked with a reduced feature set (13 features), being Gradient Boosting the best results. On the other hand, for multiclass classification, a broader set (28 features) was necessary, and the finest outcome came from HistGradient Boosting.

One point to consider is although some of the limitations of the work were previous knowledge, the fact that we can only use scikit-learn machine learning algorithms and the time factor, made the project more challenging. With more time, another thing to consider for future improvements could be exploring imbalanced learning methods even further, especially on binary classification and trying different combinations of encoding on the categorical features. We could have also spent more time on feature engineering and feature selection. In general, there is always something different to experiment with in every step when the topic is a machine learning problem. Furthermore, the multiclass analysis revealed aspects that could require additional exploration, particularly regarding the observed inconsistencies in the one-vs-all approach. To gain deeper insights into this matter, a viable approach would be to conduct a comprehensive analysis of the results. The examination would aim to comprehend the decline in specific model scores and pinpoint potential issues that may be contributing to these variations.

With that in mind, despite the minor difficulties this project was able to successfully demonstrate the application of machine learning algorithms in healthcare, particularly in making predictive models for hospital readmissions among diabetic patients.

References

sklearn.naive_bayes.BernoulliNB. (n.d.). Scikit-learn.

https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html

A guide to appropriate use of Correlation coefficient in medical research.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/

The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifiers.

https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01423-y

Imblearn. (n.d). Imbalanced learn.

https://imbalanced-learn.org/stable/references/index.html

mlxtend.feature_selection.SequentialFeatureSelector. (n.d). Mlxtend.

https://rasbt.github.io/mlxtend/api_subpackages/mlxtend.feature_selection/#sequentialfeatureselector

sklearn.ensemble.HistGradientBoostingClassifier. (n.d.). Scikit-learn.

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

sklearn.multiclass.OneVsRestClassifier. (n.d.). Scikit-learn.

https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

Annex

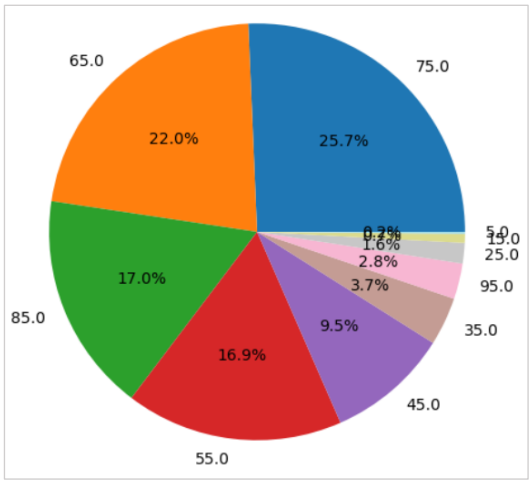


Fig. 1A – Distribution of age

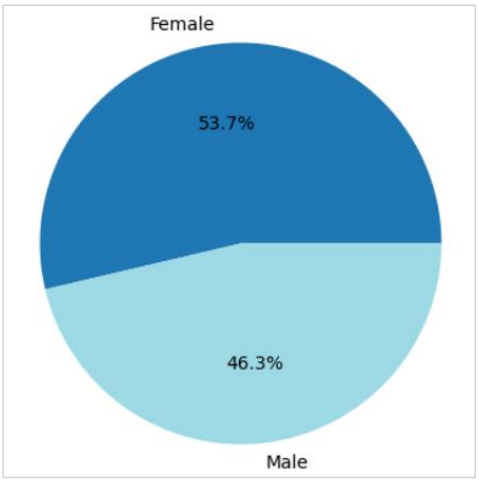


Fig. 2A – Distribution of gender

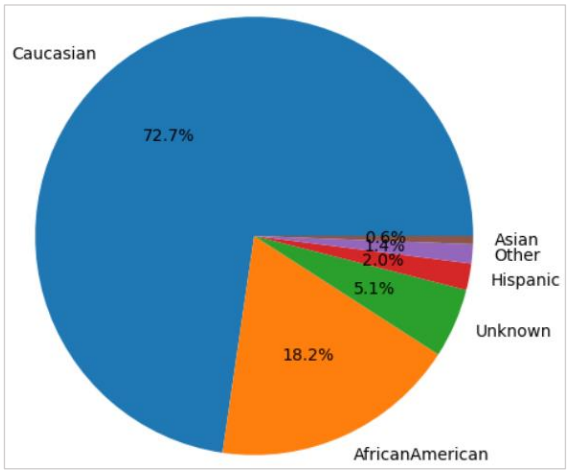


Fig. 3A – Distribution of race

Attribute	Missing rate
patient_id	0.0%
race	7.12%
gender	0.0%
age	4.99%
weight	96.85%
payer_code	39.59%
outpatient_visits_in_previous_year	0.0%
emergency_visits_in_previous_year	0.0%
inpatient_visits_in_previous_year	0.0%
admission_type	5.2%
medical_specialty	49.02%
average_pulse_bpm	0.0%
discharge_disposition	3.64%
admission_source	6.62%
length_of_stay_in_hospital	0.0%
number_lab_tests	0.0%
non_lab_procedures	0.0%
number_of_medications	0.0%
primary_diagnosis	0.02%
secondary_diagnosis	0.37%
additional_diagnosis	1.42%
number_diagnoses	0.0%
glucose_test_result	94.82%
a1c_test_result	83.27%
change_in_meds_during_hospitalization	0.0%
prescribed_diabetes_meds	0.0%
medication	0.0%
readmitted_binary	0.0%
readmitted_multiclass	0.0%

