Predicting Kidney Transplant Allocation in Brazil

Github repository: https://github.com/lross4/data1030_project

Lindsey Ross

Brown University

DATA 1030: Hands-On Data Science

Professor Zsom

9 December 2022

Introduction

The goal of this project is to create a machine learning (ML) model that predicts whether or not a new patient added to a kidney transplant waitlist will receive a kidney from a deceased donor. The target variable is a categorical variable called 'Transplant_Y_N', and it contains 2 classes — 'Y' indicates that the patient received a kidney transplant and 'N' indicates that the patient did not receive a kidney transplant. This project develops a machine learning pipeline and tests several different machine learning algorithms to solve this binary classification problem.

The dataset for this project was retrieved from a public repository on Kaggle. The data includes information about patients that were added to the waiting list of the São Paulo State Organ Allocation System (SP-OAS) in Brazil from 2000-2017 [1]. This project could have a significant impact on Brazil's healthcare system and the lives of patients. Maintaining one's position on a kidney transplant list requires recurrent tests and procedures, placing economic strain on the healthcare system and the patient [2]. Being able to predict whether a new patient will receive a transplant will help medical professionals prioritize resources for those in need and devise individualized treatment plans.

The dataset contains 48,153 data points and 52 features, plus the target variable 'Transplant_Y_N'. The features included in the dataset were not well-documented in Kaggle, but a detailed description of each feature was created and is stored in the document, "waitlist_data_descriptions.txt", in the GitHub repository. The features describe personal characteristics, medical history, family history, and lab results of the patients on the waitlist.

This dataset was previously used in one study from Sapiertein Silva et al., in which the researchers developed a model to predict wait time until a kidney transplant. They produced univariate and multivariate Cox regression models which showed that certain features—age, subregion, cPRA, and frequency of certain HLA measurements—were associated with changes in the likelihood of receiving a kidney transplant. The model had some predictive value as evidenced by its concordance statistic, or c-index, of 0.70 [2]. While this study provides indications of which features might be important in predicting kidney transplants, the fact that it describes a regression problem limits the ability to compare results directly to this project.

EDA

The balances of each class of the target variable were calculated. 28.5% of patients received a kidney transplant and 71.5% of patients did not. Further exploratory data analysis was performed with this target variable, such as comparing the fractions of patients who received transplants and those who did not across three age categories.

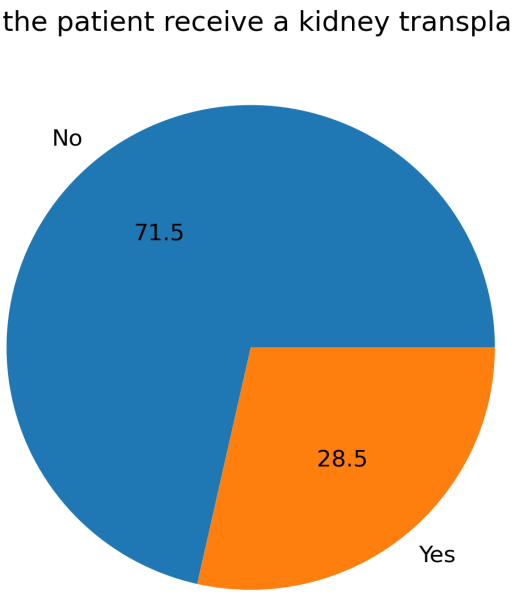

Figure 1. Visualization of the balances of the target variable, 'Transplant_Y_N'.

