Data Science Program

Capstone Report - Spring 2024

Predicting Blood Transfusions for Coronary Artery Bypass Graft Patient

Jenny Hsiao-Tien Tsai,
Jichong Wu,
Puneet Gupta

Supervised by
Amir Jafari

Abstract

Blood transfusion during or after coronary artery bypass graft (CABG) surgery is associated with an increased risk for morbidity and mortality. There is a need to develop patient-specific risk prediction tools for blood transfusions in order to improve perioperative patient optimization, patient education, patient selection, patient outcomes, and clinical guidelines. This study used machine learning methods to 1) identify patient factors that influence the risk for requiring a perioperative blood transfusion and 2) develop and assess the performance of blood transfusion risk prediction models. Eight typical classification models were constructed and compared using the CABG surgery dataset in 2015-2022 from the American College of Surgeons National Surgical Quality Improvement Program (ACS-NSQIP). Many iterations of different data processing techniques, feature selection, and engineering methods were performed.

Across all data science and machine learning methods including the classical and neural networks models in our experiments, Gradient Boosting consistently lead the performance metrics even compared with deep neural network models such as FNNs and CNNs. The top five most important features for the Gradient Boosting model using the synthetic data generation method were disseminated cancer, chronic steroid use, diabetes, age, and preoperative albumin.

To the best of our knowledge, this is one of the first studies to demonstrate that AI-based models can perform well in predicting which patients will need a blood transfusion within the intraoperative or acute postoperative period following CABG surgery. Additionally, this study showed that AI can be used to identify patient risk factors for a perioperative blood transfusion. Further studies are needed to continually improve model performances in order to increase the likelihood that they improve patient outcomes and are cost-effective. These models additionally need to be externally validated prior to clinical translation.

Contents

List of Tables

List of Figures

1 Introduction

Coronary Artery Bypass Graft (CABG) is a common cardiac surgery but continues to have many associated risks, including major bleeding which might need blood transfusion. Previous research has shown that blood transfusion during CABG surgery is associated with an increased risk for mortality after surgery. Specially, post-operative blood transfusion after CABG is associated with higher odds of readmission and heart failure within 30-days.

To lower the risk of mortality after surgery, there is a need to develop models that preoperatively predict which patients will need an intra-operative or post-operative blood transfusion. This will not only help to improve patient selection and patient education, but also physician preoperative awareness and perioperative guidelines for CABG patients. Therefore, the goal of this project is to explore different approaches and find the models that can best make predictions, including feature selection/engineering, classical statistical models, and neural networks.

2 Problem Statement & Project Objectives

The objectives of the project are three-fold. The first objective is to develop models that can best predict whether a CABG patient will need blood transfusions. Second, we also look to experiment with various data science techniques to be applied in our models in order to achieve best performance, including feature selection, feature engineering, and synthetic data generation. Lastly, we aim to build a full set of modules and functions to be reused in the future beyond the current project. The modularized codes include but not limited to data preprocessing, feature selection, feature engineering, and modeling.

3 Related Work

3.1 Cardiac Surgery and Blood Transfusion

Research have been conducted to investigate factors that can help to predict major bleeding (Gao, et al., 2022) and the need for red blood cell transfusion after cardiac surgery (Li, et al., 2024). In one of the studies (Tschoellitsch, Bock, Mahecic, Hofmann, & Meier, 2022) that is most relevant to the current project, the researchers employed machine learning models to predict perioperative allogeneic blood transfusion for cardiac patients. The best model (Random Forest) showed good performance (RUC ranged from .76 - .86), however, the study has several limitations. For example, the data was from a single adult cardiac surgery center in Austria with a relatively small sample size (N = 3782), thus the results may not be generalizable to other samples with different demographics or nationalities. Moreover, the studies only predicted allogeneic blood transfusion (i.e., transfusion of more than 10 units of packed red blood cells (pRBC)), while blood transfusion regardless of volume has been associated with many known risks. Lastly, the study only tested the basic machine learning models (e.g., tree-based models), and it is likely that the performance can be significantly improved using more advanced techniques and deep neutral networks.

To address this research gap, the current project will use the national medical database in the U.S. with a large sample size of over 8,000 data points. Additionally, we will predict blood transfusion regardless of volume. Lastly, we will experiment with various approaches in order to optimize the performance, including feature selection, feature engineering, synthetic data generation, and deep neural networks.

3.2 Using Deep Learning Neural Networks for Tabular Data

Deep neural networks models have shown excellent performance and especially when processing complex data such as image, text and sound. However, their adaptation to tabular data tasks remains highly challenging (Vadim Borisov, 2022). The datasets of this project are 2-dimensional tabular data and the purpose of our study is to compare model performance between classic models and deep learning models using neural networks with synthetic data generation and other techniques.

Gorishniy et al. performed a review of major deep learning models for tabular data and concluded that a ResNet-like architecture and a simple adaptation of the Transformer architecture outperform other NN models, but there is no sufficient evidence that neural networks are better than classical models such as gradient boosting decision trees (Y. Gorishniy, 2021). Similarly, Shwartz-Ziv and Armon published a study rigorously comparing a number of neural network models – including TabNet, NODE, and Net-DNF – with XGBoost on various datasets. Their conclusion was that XGBoost outperforms those deep learning models across datasets they used, and they also demonstrated XGBoost requires much less tuning (R. Shwartz-Ziv, 2021). The authors also noted that an ensemble of neural network models and XGBoost performs the best and better than XGBoost alone.

Some studies noted the fact that it is sometimes unclear why neural network models cannot achieve the same level of predictive quality as in other domains such as image classification and natural language processing. Major issues identified by related work include: 1) low quality training data due to missing values, inclusion of outliers, erroneous or inconsistent data, and small data size (A. Sanchez-Morales, 2020) and (Veeramachaneni, 2018); 2) missing, complex or irregular spatial dependencies. There is little spatial correlation between the features in a tabular dataset or the dependencies are rather complex or irregular. The structure and relationships between features will be learned from scratch when training tabular data using neural networks, thus the inductive biases used for homogeneous data, such as Convolutional Neural Networks (CNNs), are unsuitable for modelling tabular data type (Y. Zhu, 2021); 3) dependency on preprocessing. Some studies indicate that tabular data and deep neural networks performance may strongly depend on selected preprocessing strategy and popular techniques to process categorical features such as one-hot encoding or ordinal encoding methods can lead to a very sparse feature matrix or introduce a synthetic ordering of previously unordered values, therefore very challenging (Khoshgoftaar, 2020). Also preprocessing methods for tabular data may cause information loss leading to reduced predictive performance (E. Fitkov-Norris, 2012); 4) importance of single features. In contrast to deep neural networks, decision-tree algorithms can handle varying feature importance exceptionally well by selecting single features with a threshold and ignoring the rest, while in a typical deep neural network prediction of class requires a coordinated change in many features (Segal, 2018).

Lastly, some studies landed on more promising outlook for using neural networks for tabular data and a hybrid of classical gradient boosting and deep neural network methods may have better performance. While it was concluded that Net-DNF do not consistently beat XGBoost, their results indicate that Net-DNF performance score is not far behind gradient boosting of decision trees. Therefore, Net-DNF offers a meaningful step toward effective usability of processing tabular data with neural networks (L. Katzir, 2021). A hybrid approach is also recommended by Popov et al., in which the authors introduce Neural Oblivious Decision Ensembles (NODE), a new deep learning architecture designed to work with any tabular data. The proposed NODE architecture benefiting from end-to-end gradient-based optimization and the power of multi-layer hierarchical representation learning (S. Popov, 2019). Similarly, a proposed hybrid methods of using gradient boosting and deep neural networks named SAINT performed attention over rows and columns for a tabular dataset, and outperform other deep learning models and gradient boosting models. An enhanced embedding method and a new contrastive self-

supervised pre-training method for scarce target labels were used. (G. Somepalli, 2021). Clements, et al. created a novel approach using deep recurrent and causal convolution-based neural networks to address credit risk monitoring with tabular financial data. The deep neural network models outperformed the benchmark non-sequential tree-based model, achieving significant financial savings and earlier detection of credit risk (J. M. Clements, 2020).

3.3 Bayesian Networks and Data Synthesizer

A Bayesian network is a graphical model of the joint probability distribution for a set of variables. The attributes are called *nodes* in the graph, and a conditional relationship between any two attributes is represented as an *edge* between the two nodes. Nodes and edges construct a Bayesian network, and multiple Bayesian networks can be averaged to form a Bayesian model (Young, Graham, & Penny, 2009). Bayesian networks are typically used to draw probabilistic inference about one attribute in the network given the values of other attributes, and therefore are suitable to be used for missing data imputation (Di Zio, Scanu, Coppola, Luzi, & Ponti, 2004) as well as synthetic data generation (Kaur, et al., 2021) (Baowaly, Lin, Liu, & Chen, 2019).

Data synthesizer is a tool that takes a dataset as input and generates a structurally and statistically similar synthetic dataset using Bayesian Networks (Ping, Stoyanovich, & Howe, 2017). DataSynthesizer consists of three modules — DataDescriber, DataGenerator and ModelInspector. DataDescriber collects the user-provided information about data, such as data types and correlations between attributes, and produces a data summary, adding noise to the distributions to preserve privacy. DataGenerator samples from the summary computed by DataDescriber and outputs synthetic data. ModelInspector provides statistics and plots for the users to visually inspect the similarity between the real data and the synthetic data.

To define the correlations between attributes in the dataset, DataSynthesizer can operate in one of three modes. In correlated attribute mode, a differentially private Bayesian network (Zhang, Cormode, Procopiuc, Strivastava, & Xiao, 2014) is used to capture the correlation structure between attributes, then draw samples from this model to construct the result dataset. Independent attribute mode can be used when there is insufficient data to derive a reasonable correlated model. In this mode, a histogram is created for each attribute, noise is added to the histogram to achieve differential privacy, and then samples are drawn for each attribute. Finally, for cases of extremely sensitive data, one can use random mode that simply generates type-consistent random values for each attribute (Ping, Stoyanovich, & Howe, 2017).

In the current research, we used correlated attribute mode as factors that can help to predict blood transfusion are often correlated. When correlated attribute mode is chosen, DataDescriber runs the GreedyBayes algorithm to construct Bayesian networks (BN) to model correlated attributes (***Table 1*** Algorithm for GreedayBayes.).

Table 1 *Algorithm for GreedayBayes.*

Algorithm 1 GreedyBayes(D, A, k)

Require: Dataset D, set of attributes A, maximum number of parents k

Initialize $\mathcal{N} = \emptyset$ and $V = \emptyset$.
Randomly select an attribute X_1 from A.
Add $(X_1, \emptyset)$ to $\mathcal{N}$; add X_1 to V.
for $i = 2, .., |A|$ **do**
  Initialize $\Omega = \emptyset$
  $p = min(k, |V|)$
  for each $X \in A \backslash V$ and each $\Pi \in \binom{V}{p}$ **do**
    Add (X, Π) to Ω
  end for
  Compute mutual information based on D for all pairs in Ω.
  Select (X_i, Π_i) from Ω with maximal mutual information.
  Add (X_i, Π_i) to $\mathcal{N}$.
end for
return $\mathcal{N}$

In the GreedyBayes algorithm, a Bayesian network N is constructed from input dataset D, attributes A, and the maximum number of parents node k, which defaults to 4. V is the set of visited attributes, and Π is a subset of V that will become parents of node X if added to N. Which attributes Π are selected as parents of X is determined greedily by maximizing mutual information (X , Π). The Bayesian networks constructed in this algorithm gives the sampling order for generating attribute values. When constructing noisy conditioned distributions, Lap(4(d−k)) is injected to preserve privacy, where d is the number of attributes, k is the maximum number of parents of a node, and n is the number of tuples in the input dataset.

4 Solution and Methodology

Data Source and Data Preprocessing

The data was downloaded from the Participant Use Data File (PUF) on the American College of Surgeons National Surgical Quality Improvement Program (ACS NSQIP). In this project, we focus on the data from 2015 to 2022, which has a total of 13,534 observations and around 296 variables across eight datasets (***Error! Reference source not found.2***)

Table 2 *Summary of datasets from year 2015 to 2022*

Year	# of Rows	# of Columns
2015	1678	274
2016	1657	274
2017	1612	274
2018	1821	274
2019	1639	274
2020	1493	276
2021	1702	260
2022	1932	270
Total	**13534**	**296**

After data preprocessing, including basic cleanup, imputation (mean for numeric variables and most frequent values for categorical variables), standardization, and encoding, the dataset with 41 features identified as most relevant to the current study was served as our baseline data. The target variable is Occurrences Bleeding Transfusions, which is a binary variable predicting whether the patient needs blood transfusion after surgery. The target can be further categorized into intraoperative vs. postoperative vs. no transfusion, therefore can be transformed into a 3-class variable when needed. With different analysis strategies, these features will be entered into our models to predict the target variable, and we will compare the performance with each other as well as with the benchmarks from previous research.