Fig. 4A – Missing Values

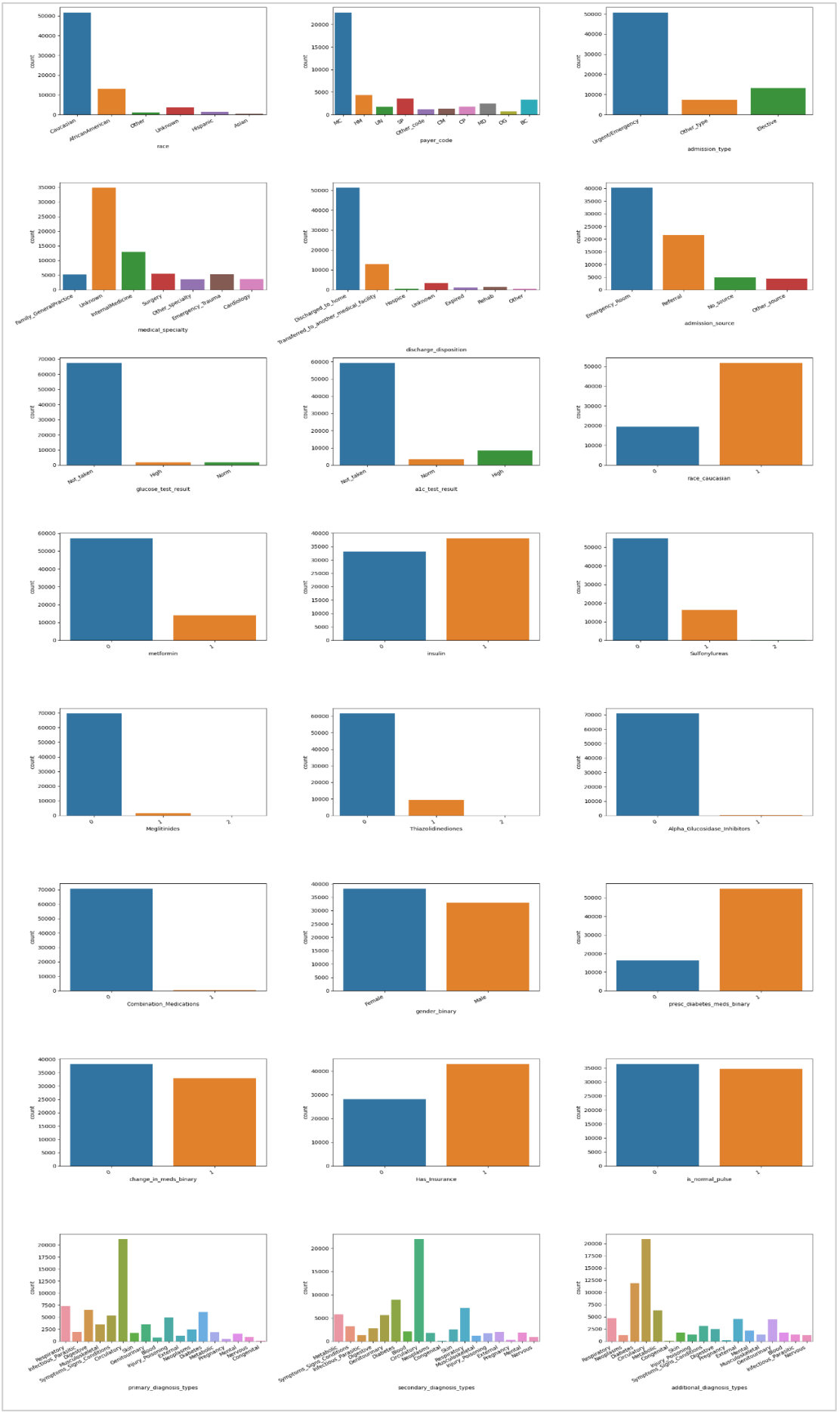


Fig. 5A – Bar charts for categorical and binary features

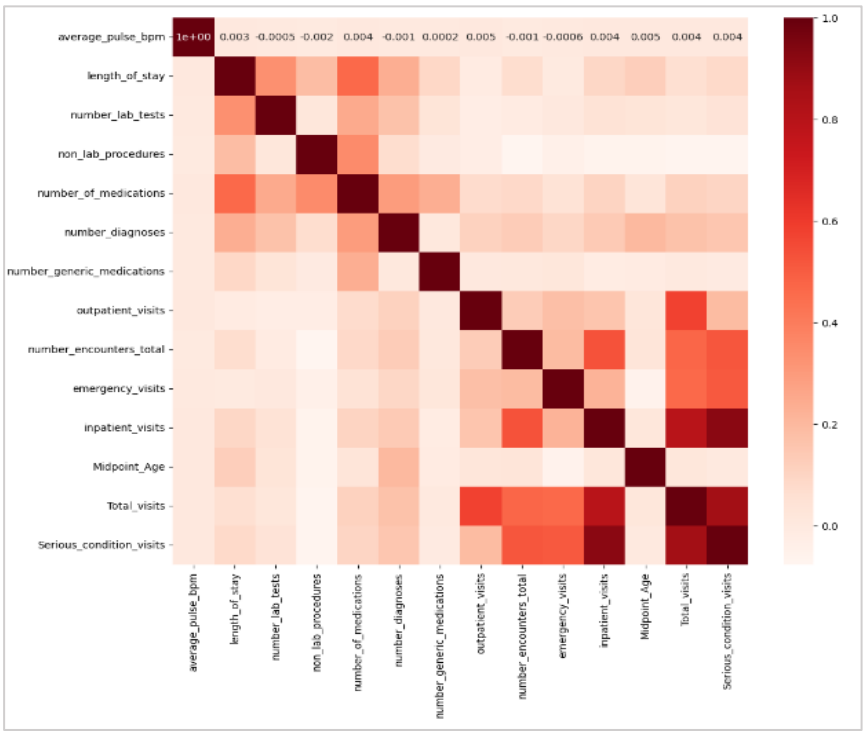
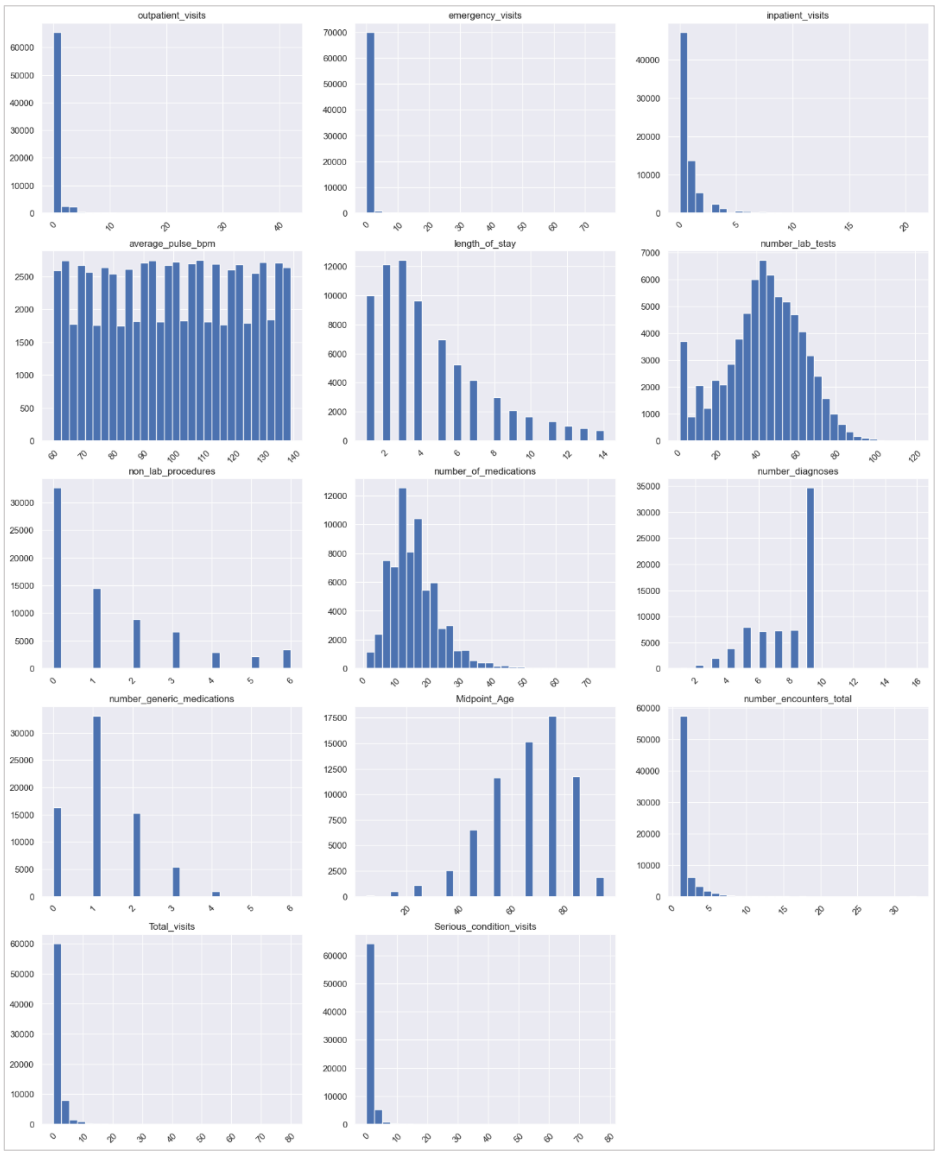