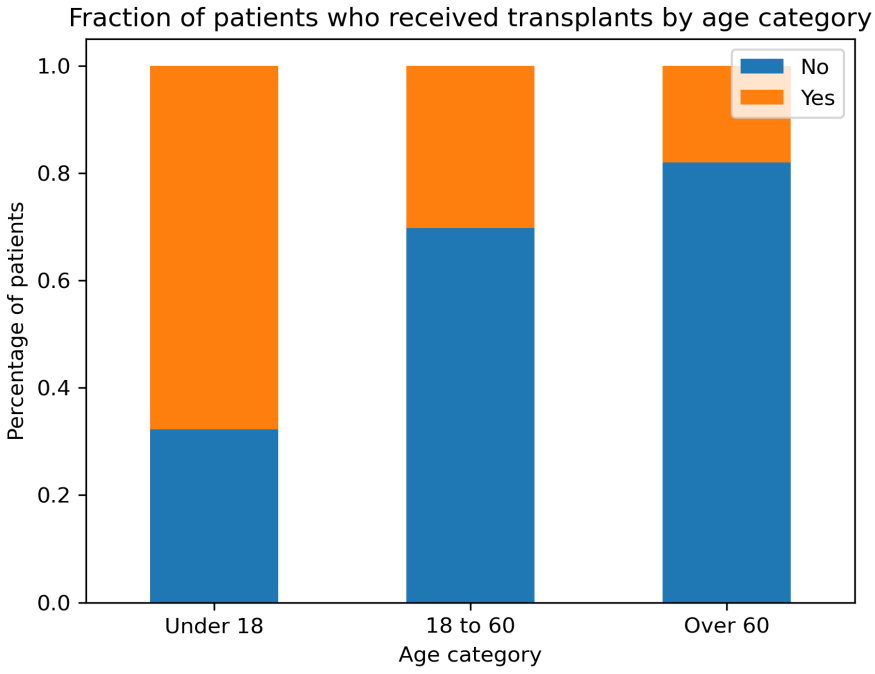

Figure 2. A higher percentage of young patients (under 18) received transplants than older patients.

Stratifying the data will not be necessary while splitting the data since the dataset is large and the target variable is relatively balanced. Additionally, this EDA demonstrates that age seems to be an important factor in the allocation of kidney transplants.

Another feature that is important for the allocation of kidneys is the cPRA value, according to the National Transplantation System of the Brazilian Ministry of Health [2]. A

higher cPRA value indicates that a patient is a worse candidate for an organ transplant. This feature was explored in relation to the target variable as well as sex.

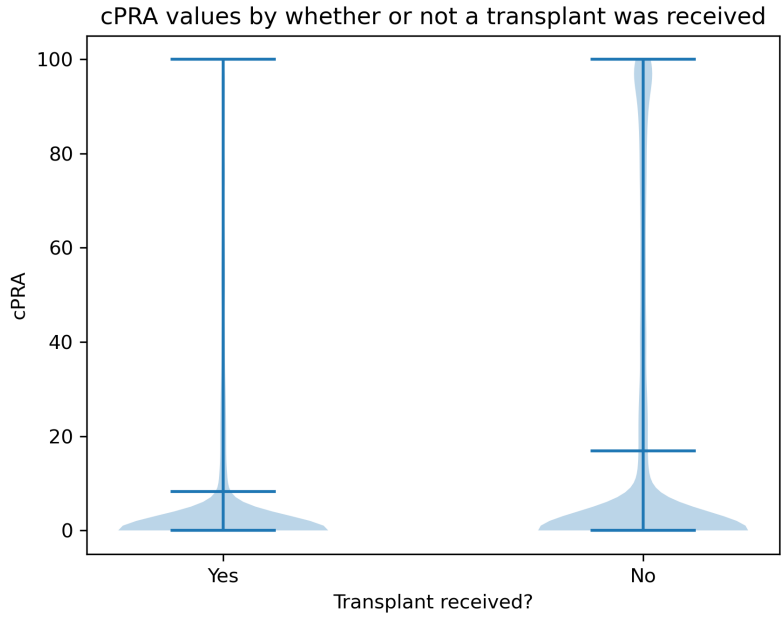

Figure 3. Plotting cPRA values whether or not a transplant was received shows that cPRA values tend to be higher for patients who did not receive a transplant. The range of cPRA values is the same for both groups: 0 to 100. The mean cPRA value is higher for patients who did not receive transplants.

