NOVA IMS Information Management School

2023

Hospital Readmission Report

Group 27

Joana Macedo – r20201498

Ana Morgado – 20230469

Gonçalo Caldeirinha - 20230469

Taha Ben Attia – 20230742

Emanuele Travisani - 20231243

Course: Machine Learning

Professors: Roberto Henriques | Ricardo Santos | Rafael Pereira

NOVA IMS Information Management School

Table of Contents

NOVA IMS Information Management School

Table of Figures

Table of Tables

Abstract

Hospital readmissions can be a problem, not only in terms of the quality of care provided but also in terms of the increase in healthcare costs.

We conducted a study to build predictive models for a classification task. The goal was to create a binary classifier that accurately predicts readmission within 30 days and a multiclass classifier that categorizes readmission into three categories: 'No' (no hospital readmission), '<30 days' (readmission within less than 30 days of the last readmission), and '> 30 days' (readmission after 30 days of the last readmission).

The ability to make predictions about readmissions can enable the adoption of preventive measures that can result in large savings.

We began by preparing our data, starting with data pre-processing, where we evaluated inconsistencies and misclassifications, outliers. We then carried out feature engineering and treating the missing values and finally feature selection. To improve the results of our model, we used resampling techniques such as SmoteNC, SmoteNN, Undersampling, and Oversampling, as well as the last two combined.

We obtained results with different models: Bayes classifier, KNN, Random Forest, Decision Trees, Logistic Regression, Neural Network, and Support Vector Machine. We achieved the best results for our binary classification with Random Forest and our multi-class classification with Support Vector Machines.

1. Introduction

Hospital readmission is not only an indicator of care quality but also a significant driver of rising healthcare costs. This project is dedicated to developing two predictive models using a training dataset comprising 71236 observations. The objective of these predictive models is to provide healthcare operators with tools that can improve the quality of patient care, reduce hospitalization rates, contribute to more sustainable healthcare spending, offer detailed insights into patient risk levels, and help hospitals customize their post-discharge care strategies.

The datasets primarily consist of patient features, such as gender, age, and weight, as well as clinical and hospital information, including the number of lab tests performed during the encounter, primary, secondary, and any additional diagnoses, and whether the patient has been prescribed diabetes medication, among other variables.

Numerous studies have explored the use of various machine learning models, such as the one conducted by BMC Medical Informatics and Decision Making, in which machine learning models were constructed to predict readmission risks among diabetic patients.

In total, three machine learning algorithms, specifically Random Forest, Naive Bayes, and Decision Tree Ensemble were employed to enhance the clinical efficiency of classification. The study suggests that the factors influencing 30-day readmission predictions in diabetic patients include the number of inpatient admissions, age, diagnosis, and gender. Consequently, these results lead us to anticipate similar predictors in our model.

To navigate through the various stages of our project, we used the CRISP-DM methodology (Cross Industry Standard Process for Data Mining) which consists of six different phases as shown in Figure 7. As a way of mapping our progress throughout the report, we will make references to each of these phases. This way and as a conclusion to the Introduction, we can say that we have concluded the Business Understanding Phase.

2. Data Exploration and Preprocessing

2.1 First Exploration

The first step in our project was to explore the data. This corresponds to the Data Understanding phase of the CRISP-DM model (Figure 7) which can help us identify potential problems. We started by importing all the libraries needed for the analysis, followed by an overall review of the first 5 rows of our data, where we identified some issues that are going to be addressed soon. We looked at how many rows and columns the dataset had and set *'encounter_id'* as the index. In addition, we observed the data types and missing values of our variables.

2.2 Statistical Information

The next step was to collect some statistical information about our variables. The numerical variables (Table 8) showed that both the 75th percentile of *'outpatient_visits_in_previous_year'* and *'emergency_visits_in_previous_year'* are not above zero, while the maximum number of visits is 42 and 76 respectively, which means that most people make very few visits to the hospital. Analyzing the maximum values of *'number_lab_tests'* and *'number_of_medications'*, we obtain 121 and 75, but only 25% of patients exceed 57 and 20, respectively. This initial analysis allows us to hypothesize that these values are outliers. The summary statistics for categorical variables (Table 9) show the number of observations, the number of unique values, and the most common value. In this analysis, we found that the variable *'country'* has only one value, namely 'USA'. Since this variable does not contribute anything to our analysis, we will remove it from our dataset as well as the test dataset.

Regarding the variable *'medication'*, we find that 'insulin' is the most common medication among the patients with about 30%, while about 77% have a doctor's medical prescription for diabetes. This could confirm the contribution of this particular condition to readmissions. We also considered that *'weight'* would not be relevant for our analysis since it contained too many missing values (98%) which made it unreasonable for us to impute. Taking this into consideration we decided to remove this variable from our analysis.

To ease our data pre-processing phase, we decided it would be beneficial to separate our dataset into metric and non-metric features, allowing us to treat separately each type of variable.

2.3 Train and Validation Split

After this initial analysis and before starting to input any values, we had to divide our dataset into training and validation. This is done to prevent data leakage and overfitting of our model. The validation set will also be useful for us to check the performance of our model during training. There are several ways to do this, such as train_test_split, where we set the desired ratio of train and validation, in this case, 30% for the validation set and 70% for the train. In addition, we have defined the stratified method to maintain the same proportion of 0s and 1s as in the original data set. We also set random_state to 42, which initializes the seed for the random function used to split the dataset.

2.4 Incoherences and Misclassifications

Then we entered the 'Data Preparation' phase of our project where we will address the problems observed before as well as potential new ones. By thoroughly examining for inconsistencies and misclassifications and rectifying them, we can significantly enhance the accuracy of our results. After analyzing the dataset, we noticed some misclassifications in the observations, so we proceeded to correct them (Table 10). The differences in some of our decisions for each variable are due to the variable itself. For instance, the omission of the variable *'payer_code'* does not mean that there is a variable missing for that entry, it might just mean that the patient corresponding to that entry did not have any insurance plan. The same logic goes to the variables *'secondary_diagnosis'* and *'additional_diagnosis'* (the patient might not have had more than one diagnosis), *'glucose_test_result'* and *'a1c_test_result'* (the patient might not have been tested for those components), *'medication'* (the patient might not have been prescribed with any medication). On the other hand, we considered that for the variable *'gender'* and in terms of the medical scope it would not have made sense to have an option other than *'female'* and *'male'*.

After this, we just had to take care of the leading spaces present in the variables *'admission_source'*. Regarding inconsistencies, we checked whether the *'primary_diagnosis'* of a patient matched the *'secondary_diagnosis'*, if so, we replaced the second with the *'additional_diagnosis'* and set the latter to 'Not Available'. Secondly, we used the same logic to check whether the *'primary_diagnosis'* or the *'secondary_diagnosis'* was the same as the *'additional_diagnosis'* and different from 'Not available', and if so, we deleted the *'additional_diagnosis'* value and set it to 'Not available'.