Fig. 6A – Spearman correlation matrix

Fig. 7A – Histograms for numerical features



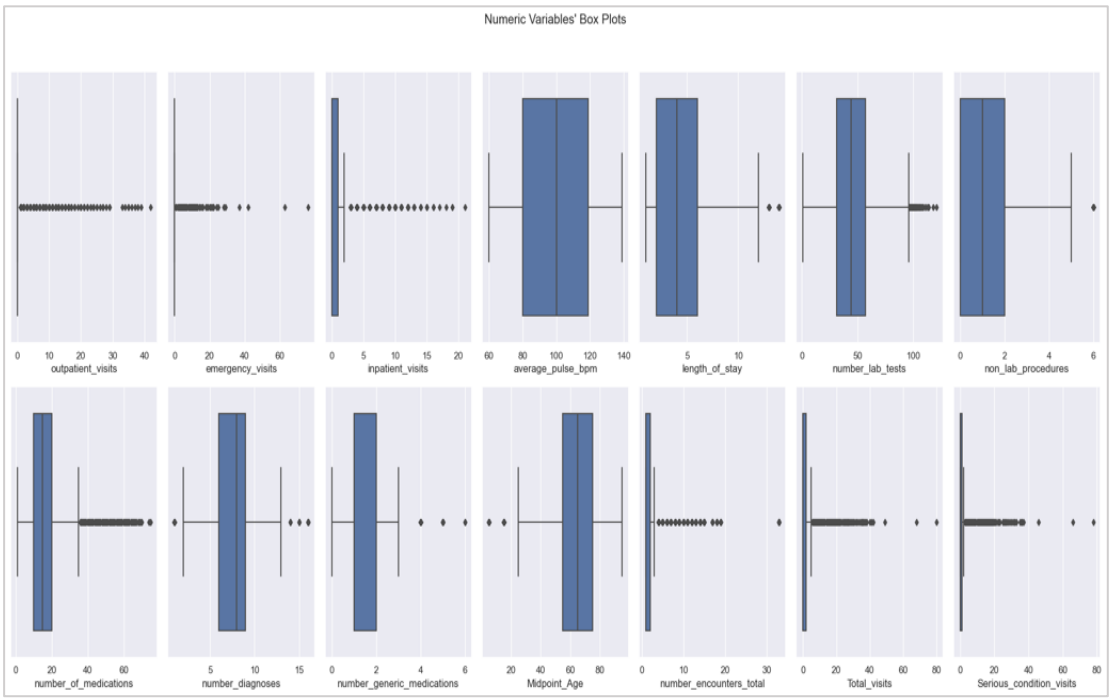


Fig. 8A – Boxplots Numerical Variables

Numerical Data

Predictor	Spearman	RFE LR	RFE RF	SFS LR	SFS RF	Lasso	Ridge	Mutual Info	ANova	PB	What to do? (One possible way to "solve")
Outpatient_visits	Keep	Discard	Discard	Keep	Discard	Keep	Keep?	Keep?	Keep	Keep	Discard
Emergency_visits	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
Inpatient_visits	Keep?	Discard	Keep	Discard	Keep	Keep?	Keep	Keep	Keep	Keep	Include in the model
Average_pulse_bpm	Keep	Discard	Discard	Keep	Discard	Discard	Keep?	Keep?	Discard	Discard	Discard
Length_of_stay	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
Number_lab_tests	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
Non_lab_procedures	Keep	Discard	Discard	Discard	Discard	Keep?	Keep?	Keep?	Keep	Keep	Discard
Number_of_medications	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
Number_diagnoses	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Try with and without
Number_generic_medications	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Discard	Discard
Midpoint_Age	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Discard	Keep	Try with and without
Total_visits	Keep?	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Keep	Discard
Serious_condition_visits	Keep?	Discard	Discard	Keep	Discard	Discard	Keep	Keep	Keep	Keep	Discard
Number_encounters_total	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Include in the model

Table 1A – Numerical variables feature selection overview (Binary case)

Categorical Data

Predictor	Chi-Square	Phik	RFE LR	RFE RF	SFS LR	SFS RF	Lasso	Ridge	Mutual Info	What to do? (One possible way to "solve")
Race	Keep	Keep?	Discard	Discard	Discard	Discard	Discard	Keep	Keep	Discard
Payer_code	Keep	Keep?	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Discard
Admission_type	Keep	Keep	Discard	Discard	Discard	Keep	Discard	Keep?	Keep	Try with and without
Medical_specialty	Keep	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Try with and without
Discharge_disposition	Keep	Keep	Keep	Discard	Discard	Keep	Keep	Keep	Keep	Include in the model
Admission_source	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Include in the model
Primary_diagnosis_types	Keep	Keep	Discard	Discard	Discard	Discard	Keep?	Keep?	Keep	Try with and without
Secondary_diagnosis_types	Keep	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Try with and without
Additional_diagnosis_types	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Try with and without
Glucose_test_result	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Try with and without
A1c_test_result	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Keep	Try with and without
Race_caucasian	Keep	Keep?	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Include in the model
Insulin	Keep	Keep?	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Try with and without
Metformin	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Keep?	Try with and without
Sulfonylureas	Discard	Keep	Discard	Discard	Discard	Discard	Keep?	Keep	Keep	Discard
Meglitinides	Keep	Keep	Discard	Discard	Discard	Keep	Keep?	Keep	Discard	Discard
Thiazolidinediones	Discard	Keep	Discard	Discard	Discard	Discard	Discard	Keep	Discard	Discard
Alpha_Glucosidase_Inhibitors	Discard	Keep	Discard	Discard	Discard	Keep	Keep	Keep	Discard	Discard
Combination_Medications	Discard	Keep	Discard	Discard	Discard	Keep	Keep?	Keep?	Discard	Discard
Gender_binary	Discard	Keep	Discard	Discard	Discard	Keep	Keep?	Keep	Keep	Discard
Presc_diabetes_meds_binary	Keep?	Keep	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Include in the model
Change_in_meds_binary	Keep?	Keep	Discard	Discard	Discard	Discard	Keep?	Keep	Keep	Discard
Has_Insurance	Keep	Keep?	Discard	Discard	Discard	Discard	Keep	Keep	Keep	Include in the model
Is_normal_pulse	Discard	Keep	Discard	Discard	Discard	Discard	Discard	Keep?	Keep	Discard