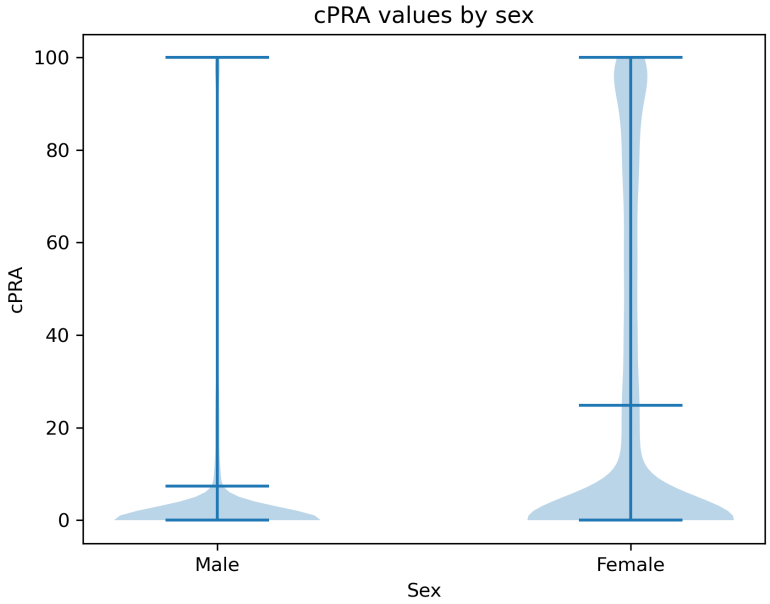

Figure 4. Plotting cPRA values by sex shows an association between being female and higher cPRA values. The mean cPRA value of females is much higher than that of males.

The most interesting and surprising discovery from this analysis was that male patients have lower cPRA values on average than female patients. The difference in mean cPRA values between males and females was even larger than the difference in the mean cPRA values between patients who received transplants and those who did not receive transplants. This discrepancy could explain why a higher percentage of males received transplants (29.5%) than females (27.0%).

Methods

This dataset is independent and identically distributed; it does not contain any group structure, nor does it contain time-series data. The intended use of this model would be to predict whether or not a new patient added to the SP-OAS waitlist will receive a kidney transplant. Several features had to be removed from the dataset due to the fact that they would not be available at the time of registration, e.g. whether a patient died, was removed from the waitlist, or their amount of time on the waitlist.

After removing these features, the dataset was divided into the feature matrix and target variable. Utilizing SciKit-Learn's powerful library of ML tools, a pipeline was developed that splits and preprocesses input data, then trains a ML model and performs cross-validation and hyperparameter tuning. The pipeline is performed 5 times for each unique ML algorithm using 5 different random states. The pipeline returns the best models, test scores, and baseline scores for each random state for each ML algorithm.

X and y were first split into 'other' and 'test' sets using a basic 80-20 split. The 'other' set was then split using k-fold with 5 folds. K-fold was selected so that cross validation could be performed while training the model. Preprocessing the dataset involved grouping the features by type. There were 15 categorical features, 2 ordinal features, and 16 continuous features. All of the categorical features were preprocessed with SciKit-Learn's OneHotEncoder. Both ordinal features were preprocessed with SciKit-Learn's OrdinalEncoder, and all of the continuous features were preprocessed with the StandardScaler to create a mean of 0 and standard deviation of 1. SciKit-Learn's SimpleImputer was also used on the continuous features for any algorithm other than XGBoost in order to replace missing values.

After splitting and preprocessing, for any pair of features with a Pearson correlation coefficient with an absolute value greater than 0.99, one of the two features was dropped. After this step, the feature matrix had 46 features.

Logistic regression, linear support vector classification (SVC), random forest, and XGBoost classification were used. For logistic regression, the penalty parameter was tuned with values 'None', 'L1', and 'L2', and C was tuned with a list of twelve log-spaced values ranging from 10^{-2} to 10^{7}. For linear SVC, penalty was tuned with two values ('L1' and 'L2') and C was tuned with a log-spaced list of 7 values between 10^{-3} and 10^{3}. A linear SVC was chosen instead of non-linear SVC because the dataset was large (over 48,000 data points), and the non-linear model had drastically higher runtime. For random forest classification, maximum depth and maximum features were the two parameters tuned. Maximum depth was tuned with the following values: [1,3,5,7,10,30,50,70,100]. The maximum features value was tuned with a

linearly spaced list of 10 values between 0.1 and 1. For the XGBoost classifier, 3 parameters were tuned, 'max_depth' with values [3, 5, 10, 30, 100], 'colsample_bytree' with a linearly spaced list of 4 values between 0.7 and 1, and 'subsample' with a linearly spaced list of 5 values between 0.65 and 1.

Accuracy score was used to evaluate the model's performance because it is an easily interpretable metric for classification problems. F-beta scores with beta=0.5 were also calculated. Uncertainties due to splitting and/or non-deterministic ML methods were measured by calculating the standard deviations of the test scores for each model that were found across all 5 random states tested.

