Predict Blood Donations

Ashley Handoko (axh151530), Michael Meadors (mtm150030)

Introduction

Blood donations are vital towards ensuring people across the world have access to blood that they might need. Every two seconds, blood is needed in the United States. Blood donations are used for a variety of reasons-- treating cancer patients, trauma patients, sickle cell patients, burn patients, patients with chronic diseases, and more [1].

Being able to predict blood donation could improve access to blood donations across the world.
To address the problem, we utilized the Blood Transfusion Service Center Data Set [2]. The dataset and problem was taken from a competition hosted by Driven Data. Our goal was to predict the probability that a given donor would give blood in March 2007, given information about their donation history.

Problem Definition and Algorithm

2.1 Task Definition

The dataset we used was the Blood Transfusion Service Center Data Set from the UCI Machine Learning Repository and created by Prof. I-Cheng Yeh [2]. The data in the dataset was taken from the Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The dataset is composed of 748 instances, representing 748 random donors of the blood donor database in Hsin-Chi City. There were 5 attributes and 4 features

Attributes
R (Recency) – Months since last donation (Numerical)
F (Frequency) – Total number of donations (Numerical)
M (Monetary) – Total volume of blood donated in c. c. (Numerical)
T (Time) – Months since first donation (Numerical)

Classification Attribute – Whether the person donated blood in March 2007
(Binary Variable: 1 represents donating blood; 0 represents not donating blood)

Experimental Evaluation

3.1 Pre-Processing Techniques

The dataset we utilized had no missing values, so we did not have to solve that problem. We also tried to create additional features from the already given attributes. We used different combinations of features by including some and dropping other into we found the best combination of features for a classifier.

Created Features

AD (Average Donation): Monetary / Time

TB (Time between first and last donation): Time - Recency

DR (Donation Rate (blood donated per month)) = Frequency / Volume

Average Donation and Donation Rate in particular were found to increase accuracy and decrease log loss in many cases.

We split the dataset into separate test and training sets, testing different splits to see the effect on the results.

3.2 Methodology

Based on previous projects with different assignments, we were able to identify some algorithms we believed were more inclined to more accurately predict. We focused on testing these 6 algorithms--Decision Tree, Random Forest, Neural Network, AdaBoost, Gradient Boost, and XGBoost. We used GridSearchCV to test algorithms with different parameters. Additionally, we tried a variety of preprocessing methods to see how it affected the accuracy. We split the dataset into 80:20 ratio for train:test with 5-fold cross-validation to determine accuracy to get the following results.