Table 2A – Categorical variables feature selection overview (Binary case)

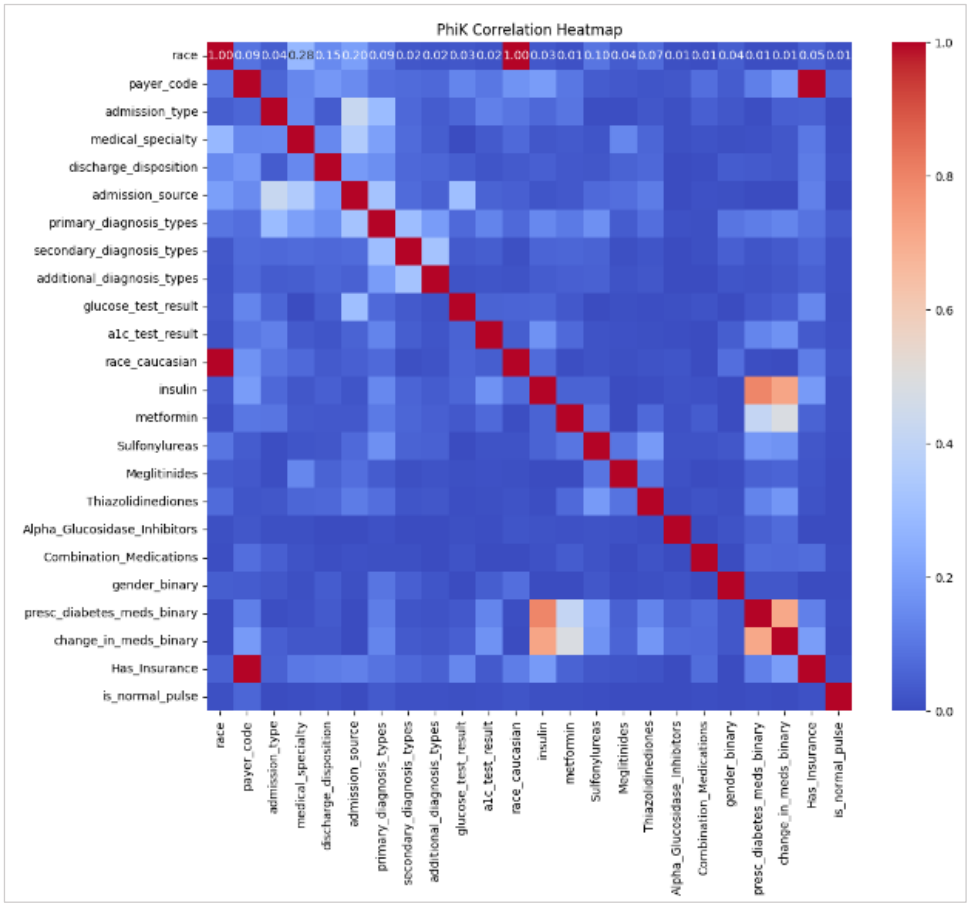


Fig. 9A – Phik correlation Matrix for the categorical variables (Binary case)

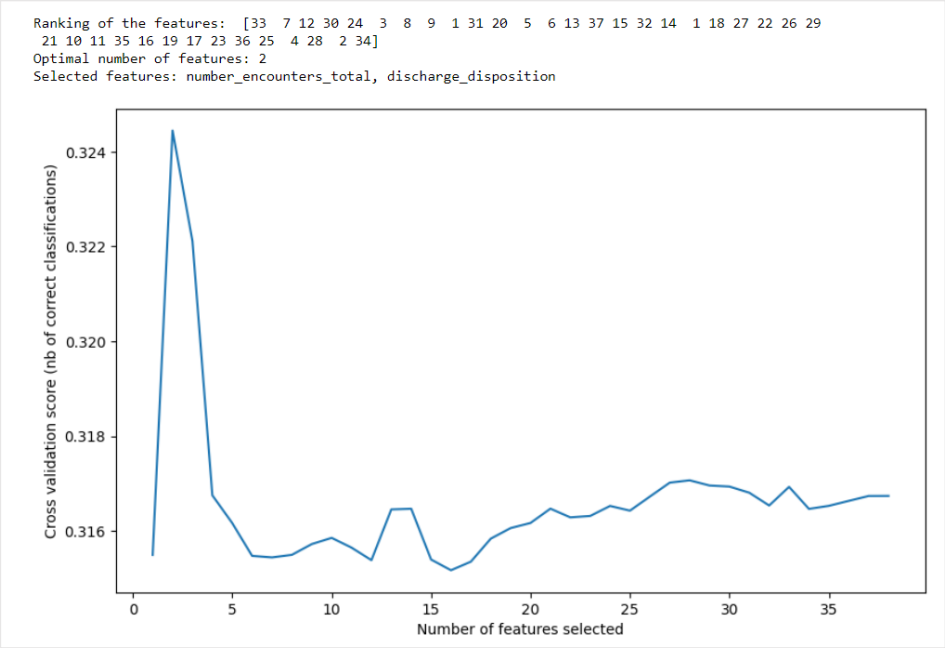


Fig. 10A – Recursive Feature Elimination evaluation with Logistic Regression (Binary case)

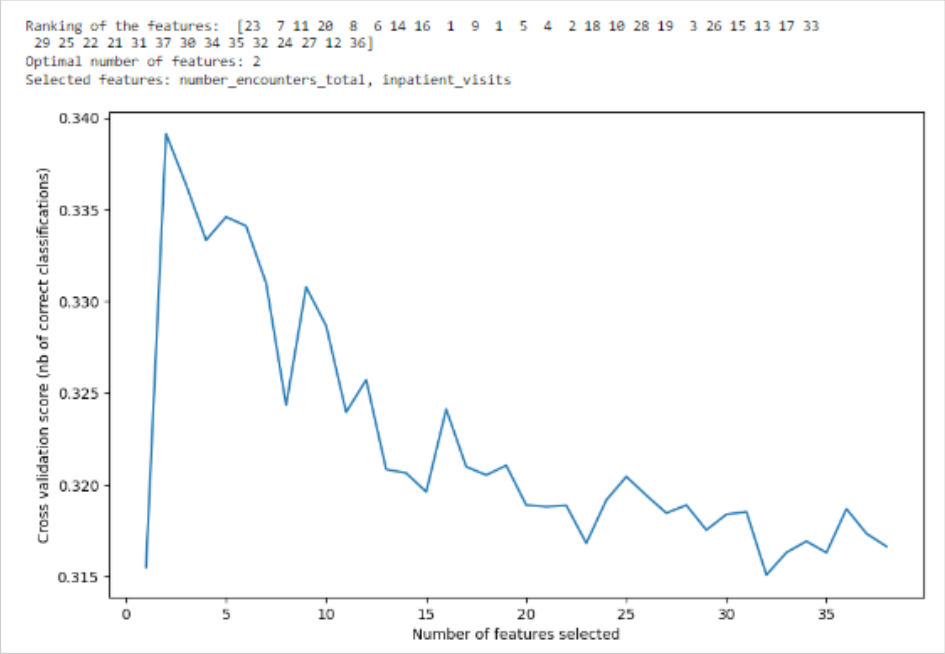


Fig. 11A – Recursive Feature Elimination evaluation with Random Forest (Binary case)

