Predicting Hospital Readmissions:

A Predictive Analytics Approach

Jolene M Branch, Andrea J Fox, and Samuel P Sears

Data Science Program, Bellevue University

DSC630 – Predictive Analytics

Dr. Brett Werner

November 21, 2020

Executive Summary

The need for repeated episodes of hospital care disrupts the patient's life, costs the healthcare industry billions of dollars each year, places demand on hospital bed capacity, and threatens the viability of hospitals with high readmission rates in the form of Medicare linking payment to the quality of hospital care.

This paper shares the results of an attempt to improve upon previous work in predicting hospital readmissions for diabetic patients. Through use of predictive analytics and iterative model design changes via the CRISP-DM process, the team was able to develop a model that beat the industry baseline cost-per-prediction.

While this model has not been implementation tested sufficiently to prompt changes in clinical practice, it serves to add to the pool of knowledge of predictive analytics in the highly publicized and monitored field of hospital readmission prevention.

Technical Report

Introduction/Background of the Problem

The rising cost of healthcare in the United States has the potential to stall the economy as well as threaten national security (due to need to balance the budget by making cuts elsewhere). “Rising health-care costs stall Americans’ dreams of buying homes, building families and saving for retirement.” (Leonhardt, 2019). This project will look at demographic and clinical variables commonly collected on hospital admissions and determine their effectiveness in predicting hospital readmissions.

Background of the Problem

Inadequate social resources and suboptimal care transitions, especially in the form of poor or misunderstood discharge instructions, have been implicated for years as possible causes of readmission to the hospital within 30 days. Readmissions to the hospital also cause capacity challenges that can force medical centers to expand the number of beds (at great cost to the facility) (Hospital, n.d.).

Problem Statement

The need for repeated episodes of hospital care disrupts the patient's life, costs the healthcare industry billions of dollars each year, places demand on hospital bed capacity, and threatens the viability of hospitals with high readmission rates in the form of Medicare linking payment to the quality of hospital care.

Through analysis of the features associated with hospital readmissions, the goal was to determine which factors correlate strongly with hospital readmissions and create a predictive model which can accurately predict whether a patient will be readmitted based on those features.

Methods

Since this is classification problem, model selection was based accordingly. Further, data preparation and feature engineering were done with the assumption that a classification algorithm would be used.

Data Exploration

This dataset is comprised of 10 years of care at 130 US hospitals and integrated delivery networks between 1999 and 2008 (Beata, 2014). Over 50 features are present representing patient and hospital outcomes including, but not limited to, age, gender, race, admission type, discharge type, time in hospital, and whether the patient was readmitted or not. Records are indexed by unique encounter id and patients are indexed by a unique patient id.

With several categorical variables in the dataset, readmission rates for each category in the categorical variables were analyzed. The mean readmission rate for the entire data set was 11.2%. This was then compared to the readmission rates of each category to see if there were any that were significantly above or below average.

Encounters that are present in the dataset satisfied several criteria. First, the encounter had to be a hospital admission. Second, the encounter needed to have diabetes included in the diagnosis. Third, the length of stay had to be between one and 14 days. Finally, lab tests and medications had to have been given during the encounter. In all, over 100,000 records were included in the dataset. The target variable, which tracked whether the patient was readmitted, was a categorical variable including the values of no, greater than 30 days, or less than 30 days. Considering the last two variables as yes made the target variable binary.

The next step was exploration of correlations of various features against the target feature, including weight, admission and discharge, length of stay, and diagnosis. In all these explorations, there was no clear evidence of a variable or value of a variable having strong association with readmission.

Initial Data Preparation

A derived binary feature was created that differentiated between patients with a readmission within 30 days and those without one. This required combining the non-readmissions and the readmissions greater than 30 days, which were treated as a ‘0.’ This binary categorical variable was treated as the target variable for predicting whether a patient was readmitted or not.

Type conversions required included one hot encoding of categorical variables and binning ages and using the lowest age of the range as the value. One hot encoding categorical variables created a new variable for each unique value of the categorical variable. Because many of these categorical variables only had four of fewer unique values, it did not add an overwhelming number of variables. One categorical variable that did have lots of unique values was the medical specialty variable which represented the medical specialty of the observing doctor. To avoid having too many dimensions from one hot encoding of this variable, only the top 10 medical specialties were retained as values with all other values converted to an ‘other’ value. Since nearly all weight values were absent, the weight value was converted to a binary response, indicating simply presence of weight in the record or not.