Table 3 *List of 41 most relevant features selected by expert*

#	Name	Definition
1	PUFYEAR	Year of PUF
2	SEX	Gender
3	RACE_NEW	Race
4	INOUT	Inpatient/outpatient
5	AGE	Age of patient with patients over 89 coded as 90+

6	ANESTHES	Principal anesthesia technique
7	BMI	Body Mass Index (calculated from HEIGHT and WEIGHT)
8	DIABETES	Diabetes mellitus with oral agents or insulin
9	SMOKE	Current smoker within one year
10	DYSPNEA	Dyspnea
11	FNSTATUS2	Functional health status Prior to Surgery
12	VENTILAT	Ventilator dependent
13	HXCOPD	History of severe COPD
14	ASCITES	Ascites
15	HXCHF	Heart failure (CHF) in 30 days before surgery
16	HYPERMED	Hypertension requiring medication
17	RENAFAIL	Acute renal failure (pre-op)
18	DIALYSIS	Currently on dialysis (pre-op)
19	DISCANCR	Disseminated cancer
20	WNDINF	Open wound/wound infection
21	STEROID	Immunosuppressive Therapy
22	WTLOSS	Malnourishment
23	BLEEDIS	Bleeding disorder
24	TRANSFUS	Preop Transfusion of >= 1 unit of whole/packed RBCs in 72 hours prior to surgery
25	PRSODM	Pre-operative serum sodium
26	PRBUN	Pre-operative BUN
27	PRCREAT	Pre-operative serum creatinine
28	PRALBUM	Pre-operative serum albumin
29	PRBILI	Pre-operative total bilirubin
30	PRSGOT	Pre-operative SGOT
31	PRALKPH	Pre-operative alkaline phosphatase
32	PRWBC	Pre-operative WBC
33	PRHCT	Pre-operative hematocrit
34	PRPLATE	Pre-operative platelet count
35	PRPTT	Pre-operative PTT
36	PRINR	Pre-operative International Normalized Ratio (INR) of PT values
37	EMERGNCY	Emergency case
38	ASACLAS	ASA classification
39	OPTIME	Total operation time
40	TOTHLOS	Length of total hospital stay
41	OTHERCPT1	Other CPT code 1

Exploratory Data Analysis

a. *Demographics*

Among the 13,534 patients in the eight-year combined dataset, nearly 80% are male (see Figure 1). The mean age is 65.73 with a standard deviation of 9.82 (see Figure 2). As for ethnicity composition (see Table 3), nearly half of the patients are white (48%) while over a third did not report their ethnicity (44%). Body Mass Index (BMI) were calculated based on HEIGHT and WEIGHT, indicating the signs of overweight with a mean BMI of 29.26 (see Figure 3 and Table 4).

Figure 1 *Gender breakdown*

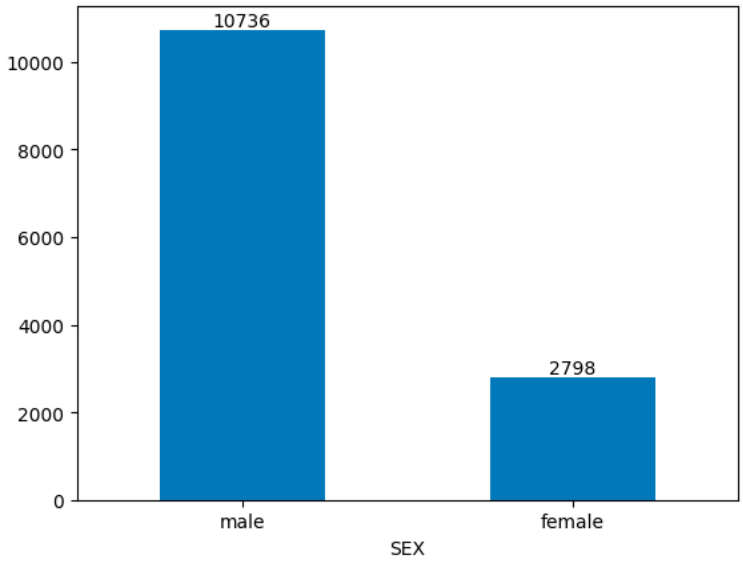


Figure 2 *Age distribution*

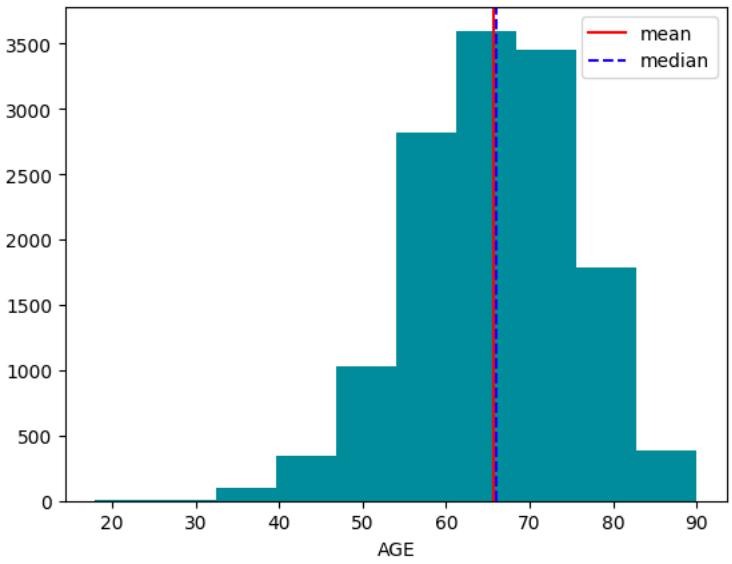


Table 4 *Ethnicity composition*

RACE_NEW	Counts
White	6488
Unknown/Not Reported	5918
Black or African American	606
Asian	381
Some Other Race	60
American Indian or Alaska Native	38
Native Hawaiian or Pacific Islander	36
Native Hawaiian or Other Pacific Islander	7

Figure 3 *BMI distribution*

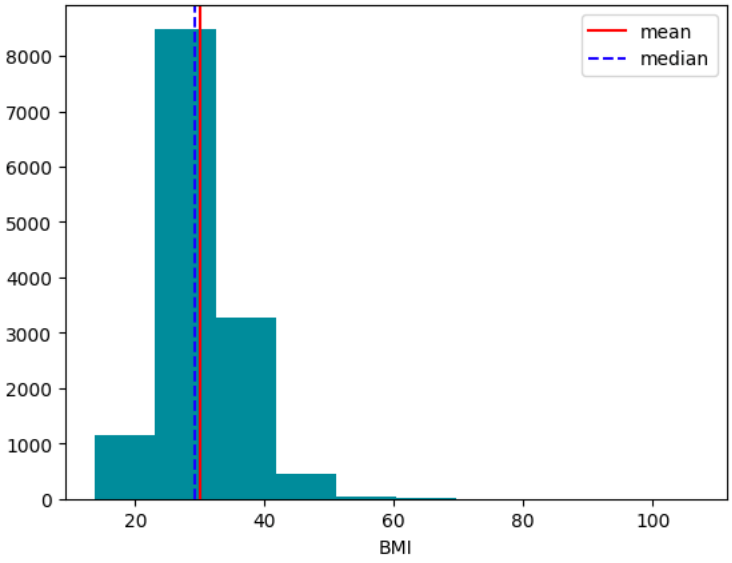


Table 5 *BMI statistics*

Mean	29.94
STD	5.76
Median	29.26
Max	106.82
Min	13.82
Skewness	1.15

b. *Target variable analysis - OTHBLEED*

Among all CABG patients, around half of the patients had blood transfusion (52.8%) and the other half did not (see Figure 4), therefore the target variable OTHBLEED is balanced. If further broken down into intra- vs. postop-blood transfusion, 86.5% of patients had blood transfusions *during* the surgery and only 13.5% had blood transfusion *after* the surgery.

Figure 4 *Bleeding Occurrence breakdown (binary)*

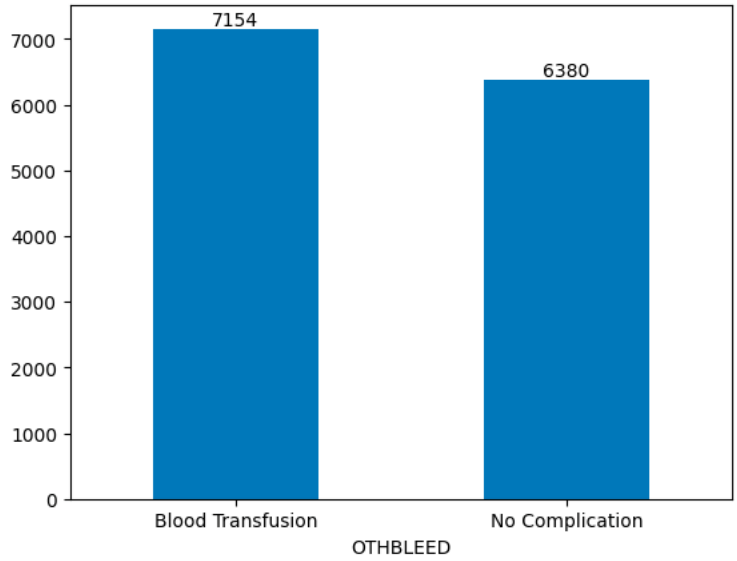
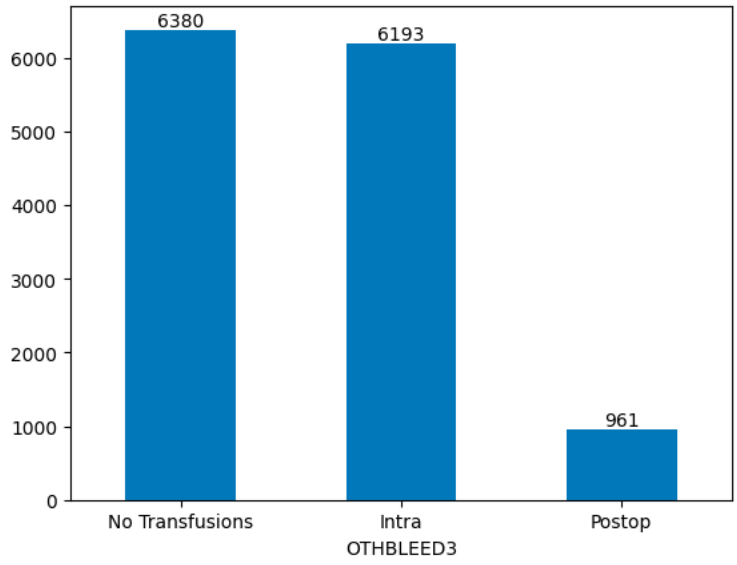


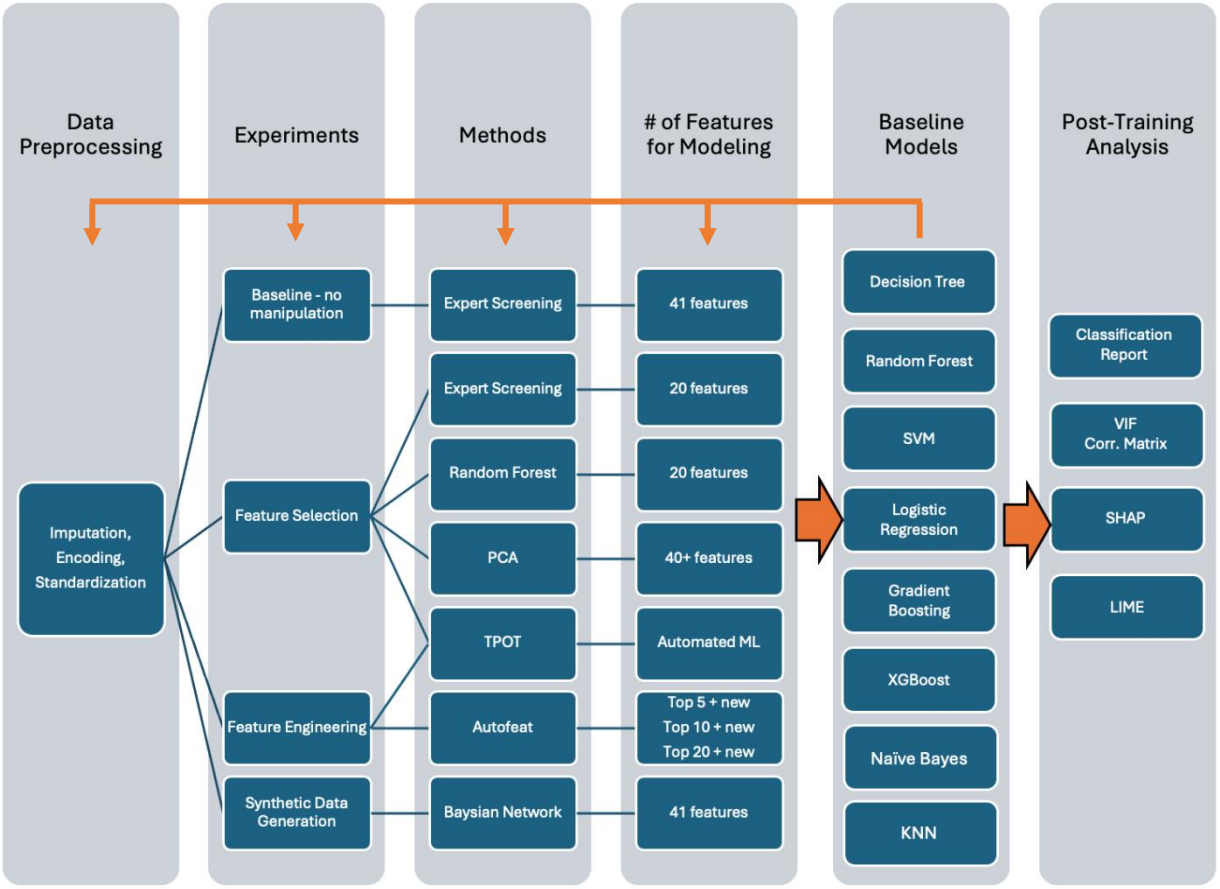
Figure 5 *Bleeding Occurrence breakdown (3-class)*



Analysis Strategy

Figure 6 shows the analysis strategy for the current project. After data preprocessing, we first entered the data into eight models, and used the results as our baseline benchmark. Next, we experimented with a different technique and method, and then entered the modified data into our models. In each iteration, we compared the new model performance with the baseline results and may start again from any of previous steps. For example, new data will be added to compare the prediction results when they become available. Or new feature engineering methods will be applied so we run the same data again starting from the third column ("methods).