2.5 Duplicates

We searched for duplicate values and as expected, we noticed that we have *'patient_id'* repeated 9600 times, we will proceed with this variable for further analysis. Other than that, there were no duplicate entries in our dataset.

2.6 Outliers

Outliers were the second problem we had to deal with, as these are observations that differ significantly from the other observations in the dataset. They can lead to very biased results and mislead our models. We created histograms (Figure 8) and box plots (Figure 9) to better visualize these problems. After seeing this and analyzing some of the variables individually, we manually deleted some of the outliers in 6 of our variables. It was found that 2.09% of the observations would be deleted.

To achieve a more accurate removal of outliers, we decided to combine manual deletion with the interquartile method, where the quartiles and interquartile's were calculated, the upper and lower limits were defined and finally the filters for the removal of outliers were created. The combination of the two methods resulted in a removal of 1.19% of the observations, so we decided to delete them using the combination of the two methods.

2. Feature Engineering

We then found that the variable *'discharge_disposition'* included patients who had already died, 'expired'. As these patients do not contribute with any information to our models and we know that they will not be readmitted to the hospital, we decided to delete them from our train dataset. With this, we did the same for validation but stored these in a new data frame that contained only these patients. For our test dataset, we created a new data frame called 'results' that will store our final results, taking into consideration the patients that are 'expired', administering values of 0 on the *'readmitted_binary'* column and leaving the other empty.

2.1 Dummy Variables

As a first method of encoding, we set our binary variables to 1s and 0s. We did this to convert our categorical variables into dummy variables since most models only work with numeric data, and this allows the model to read the data and perform calculations and comparisons. The variables with the values 'yes' and 'no' were set to 1 and 0 respectively, namely the variables *'change_in_meds_during_hospitalization'*, *'prescribed_diabetes_meds'* and *'readmitted_binary'*. For the variable *'gender'* we decided to set 'Female' to 1 and 'Male' to 0.

2.2 Creating New Variables

This is a crucial aspect of our analysis, since by creating new features or modifying the existing ones we are enhancing our model's performance and ability to capture meaningful patterns. This will also help us reduce dimensionality in the data.

New Variable	Method of Creation
insurance	1, if the patient has a payer_code, 0 otherwise
caucasian	1, if the patient is caucasian, 0 otherwise
diagnosis_count	count of all diagnosis, primary, secondary and additional
diabetes_type	Type I diabetes if the patient takes only insulin, Unknown if it has no medication, Type II otherwise*
total_procedures	number of the lab and non-lab tests
total_visits	sum of all visits, either outpatient, inpatient or emergency
encounter_count	sum of each time the same patient had a different encounter
avg_length_of_stay	average amount of days the same patient spent in the hospital
total_visits_pp	sum of total_visits by patient

Table 1 - New Variables

*This variable was created taking into consideration the information provided on medications for type 2 diabetes (Healthline, n.d.).

Grouping variables into categories was also an approach done, to simplify and improve decision-making. We created a new variable called *'age_group'* where we grouped the variable *'age'* into categories, called 'Child', 'Teen', 'Young Adult', 'Adult' and 'Elderly' (Table 11). We also grouped the three variables *'primary_diagnosis'*, *'secondary_diagnosis'*, and *'additional_diagnosis'* into categories according to the ICD-9 code set [1]. The next group of categories we made was regarding *'medical_specialty'*, following the information provided on (*Specialty Profiles | Careers in Medicine*, n.d.) [3], (Table 12). Having done this, we decided to remove the variables used to build the new ones since they would cause redundancy. Following this we renamed some variables for simplicity.

3. Visualizations

To gain a deeper understanding of the data, we created various graphs and visualizations for our machine-learning project. First, we decided to create a bar chart to capture and quantify the readmissions of patients according to their gender. In the graph on the right (Figure 2), we see that for both genders the predominance is of not being readmitted to the hospital within 30 days. For both, being readmitted and not being readmitted, we can see that there are slightly more males than females. In the next visualization (Figure 1), we have shown the distribution of diagnosis types of patients, distinguishing between three different diagnosis categories. The most common diagnoses are nutritional and metabolic diseases (240-279) and diseases of the circulatory system (390-459).

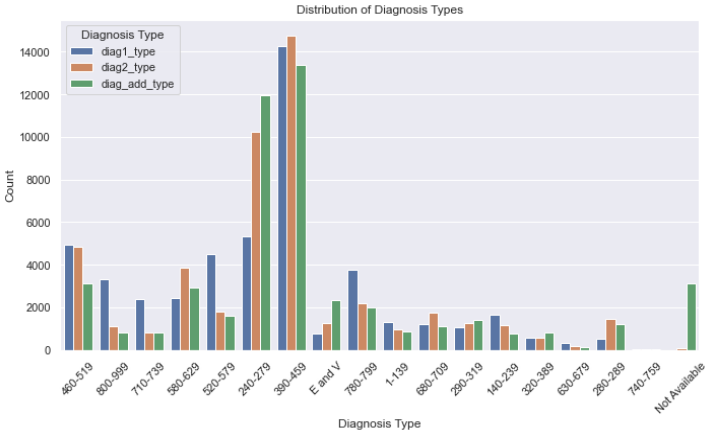

Figure 1 - Count of Diagnosis Types

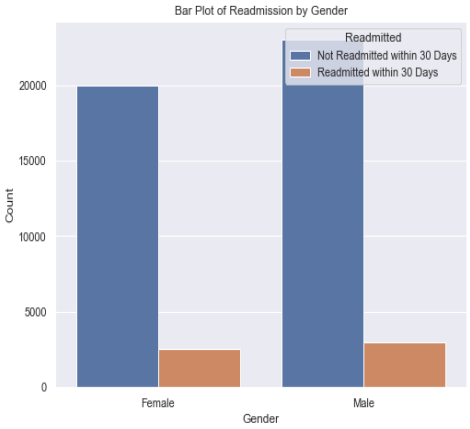

Figure 2 - Readmission by Gender

In Figure 3, we have used boxplots to illustrate and compare the duration of hospital stays in the different age groups. The boxplot on the far left clearly shows that the mean duration of hospitalization is higher for older people than for other age groups. In addition, 25% of older patients have a length of stay of more than 6 days. We used violin plots as a further visualization method. These plots combine a boxplot with a kernel density plot and highlight peaks in the data distribution. In Figure 4, a violin plot was created for the variables *'caucasian'* and *'number of medications'*. The shape of the distribution, which is narrow at both ends and wide in the middle, shows that the values for the number of medications are predominantly concentrated around the median for both caucasian and not caucasians.