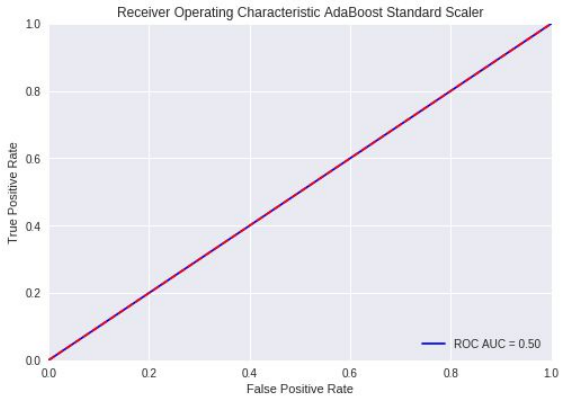
3.3 Results

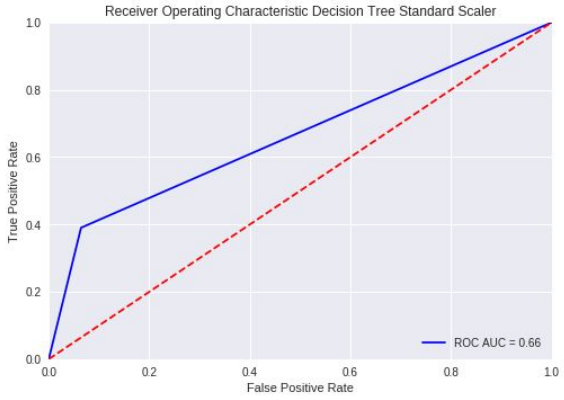
Algorithm	Preprocess	Best Parameters	Avg Precision	Avg Recall	Avg F1	Accuracy_Score	ROC_AUC
Decision Tree	None	{'max_depth': 5, 'max_features': 2, 'min_samples_leaf': 10, 'min_samples_split': 100} Features: R, F, M, T	0.77	0.78	0.77	0.78	0.67
	StandardScaler	{'max_depth': 10, 'max_features': 'sqrt', 'max_leaf_nodes': None, 'min_samples_leaf': 10, 'min_samples_split': 25}	0.79	0.80	0.79	0.80	0.66

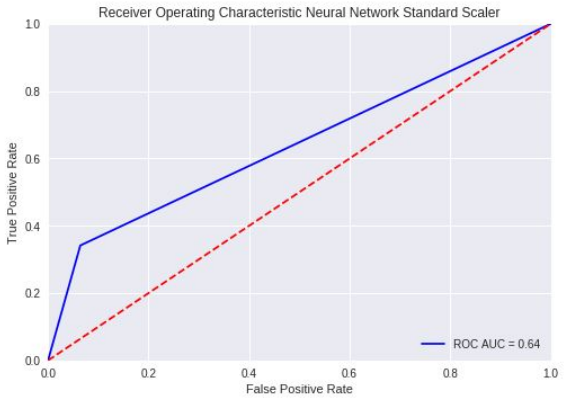
		Features: R, F, M, T, DR					
	Normalizer	{'max_depth': 5, 'max_features': 2, 'min_samples_leaf': 2, 'min_samples_split': 15} Features: R, F, M, T, AD, TB, DR	0.76	0.75	0.75	0.7733333333	0.67
Random Forest	None	{'criterion': 'gini', 'max_depth': 5, 'max_features': 4, 'n_estimators': 20} Features: R, F, T, AD, TB, DR	0.76	0.78	0.76	0.78	0.65
	StandardScaler	{'criterion': 'entropy', 'max_depth': 5, 'max_features': 3, 'n_estimators': 20} Features: R, F, M, T, AD, TB, DR	0.79	0.80	0.79	0.80	0.68
	Normalizer	{'criterion': 'entropy', 'max_depth': 5, 'max_features': 3, 'n_estimators': 10} v	0.76	0.77	0.75	0.7733333333	0.66
Neural Network	None	{'activation': 'relu', 'alpha': 0.01, 'hidden_layer_sizes': 1, 'learning_rate': 'adaptive', 'max_iter': 500} {'activation': 'logistic', 'alpha': 0.0001, 'early_stopping': False, 'hidden_layer_sizes': 100} Features: R, F, T, AD, TB, DR	0.77	0.78	0.76	0.78	0.61
	StandardScaler	{'activation': 'tanh', 'alpha': 0.0005, 'early_stopping': False, 'hidden_layer_sizes': 100} Features: R, F, T, AD, TB, DR	0.79	0.8	0.79	0.8	0.64
	Normalizer	{'activation': 'logistic', 'alpha': 0.0001, 'early_stopping': False, 'hidden_layer_sizes': 50, 'learning_rate': 'adaptive'} Features: R, F, T, AD, TB, DR	0.77	0.78	0.75	0.78	0.63
AdaBoost Classifier	None	{'algorithm': 'SAMME', 'learning_rate': 0.001, 'n_estimators': 1, 'random_state': 1} Features: R, F, T, AD, TB, DR	0.53	0.73	0.61	0.7266666667	0.50
	StandardScaler	{'algorithm': 'SAMME', 'learning_rate': 0.001, 'n_estimators': 1, 'random_state': 1} Features: R, F, T, AD, TB, DR	0.53	0.73	0.61	0.7266666667	0.50
	Normalizer	{'algorithm': 'SAMME', 'learning_rate': 0.001, 'n_estimators': 1, 'random_state': 1} Features: R, F, T, AD, TB, DR	0.53	0.73	0.61	0.7266666667	0.50