Figure 6 *Research Strategy of the Project*



5 Results and Discussion

5.1 Model selection and tuning

Ten typical and representative supervised learning classification models are selected for this project and will be run through at each iteration every time there is an adjustment or improvement in the method, or new or extended data become available. The selected models are Decision Tree (criterion = "gini" and "entropy"), SVM (kernel = "linear" and "rbf"), Gaussian Naive Bayes, Logistic Regression, Gradient Boosting, XGBoost, KNN, and Random Forest (top important features = 20).

The target variable (the predicted variable, dependent variable) is "OTHBLEED" in the original dataset, or Occurrences Bleeding Transfusions. We group the values "Transfusions/Intraop/Postop" together and map as "1", and "No Complication" was converted to "0". We set random state as 100, testing size 25%, k-folds 10, and hold them constant in all models for comparison. The parameters of model construction and prediction results and evaluations for each run are documented in this section below.

Iteration #1

As the first step, we include all possible features with a missing data percentage larger than 50%. We then drop "NOTHBLEED" and "DOTHBLEED" due to high collinearity with the target. NOTHBLEED, number of bleeding transfusions occurrences, is highly correlated with the target OTHBLEED (occurrences bleeding transfusions) and has a Pearson correlation coefficient of -0.99. Similarly, DOTHBLEED, days from operation until bleeding transfusions complication, has a -0.81 Pearson correlation coefficient with the target. This leaves the data to be a 4953 by 127 dimension.

Simple linear imputation is applied to fill in the missing data in order for some models to run without errors. However, data is not standardized in this first round of modeling. ***Table 6***summarizes the setup of iteration #1.

Eight typical classification models were selected to train the data, including Decision Tree, SVM, Gaussian Naive Bayes, Logistic Regression, Gradient Boosting, XGBoost, KNN, and Random Forest. Different parameters and algorithm were also compared within Decision Tree (gini vs entropy) and SVM (linear vs rbf). Error! Reference source not found. indicates that Random Forest and Gradient Boosting have the better results across several evaluation metrics (accuracy score, root mean square error, F1 score, and ROC-AUC score). ***Figure 7*** combines the ROC plots for all the models which verify the conclusion above on top performing models.

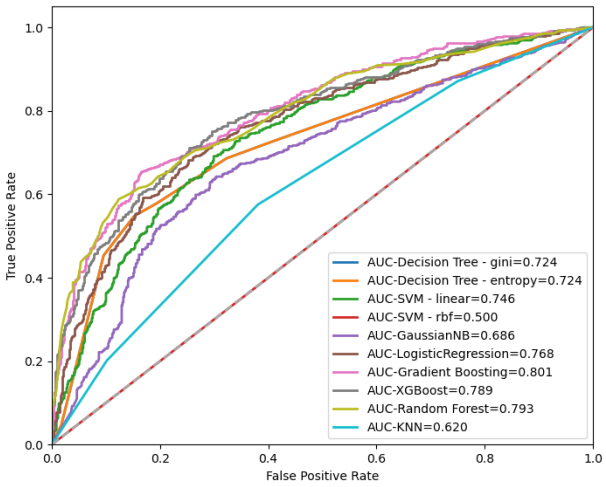
Table 6 *Summary of Iteration #1 Setup*

Data year	2018-2020
Observations	4953
Features included	126
Features manually dropped based on expert judgement	NOTHBLEED DOTHBLEED
Data preprocessing methods applied	Simple imputations
Final dataset	CABG_2018_2020_baseline.csv

Table 7 Model Results from Iteration #1

Model Name	Parameters	Accuracy	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree – gini	max_depth=3 min_samples_leaf=5	68.12	0.56	0.68	0.72
Decision Tree – entropy	max_depth=3 min_samples_leaf=5	68.12	0.56	0.68	0.72
SVM – linear	C=1.0, gamma=auto	69.49	0.55	0.69	0.75
SVM – rbf	C=1.0, gamma=0.2	55.53	0.67	0.36	0.50
Gaussian Naive Bayes		52.95	0.69	0.50	0.69
Logistic Regression		71.75	0.53	0.71	0.77
Gradient Boosting	n_estimators=300 learning_rate=0.05	73.93	0.51	0.73	0.80
XGBoost	n_estimators=100 eta=0.3	72.56	0.52	0.72	0.79
KNN	n_neighbor=3	59.97	0.63	0.60	0.62
Random Forest	n_estimators=300 feature_importances=20	73.93	0.51	0.73	0.79

Figure 7 ROC Plot with 10 Selected Models from Iteration #1



Iteration #2

Based on the first round of modeling with little data manipulation and interventions, we take it a step further to simply include more data from year 2021-2022 (***Table 8***. Results indicate very slight improvements in almost all models, with Gradient Boosting still being the best in all evaluation metrics followed by Random Forest not too far behind (Error! Reference source not found.***, Figure 8***).

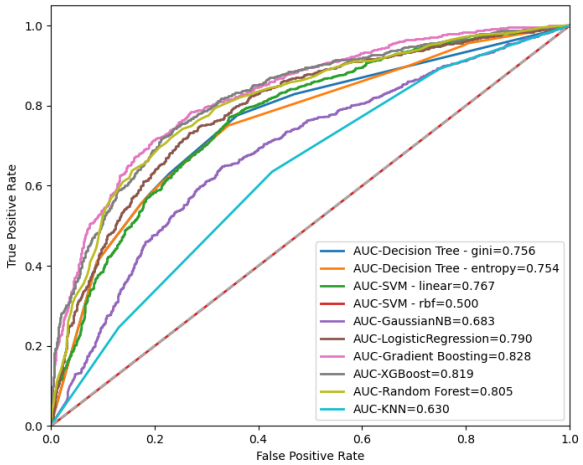
Table 8 Summary of Iteration #2 Setup

Data year	2018-2022
Observations	8587
Features included	126
Final dataset	CABG_2018_2022_baseline.csv

Table 9 Model Results from Iteration #2

Model Name	Parameters	Accuracy	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree – gini	max_depth=3 min_samples_leaf=5	70.70	0.54	0.71	0.76
Decision Tree – entropy	max_depth=3 min_samples_leaf=5	70.33	0.54	0.70	0.75
SVM – linear	C=1.0, gamma=auto	70.28	0.55	0.70	0.77
SVM – rbf	C=1.0, gamma=0.2	50.77	0.70	0.34	0.50
Gaussian Naive Bayes		57.29	0.65	0.53	0.68
Logistic Regression		72.99	0.52	0.73	0.79
Gradient Boosting	n_estimators=300 learning_rate=0.05	75.41	0.50	0.75	0.83
XGBoost	n_estimators=100 eta=0.3	75.13	0.50	0.75	0.82
KNN	n_neighbor=3	60.36	0.63	0.60	0.63
Random Forest	n_estimators=300 feature_importances=20	74.29	0.51	0.74	0.81

Figure 8 ROC Plot with 10 Selected Models from Iteration #2



Iteration #3:

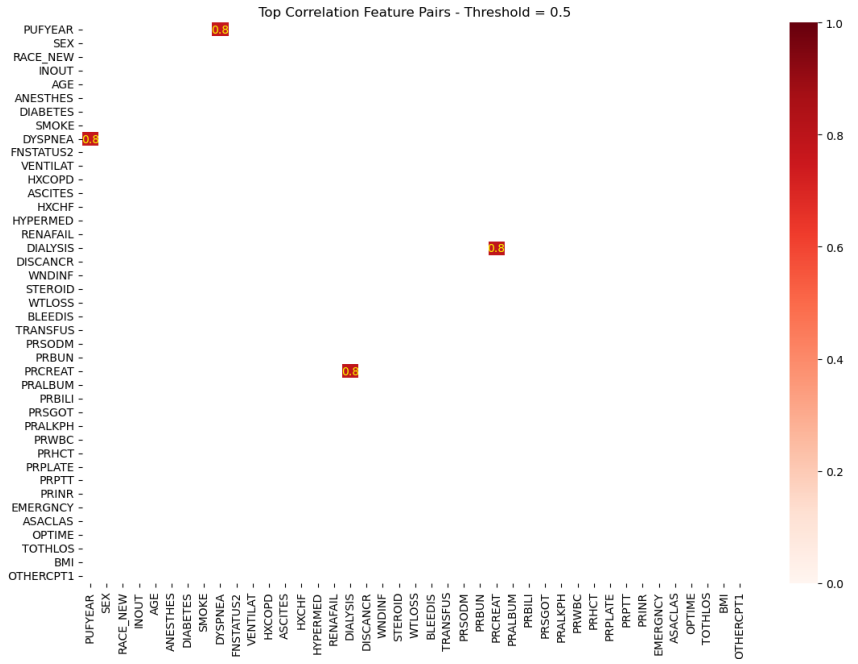
Next, we focus on cleaning up the data with more advanced data pre-processing methods including imputation, encoding, standardization to improve trainings of the model. Cross validation method is also introduced with a 10-fold parameter to enhance the mean accuracy. Further, 41 out of the 126 features from previous steps were manually screened and selected. Those obviously irrelevant features or those have little relationship with the target variable were removed based on common sense in medical and data science.

Correlation is checked for all features to identify highly correlated features which may affect model performance. “DLALYSIS” and “PRCREAT”, and “PUFYEAR” and “DYSPNEA” have the highest Pearson correlation coefficient and larger than 0.5 (***Figure 9***, left panel is a heatmap and right panel is the table of paired features).

"HEIGHT", "WEIGHT" are dropped due to multicollinearity issue with "BMI" (kept). Similarly, "ETHNICITY_HISPANIC" was dropped which is highly correlated with and a subset of feature "RACE_NEW" (kept). There are also features that may cause multicollinearity after the variance inflation factor (VIF) check but we decide to keep given their importance as a measure in understanding impacts on transfusion needs and decisions. As shown in ***Figure 10*** for example, "ASACLAS" (ASA classification, VIF = 43.6), "RACE_NEW" (VIF = 22.1), "OTHERCPT1" (Other Procedure, VIF = 9.9), and (SEX (VIF = 5.8). The data year, "PUFYEAR" has the highest VIF with a value of 100.8, we run all ten models with and without "PUFYEAR" to double check if recent or older data has a relationship with blood transfusion. The results are close to being identical between the two runs. Therefore, it is decided to drop "PUFYEAR" since it won't matter much and given its high VIF value.

Table 11 summarizes the setup for model construction iteration 3, ***Figure 111*** and ***Figure 11*** summarizes the results. Same as previous, Random Forest performs the best followed closely by Gradient Boosting. These 40 selected features now composite the baseline of our model iterations, before applying more advanced data processing and modeling methods in the following steps.