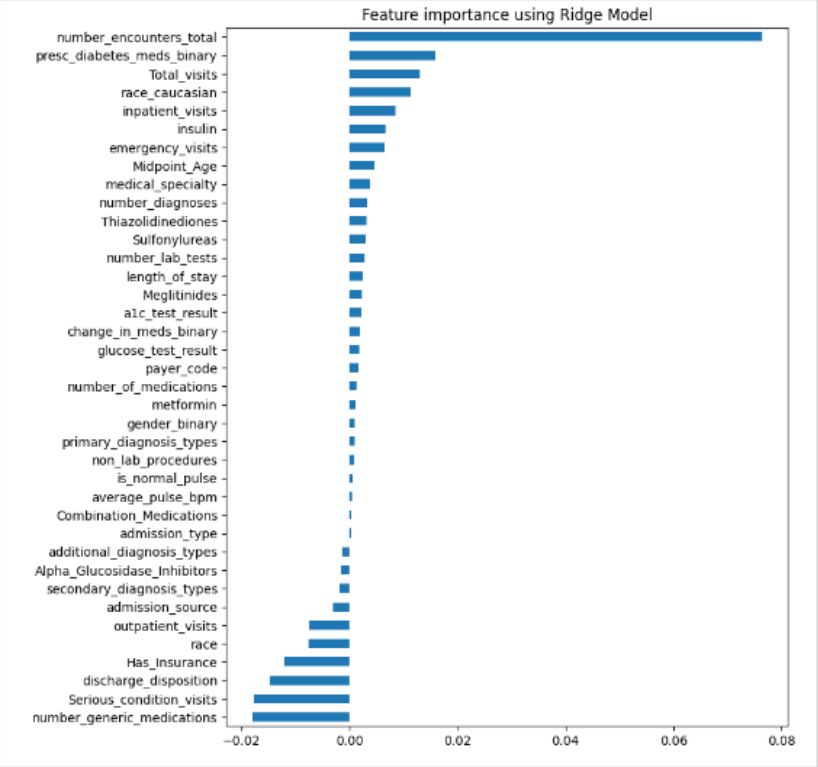


Fig. 12A – Feature Importance using Ridge model (Binary case)

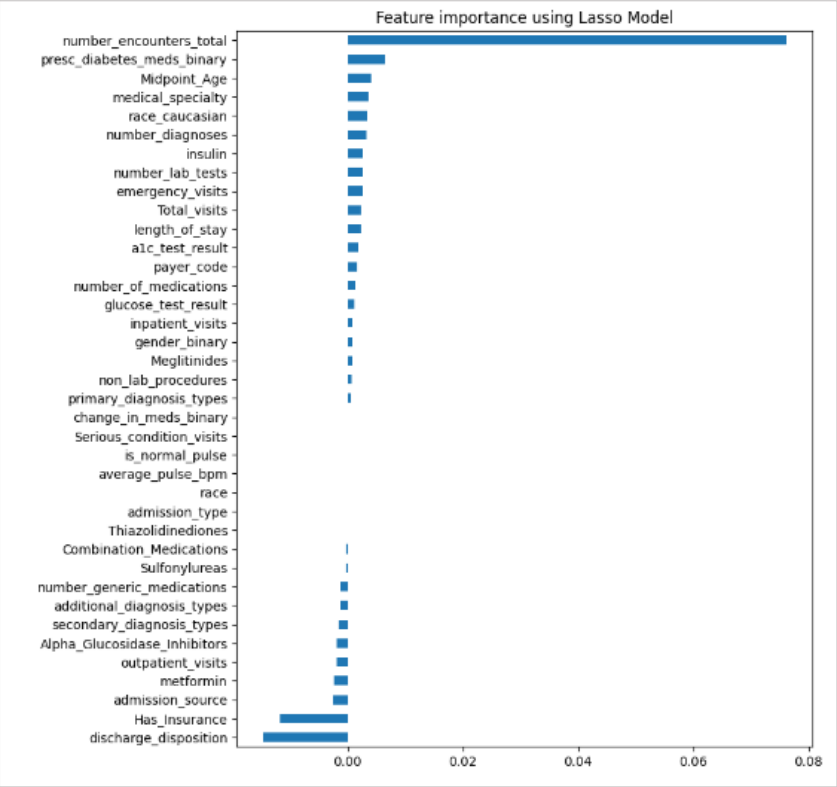
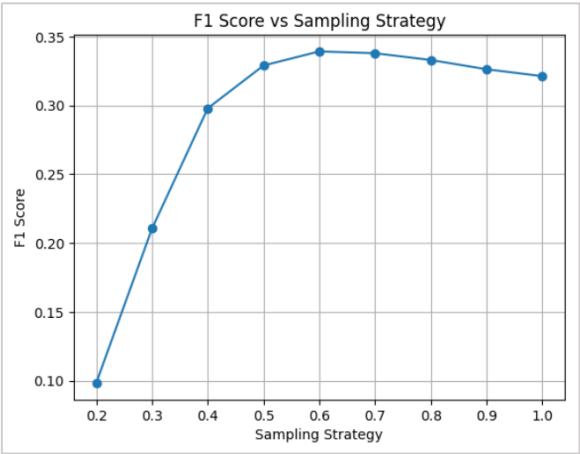


Fig. 13A – Feature Importance using Lasso regression (Binary case)



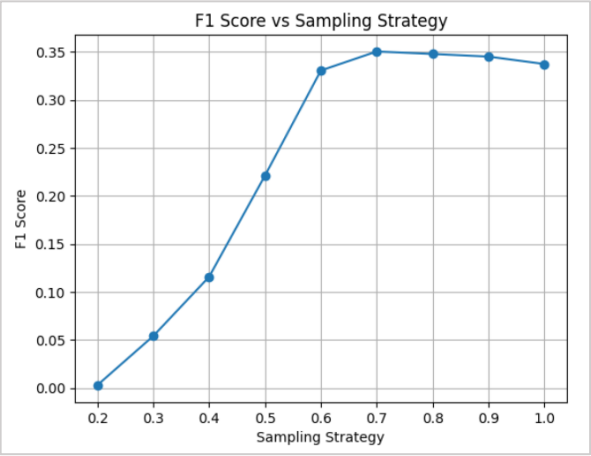


Fig. 14A – F1 score for Random Undersampling on MLP (Binary case) *(left)*

Fig. 15A – F1 score for Random Undersampling on Gradient Boosting (Binary case) *(right)*