Results

ML algo	baseline accuracy	test accuracy	accuracy number of standard deviations above baseline	baseline f0.5 score	test f0.5 score	f0.5 number of standard deviations above baseline
logistic regression	0.711 +/- 0.00409	0.727 +/- 0.00547	3.912	0.337 +/- 0.00445	0.429 +/- 0.0133	20.695
random forest classification	0.711 +/- 0.00409	0.740 +/- 0.00474	7.133	0.337 +/- 0.00445	0.449 +/- 0.0249	25.282
linear SVM classification	0.711 +/- 0.00409	0.727 +/- 0.00500	3.737	0.337 +/- 0.00445	0.396 +/- 0.0112	13.277
XGBoost classification	0.713 +/- 0.00398	0.750 +/- 0.00382	9.377	0.335 +/- 0.00433	0.523 +/- 0.0131	43.603

Table 1. Baseline and test accuracies and f0.5 scores for each ML algorithm, as well as the number of standard deviations above the baseline for each score.

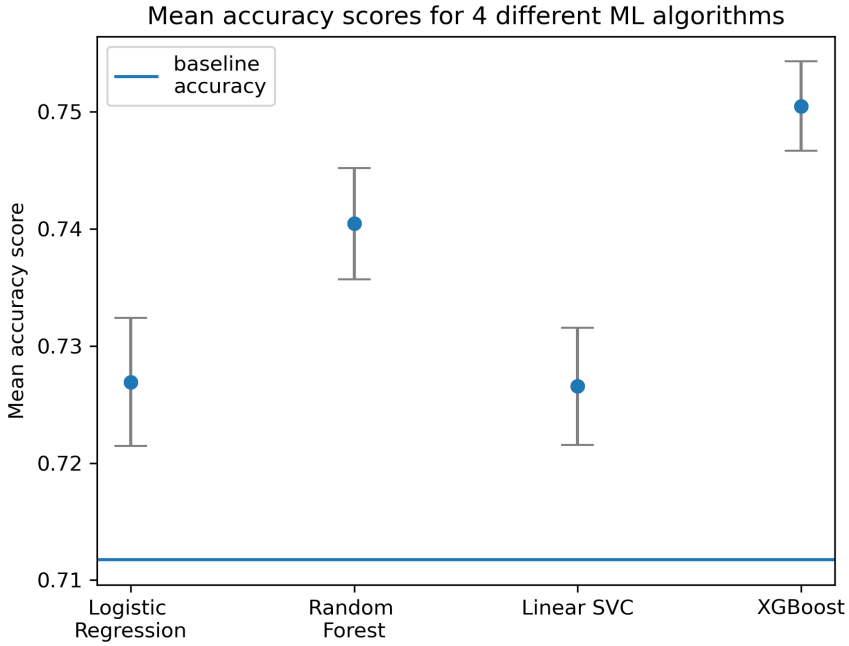

Figure 5. Displays the means and standard deviations of the test set accuracy scores for each of the four ML algorithms used in this study, as well as the baseline accuracy. Means and standard deviations were calculated across the test scores collected from each of the 5 random states for each algorithm.

As shown, the XGBoost classifier performed the best out of all four models with an accuracy score of 0.750 +/- 0.00382, which was 9.4 standard deviations above the baseline accuracy (0.713 +/- 0.00398). XGBoost also had the smallest standard deviation in test scores, indicating that it had the least amount of uncertainty due to splitting and the non-deterministic ML method itself.

Feature importances were investigated using the test set data and results. First, XGBoost's feature importance metrics were calculated and visualized.