Figure 9 Top Correlation Feature Pairs in Iteration #3



Variable_1	Variable_2	Correlation
DIALYSIS	PRCREAT	0.79
PUFYEAR	DYSPNEA	0.77
PRBUN	PRCREAT	0.49
PRALBUM	PRHCT	0.32
DIALYSIS	PRBUN	0.31
PRALBUM	TOTHLOS	-0.31
SEX	PRHCT	0.27
OTHBLEED	PRHCT	0.27
BLEEDIS	PRPTT	0.26
DIALYSIS	PRHCT	-0.24
AGE	SMOKE	-0.24
PRWBC	PRPLATE	0.24
PRCREAT	PRHCT	-0.23
PRBUN	PRHCT	-0.22
PRHCT	TOTHLOS	-0.22
OTHBLEED	OPTIME	-0.21
HXCHF	TOTHLOS	0.21
PUFYEAR	EMERGNCY	-0.18
PRALBUM	PRPTT	-0.18
OTHBLEED	OTHERCPT1	0.18

Figure 10 *Variance Inflation Factor (VIF) Values of All Features (n = 41) in Iteration #3*

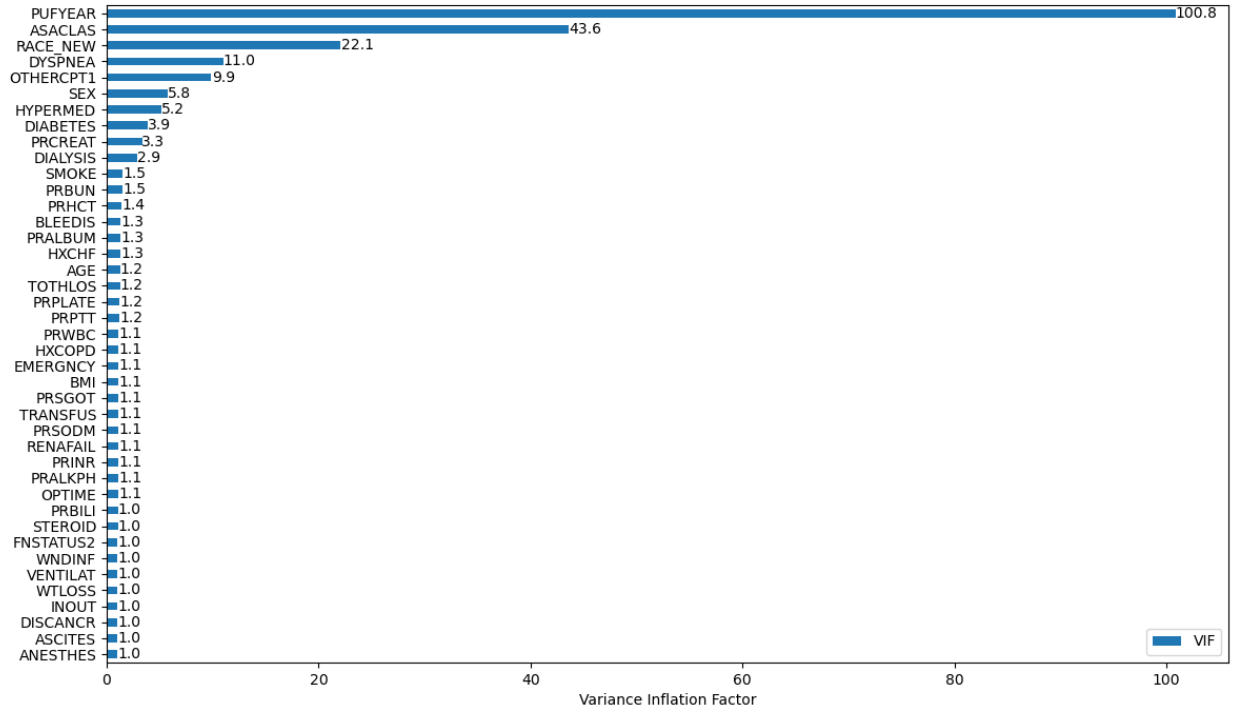


Table 10 *Summary of Iteration #3 Setup*

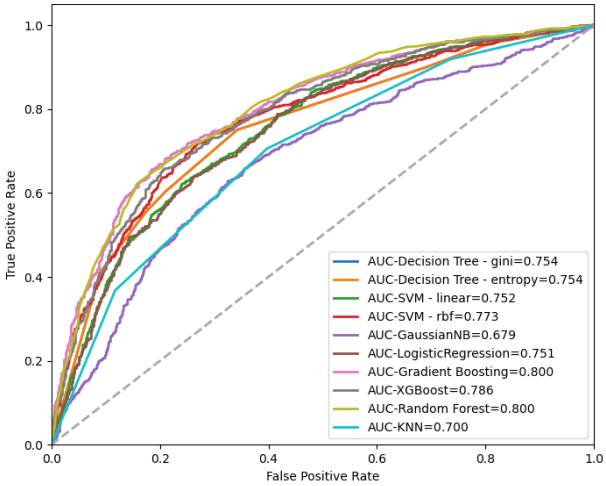
Data year	2018-2022
Observations	8587
Features included	40
Features manually dropped based on expert judgement	HEIGHT WEIGHT ETHNICITY_HISPANIC PUFYEAR
Features kept based on expert judgement	ASACLAS RACE_NEW SEX OTHERCPT1
Data preprocessing methods applied	Standardization Encoding Cross validation (10-folds)
Final dataset	CABG_5yr_preselect41.csv

Table 11 *Model Results from Iteration #3*

Model Name	Parameters	Mean Accuracy (10 folds)	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree – gini	max_depth=3 min_samples_leaf=5	69.70	0.54	0.70	0.75
Decision Tree – entropy	max_depth=3 min_samples_leaf=5	69.65	0.54	0.70	0.75
SVM – linear	C=1.0, gamma=auto	68.31	0.57	0.68	0.75
SVM – rbf	C=1.0, gamma=0.2	71.52	0.54	0.71	0.77

Gaussian Naive Bayes		54.23	0.68	0.48	0.68
Logistic Regression		68.28	0.57	0.68	0.75
Gradient Boosting	n_estimators=300 learning_rate=0.05	73.49	0.52	0.73	0.80
XGBoost	n_estimators=100 eta=0.3	72.62	0.53	0.72	0.79
KNN	n_neighbor=3	66.32	0.59	0.65	0.70
Random Forest	n_estimators=300 feature_importances=20	73.98	0.52	0.73	0.80

Figure 11 ROC Plot with 10 Selected Models from Iteration #3



Iteration #4:

Principal Component Analysis (PCA) is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional representation, preserving the most important information. It is commonly used to tackle multicollinearity and improves dimension. To have a complete comparison with all popular feature selection methods, we conduct PCA and the new dataset after PCA transformation reduced the feature dimension by one.

Error! Reference source not found.2 summarizes iteration #4 setup and ***Table 133*** and ***Figure 12*** includes the results comparison. Random Forest is the best performing model in this iteration (, however its results are significantly below iteration #3 and the same applies to all other models. We will then stop using PCA in future model constructions.

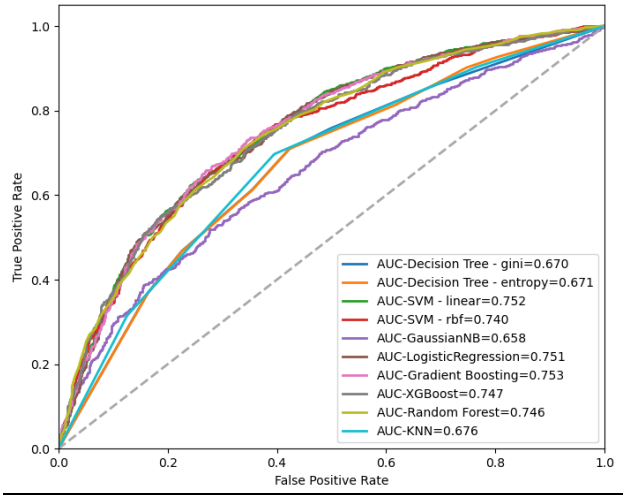
Table 12 Summary of Iteration #4 Setup

Data year	2018-2022
Observations	8587
Features included	39
Data preprocessing methods applied	PCA
Final dataset	CABG_5yr_PCA_39feature.csv

Table 13 Model Results from Iteration #4

Model Name	Parameters	Mean Accuracy (10 folds)	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree - gini	max_depth=3, min_samples_leaf=5	63.03	0.60	0.64	0.67
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	62.89	0.60	0.64	0.67
SVM - linear	C=1.0, gamma=auto	68.31	0.57	0.68	0.75
SVM - rbf	C=1.0, gamma=0.2	68.04	0.57	0.67	0.74
GaussianNB		56.46	0.66	0.53	0.66
LogisticRegression		68.27	0.57	0.68	0.75
Gradient Boosting	n_estimators=300, learning_rate=0.05	68.90	0.56	0.69	0.75
XGBoost	n_estimators=100, eta=0.3	68.78	0.57	0.68	0.75
KNN	n_neighbors=3	63.71	0.59	0.657	0.68
Random Forest	n_estimators=100, features_importances=20	69.10	0.57	0.68	0.75

Figure 12 ROC Plot with 10 Selected Models from Iteration #4



Iteration #5:

The key feature for this experiment is to use AutoFeat package to transform the original features. It automates feature engineering and selection and fit a linear prediction model. In this iteration, top 20 important features derived from previous iterations were selected to use with AutoFeat library. ***Table 14*** summaries the key information and ***Table 15*** and ***Figure 13*** displays the results. Gradient Boosting is the best performing model but it doesn't beat the top models from previous iterations.

Table 14 Summary of Iteration #5 Setup

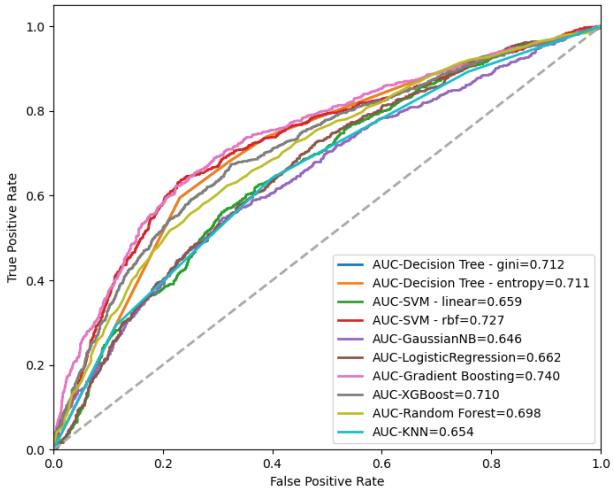
Data year	2018-2022
Observations	8587
Features included	20
Data preprocessing methods applied	AutoFeat

Final dataset	CABG_autofeat_top20.csv

Table 15 Model Results from Iteration #5

Model Name	Parameters	Mean Accuracy (10 folds)	RMSE	F1-score (macro avg)	ROC-AUC
Decision Tree - gini	max_depth=3, min_samples_leaf=5	67.45	0.56	0.68	0.71
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	67.47	0.56	0.68	0.71
SVM - linear	C=1.0, gamma=auto	60.57	0.63	0.60	0.66
SVM - rbf	C=1.0, gamma=0.2	68.99	0.55	0.69	0.73
GaussianNB		52.55	0.69	0.44	0.65
LogisticRegression		61.50	0.62	0.62	0.66
Gradient Boosting	n_estimators=300, learning_rate=0.05	68.88	0.55	0.69	0.74
XGBoost	n_estimators=100, eta=0.3	67.07	0.57	0.67	0.71
KNN	n_neighbors=3	62.00	0.62	0.62	0.65
Random Forest	n_estimators=100, features_importances=20	64.61	0.59	0.65	0.70

Figure 13 ROC Plot with 10 Selected Models from Iteration #5



Iteration #6:

TPOT is an automated machine learning tool that optimizes machine learning pipelines using genetic programming. TPOT automatically explores thousands of possible pipelines to find the best results data. This experiment tests the TPOT method and concludes that best model is ExtraTrees (***Table 16***). However, it is slightly under performed by the best models from iteration #3 – Random Forest and Gradient Boosting.

Table 16 Summary of Iteration #6 Setup and results

Data year	2018-2022
Observations	8587

Features included	40
Data preprocessing methods applied	TPOT
Final dataset	CABG_5yr_preselect41.csv
Model parameters	n_estimators = 100 generations = 5 population_size = 20 verbosity = 2
Results	Best pipeline: ExtraTreesClassifier with the following model parameters (booststrap=True, criterion=entropy, max_features=1.0, min_samples_leaf=1, min_samples_split=9, n_estimators=100) Accuracy = 72.52

Iteration #7:

Synthetic data generation is another effective feature engineering method and is used in this iteration. Again, 40 features that lead to the best models so far in iteration #3 are all included in this experiment. DataSynthesizer library that is based on Bayesian networks algorithm is used to re-generate a new dataset with the size of 1000 x 40 that feeds into our 10 models (***Table 17***).

This time, all models produce much higher results and the best is still between Gradient Boosting and Random Forest. Gradient Boosting model with synthetic data generation method significantly brings the mean accuracy to above 90%, with an error of 0.31, an F1 score of 0.8, and an ROC-AUC score of 0.93 (***Error! Reference source not found.*** and ***Figure 14***). This is our best results so far.