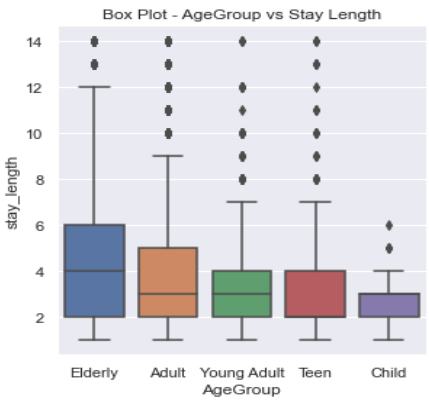

Figure 3 - Age Group vs Stay Length

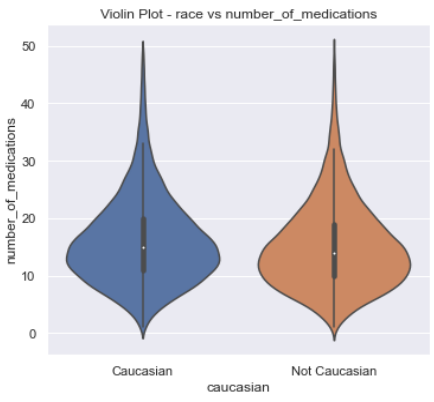

Figure 4 - Race vs Number of Medications

4. Encoding & Scaling

For the reasons already mentioned, we had to encode our non-binary variables. Our final decision was to use ordinal encoding for the variables that have a significant order, such as *'age_group'*, *'a1c'*, *'glucose'*, and *'age'*. For the remaining variables, we used target coding, where the values are replaced by the mean of our target variable, this was preferred to the one-hot due to its complexity. We opted for MinMax scaling to ensure that all our variables are within a certain range, in this case between 0 and 1 and that the relationship to the original data points is preserved. We did this for all our variables except the binary variables, as these already follow a distribution between 0 and 1.

5. Missing Values

Missing values are an important issue in our data analysis as they can lead to loss of information, skewed results, and even false conclusions. The variables with missing values, *'race'*, *'gender'*, *'age'*, *'discharge_disposition'*, *'age_group'*, *and 'primary_diagnosis'* were retained as we considered them important for the model. After careful analysis and attempting to use the mode and knn imputation methods, we opted for the latter as it preserves the underlying structure of the data by considering the relationships between features, and considers multiple features to estimate missing values, which can lead to more accurate results.

6. Resampling

We found that our dataset had a very imbalanced target variable, where our majority class (0) had 88.7% of the values and our minority class (1) had 11.30%. This can affect the performance of our machine learning models and lead to a biased model and incorrect assumptions. To solve this problem, we used a resampling technique. After trying different resampling techniques (SMOTE-NC, SMOTEENN, SMOTETomek, Undersampling, Oversampling, and both together), we decided to use Undersampling as it significantly reduces the training time, it balances our class distributions and most importantly improves the performance of our model.

7. Feature Selection

We have used various methods to reduce the dimensionality of our features and retain only the most important ones. For the filtering methods, which evaluate the features based on their intrinsic properties, we used Spearman correlation for our numeric variables (Figure 10), followed by the chi-square method for our categorical variables, which tests the association between each feature and the target variables and selects the subset of features that contribute significantly to the predictive performance of a model. For the wrapper methods that evaluate subsets against a model, we used Recursive Feature Elimination (RFE) with logistic regression, a feature selection technique that recursively fits a logistic regression model, ranking and eliminating features based on their effects to optimize prediction accuracy. The Least Absolute Shrinkage and Selection Operator (LASSO) was also used to penalize the absolute size of the coefficients by shrinking the less important features to zero and excluding them from the model.

The next method was K-Best, which selects the K most relevant features based on statistical tests, and finally, Random Forest, which evaluates the importance of the features during tree-based ensemble training. In Table 13, we can see our decision table, we started with 35 variables and after feature selection, we kept 16 for our readmitted binary classification. For the multiclass, we kept 32 variables (Table 14).

8. Results Binary Classification

When all the necessary steps were finally completed, we were able to proceed with the Modelling phase of our project. We used 7 models to help us predict the behavior of patient readmissions. Random Forest, Support Vector Machines, and Logistic Regression were expected to perform better on our data, once this one is very imbalanced and has complex relationships. K-Nearest Neighbours, Naïve Bayes, Decision Trees, and Neural Networks were expected to struggle with our type of data. We started by tuning the hyperparameters of each model using random search, which is more effective than grid search in high-dimensional spaces with many hyperparameters, once random search explores the space of hyperparameters randomly, although it does not find the most optimal solutions. We did pay attention to the models where the parameter *class_weight='balanced'* existed, allowing the model to handle our imbalanced data. After finding an initial approach for preprocessing and rebalancing (removing outliers, using undersampling, MinMax as a baseline, and imputing missing values with the Knn imputer), we started with that (Table 2) and then moved on to some modifications. We tried a different method of scaling by using standard scaling instead of MinMax, which did not improve our results (Table 3), we tried to implement the models in a dataset where the missing values were imputed with the mode (Table 4), we kept knn imputer for the reasons already mentioned. We tried a different method of resampling the data using SMOTE-NC (Table 5). We found that the difference between train and validation was smaller when using SMOTENC with decision trees, but once the values on the train are higher with undersampling, it means that the model performs better in this case, and since the difference is not very large, we chose this method as our final resampling technique.

KNN	Train	Validation	Diference
Bayes	0.459	0.263	0.196
KNN	0.645	0.274	0.371
Decision Trees	0.31	0.281	0.029
Logistic Regression	0.317	0.287	0.03
Neural Networks	0.668	0.319	0.349
SVM	0.32	0.297	0.023
Random Forest	0.323	0.307	0.016

Table 2 - Results KNN

Standard Scaling	Train	Validation	Diference
Bayes	0.454	0.263	0.191
KNN	0.631	0.264	0.367
Decision Trees	0.317	0.282	0.035
Logistic Regression	0.305	0.284	0.021
Neural Networks	0.631	0.309	0.322
SVM	0.32	0.297	0.023
Random Forest	0.371	0.317	0.054

Table 3 - Results Standard Scaling

Mode	Train	Validation	Difference
Bayes	0.482	0.273	0.209
KNN	0.636	0.269	0.367
Decision Trees	0.31	0.281	0.029
Logistic Regression	0.31	0.28	0.03
Neural Networks	0.622	0.282	0.34
SVM	0.32	0.297	0.023
Random Forest	0.324	0.304	0.02

Table 4 - Results Mode

SmoteNC	Train	Validation	Diference
Bayes	0.468	0.245	0.223
KNN	0.885	0.217	0.668
Decision Trees	0.275	0.26	0.015
Logistic Regression	0.3	0.276	0.024
Neural Networks	0.776	0.27	0.506
SVM	0.279	0.249	0.03
Random Forest	0.36	0.305	0.055

Table 5 - Results Smote NC

Finally, with the scaling, resampling, and method of imputation of missing values chosen, we tried not removing any outliers (Table 6). We did not stick to this approach once our models can be very sensitive to these and lead to an adverse performance on them.