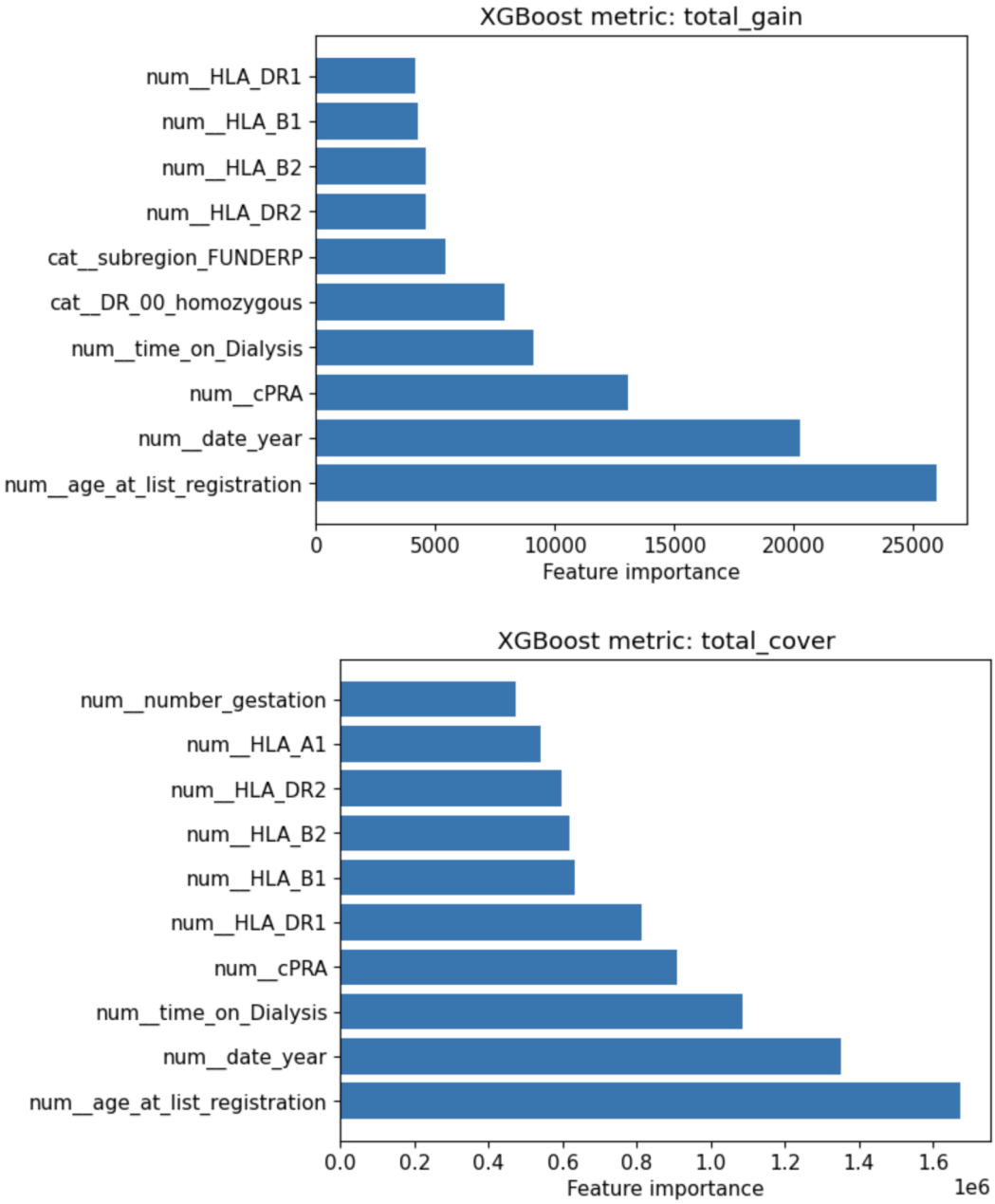

Figure 6. Age, year, cPRA, and time on dialysis, as well as several of the HLA-related features have high values for feature importance using both the total gain and total cover metrics.

Next, SHAP values were calculated for the XGBoost model using the TreeExplainer. A summary plot was created to show global feature importances based on the SHAP values.