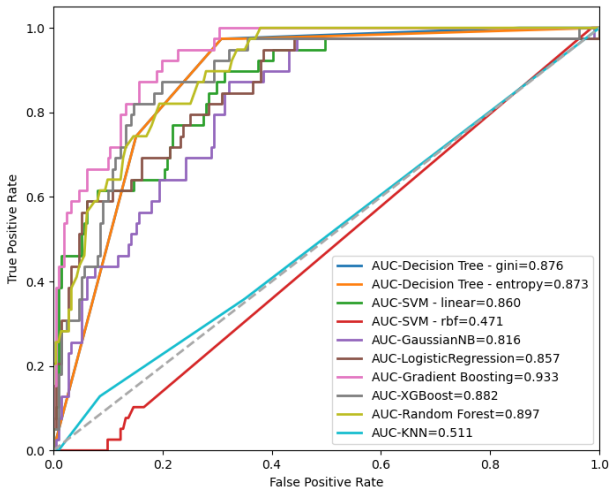
Table 17 *Summary of Iteration #7 Setup*

Data year	2018-2022
Observations	1000
Features included	40
Data preprocessing methods applied	Synthetic data generation – Bayesian networks
Final dataset	CABG_synthetic_Bayesian.csv

Table 18 *Summary of Iteration #7 Setup and results*

Model Name	Parameters	Mean Accuracy (10 folds)	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree - gini	max_depth=3, min_samples_leaf=5	86.90	0.41	0.74	0.88
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	86.90	0.41	0.74	0.87
SVM - linear	C=1.0, gamma=auto	87.60	0.36	0.76	0.86
SVM - rbf	C=1.0, gamma=0.2	83.90	0.39	0.46	0.47
GaussianNB		79.60	0.53	0.62	0.82
LogisticRegression		86.60	0.34	0.77	0.86
Gradient Boosting	n_estimators=300, learning_rate=0.05	90.80	0.31	0.80	0.93
XGBoost	n_estimators=100, eta=0.3	88.80	0.38	0.72	0.88
KNN	n_neighbors=3	81.20	0.46	0.52	0.51
Random Forest	n_estimators=100, features_importances=20	86.60	0.35	0.71	0.90

Figure 14 ROC Plot with 10 Selected Models from Iteration #7



Iteration #8:

Lastly, we include even more data from 2015-2017 and test if older data will even improve the current results even more. This brings the total observations to 13,534 from over 8,000 (***Error! Reference source not found.***), but it does not significantly improve the modeling results – the best model is still Gradient Boosting followed closely by Random Forest (

20 and ***Figure 15***).

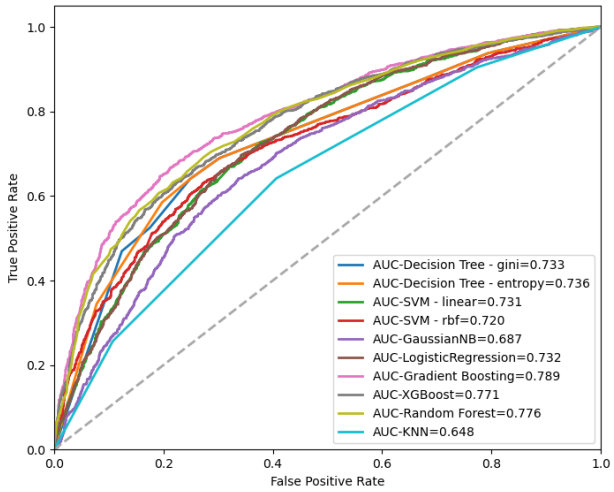
Table 19 Summary of Iteration #8 Setup

Data year	2015-2022
Observations	13534
Features included	40
Final dataset	CABG_8yr_preselect41.csv

Table 20 Summary of Iteration #8 Setup and results

Model Name	Parameters	Mean Accuracy (10 folds)	RMSE	F1 (macro avg)	ROC-AUC
Decision Tree - gini	max_depth=3, min_samples_leaf=5	69.27	0.55	0.69	0.73
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	69.28	0.55	0.69	0.74
SVM - linear	C=1.0, gamma=auto	66.91	0.57	0.67	0.73
SVM - rbf	C=1.0, gamma=0.2	66.43	0.58	0.64	0.72
GaussianNB		54.99	0.66	0.53	0.69
LogisticRegression		66.90	0.57	0.68	0.73
Gradient Boosting	n_estimators=300, learning_rate=0.05	72.33	0.52	0.73	0.79
XGBoost	n_estimators=100, eta=0.3	70.97	0.54	0.70	0.77
KNN	n_neighbors=3	61.00	0.62	0.62	0.65
Random Forest	n_estimators=100, features_importances=20	72.23	0.54	0.70	0.78

Figure 15 ROC Plot with 10 Selected Models from Iteration #8



5.2 Results and interpretation

By all data science and machine learning methods in our experiments, Gradient Boosting, Random Forest, and XGBoost are consistently leading the performance metrics in all iterations with Gradient Boosting being slightly better. The best model construction is from iteration #3 with data covered from 2018-2022, proper data cleaning methods applied, and 40 features included. In this run, Gradient Boosting performs the best. While it has a slightly lower average accuracy of 73.49 compared with 73.98 from Random Forest, it's better in all other important model evaluation metrics in RMSE, F1-score, and the ROC-AUC score (***Error! Reference source not found.***), which are typically weighted more heavily than accuracy in modeling evaluation.

Table 21 Best Performing Models – Gradient Boosting vs Random Forest vs XGBoost from Iteration #3

Model Name	Mean Accuracy (10 folds)	RMSE	F1-score (macro avg)	ROC-AUC score
Gradient Boosting	73.49469	0.51751	0.73204	0.80018
Random Forest	73.98384	0.52020	0.72878	0.80013
XGBoost	72.62148	0.53128	0.71766	0.78577

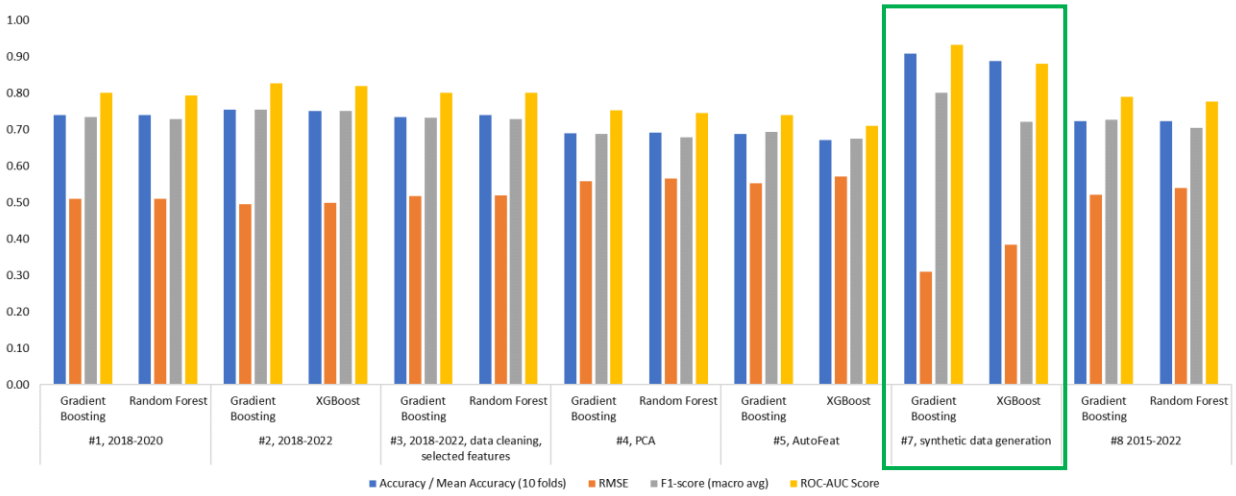
Focusing on just Gradient Boosting and other top 2 models across all iterations, model accuracy ranges from 67.07 - 90.80, root-mean-square error (RMSE) 0.31 – 0.57, F1-score 0.67 – 0.80, and ROC-AUC score has an upper bound of 0.93 and a lower bound of 0.71 (***Table 22*** *and* ***Figure 16***). Our results are consistent with our literature review in which others previous work has a range of ROC-AUC from 0.76 – 0.86. The synthetic data generation method enables significant improvements and bring the ROC-AUC scores in our results to 0.93 as the highest (Gradient Boosting from iteration #3).

Table 22 Top 2 Models Comparison in All Iterations

Iteration	Top 2 Models	Accuracy / Mean Accuracy (10 folds)	RMSE	F1-score (macro avg)	ROC-AUC Score

#1. 2018-2020	Gradient Boosting	73.93059	0.51058	0.73430	0.80146
	Random Forest	73.93059	0.51058	0.72932	0.79298
#2. 2018-2022	Gradient Boosting	75.40755	0.49591	0.75391	0.82770
	XGBoost	75.12809	0.49872	0.75123	0.81872
#3. 2018-2022 data cleaning selected features	Gradient Boosting	73.49469	0.51751	0.73204	0.80018
	Random Forest	73.98384	0.52020	0.72878	0.80013
#4. PCA	Gradient Boosting	68.90664	0.55821	0.68838	0.75256
	Random Forest	69.10431	0.56567	0.67935	0.74613
#5. AutoFeat	Gradient Boosting	68.88298	0.55276	0.69395	0.73976
	XGBoost	67.06653	0.57059	0.67438	0.71022
#7. synthetic data generation	Gradient Boosting	90.80000	0.30984	0.80066	0.93341
	XGBoost	88.80000	0.38471	0.72188	0.88152
#8. 2015-2022	Gradient Boosting	72.32897	0.52056	0.72637	0.78931
	Random Forest	72.22554	0.53924	0.70484	0.77600

Figure 16 Top 2 Models Comparison in All Iterations

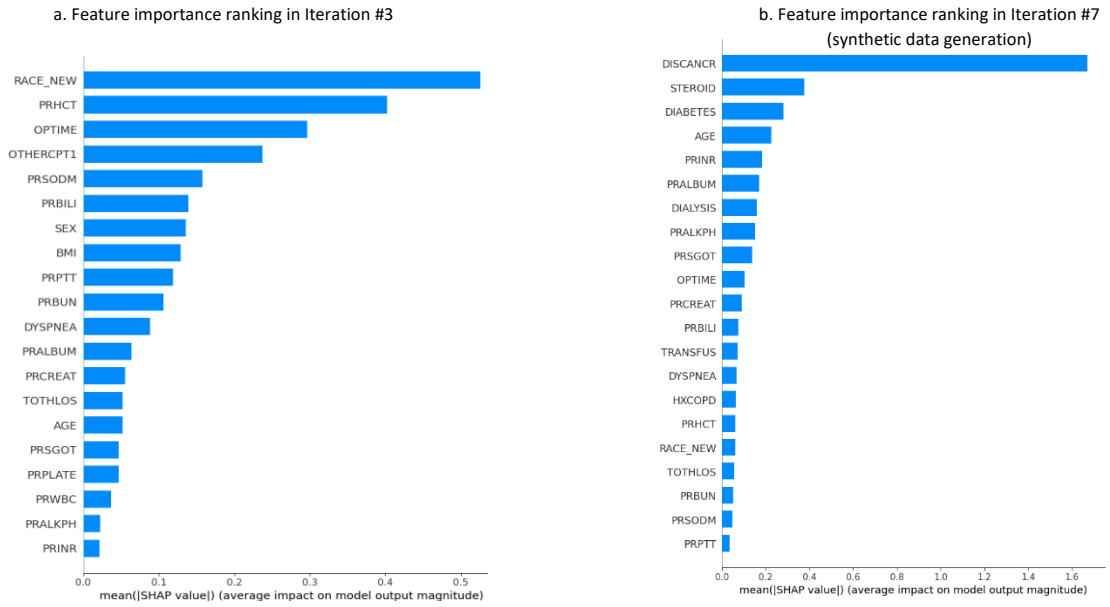


6 Discussion - understanding what features have higher impacts on model prediction

Figure 17 lists the ranking of the most importance 20 features from the best performing model – Gradient Boosting. Panel b on the right is what will be focused on in the following discussions since it has best results, but we do note that the feature data is transformed after synthetic generation. Therefore, we do not want to completely rule out any interesting observations on feature contributions to model prediction from iteration #3, which is the second best and the feature data are closer to their original values. We decide to document the feature importance and impacts analysis from panel a from iteration #3 in the Appendix as a background reference.

Back to panel b in ***Figure 17***, by ranking, DISCANCER[1], STEROID[2], DIABETES[3], AGE, and PRALBUM[4] are the top five important features that have made significant contributions to our Gradient Boosting model prediction from iteration #7 using the synthetic data generation method. Among which, DISCANCER has a significantly higher impact on predictions than other features.

Figure 17 *Feature Importance TOP 20 from Gradient Boosting in Iteration #3 vs Iteration #7*



Beeswarm plots can be used to highlight these important feature relationships. It indicates the relationship between feature values (high is red, blue is low) and its contribution to prediction classes (below zero or on the left falls into zero class or no transfusion, above zero or on the right belongs to one or need transfusion), as shown in ***Figure 18***. For example, the top 1 feature that impact the predicting results the most – DISCANCR. Those high density red dots on the left meaning many high values of DISCANCER (which is 1 since it's a categorical data) are contributing to high probability of predicting a zero class, or no transfusion. A negative feature impact on prediction class is observed.