Without Removing Outliers	Train	Validation	Diference
Bayes	0.449	0.259	0.19
KNN	0.648	0.253	0.395
Decision Trees	0.324	0.285	0.039
Logistic Regression	0.316	0.286	0.03
Neural Networks	0.692	0.3	0.392
SVM	0.322	0.294	0.028
Random Forest	0.326	0.306	0.02

Table 6 - Results Without Removing Outliers

Additionally, we plotted the Receiver Operating Characteristic (ROC) curve to further evaluate our models, based on the true positive rates and the false positive rates (Figure 5). Meaning, the rate of correct predictions on readmissions to the hospital and the rate of wrong predictions of readmissions. Here we can see the model's performance across different cutoff points for classifying predictions on readmissions and no readmission. Neural Network gives us the best value of Area Under the Curve, meaning overall this one has the best discrimination ability compared with others.

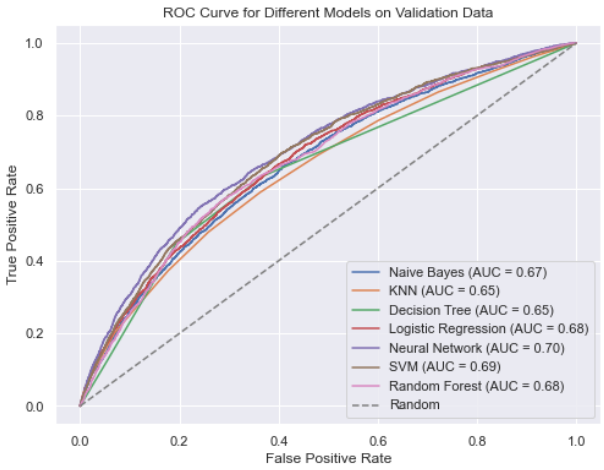

Figure 5 - ROC curve

Our final choice of model was Random Forest, as it not only provided the best results for train and validation but also offered several advantages that we considered important. As an ensemble model, Random Forest combines the predictions of different models, in this case, decision trees, which in many cases leads to better performance than a single model. This happens because the prediction errors of the different models cancel each other out, resulting in a more reliable final prediction. In addition, the random forest model can also handle linear and non-linear relationships, which was relevant to our project. After further research on similar problems also dealing with healthcare and hospitalization, we were even more confident with our choice, as the Random Forest model has been used as the final prediction model in many different studies. Some of the reasons we gathered for these choices were not only its results, but also the model's capacity to handle high-dimensional data and its techniques to reduce overfitting and select the features with the most importance.

9. Results Multiclass Classification

For our multiclass classification, where we wanted to predict whether a patient would be readmitted within less than 30 days, more than 30 days, or not at all, we first performed the same steps as for our binary classification, including preprocessing and feature engineering. We adapted a visualization to the new context, which you can see in Figure 6, where we again have two predominant classes, not readmitted and readmitted within more than 30 days, compared to readmitted within less than 30 days. We can also see that men always have a slightly higher score than women.

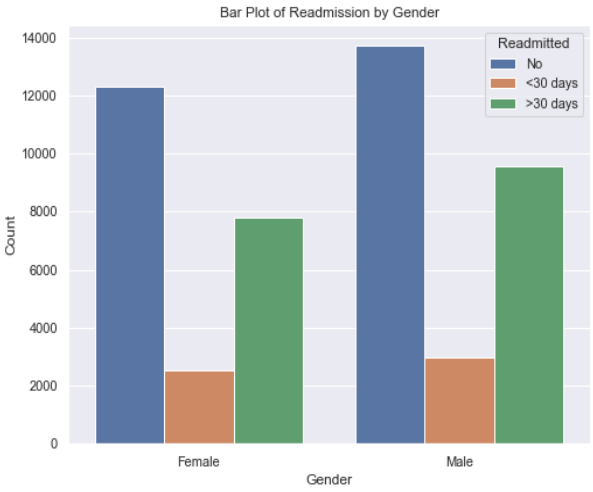

Figure 6 - Multiclass Readmission By Gender

Regarding encoding our target variable, we found that using the label encoder, which gives categories a numerical order without considering an inherent order among categories, is the best option. Another small but very important modification we had to do was regarding the f1-score, we added the parameter *average='weighted'*, which calculates the f1-score for each of our classes, and then computes the weighted average based on the number of true instances for each class, this will help balance our classes. Additionally, on the Logistic Regression, we used the parameter *multi_class= 'multinomial'*, for the model to be able to handle classification with more than one class. By default, this model uses the parameter 'ovr' (One-vs-Rest), we tried using both. After making similar attempts as the ones done in the binary classification, we concluded that by using the same baseline as the binary classification we had our best results. This included using undersampling, knn imputation, and MinMax scaling.

After having all our results, (Table 7), we can see that with SVM we reach our highest values on training and validation but have a slightly bigger difference than some others, this can indicate that the model is performing well on the training data but worse on our unseen data. Now for KNN where we have the smallest difference between train and validation but very low values on both, this can indicate that the model is not complex enough to capture all patterns of the data.

UnderSampling	Train	Validation	Diference
Bayes	0.4	0.5	-0.1
KNN	0.47	0.471	-0.001
Decision Trees	0.551	0.541	0.01
Logistic Regression	0.576	0.532	0.044
Neural Networks	0.569	0.571	-0.002
SVM	0.628	0.649	-0.021
Random Forest	0.93	0.568	0.362

Table 7 – Multiclass Results using UnderSampling

We chose Support Vector Machines since it works well with multiple classes using the 'One-vs-Rest', where the model is trained for each class considering one versus all the others. It can handle well our overfitting using the parameter *class_weight='balanced'* and it has robustness with high dimensional data, this is useful since our data handles medical data which contains many attributes.

10.Conclusion

Our aim in this project was to predict whether a patient would be readmitted to hospital within 30 days of discharge and, on the other hand, whether a patient would not be readmitted, whether they would be readmitted within less than 30 days or more than 30 days. The data provided contained some challenges for our data analysis, not only insignificant features for the analysis as well as many misclassifications within the same feature, but the main problem we encountered was the unbalanced target, not only in the binary classification but also in the multiclass. We embarked on the analysis with the aim of finding out how these features contribute to the predictions of our target.

After carefully analysing and cleaning the data – dealing with outliers, inconsistencies, misclassifications, missing values, creating new variables, and resampling - we were able to make our predictions using multiple models. We tried several attempts with different methods of preprocessing - filling missing values with mode and knn imputer, removing or not outliers – using different scaling methods, standard and MinMax, and trying different resampling techniques and changing the hyperparameters of the models, we reached the best solution.

With Random Forest, we obtained the best results for binary classification, but the models did not perform as well as expected, which leads us to conclude that the model's performance will not be good enough when the given data is hard to manage, or the preprocessing stage was poorly performed. For our multiclass classification, we chose Support Vector Machines where we saw that our model performs better than binary classification. This could be because the model has more classes to learn from and can better capture the specific patterns associated with them. The multiclass also has a more balanced target variable than the binary variable, which may also be an important factor in this difference.