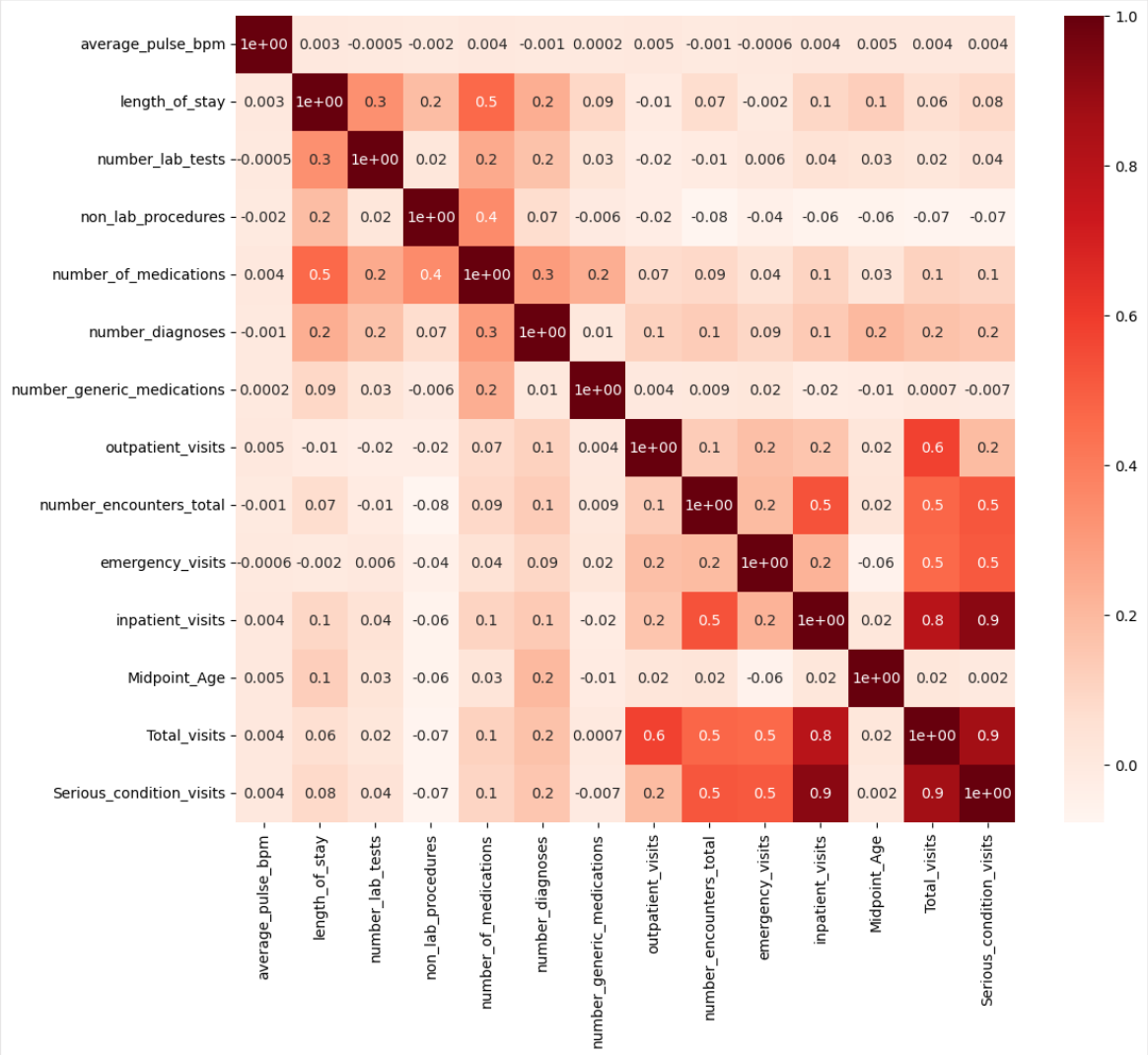


Fig 16A – Spearman Correlation Matrix (Multiclass case)

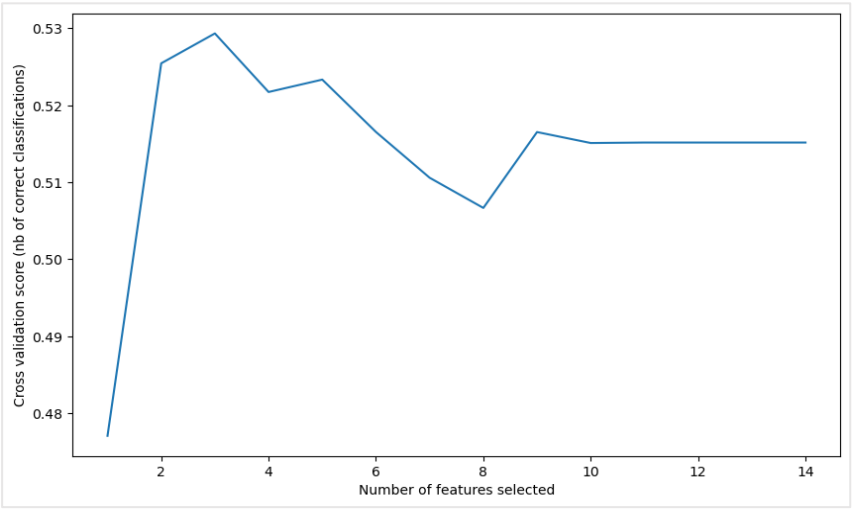


Fig. 17A – Recursive Feature Elimination (RFE) with Decision Trees (Multiclass case)

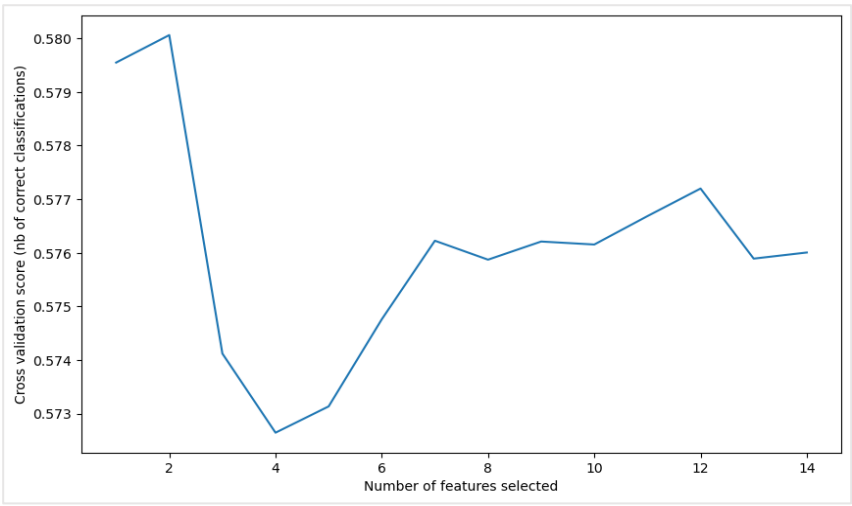


Fig. 18A - Recursive Feature Elimination (RFE) with Logistic Regression (Multiclass case)