Similarly, STEROID, PRALKPH, OPTIME all show strong relationship of a native impact on prediction – the higher their values, the more they contribute to high probability of the zero class, the more likely no transfusion is needed. On the contrary, PRSGOT, PRCREAT, PRHCT, and RACE_NEW indicate a strong positive impact on prediction, that is the higher values of these features, the more likely transfusion is needed. ***Table 23*** explained the relationship and impacts on target class for each of the top 20 features. Yellow highlighted features indicate a stronger impact in model prediction.

[1] Disseminated cancer
[2] Immunosuppressive Therapy
[3] Diabetes mellitus with oral agents or insulin
[4] Days from Albumin Preoperative Labs to Operation

Figure 18 *Beeswarm Plot of Important Feature Relationships from Gradient Boosting from Iteration #7*

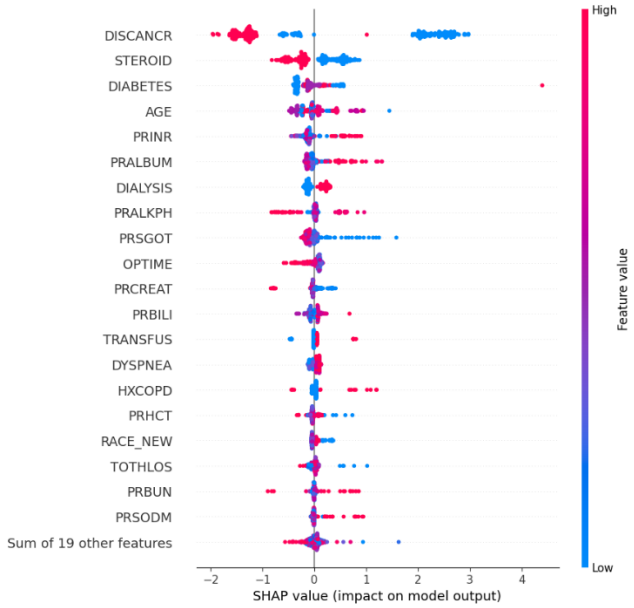
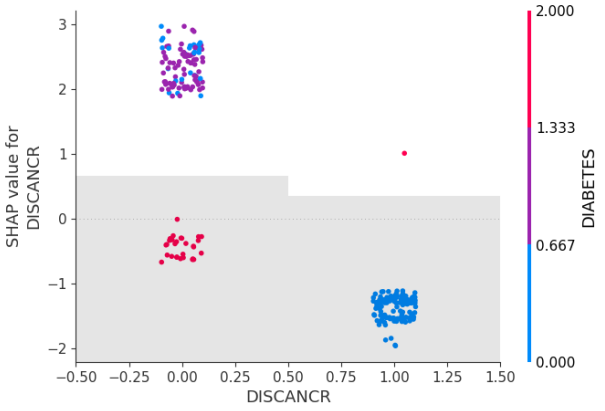


Table 23 *TOP 20 Important Features and Their Impacts on Gradient Boosting Model Prediction*

Feature	Definition	Data Type	Min - Max	Feature Contribution Relationship	Feature Impacts on Target Class
DISCANCR	Disseminated cancer	Categorical	No, Yes	Negative	Yes (1) --> lower chance for transfusion
STEROID	Immunosuppressive Therapy	Categorical	No, Yes	Negative	Yes (1) --> lower chance for transfusion
DIABETES	Diabetes mellitus with oral agents or insulin	Categorical	No MODERATE EXERTION AT REST	Positive	AT REST (1) --> higher chance for transfusion
AGE	Age of patient with patients over 89 coded as 90+	Numerical	[18, 90+]	Positive	The older, the higher chance for transfusion
PRINR	Days from INR Preoperative Labs to Operation	Numerical	[0.8, 4.8]	Positive	The higher, the higher chance for transfusion
PRALBUM	Pre-operative serum albumin	Numerical	[1.1, 9.9]	Positive	The higher, the higher chance for transfusion
DIALYSIS	Currently on dialysis (pre-op)	Categorical	No, Yes	Positive	No strong impact
PRALKPH	Pre-operative alkaline phosphatase	Numerical	[0, 89]	Unclear	Unclear
PRSGOT	Days from SGOT Preoperative Labs to Operation	Numerical	[0, 90]	Negative	The lower, the higher chance for transfusion
OPTIME	Total operation time	Numerical	[0, 1214]	Negative	The less, the higher chance for transfusion
PRCREAT	Days from Creatinine Preoperative Labs to Operation	Numerical	[0, 88]	Negative	The lower, the higher chance for transfusion
PRBILI	Days from Bilirubin Preoperative Labs to Operation	Numerical	[0, 89]	Positive	The higher, the higher chance for transfusion
TRANSFUS	Preop Transfusion of >= 1 unit of whole/packed RBCs in 72 hours prior to surgery	Categorical	No, Yes	Positive	The higher, the higher chance for transfusion
DYSPNEA	Dyspnea	Categorical	AT REST MODERATE EXERTION No	Positive	The higher, the higher chance for transfusion
HXCOPD	History of severe COPD	Categorical	No, Yes	Positive	The higher, the higher chance for transfusion
PRHCT	Pre-operative hematocrit	Numerical	[0, 88]	Negative	The lower, the higher chance for transfusion
RACE_NEW	New Race	Categorical	American Indian or Alaska Native (0) Asian (1) Black or African American (2) Native Hawaiian or Pacific Islander (3) Native Hawaiian or Other Pacific Islander (4) Some Other Race (5) Unknown/Not Reported (6) White (7)	Negative	The lower, the higher chance for transfusion
TOTHLOS	Length of total hospital stay	Numerical	[0, 64]	Negative	The lower, the higher chance for transfusion
PRBUN	Days from BUN Preoperative Labs to Operation	Numerical	[2, 198.88]	Unclear	Unclear
PRSODM	Pre-operative serum sodium	Numerical	[120, 156]	Positive	The higher, the higher chance for transfusion

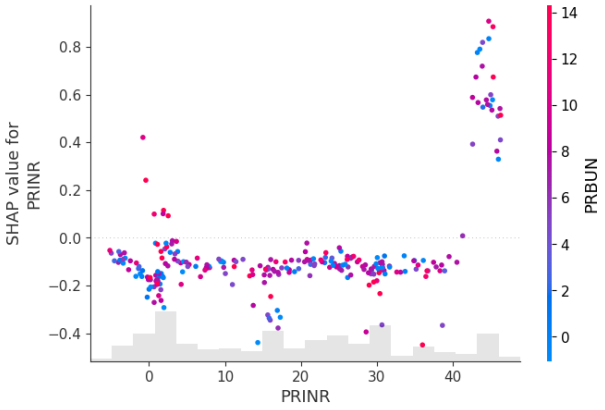
Lastly, we focus on several selected features to study its impact on prediction together with their most highly related feature (***Figure 19*** – ***Figure 28***).

Figure 19 Relationship Between DISCANCR and DIABETES and Their Impact on Prediction



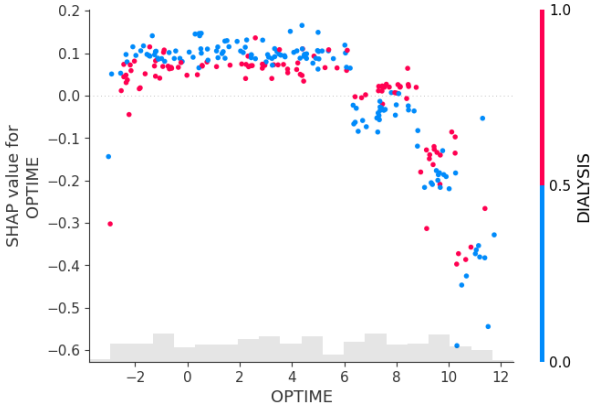
When DISCANCR (Disseminated cancer) = 0 and DIABETES = 1 (MODERATE EXERTION), these observations tend to have a higher positive SHAP value meaning they are more likely needed for blood transfusion.

Figure 20 Relationship Between PRINR and PRBUN and Their Impact on Prediction



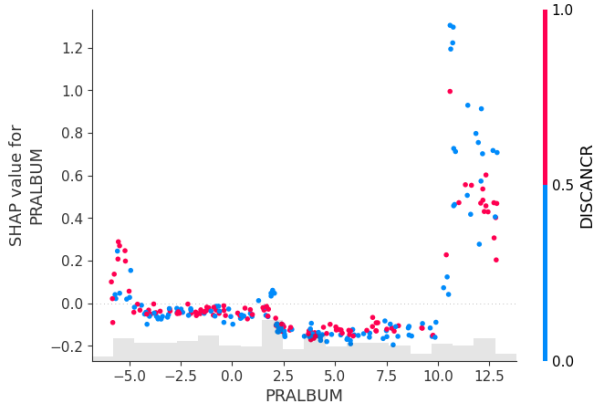
A small sample of observations indicate that when PRINR (Days from INR Preoperative Labs to Operation) values are high, the higher value its most related feature PRBUN (Days from BUN Preoperative Labs to Operation) is, the more the more likely these observations require blood transfusion.

Figure 21 Relationship Between OPTIME and DIALYSIS and Their Impact on Prediction



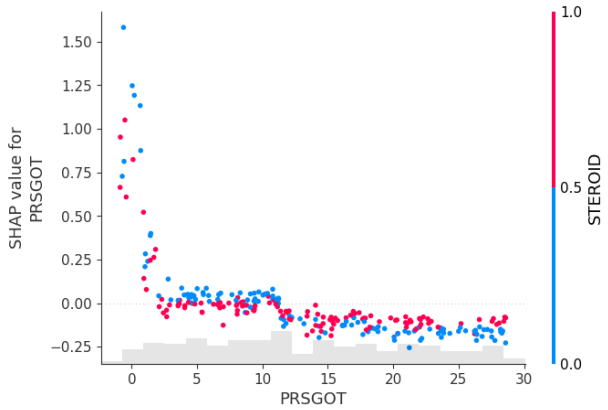
A strong negative relationship is observed: OPTIME (Total operation time) is negatively associated with blood transfusion, the longer OPTIME, the less likely blood transfusion is needed. Secondly, If DIALYSIS (Currently on dialysis (pre-op)) is “No”, it’s more likely that no blood transfusion is more likely to be needed.

Figure 22 *Relationship Between PRALBUM and DISCANCR and Their Impact on Prediction*



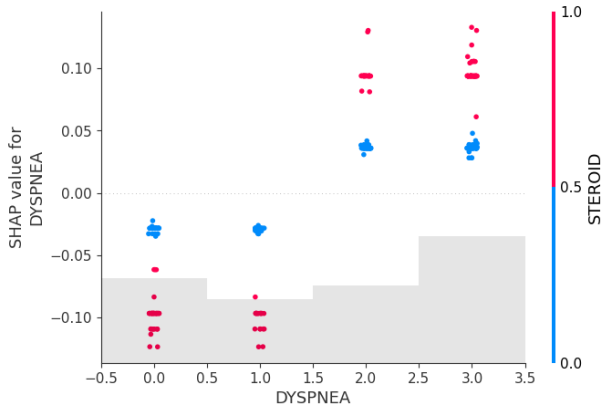
Some observations indicate that when PRALBUM (Pre-operative serum albumin) values are high, blood transfusion tends to be needed. And for those observations, DISCANCR (disseminated cancer) = 0 is more likely to cause transfusion than DISCANCR = 1.

Figure 23 *Relationship Between PRSGOT and STEROID and Their Impact on Prediction*



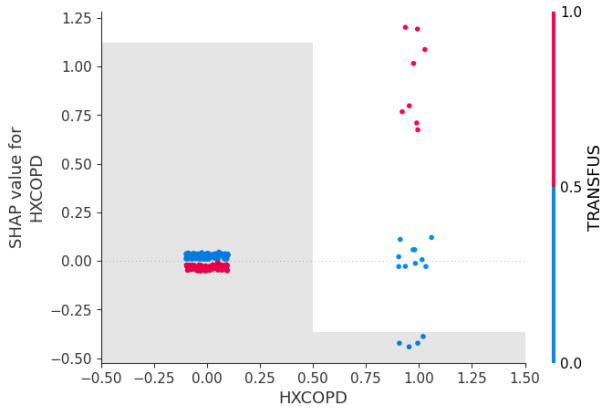
A small sample of observations indicate that when PRSGOT (Days from SGOT Preoperative Labs to Operation) values are low, blood transfusion is more likely to be needed.

Figure 24 *Relationship Between DYSPNEA and STEROID and Their Impact on Prediction*



When DYSPNEA occurs, it is more likely that blood transfusion tends to be needed. On top of that, if STEROID (Immunosuppressive Therapy) is "Yes", transfusion is even more likely.