Early analysis in data exploration and review of Andrew Long’s project resulted in the discovery that hospice patients (or those expected to expire within the next six months) and patients who died were contained within the dataset (Long, 2020). This discovery occurred early in the process and all instances with one of the six hospice/death classifications were removed.

This would be a critical piece of information for model deployment as the model would not be applicable or usable for these situations.

Movement back and forth between the data preparation and data exploration stages ultimately resulted in a dataset with 9 non-binary numeric predictor variables and 133 binary categorical predictors. With the target variable, the dataset used for predictive modeling contained 143 variables.

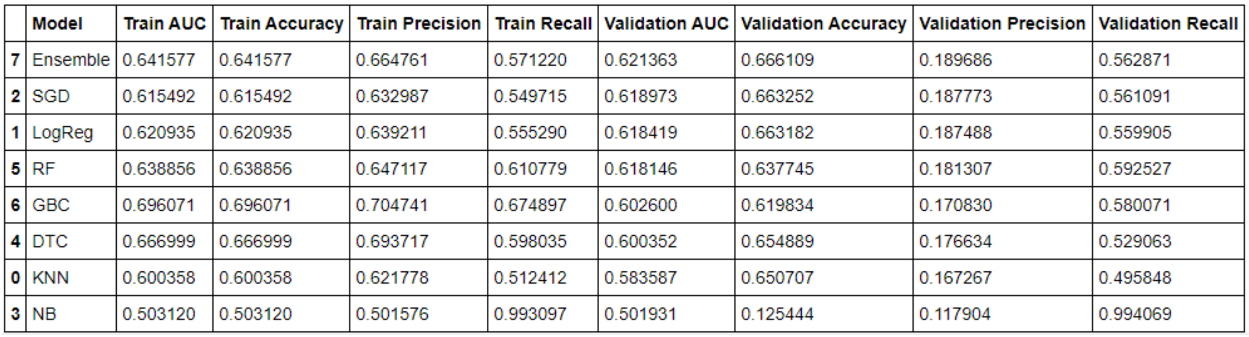
Initial Modeling

To start, the data was randomly split into 15% test, 15% validation, and 70% training. Because our data set had far more patients not readmitted than readmitted, the training data was balanced so it would have equal proportions of the target variable. The training data was then scaled using unit variance to improve the performance of the classification models. Finally, the training data was used to fit 7 different classification models including K-Nearest Neighbors, Logistic Regression, Stochastic Gradient Descent, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, and a heterogeneous ensemble of all the models. To evaluate the models, a function was created to calculate and display AUC, accuracy, precision, recall, and a confusion matrix. The fit models were used to make predictions on the training dataset to evaluate the fit of the model and then again on the test data set to evaluate the generalization of the model to new data.

The results of the initial modeling in Table 1 showed relatively promising scored for several of the models. In terms of validation AUC, accuracy, and precision, the heterogeneous ensemble model was the top performer with 0.6214 AUC, 66.6% accuracy, and 18.97% precision. However, all models except Naïve Bayes had similar performance. The Naïve Bayes classifier almost always predicted a hospital readmission, which resulted in a positive of having fewer costly False Negatives, but very poor accuracy.

Table 1

Results of Initial Modeling



	Model	Train AUC	Train Accuracy	Train Precision	Train Recall	Validation AUC	Validation Accuracy	Validation Precision	Validation Recall
7	Ensemble	0.641577	0.641577	0.664761	0.571220	0.621363	0.666109	0.189686	0.562871
2	SGD	0.615492	0.615492	0.632987	0.549715	0.618973	0.663252	0.187773	0.561091
1	LogReg	0.620935	0.620935	0.639211	0.555290	0.618419	0.663182	0.187488	0.559905
5	RF	0.638856	0.638856	0.647117	0.610779	0.618146	0.637745	0.181307	0.592527
6	GBC	0.696071	0.696071	0.704741	0.674897	0.602600	0.619834	0.170830	0.580071
4	DTC	0.666999	0.666999	0.693717	0.598035	0.600352	0.654889	0.176634	0.529063
0	KNN	0.600358	0.600358	0.621778	0.512412	0.583587	0.650707	0.167267	0.495848
3	NB	0.503120	0.503120	0.501576	0.993097	0.501931	0.125444	0.117904	0.994069

Note. The models used were Ensemble, Stochastic Gradient Descent (SGD), Logistic Regression (LogReg), Random Forest (RF), Gradient Boosting Classifier (GBC), Decision Tree (DTC), K-Nearest Neighbors (KNN), and Naïve Bayes (NB).