11.Limitations

As for limitations, we were confronted with some. In the results of our models in the readmitted binary, we were surprised by the large amount of overfitting, which means that our model learns the training data too well, but fails on the unseen data, making it difficult to achieve similar results on both. This means that while we have gotten a handle on overfitting, our model may not capture all patterns well enough, and we have many constraints that make it difficult for our model to learn. The model predicts the majority class well but has problems with the minority class. In the final model we had 0.94 of precision at 0s and 0.21 at 1s. These problems may be related to the unbalanced features of our dataset, which are common in healthcare datasets, with a majority class of patients who are not readmitted and a minority class of patients who are readmitted. Another limitation concerned the use of GridSearch, which can be computationally intensive and take a long time to run, partly due to our very complex models and the size of the parameters given.

12.Figures

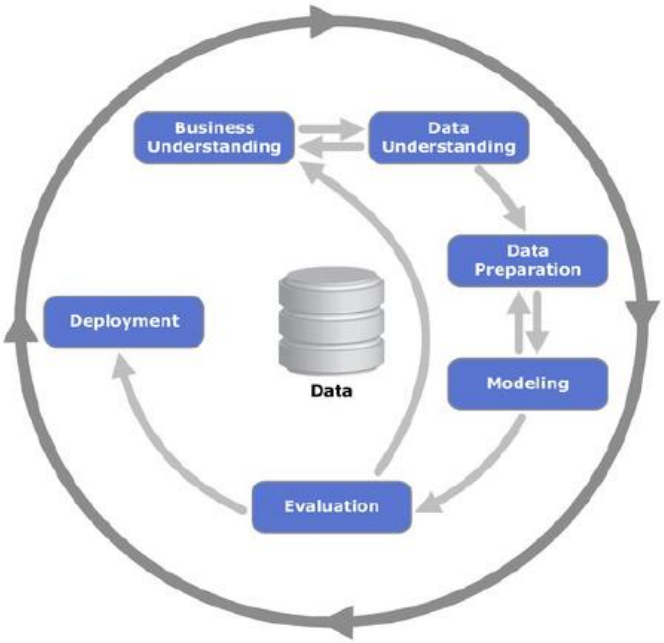

Figure 7 - CRISP-DM

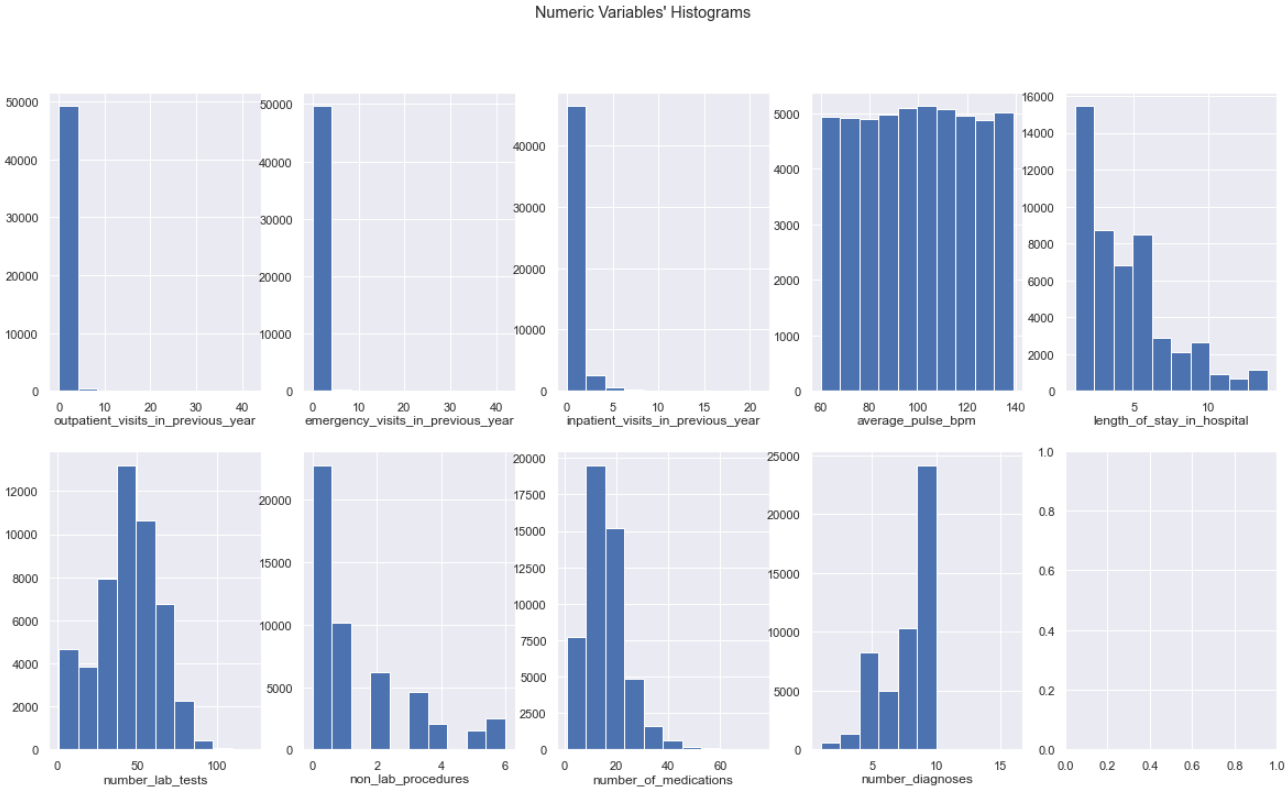

Figure 8 - Histograms

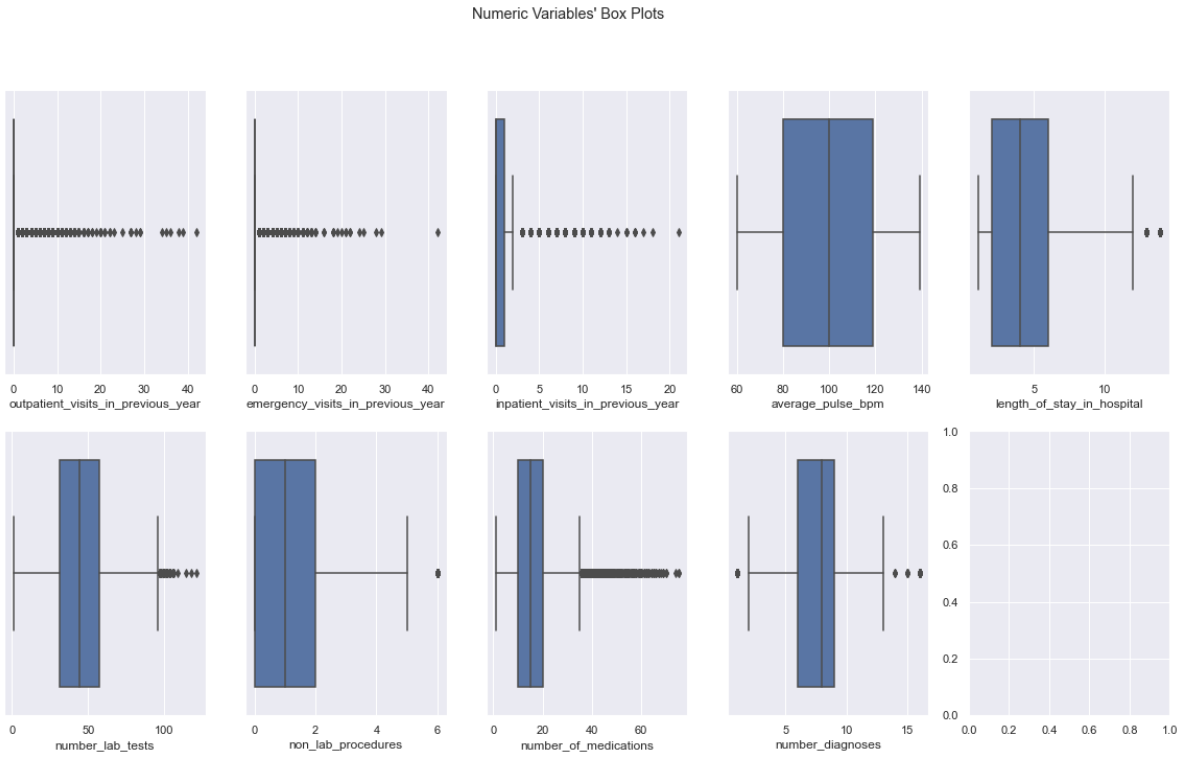

Figure 9 - Box Plots

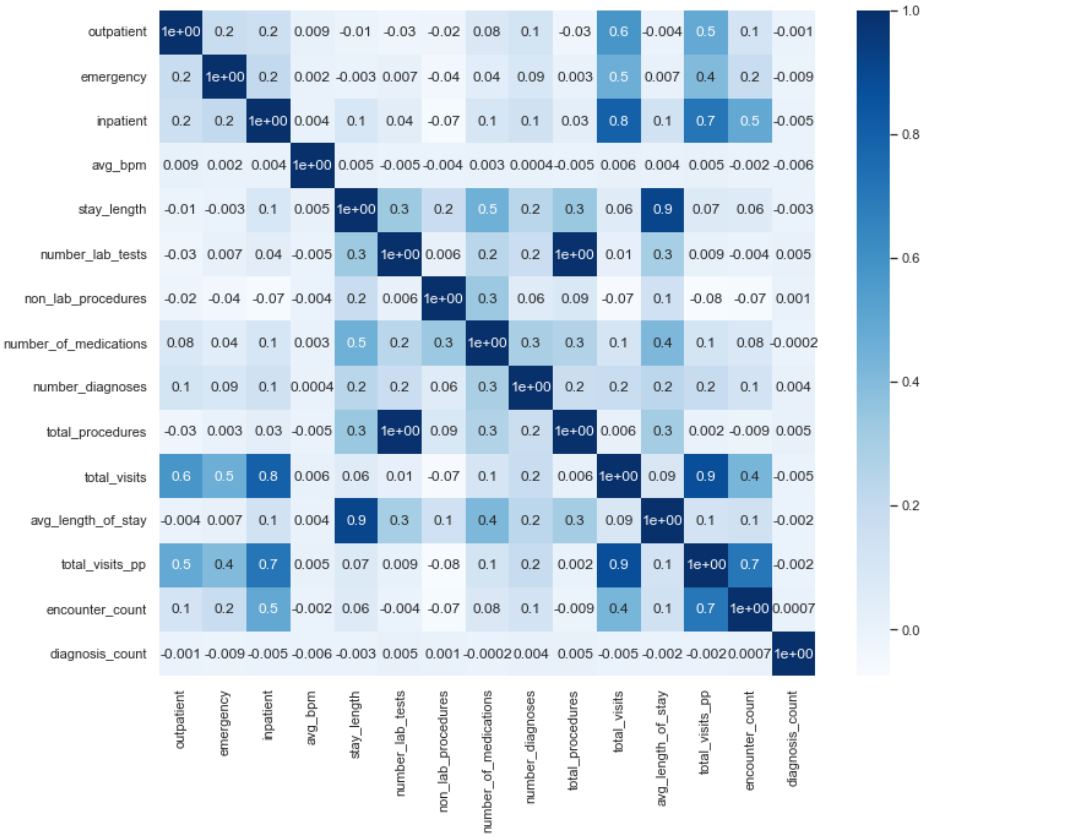

Figure 10 - Spearman Correlation

13.Tables

	count	mean	std	min	25%	50%	75%	max
patient_id	71236.0	5.430228e+07	3.879585e+07	135.0	23396510.25	45305631.0	87558374.25	189502619.0