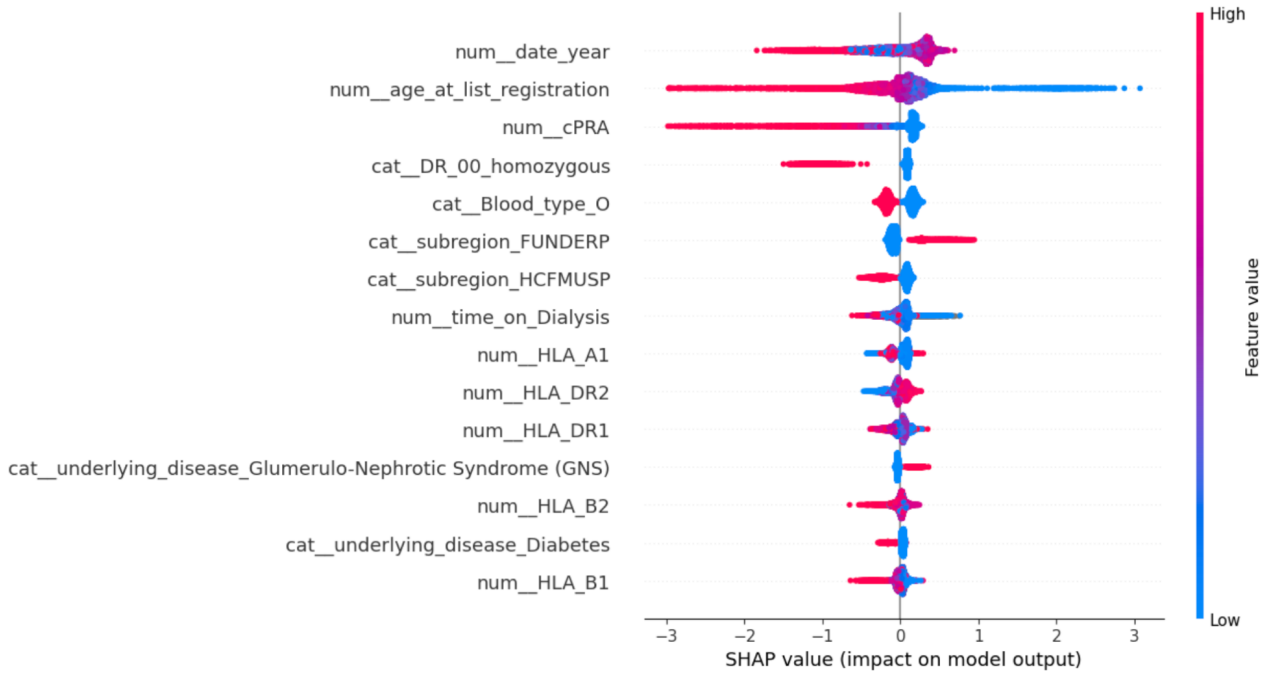

Figure 7. Global feature importance based on SHAP values show that year, age, cPRA, and time on dialysis appear within the top fifteen most important features once again.

Finally, permutation feature importance was calculated and visualized with a box plot.

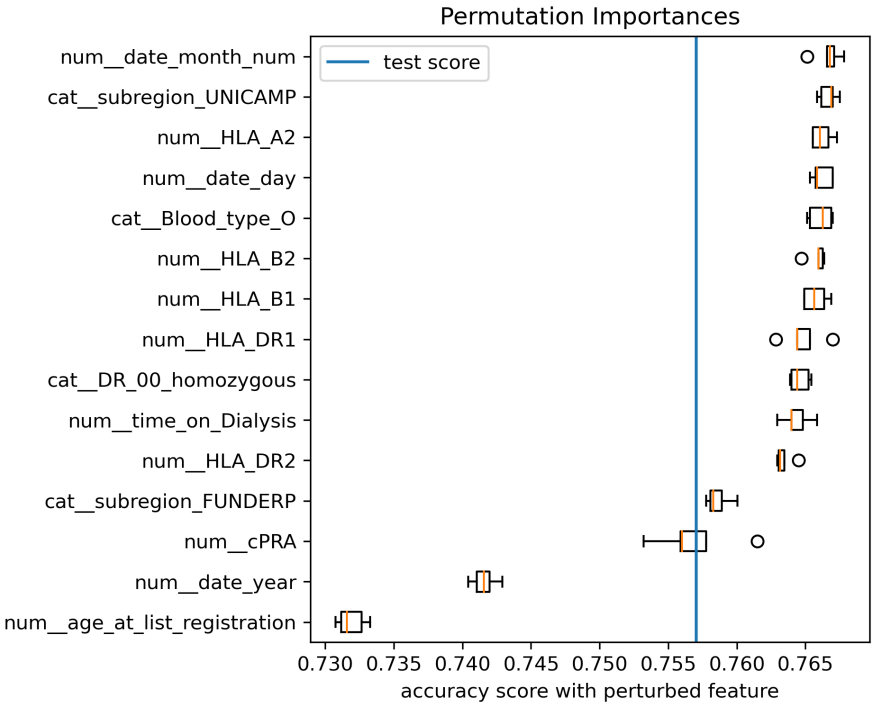

Figure 8. Global permutation feature importances. Perturbation of only three features—age, year, and cPRA—resulted in a reduction in the accuracy score when compared to the best accuracy score for the best XGBoost model.