Updated Data Preparation

One of the first ideas to improve the models was to include the diagnosis codes from the dataset. The data had three diagnosis code fields for each observation. The issue was that there were simply too many to explode into dummy variables. Further, the diagnoses were not based on the current ICD-10 medical diagnosis categorization system which prompted retrieval of the previous ICD-9 system codes. Each diagnosis code was mapped to its respective chapter of the ICD-9 as a way of categorizing since each chapter contained similar and related diagnosis codes. Each of the diagnosis field's values was mapped to one of 19 ICD-9 chapters and the columns were exploded into additional binary dummy variables. As a result, the second iteration of modeling contained 196 predictor variables: 9 non-binary numeric, and 187 binary categorical.

Along with the updated modeling, further investigation of the business problem motivated the group to create a cost function for comparing models. Treating someone as if they are going to require admission by keeping him/her an additional day in the hospital was a false positive and was calculated by averaging the cost of an additional day in a for-profit and non-profit facility and taking 75% (or the estimated cost to the hospital) of that amount, which was $1780. Readmitting a person who was initially not expected to require readmission was treated as a false negative. This amount was found by taking the average cost for readmission (for any diagnosis, from 2016), which was $14,400.

Figure 1

Cost Function

$$Cost = (0)TP + (0)TN + (14400)FN + (1780)FP$$

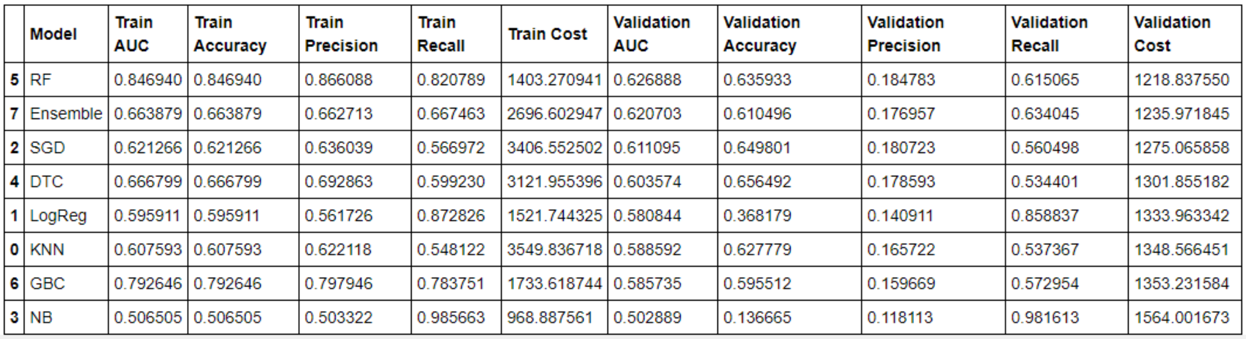
Note. This is the formula used to calculate total cost as an evaluation metric for the model.

Updated Modeling

For the second round of modeling, the same data split, balancing, and scaling took place to arrange a training, test, and validation dataset. A considerable amount of time was spent tuning hyperparameters of the various models to optimize cost. For example, in the Gradient Boosting Classifier, the number of estimators used was increased to improve performance in accordance with sklearn documentation (sklearn, n.d.). Further, for the Random Forest Classifier, the max depth of the trees was increased to allow for more splits, but a minimum impurity decrease was also set to ensure the splits were necessary. Other parameters were experimented with as well including the alpha in Stochastic Gradient Decent, class weights for Logistic Regression, and different model combinations for our majority vote heterogeneous ensemble model. As a result of our efforts tuning the models and including the diagnoses in the modeling data, we were able to get our validation AUC of our top model to improve slightly and get a cost per prediction of $1,218.84 on the Random Forest Classifier, as seen in Table 2.

Table 2

Results After Tuning and Including Diagnoses



	Model	Train AUC	Train Accuracy	Train Precision	Train Recall	Train Cost	Validation AUC	Validation Accuracy	Validation Precision	Validation Recall	Validation Cost
5	RF	0.846940	0.846940	0.866088	0.820789	1403.270941	0.626888	0.635933	0.184783	0.615065	1218.837550
7	Ensemble	0.663879	0.663879	0.662713	0.667463	2696.602947	0.620703	0.610496	0.176957	0.634045	1235.971845
2	SGD	0.621266	0.621266	0.636039	0.566972	3406.552502	0.611095	0.649801	0.180723	0.560498	1275.065858
4	DTC	0.666799	0.666799	0.692863	0.599230	3121.955396	0.603574	0.656492	0.178593	0.534401	1301.855182
1	LogReg	0.595911	0.595911	0.561726	0.872826	1521.744325	0.580844	0.368179	0.140911	0.858837	1333.963342
0	KNN	0.607593	0.607593	0.622118	0.548122	3549.836718	0.588592	0.627779	0.165722	0.537367	1348.566451
6	GBC	0.792646	0.792646	0.797946	0.783751	1733.618744	0.585735	0.595512	0.159669	0.572954	1353.231584
3	NB	0.506505	0.506505	0.503322	0.985663	968.887561	0.502889	0.136665	0.118113	0.981613	1564.001673