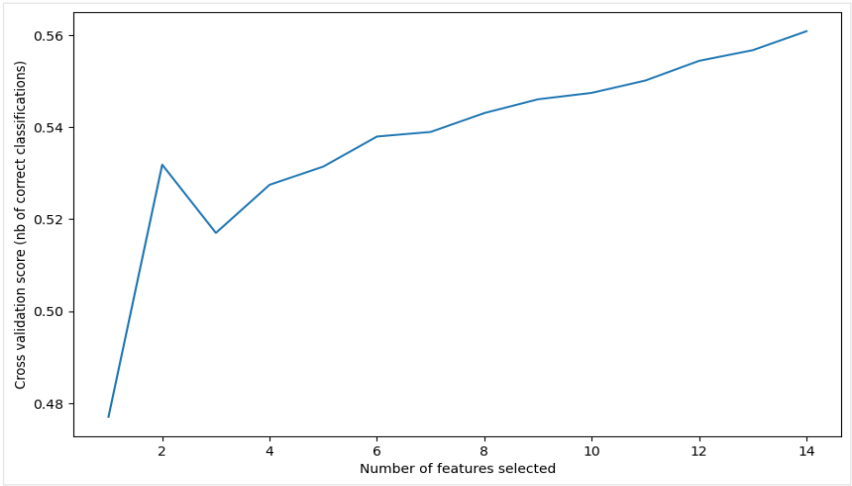


Fig. 19A - Recursive Feature Elimination (RFE) with Random Forest (Multiclass case)

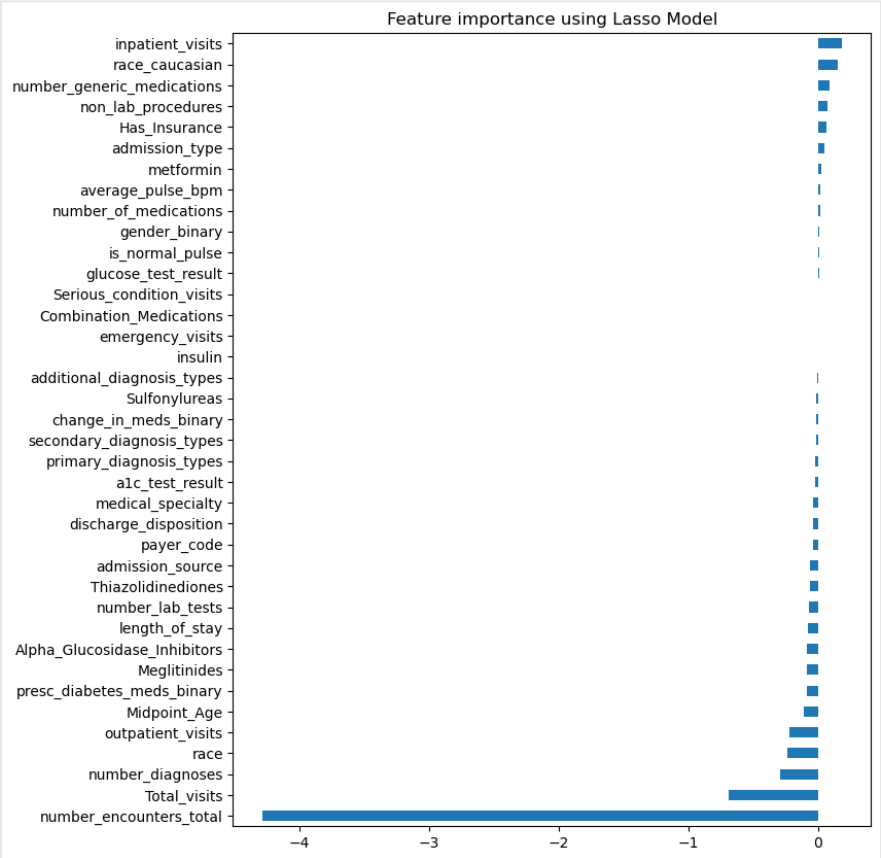


Fig. 20A – Feature Importance using Lasso regression (Multiclass case)

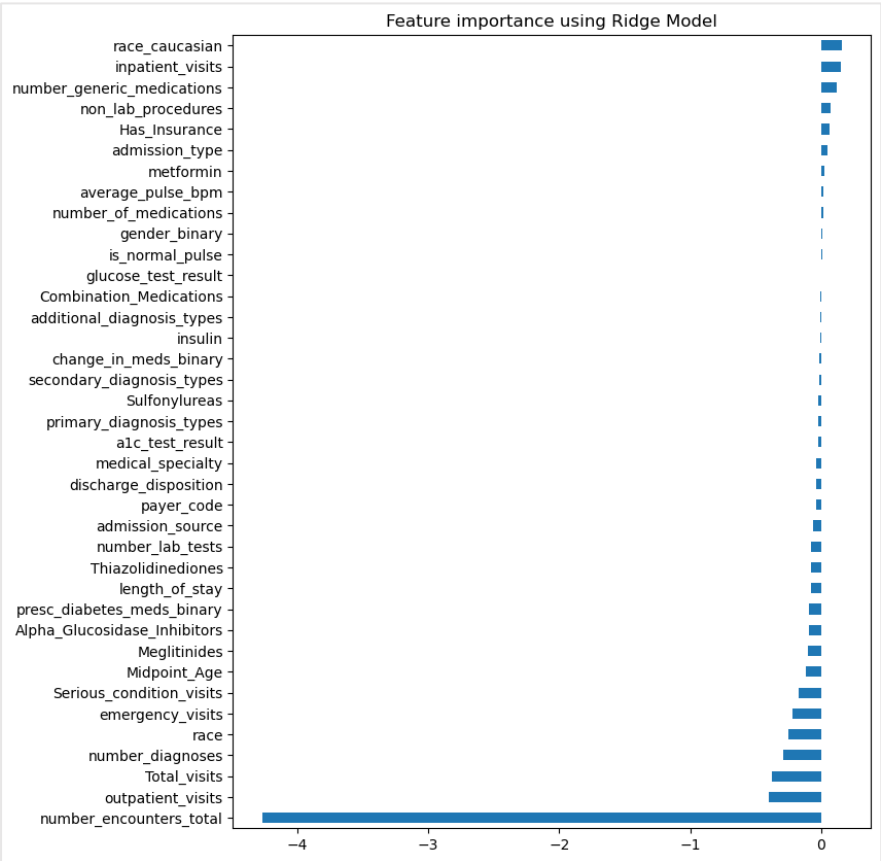


Fig. 21A – Feature Importance using Ridge regression (Multiclass case)