outpatient_visits_in_previous_year	71236.0	3.695884e-01	1.287469e+00	0.0	0.00	0.0	0.00	42.0
emergency_visits_in_previous_year	71236.0	1.962491e-01	9.108537e-01	0.0	0.00	0.0	0.00	76.0
inpatient_visits_in_previous_year	71236.0	6.401539e-01	1.267271e+00	0.0	0.00	0.0	1.00	21.0
average_pulse_bpm	71236.0	9.961122e+01	2.304052e+01	60.0	80.00	100.0	119.00	139.0
length_of_stay_in_hospital	71236.0	4.391024e+00	2.988739e+00	1.0	2.00	4.0	6.00	14.0
number_lab_tests	71236.0	4.309565e+01	1.964292e+01	1.0	31.00	44.0	57.00	121.0
non_lab_procedures	71236.0	1.340923e+00	1.706664e+00	0.0	0.00	1.0	2.00	6.0
number_of_medications	71236.0	1.599545e+01	8.122347e+00	1.0	10.00	15.0	20.00	75.0
number_diagnoses	71236.0	7.421023e+00	1.937809e+00	1.0	6.00	8.0	9.00	16.0

Table 8 – Numeric Variables Data Description

	count	unique	top	freq
country	71236	1	USA	71236
race	67682	6	Caucasian	50693
gender	71236	3	Female	38228
age	67679	10	[70-80)	17359
weight	71236	10	?	68990
payer_code	71236	18	?	28201
admission_type	67530	7	Emergency	37742
medical_specialty	71236	69	?	34922
discharge_disposition	68646	25	Discharged to home	42256
admission_source	66518	16	Emergency Room	40319
primary_diagnosis	71236	687	428	4776
secondary_diagnosis	71236	699	276	4694
additional_diagnosis	71236	747	250	8070
glucose_test_result	3688	3	Norm	1806
a1c_test_result	11916	3	>8	5705
change_in_meds_during_hospitalization	71236	2	No	38326
prescribed_diabetes_meds	71236	2	Yes	54890
medication	71236	303	['insulin']	21715
readmitted_binary	71236	2	No	63286
readmitted_multiclass	71236	3	No	38405