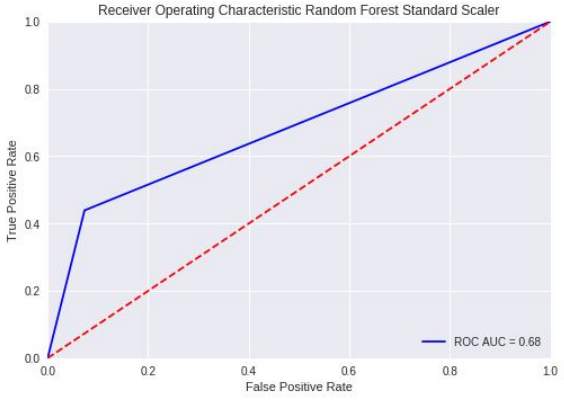
Gradient Boosting Classifier	None	{'learning_rate': 0.05, 'min_samples_leaf': 15, 'min_samples_split': 5, 'n_estimators': 20} Features: R, F, T, AD, DR	0.77	0.79	0.77	0.7866666667	0.67
	StandardScaler	{'learning_rate': 0.05, 'min_samples_leaf': 20, 'min_samples_split': 5, 'n_estimators': 50} Features: R, F, T, AD, DR	0.77	0.79	0.77	0.7866666667	0.67
	Normalizer	{'learning_rate': 0.05, 'min_samples_leaf': 20, 'min_samples_split': 50, 'n_estimators': 50} Features: R, F, T, AD, DR	0.77	0.79	0.77	0.7866666667	0.67
XGBoost	None	{'learning_rate': 0.001, 'max_delta_step': 0, 'min_child_weight': 10, 'n_estimators': 20} Features: R, F, T, AD, TB, DR	0.79	0.80	0.79	0.80	0.72
	StandardScaler	{'booster': 'gbtree', 'learning_rate': 0.5, 'max_delta_step': 0, 'n_estimators': 10} Features: R, F, T, AD, TB, DR	0.81	0.82	0.81	0.82	0.72
	Normalizer	'booster': 'gbtree'{'learning_rate': 0.5, 'max_delta_step': 0, 'n_estimators': 10} Features: R, F, T, AD, TB, DR	0.81	0..82	0.81	0.82	0.72

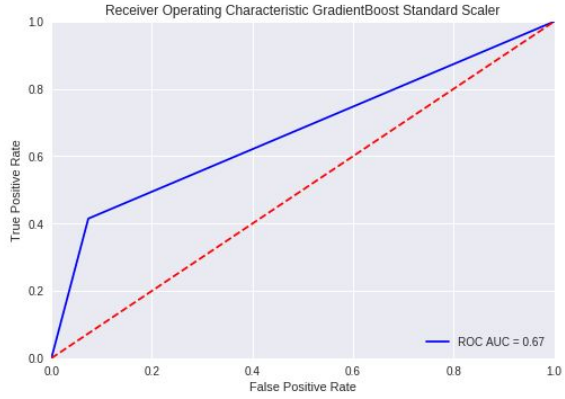
Receiver Operating Characteristic Graphs for Algorithms Preprocessed with Standard Scaler

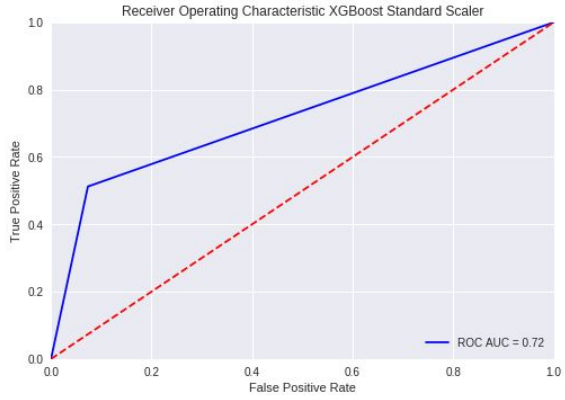












3.4 Discussion

By looking at the results and comparing all the evaluation metrics, XGBoost in combination with standard scaler and modified features ended up being the best method for the dataset.