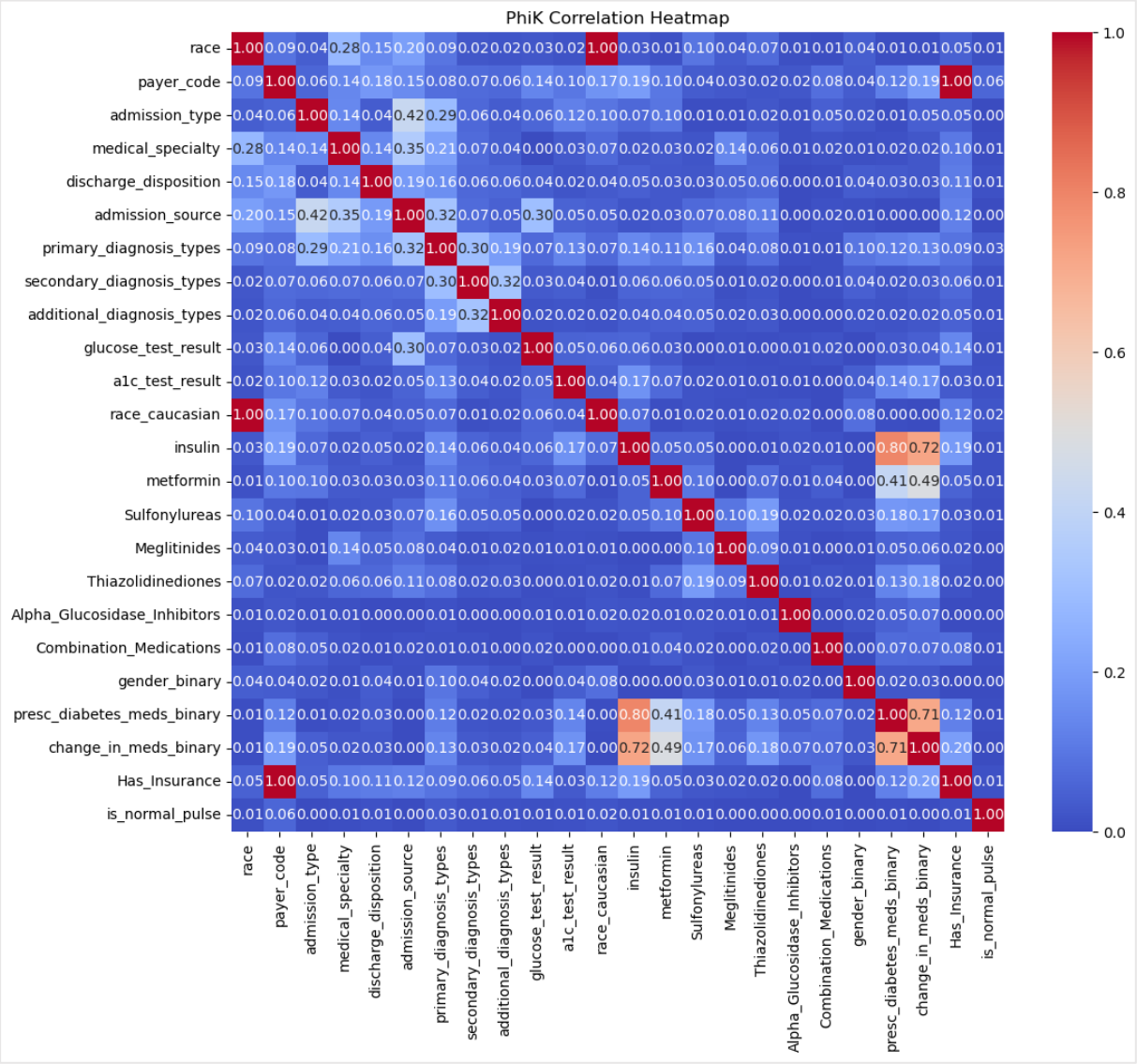


Fig. 22A – Phik Correlation Matrix (Multiclass case)

Numerical Data

Predictor	Spearman	RFE LR	RFE DT	RFE RF	SFS LR	SFS RF	Lasso	Ridge	Mutual Info	ANova	What to do?
Outpatient_visits	Keep	Keep	Discard	Keep	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Emergency_visits	Keep	Keep	Discard	Keep	Discard	Keep	Discard	Keep	Keep	Keep	Keep
Inpatient_visits	Keep?	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep
Average_pulse_bpm	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Discard	Discard
Length_of_stay	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Number_lab_tests	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Non_lab_procedures	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Number_of_medications	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Number_diagnoses	Keep	Discard	Keep	Keep	Discard	Keep	Keep	Keep	Keep	Keep	Keep
Number_generic_medications	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Midpoint_Age	Keep	Discard	Discard	Keep	Discard	Keep	Keep	Keep	Discard	Keep	Keep
Total_visits	Keep?	Keep	Discard	Keep	Discard	Keep	Keep	Keep	Keep	Keep	Discard
Serious_condition_visits	Keep?	Keep	Discard	Keep	Discard	Keep	Discard	Keep	Keep	Keep	Discard
Number_encounters_total	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep

Table 3A – Feature Selection overview for numerical data (Multiclass case)

Categorical Data

Predictor	Chi-Square	Phik	Lasso	Ridge	Mutual Info	Anova	What to do?
Race	Keep	Keep?	Keep	Keep	Keep	Keep	Keep
Payer_code	Keep	Keep?	Keep	Keep	Keep	Keep	Discard
Admission_type	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Medical_specialty	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Discharge_disposition	Keep	Keep	Keep	Keep	Keep	Keep	Keep
Admission_source	Keep	Keep	Keep	Keep	Keep	Discard	Keep
Primary_diagnosis_types	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Secondary_diagnosis_types	Keep	Keep	Keep	Keep	Keep	Keep	Keep
Additional_diagnosis_types	Keep	Keep	?	?	Discard	Keep	Keep
Glucose_test_result	Keep	Keep	?	?	Keep	Discard	Discard
A1c_test_result	Keep	Keep	Keep	Keep	Keep	Discard	Keep
Race_caucasian	Keep	Keep?	Keep	Keep	Keep	Discard	Discard
Insulin	Keep	Keep?	?	?	Keep	Keep	Discard
Metformin	Keep	Keep	Keep	Keep	Keep	Discard	Keep
Sulfonylureas	Keep	Keep	Keep	Keep	Discard	Discard	Keep
Meglitinides	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Thiazolidinediones	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Alpha_Glucosidase_Inhibitors	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Combination_Medications	Discard	Keep	Discard	Keep	Discard	Discard	Discard
Gender_binary	Keep	Keep	Keep	Keep	Discard	Keep	Keep
Presc_diabetes_meds_binary	Keep	Keep?	Keep	Keep	Discard	Keep	Discard
Change_in_meds_binary	Keep	Keep?	Keep	Keep	Discard	Keep	Keep
Has_Insurance	Keep	Keep	Keep?	Keep	Discard	Discard	Keep
Is_normal_pulse	Discard	Keep	?	?	Discard	Discard	Discard

Table 4A – Feature selection overview for Categorical data (Multiclass case)