As demonstrated by the three different global feature importance metrics, a few features—year, cPRA, age, time on dialysis, and HLA-related features—are repeatedly found to be among the most influential features for the model. One unexpected result from the permutation feature importance calculations was the discovery that perturbing many of the features actually improved the model's accuracy. This is a sign of a poor model because shuffling a feature should always reduce the model's accuracy if the model is using that feature properly.

The SHAP values and TreeExplainer were also used to calculate local feature importances for three patients. Two such force plots are included below.

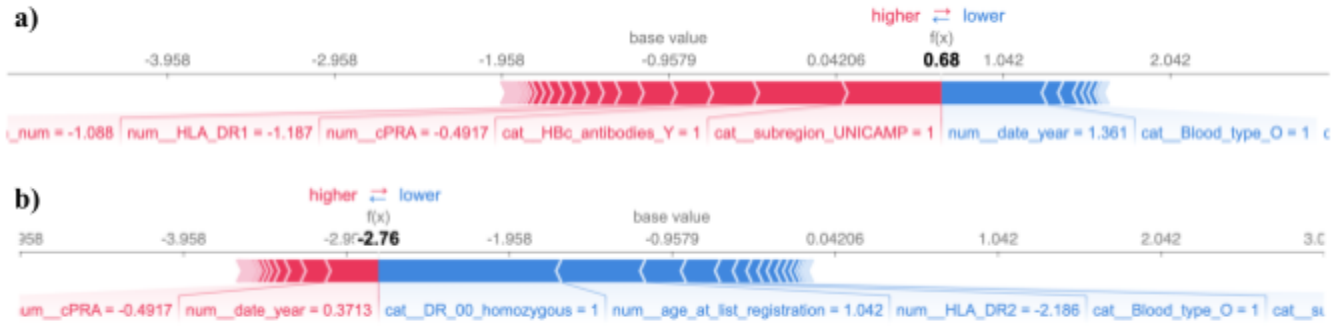

Figure 9. a) SHAP force plot for a patient whose true and predicted classes were 1 (received a transplant). **b)** SHAP force plot for a patient whose true and predicted classes were 0 (did not receive a transplant).

Because of the concerning findings from permutation feature importance, which suggested that many of the features in the model are actually reducing its accuracy, I do not recommend that this model should be used to determine who should and should not receive kidney transplants. With improvements, I believe that a model like this could be useful for the purpose of telling a new patient who is added to the waitlist whether or not they are likely to receive a transplant at some point in time. However, with the current parameters, this model is relatively inaccurate and has a very low recall score, so it performs poorly at identifying patients who should receive kidney transplants.

Outlook

In future iterations of this project, one way to improve the performance of this model could be to use the reduced features approach combined with XGBoost. The reduced features approach can help the model better handle missing values in continuous features, and it could improve the accuracy of the model when used in conjunction with XGBoost. Another method that could improve model performance would be using nonlinear SVM classification on a more powerful machine, as well as tuning the critical probability.

One longer-term way to improve this model would be to increase its generalizability by removing Brazil-specific features. Currently, some of the features used in the model are specific to Brazil (e.g. the subregion of Brazil), which makes this model only applicable for use in Brazil. Removing those features and increasing the generalizability of the model would allow it to benefit significantly more people on kidney transplant waitlists globally.

References

[1] Modelli, Luís Gustavo. (2021). Waitlist to Kidney Deceased Donor Transplant in Brazil. Retrieved October 7, 2022 from https://www.kaggle.com/datasets/gustavomodelli/waitlist-kidney-brazil

[2] Sapiertein Silva JF, Ferreira GF, Perosa M, Nga HS, de Andrade LGM. A machine learning prediction model for waiting time to kidney transplant. PLoS One. 2021 May 20;16(5):e0252069. doi: 10.1371/journal.pone.0252069. PMID: 34015020.