Table 9 – Categorical Variables Data Description

Variable	Old Value	New Value
whole dataset	?	NaN
gender	Unknown/Invalid	NaN
admission_type, admission_source	Not Mapped	Not Available
admission_type, admission_source	NaN	Not Available
glucose_test_result, a1c_test_result	NaN	Not Available
medication	[]	No medication
change_in_meds_during_hospitalization	Ch	Yes
medical_specialty	NaN	Unknown
payer_code	NaN	Not Insured
secondary_diagnosis, additional_diagnosis	NaN	Not Available

Table 10 – Treating Data Incoherences

Age Group	Age Category
[0-10)	Child
[10-20)	Teen
[30-40), [40-50), [50-60)	Adult
[60-70), [70-80), [80-90), [90-100)	Eldery

Table 11 – Regrouping the Categories from 'age'

Sub-category	Category
Surgeon , Surgery-Cardiovascular , Surgery-Cardiovascular/Thoracic , Surgery-Colon&Rectal , Surgery-General , Surgery-Maxillofacial , Surgery-Neuro , Surgery-Pediatric , Surgery-Plastic , Surgery-Thoracic , Surgery-Vascular , SurgicalSpecialty	GeneralSurgery
Cardiology , DCPTEAM , Endocrinology , Endocrinology-Metabolism , Gastroenterology , Hematology , Hematology/Oncology , Hospitalist , InfectiousDiseases , InternalMedicine , Nephrology , Oncology , Proctology , Pulmonology , Rheumatology , SportsMedicine , Urology	InternalMedicine
Gynecology , Obsterics&Gynecology-GynecologicOnco , Obstetrics , ObstetricsandGynecology	Obstetrics&Gynecology
Cardiology-Pediatric , Pediatrics , Pediatrics-AllergyandImmunology , Pediatrics-CriticalCare , Pediatrics-EmergencyMedicine , Pediatrics-Endocrinology , Pediatrics-Hematology-Oncology , Pediatrics-InfectiousDiseases , Pediatrics-Neurology , Pediatrics-Pulmonology , Psychiatry-Child/Adolescent	Pediatrics
Anesthesiology , Anesthesiology-Pediatric	Anesthesiology
Radiologist , Radiology	Diagnostic Radiology
Orthopedics , Orthopedics-Reconstructive	OrthopedicSurgery
Dentistry , Podiatry , Psychology , Resident , Speech , OutreachServices	Other
PhysicalMedicineandRehabilitation , PhysicianNotFound	PhysicalMedicineandRehabilitation

Table 12 – Regrouping the Categories from 'Medical Specialty'

Variables	Spearman	Chi-Square	RFE Logistic	Lasso	Kbest	Random Forest	% Keep	Decision
race	-	Keep	Discard	Keep	Discard	Discard	0.4	Discard
gender	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
age	-	Keep	Discard	Keep	Discard	Keep	0.6	Keep
payer_code	-	Keep	Discard	Keep	Discard	Discard	0.4	Discard
outpatient	Keep	-	Discard	Discard	Discard	Discard	0.2	Discard
emergency	Keep	-	Keep	Keep	Keep	Discard	0.8	Keep
inpatient	Keep	-	Keep	Keep	Keep	Discard	0.8	Keep
admission_type	-	Discard	Discard	Keep	Keep	Discard	0.4	Discard
medical_specialty	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
avg_bpm	Discard	-	Discard	Keep	Discard	Keep	0.4	Discard
discharge_disposition	-	Keep	Keep	Keep	Keep	Keep	1	Keep
admission_source	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
stay_length	Keep	-	Discard	Keep	Keep	Keep	0.8	Keep
number_lab_tests	Keep	-	Discard	Keep	Discard	Keep	0.6	Keep
non_lab_procedures	Keep	-	Discard	Keep	Discard	Discard	0.4	Discard
number_of_medications	Keep	-	Discard	Keep	Discard	Keep	0.6	Keep
number_diagnoses	Keep	-	Discard	Keep	Discard	Discard	0.4	Discard
glucose	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
a1c	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
meds_change	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
prescribed_diabetes_med	-	Keep	Discard	Keep	Discard	Discard	0.4	Discard
medication	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
caucasian	-	Discard	Discard	Keep	Discard	Discard	0.2	Discard
insurance	-	Discard	Discard	Keep	Keep	Discard	0.4	Discard
diagnosis_count	Discard	-	Discard	Keep	Discard	Discard	0.2	Discard
total_procedures	Discard	-	Discard	Keep	Discard	Keep	0.4	Discard
total_visits	Keep	-	Discard	Keep	Keep	Discard	0.6	Keep
age_group	-	Keep	Discard	Keep	Discard	Discard	0.4	Discard
encounter_count	Keep	-	Keep	Keep	Keep	Keep	1	Keep
avg_length_of_stay	Discard	-	Discard	Keep	Keep	Keep	0.6	Keep
total_visists_pp	Discard	-	Discard	Keep	Keep	Keep	0.6	Keep
diag1_type	-	Keep	Discard	Keep	Discard	Keep	0.6	Keep
diag2_type	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
diag_add_type	-	Keep	Discard	Keep	Discard	Keep	0.6	Keep
diabetes_type	-	Keep	Discard	Keep	Discard	Discard	0.4	Discard

Table 13 – Feature Selection Results for Binary Target

Variables	Spearman	Chi-Squar	RFE Logist	Lasso	Kbest	Random F	% Keep	Decision
race	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
gender	-	Discard	Keep	Keep	Discard	Discard	0.4	Discard
age	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
payer_cod	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
outpatient	Keep	-	Keep	Keep	Discard	Discard	0.6	Keep
emergency	Keep	-	Keep	Keep	Discard	Discard	0.6	Keep
inpatient	Keep	-	Keep	Keep	Keep	Discard	0.8	Keep
admission_	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
medical_sp	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
avg_bpm	Discard	-	Keep	Keep	Discard	Keep	0.6	Keep
discharge_	-	Keep	Keep	Keep	Keep	Keep	1	Keep
admission_	-	Keep	Keep	Keep	Keep	Discard	0.8	Keep
stay_lengt	Keep	-	Keep	Keep	Discard	Keep	0.8	Keep
number_la	Keep	-	Keep	Keep	Discard	Keep	0.8	Keep
non_lab_p	Keep	-	Keep	Keep	Discard	Keep	0.8	Keep
number_o	Keep	-	Keep	Keep	Keep	Keep	1	Keep
number_di	Keep	-	Keep	Keep	Discard	Keep	0.8	Keep
glucose	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
a1c	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
meds_cha	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
prescribed	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
medicatio	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
caucasian	-	Keep	Keep	Keep	Keep	Discard	0.8	Keep
insurance	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
diagnosis_	Discard	-	Keep	Discard	Discard	Discard	0.2	Discard
total_proc	Discard	-	Keep	Discard	Discard	Keep	0.4	Discard
total_visits	Keep	-	Keep	Keep	Keep	Discard	0.8	Keep
age_group	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep
encounter_	Keep	-	Keep	Keep	Keep	Keep	1	Keep
avg_length	Discard	-	Keep	Keep	Keep	Keep	0.8	Keep
total_visit	Discard	-	Keep	Keep	Keep	Keep	0.8	Keep
diag1_type	-	Keep	Keep	Keep	Keep	Keep	1	Keep
diag2_type	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
diag_add_	-	Keep	Keep	Keep	Discard	Keep	0.8	Keep
diabetes_t	-	Keep	Keep	Keep	Discard	Discard	0.6	Keep