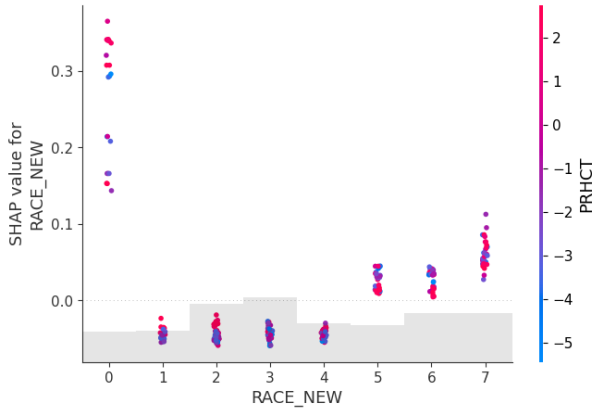
Figure 25 *Relationship Between HXCOPD and TRANSFUS and Their Impact on Prediction*



A small sample of observations indicate that HXCOPD (History of severe COPD) may lead to higher chance of blood transfusion, and if TRANSFUS (Preop Transfusion of >= 1 unit of whole/packed RBCs in 72 hours prior to surgery) = "Yes", it's even more likely (those red dots in the upper right quadrant.

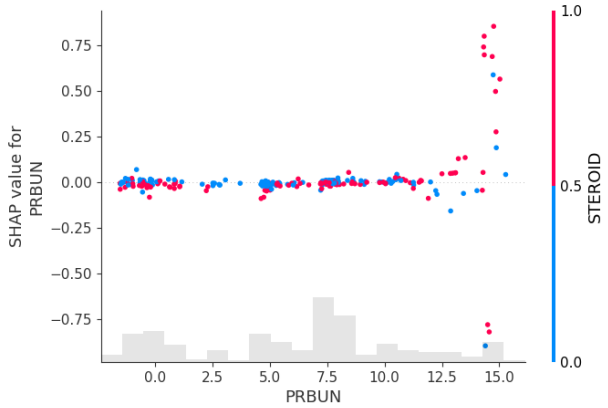
The opposite has a much stronger pattern: no HXCOPD plus no TRANSFUS have little impact on transfusion.

Figure 26 *Relationship Between RACE_NEW and PRHCT and Their Impact on Prediction*



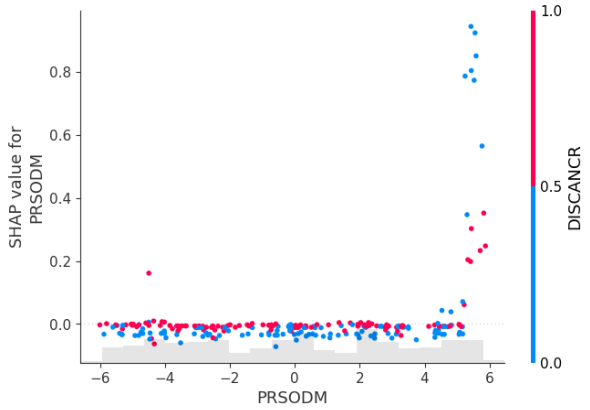
A small sample of observations with RACE_NEW = 0 (American Indian or Alaska Native) present higher chance for blood transfusion.

Figure 27 *Relationship Between PRBUN and STEROID and Their Impact on Prediction*



A small sample of observations indicate that high value of PRBUN (Days from BUN Preoperative Labs to Operation) may have a higher chance leading to blood transfusion; among those, STEROID (Immunosuppressive Therapy) = "Yes" seems to have an even higher likelihood.

Figure 28 Relationship Between PRSODM and DISCANCR and Their Impact on Prediction



A small sample of observations indicate higher PRSODM (Pre-operative serum sodium) may have a higher chance leading to blood transfusion, especially for those DISCANCR (Disseminated cancer) = "No".

The following research activities can be conducted to confirm our research findings and continue to improve model prediction performance.

- Use automated libraries or other methods to repeat some data processing and feature engeering methods we used with our own code and compare the results. For example, Caret for data cleaning, and Featurewiz, Featuretools for feature engineering; GANs for synthetic data generation.
- Break down the target label into 3 classes – same as the original data (Transfusions; Intraop/Postop; No Complication) – instead of the current two (Yes and No) and see if this would improve model predictions.
- Use deep learning and neural networks methods to build more advanced models to further improve the performance.

7 Neural Networks

In addition to the classical models, we aimed to employ two deep neural networks – Fully-Connected Neural Networks (FNN) and Convolutional Neural Networks (CNN) to predict the need for perioperative blood transfusions for CABG patients. Additionally, we used two approaches to generate synthetic data to train these neural networks – DataSynthesizer and REaLTabFormer (Realistic Relational and Tabular Data using Transformers). Data Synthesizer is based off Bayesian Networks, which are probabilistic graphical models that represent probabilistic relationship between variables. While REaLTabFormer uses a sequence-to-sequence (Seq2Seq) model for generating synthetic relational datasets and uses GPT-2 for non-relational tabular data.

In each type of neural network, we designed different models (e.g., different number of layers, activation functions, loss functions, etc.) and tested them with the original dataset, then we re-ran the models with synthetic datasets from DataSynthesizer and REaLTabFormer and compared the results to see which combination yields the best performance.

7.1 Fully-Connected Neural Networks (FNNs)

In FNNs, we designed eight models varying in complexity (5-layer vs. 7-layer with more neurons), optimizers (SGD vs. Adam), output activation functions (sigmoid vs. softmax) and their corresponding loss

functions (binary cross entropy vs. categorical cross entropy) to see which one(s) makes the best predictions. To evaluate the model performance, we looked at metrics across accuracy, f1 score, area under the curve (AUC), rooted mean squared error (rMSE). The best model(s) are determined jointly by f1 score and accuracy score with rMSE and AUC as supplementary benchmarks.

7.1.1 FNNs with Original Data

Results from FNN with the original dataset were shown in Table 2. Accuracy scores and f1 scores were landed in the range from .68 to .72, with a lowest rMSE of .46 and a highest AUC of .78. The best model was the five-layer design with SGD as optimizer and softmax as output activation function.

Table 24 FNN results with original dataset.

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.7215	0.6888	0.4794	0.7764
FNN-5layer-Adam-sigmoid	0.7218	0.6961	0.4928	0.7756
FNN-7layer-SGD-sigmoid	0.7174	0.6954	0.4850	0.7741
FNN-7layer-Adam-sigmoid	0.7174	0.6782	0.5735	0.7752
FNN-5layer-SGD-softmax	0.7229	0.7200	0.4643	0.7782
FNN-5layer-Adam-softmax	0.7218	0.7179	0.4975	0.7815
FNN-7layer-SGD-softmax	0.7181	0.7150	0.4780	0.7780
FNN-7layer-Adam-softmax	0.7185	0.7118	0.5587	0.7766

7.1.2 FNNs with Synthetic Data from REaLTabFormer

The eight models showed slightly improved performance with the synthetic data from REaLTabFormer (see Table 3). The accuracy scores and f1 scores ranged from .72 to .74, with a lowest rMSE of .45 and a highest AUC of .80. The best models, once again, were the 5-layer model with SGD using either sigmoid or softmax function.

Table 25 FNN results with synthetic dataset from REaLTabFormer.

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.7362	0.7420	0.4789	0.8006
FNN-5layer-Adam-sigmoid	0.7359	0.7394	0.4650	0.8007
FNN-7layer-SGD-sigmoid	0.7303	0.7392	0.4819	0.7939
FNN-7layer-Adam-sigmoid	0.7351	0.7315	0.5577	0.7972
FNN-5layer-SGD-softmax	0.7381	0.7380	0.4506	0.8008
FNN-5layer-Adam-softmax	0.7355	0.7355	0.4698	0.7999
FNN-7layer-SGD-softmax	0.7362	0.7362	0.4787	0.7995
FNN-7layer-Adam-softmax	0.7303	0.7286	0.5575	0.7934

7.1.3 FNNs with Synthetic Data from DataSynthesizer

Finally, results from FNN with the synthetic data from DataSynthesizer showed significantly improved performance (see Table 4). The accuracy scores and f1 scores ranged from .86 to .87, with a lowest rMSE of .42 and a highest AUC of .94. The best models were the 5-layer SGD model with sigmoid function and the 7-layer model SGD model with softmax function. However, although the 5-layer SGD model with sigmoid function had high

accuracy and f1 score, its rMSE was the highest across all models across three datasets (rMSE = .67). An interesting observation was found with DataSynthesizer synthetic data that rMSE were relatively high with SGD optimizer across the board (rMSE =.55~.67) compared with same design with Adam optimizer (rMSE =.42~.45).

Table 26 FNN results with synthetic dataset from DataSynthesizer.

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.8718	0.8776	0.6790	0.9386
FNN-5layer-Adam-sigmoid	0.8689	0.8772	0.4340	0.9474
FNN-7layer-SGD-sigmoid	0.8685	0.8774	0.5853	0.9358
FNN-7layer-Adam-sigmoid	0.8626	0.8712	0.4222	0.9456
FNN-5layer-SGD-softmax	0.8751	0.8749	0.6119	0.9421
FNN-5layer-Adam-softmax	0.8696	0.8696	0.4286	0.9495
FNN-7layer-SGD-softmax	0.8762	0.8762	0.5596	0.9401
FNN-7layer-Adam-softmax	0.8670	0.8667	0.4513	0.9474

7.1.4 Convolutional Neural Networks (CNNs)

While CNN is more suitable for handling image data, we also trained our preprocessed datasets with CNN just for comparison. We first transformed the 2-dimensional tabular dataset to 4-dimensional to meet the data format required by CNN. Two datasets derived from different synthetic data generation techniques (REaLTabFormer and DataSynthesizer) were used to run the CNN model and the accuracy was far lower than the FNN models (see Table 5). This results echo the findings and conclusion from related work done by others and mentioned in the earlier literature review section (Y. Gorishniy, 2021, R. Shwartz-Ziv, 2021, Y.Zhu, 2021)

Table 27 CNN results with across different preprocessed datasets.

Model	dataset	accuracy
CNN-2D-2layer-noPooling-ReLU	synthetic dataset 2015-2022 from REaLTabFormer	0.6681
	synthetic dataset 2015-2022 from DataSynthesizer	0.5785

8 Conclusion

Gradient Boosting, Random Forest, XGBoost present constant model performance across all methods and iterations in the order of ranking. It appears adding significantly more recent (2021-2022) or older data (2015-2017) may bring little improvement in model performance. Properly applying data processing techniques and feature selection and engineering methods will help improve modeling results.

Synthetic data generation method (DataSynthesizer using Bayesian networks) may significantly improve model performance. Together with #3 above, our model evaluation metrics have superior to previous modeling work (ROC-AUC ranged from 0.76 - 0.86) with best model results of an accuracy score of 90.80, RMSE 0.31, F1-score 0.80, and ROC-AUC comes to 0.93.

DISCANCR, STEROID, DIABETES, AGE, and PRINR are the top five most important features that have a larger impact on predicting blood transfusions. DISCANCR also presents a far more important influence, whose feature importance values are four times higher than the second place STEROID.

DISCANCR and STEROID have a negative contribution relationship to the prediction target classes, while DIABETES, AGE, and PRINR have a positive impact. This means that for example, the higher value of DISCANCR (1), the more it contributes to a higher probability of predicting of the lower level class for the target (blood transfusion = 0), meaning the less likely blood transfusion would occur. The other example is AGE, and it has a positive impact on prediction target classes. The higher AGE value, the more it contributes to a higher probability of predicting of the higher level class for the target (blood transfusion = 1), meaning higher likelihood for blood transfusion.

Page 23-25 examined selected pairs among the top 20 features and their impacts on model prediction. A strong negative relationship is observed between OPTIME (total operation time) and blood transfusion: the longer the operation, the less likely blood transfusion occurs. In addition, for those having long operation time samples, if DIALYSIS = 0 (Currently on dialysis (pre-op) is "No"), the more likely blood transfusion occurs.

This study aimed to employ modern techniques to develop models that can best predict the need for blood transfusions among CABG patients. In some cases, deep neural networks combined with data synthesis techniques have shown to significantly improve model performance. Especially in FNNs, regardless of model complexity and design, models trained with synthetic data generated from DataSynthesizer had best performance across the board, with f1 score and accuracy ranged from .86 to .87, with a lowest rMSE of .42 and a highest AUC of .94. Future research should look into different methodologies to generate synthetic data for training and developing models, both tree-based models and deep neural networks, that can help inform guidelines for major high-risk surgeries.

9 References

A. Sanchez-Morales, J.-L. S.-G.-A.-G.-V. (2020). Improving deep learning performance with missing values via deletion and compensation. *Neural Computing and Applications*, vol. 32, no. 17, pp. 13 233–13 244.

Baowaly, M., Lin, C., Liu, C., & Chen, K. (2019, March). Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association, 26*(3), 228-241.

Carpenter, G. G. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Neural Networks and Natural Intelligence*, 37, 54-115. DOI: https://doi.org/10.7551/mitpress/4934.003.0008.

Di Zio, M., Scanu, M., Coppola, L., Luzi, O., & Ponti, A. (2004). Bayesian networks for imputation. *Journal of the Royal Statistical Society, 167*, 309-322.