XGBoost works as a combination of multiple trees, and looking at the results it is evident that ensemble methods lead to better performance. For example, in comparison to ensemble methods, using a neural

network did not perform as well. This is understandable due to the nature of ensemble methods. Ensemble methods work by combining weak learners to make a stronger learner. By utilizing a combination of weaker learners, ensemble methods are able to remove some of the bias and variance that other learners have from overfitting and create a more optimized model.

Utilizing the standard scaler also proved effective at improving the accuracy of the models. Preprocessing the data lended to better performance due to the size of the data and the range of each of the attributes. Typically, learners perform better when the training data is preprocessed. In our dataset, although all the attributes were integer values, the values could range from being double digits in some attributes(months since last donation, number of donations, months since first donation) and thousands in another (total volume donated). As a result of the standard scaler, all of the attributes were transformed to have standard 0 mean and standard deviation of 1.

Related Work

The same problem has been addressed in the paper, Predicting Blood Donations using Machine Learning Techniques [3]. The authors of the paper were determining whether or not accuracy improved with clustering or not across various machine learning techniques. The result of their paper was that non-clustering methods seemed to have higher accuracy versus the same techniques that were clustered. This conclusion was helpful, as it further solidified our own methods, which did not use clustering.

Another look into the problem by Ashish addressed the strong correlations between number of donations and total volume donated and number of donations and months since first donation [4]. This corresponds with our findings that creating to features to train on-- Average Donation and Donation Rate, led to higher accuracy, as they both take into consideration these attributes.

Conclusion

In conclusion, with our work we were able to predict whether someone would donate blood based on their previous donation history. Overall, we were able to determine that average donation and donation rate in particular were found to increase accuracy and decrease log loss in many cases for our model. This is relevant, because it provides insight to what might have more weight when determining that a donor donates blood again.

While our project focused of what affects a donors probability of donating, in the future more research could be done to further investigate other reasons why people choose not to donate blood regularly. This could be very important in the future as having blood donations is always necessary and can help people across the world.

Contribution of Team Members

Ashley
We both worked to fine tune the parameters for each of the algorithms. I worked on compiling the results and aggregating the data into a table, as well as generating the ROC-AUC curve graphs to add to the final report. Additionally, I helped contribute to creating the final report and doing research on blood donation need and machine learning research done on the same topic. I also, worked on adding the logic to export the data as a csv with the expected probability.

Michael
We both worked to test the best parameters and features for each of the algorithms. I was able to create new features--average donation, time between first and last donation, and donation rate that were used to increase the evaluation metrics used for the results. Also, I worked on updating the data table for the best parameters and their evaluation metrics. Additionally, I helped finalize the code for this project. I also worked on contributing to the final report and revising.

References

[1] "How Blood Donations Help Patients," American Red Cross Blood Services. [Online]. Available: https://www.redcrossblood.org/donate-blood/how-to-donate/how-blood-donations-help.html. [Accessed: 24-Nov-2018].

[2] I.-C. Yeh, "Blood Transfusion Service Center Data Set ," UCI Machine Learning Repository: Flags Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Blood Transfusion Service Center. [Accessed: 24-Nov-2018].

[3] Bahel, D., Ghosh, P., Sarkar, A., Lanham, M. and Lafayette, W. (2018). Predicting Blood Donations Using Machine Learning Techniques. [online] Semanticscholar.org. Available at: https://www.semanticscholar.org/paper/Predicting-Blood-Donations-Using-Machine-Learning-Bahel-Ghosh/8c790fc526cfd07400bfd3107bf1cd5bfb021020 [Accessed 24 Nov. 2018].]

[4] Ashish (2018). Predict Blood Donation. [online] My thoughts & learnings. Available at: https://edumine.wordpress.com/2016/08/28/predict-blood-donation-part-1/ [Accessed 29 Nov. 2018].