Table 14 – Feature Selection Results for Multiclass Target

14. Annexes

Self Study Contents

K-best

K-Best/SelectKBest uses statistical tests to rank the features based on their relationship with the output variable. The K features with the highest scores are then selected to be included in the final set of features. In our case we used mutual_info_classif as the score function to evaluate feature importance since it is usually used for classification problems and computes mutual information between two discrete variables.

SMOTE-NC (Synthetic Minority Over-sampling Technique for Nominal and Continuous)

This is an extension of SMOTE, which is a resampling technique useful in the presence of class imbalance in the dataset, which means that there is a very different number of observations in each class. SMOTE-NC works with Datasets that have both nominal and continuous observations. The baseline approach is the same as SMOTE which generates synthetic samples (instances of the data created by the algorithm) using instances from the minority class and then uses them to identify other samples inside the class, called neighbors, which are very similar to the synthetic one. The algorithm then uses these new neighbors to create additional synthetic samples. SMOTE-NC has a different approach for categorical and continuous data. For categorical data, the generation of synthetic samples takes into consideration the distribution of categories within the class.

SMOTENN (Synthetic Minority Over-sampling Technique with Nearest Neighbors)

This case is a variation of SMOTE. The main difference is that SMOTE generates random samples without taking into consideration the distribution of the neighbors. SMOTE-NN, for each instance in the minority class, selects its K closest neighbors (where K is a given parameter) and then generates synthetic samples between the initial instance and its neighbors. This type of selection improves the diversity of synthetic samples.

SMOTETomek

Tomek Links is another technique useful in the presence of class imbalance in the dataset. Tomek Links are a pair of samples, one from the minority class and one from the majority class, but they are still very close to each other for their characteristics. Removing Tomek Links we can improve the separation between classes. Our purpose was to combine SMOTE and Tomek Links, starting with SMOTE and then removing the Tomek Links to eliminate samples that could cause ambiguity in the classification.

UnderSampling

Undersampling is a technique that removes samples from the majority class until it matches the size of the minority class. It is useful when dealing with a large data set where the majority class is well

enough represented so that its removal doesn't lead to a significant information loss. By applying this technique, we are not only able to tackle imbalanced data but also decrease the training time.

OverSampling

Oversampling is a technique that creates new observations belonging to the minority class and adds them to the dataset. By doing so, it is able to balance the distribution of classes. It is especially useful when the amount of data is limited and there is a high cost in discarding observations. On the other hand, it can also lead to overfitting.

KNN imputer

KNN imputer is used in to predict missing values of a dataset, it uses the k-nearest neighbors using the distance from the missing value to its neighbors with a value. The average of the neighbors is calculating, and it is used to predict the missing value.

Ordinal Encoding

This type of encoding preserves the order of the categories within each feature. It gives sequential numbers to each category of the feature.

Target Encoding

Target Encoder encodes the categorical variables based on values from the target feature. It is useful for high cardinality features, where there are many categories within the feature. The 'smothing' parameter can be used to avoid overfitting and balance the encoded value between the mean of the target variable and the global mean.

Label Encoding

This type of encoding attributes a numerical value to each category without considering the order they have, giving values between 0 and the number of classes minus 1.

15. References

[1] *ICD-9-CM - Medical Codes*. (n.d.). https://www.findacode.com/code-set.php?set=ICD9 , Accessed on 10/12/2023

[2] Cherney, K. (2023, March 13). *A complete list of diabetes medications*. Healthline. https://www.healthline.com/health/diabetes/medications-list#type-2-diabetes, Accessed on 10/12/2023

[3] *Specialty Profiles | Careers in Medicine*. (n.d.). Careers in Medicine. https://careersinmedicine.aamc.org/explore-options/specialty-profiles, Accessed on 10/12/2023

[4] *3.1. Cross-validation: evaluating estimator performance*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html, Accessed on 10/12/2023

[5] Couronné, R., Probst, P., & Boulesteix, A. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, *19*(1). https://doi.org/10.1186/s12859-018-2264-5, Accessed on 10/12/2023

[6] *Random Decision Forests*. (n.d.). Tin Kam Ho. http://vision.cse.psu.edu/seminars/talks/2009/random_tff/odt.pdf, Accessed on 10/12/2023

[7] Shang, Y., Jiang, K., Wang, L., Zhang, Z., Zhou, S. P., Liu, Y., Dong, J., & Wu, H. (2021). The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifiers. *BMC Medical Informatics and Decision Making*, *21*(S2). https://doi.org/10.1186/s12911-021-01423-y, Accessed on 10/12/2023

[8] *Multiclass Receiver Operating Characteristic (ROC)*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html, Accessed on 10/12/2023

[9] Pirge, G. (2021, December 25). Performance Comparison of Multi-Class Classification Algorithms. *Medium*. https://gursev-pirge.medium.com/performance-comparison-of-multi-class-classification-algorithms-606e8ba4e0ee, Accessed on 10/12/2023

[10] *6.4. Imputation of missing values*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/impute.html#knnimpute, Accessed on 10/12/2023

[11] *6.9. Transforming the prediction target (y)*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/preprocessing_targets.html#preprocessing-targets, Accessed on 10/12/2023

[12] D, K. (2023, February 16). Optimizing performance: SelectKBest for efficient feature selection in machine learning. *Medium*. https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48#a64d, Accessed on 10/12/2023

[13] *sklearn.feature_selection.SelectKBest*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html, Accessed on 10/12/2023

[14] Sole. (2023, November 3). *The role of undersampling in tackling imbalanced datasets in machine learning*. Train in Data Blog. https://www.blog.trainindata.com/undersampling-techniques-for-imbalanced-data/, Accessed on 10/12/2023

[15] Sole. (2023a, November 3). *Exploring oversampling techniques for imbalanced datasets*. Train in Data Blog. https://www.blog.trainindata.com/oversampling-techniques-for-imbalanced-data/, Accessed on 10/12/2023