E. Fitkov-Norris, S. V. (2012). Evaluating the impact of categorical data encoding and scaling on neural network classification performance: the case of repeat consumption of identical cultural goods. *International Conference on Engineering Applications of Neural Networks*, pp. 343–352.

G. Somepalli, M. G. (2021). SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv preprint arXiv:2106.01342.

Gao, Y., Liu, X., Wang, L., Wang, S., Yu, Y., Ding, Y., . . . Ao, H. (2022, July). Machine learning algorithms to predict major bleeding after isolated coronary artery bypass grafting. *Frontiers in Cardiovascular Medicine, 9*, doi: 10.3389/fcvm.2022.881881.

Horvath, K., Acker, M., Chang, H., & Bagiella et al, E. (2013, June). Blood transfusion and infection after cardiac surgery. *The Annals of Thoracic Surgery, 95*(6), 2194-2201.

J. M. Clements, D. X. (2020). Sequential deep learning for credit risk monitoring with tabular financial data. arXiv preprint arXiv:2012.15330.

Kaur, D., Sobiesk, M., Patil, S., Liu, J., Bhagat, P., Gupta, A., & Markuzon, N. (2021, March). Application of Bayesian networks to generate synthetic health data. *Journal of the American Medical Informatics Association, 28*(4), 801-811. doi: 10.1093/jamia/ocaa303.

Khoshgoftaar, J. T. (2020). "Survey on categorical data for neural networks. *Journal of Big Data*, vol. 7, pp. 1–41.

L. Katzir, G. E.-Y. (2021). Net-DNF: Effective deep modeling of tabular data. *International Conference on Learning.*

Li, Q., Lv, H., Chen, Y., Shen, J., Shi, J., & Zhou, C. (2024, April). Development and validation of a machine learning prediction model for perioperative red blood cell transfusions in cardiac surgery. *International Journal of Medical Informatics, 184*, 105343.

Macukow, B. (2016). Neural Networks – State of Art, Brief History, Basic Models and Architecture. *Computer Information Systems and Industrial Management*, vol 9842. https://doi.org/10.1007/978-3-319-45378-1_1.

Martin T. Hagan, H. B. (2014). *Neural Network Design (2nd Edition).*

Mufarrih, S., Mahmood, F., Qureshi, N., Yunus, R., & Matyal et al, R. (2023, Mar). Timing of blood transfusions and 30-day patient outcomes after coronary artery bypass graft surgery. *Journal of Cardiothoracic and Vascular Anesthesia, 37*(3), 382-391, doi: 10.1053/j.jvca.2022.11.029.

Ping, H., Stoyanovich, J., & Howe, B. (2017). DataSynthesizer: Privacy-preserving synthetic datasets. *International Conference on Scientific and Statistical Database Management*, (pp. 1-5).

R. Shwartz-Ziv, A. A. (2021). Tabular Data: Deep Learning is Not All You Need. arXiv preprint arXiv:2106.03253.

S. Popov, S. M. (2019). Neural oblivious decision ensembles for deep learning on tabular data. *arxiv:1909.06312*.

Schmidhuber, S. H. (1997). Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780.

Segal, I. S. (2018). "Regularization learning networks: deep learning. *Advances in Neural Information Processing Systems*, pp. 1379–1389.

Tschoellitsch, T., Bock, C., Mahecic, T., Hofmann, A., & Meier, J. (2022, Sep). Machine learning-based prediction of massive perioperative allogeneic blood transfusion in cardiac surgery. *European Journal of Anaesthesiology, 39*(9), 766-773, doi: 10.1097/EJA.0000000000001721.

Vadim Borisov, T. L. (2022). Deep Neural Networks and Tabular Data: A Survey. *IEEE*. https://arxiv.org/pdf/2110.01889.

Veeramachaneni, L. X. (2018). Synthesizing Tabular Data using Generative Adversarial Networks. arXiv preprint arXiv:1811.11264.

Warren McCulloch, W. P. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5, 115-133.

Y. Gorishniy, I. R. (2021). Revisiting deep learning models for tabular data. arXiv preprint arXiv:2106.11959.

Y. Zhu, T. B. (2021). Converting tabular data into images. *Scientific Reports*, vol. 11, no. 1, pp. 1–11.

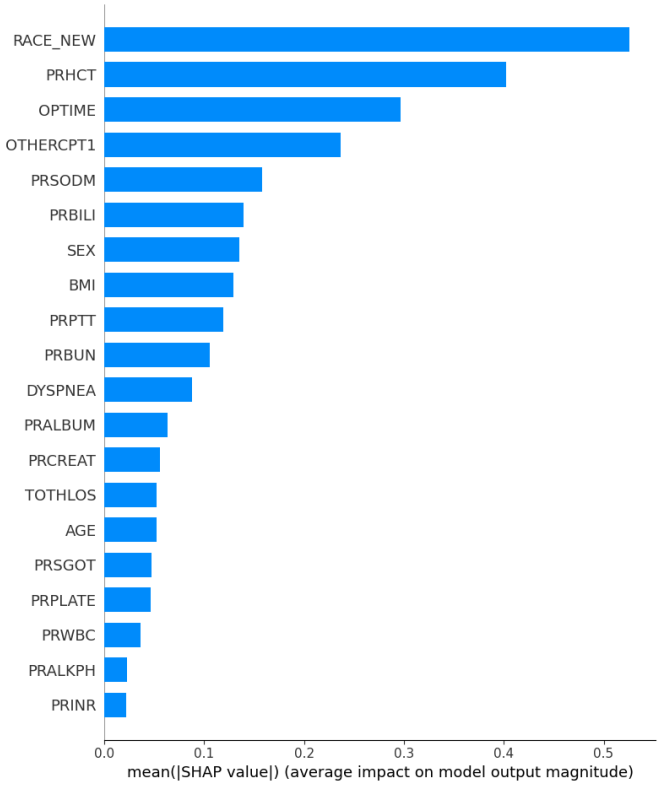
Young, J., Graham, P., & Penny, R. (2009). Using Bayesian networks to create synthetic data. *Journal of Official Statistics, 25*(4), 549-567.

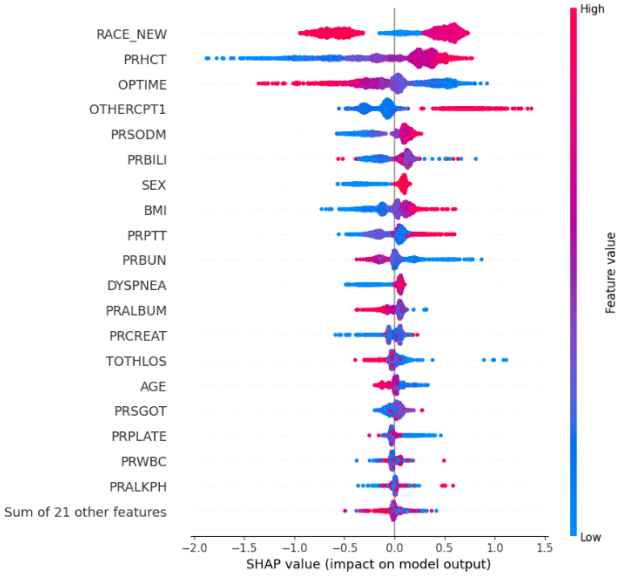
Yuchen Wu, J. F. (2018). Development and Application of Artificial Neural Network. *Wireless Personal Communications: An International Journal, Volume 102, Issue 2*, 1645-1656. Retrieved from https://doi.org/10.1007/s11277-017-5224-x

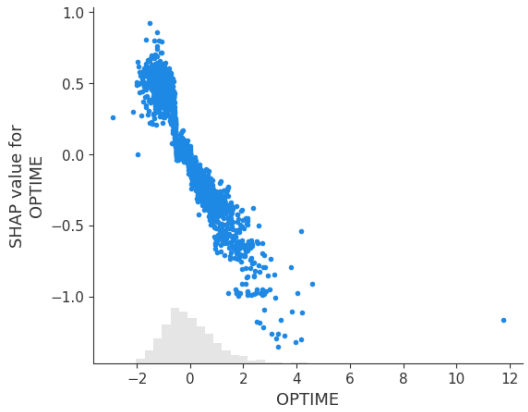
Zhang, J., Cormode, G., Procopiuc, C., Strivastava, D., & Xiao, X. (2014). PrivBayes: private data release via bayesian networks. *SIGMOD International Conference on Management of Data*, (pp. 1423-1434).

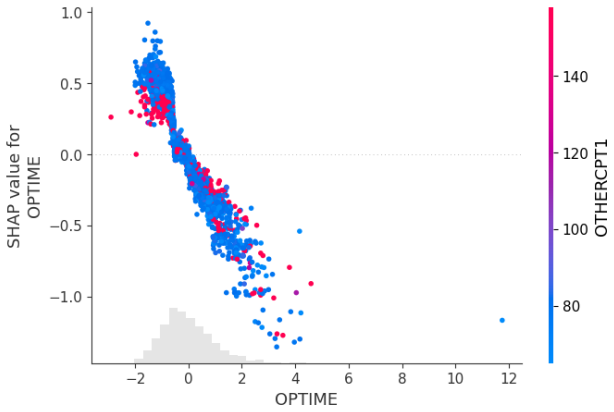
10 Appendix

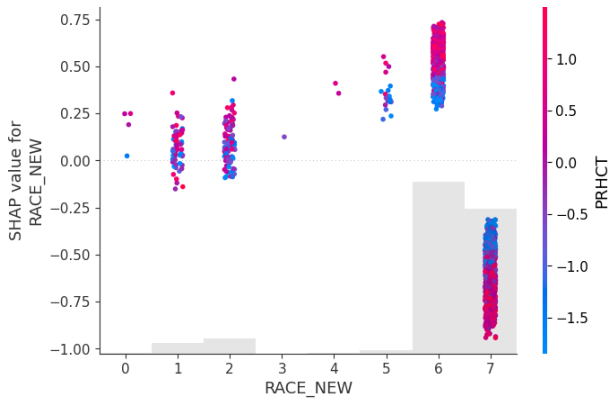
Feature Impacts analysis from Gradient Boosting Model in Iteration #3

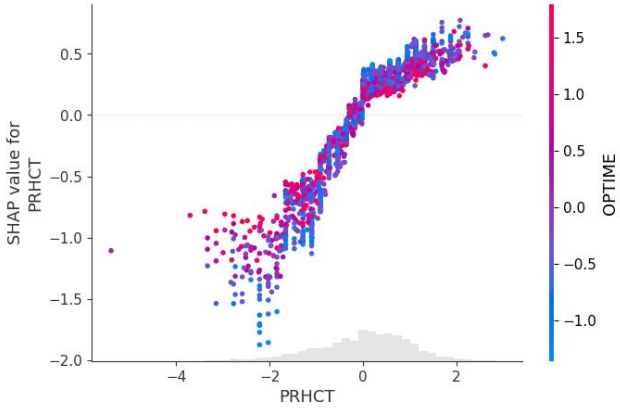


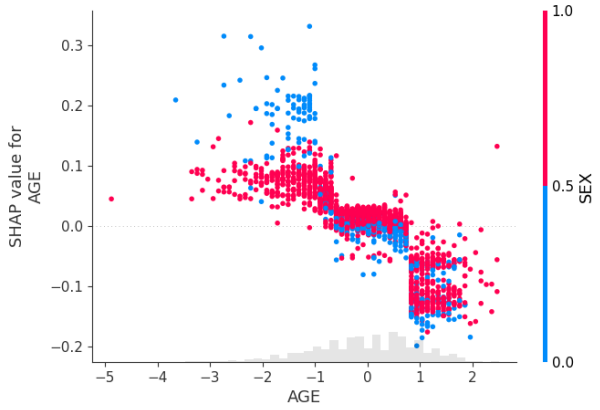


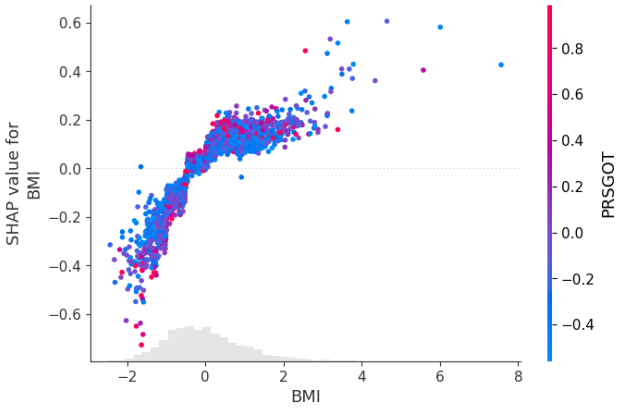

SHAP value for OPTIME
OPTIME


SHAP value for OPTIME
OPTIME
OTHERCPT1


SHAP value for RACE_NEW
RACE_NEW
PRHCT


SHAP value for PRHCT
PRHCT
OPTIME


SHAP value for AGE
AGE
SEX


SHAP value for BMI
BMI
PRSGOT