Random Forest Using H2O

One aspect of data preparation that was thought to be able to be improved upon was the one hot encoding of the categorical data. This was a requirement of the sklearn decision tree and random forest algorithms, but the team hypothesized that it was suboptimal. The team learned that decision trees may not perform well in these situations because the gain in purity from a split of one of these dummy variables would be marginal. Therefore, the dummy variables would be less likely to be selected earlier on in the tree (Ravi, 2019).

A package called H2O was found that allowed the team to skip the one-hot encoding step of data preparation and treat the categorical variables as truly categorical. All the same steps for the H2O Random Forest were used as for the sklearn random forest with one exception; All categorical variables were left in their original form. This resulted in less dimensions where there were 45 predictors instead of 196. Unfortunately, from a cost perspective, the H2O random forest algorithm did not outperform the sklearn random forest.

Cost Sensitive Random Forest Using CostCla

Another idea was to utilize the cost function that had been created as an input parameter to a model. In the end, the team goal had become to minimize this cost per prediction. After some digging, the CostCla model was found. This module was designed for cost-sensitive machine learning classification as part of a PhD research project by Alejandro Bahnsen (Bahnsen, 2016). The algorithm took as input a cost matrix where cost of each type of classification was given for each observation. Even though this algorithm was designed for observation-dependent cost situations where the cost function may be different for each observation, the team wanted to try implementation in the case where the cost was the same for each observation. Initially, the algorithm was tested by simply giving each observation the exact cost outlined previously. This did not result in a better cost-per-prediction than the team's current top model, so a loop was designed to try different ratios for costs of misclassifications. Interestingly, the best result came from the models in which cost was equal; the best result came when cost of false negative and false positive were both 6. Still, this was not better than sklearn's random forest in terms of cost per prediction.

Results

Models were compared by cost per prediction. Using cost as the deciding factor, the best model was the random forest, with $1218 per prediction. To compare how well this model would do in comparison to baseline models of always predicting a readmission, always predicting non-readmission, and random guessing, the cost per prediction was calculated for each. The lowest cost per prediction for these baseline models came from always predicting that the patient would be readmitted which had a cost per prediction of $1570.85. Therefore, our best model in terms of cost, the random forest classifier with sklearn, was over $350.00 more efficient at classifying the patients than the best baseline predictor.

Table 3

Baseline Models Compared by Cost-Per-Prediction

Always predict negative	**Count**	**Cost**	**Per Prediction**
TP	0	$ -	
TN	12663	$ -	
FP	0	$ -	
FN	1686	$ 24,278,400.00	
	14349	**$ 24,278,400.00**	**$ 1,691.99**
Always predict positive			
TP	1686	$ -	
TN	0	$ -	
FP	12663	$ 22,540,140.00	
FN	0	$ -	
	14349	**$ 22,540,140.00**	**$ 1,570.85**
Random			
TP	843	$ -	
TN	6331.5	$ -	
FP	6331.5	$ 11,270,070.00	
FN	843	$ 12,139,200.00	
	14349	**$ 23,409,270.00**	**$ 1,631.42**

Discussion/Conclusion

Overall, each round of modeling had slight differences in the output but there was nothing certain or validity tested enough to suggest that the team had created a prediction model that could rival those most recently published. (Liu, et al., 2020). The Random Forest Classifier had a cost-per-prediction of $1,218.84, which was $352.01 less than the best baseline predicter of always predicting a positive readmission. This was the top performing model. The second highest performing model was the Heterogeneous Ensemble of all the updated models, excluding Naïve Bayes and Random Forest, with a cost of $1,235.97 per prediction. The third highest performing model was the Cost Sensitive Random Forest using Costcla which had a cost of $1,262.19 per prediction.

Some outstanding questions that were discussed after the modeling was complete related to what the implications would be if able to successfully predict the readmission rate. Does it mean keeping the patient in the hospital for an additional day for observation? Would it mean improving upon discharge instructions or scheduling follow-up appointments? This would most likely be up to the business the project was completed for. So, the question remains unanswered. Another thing considered was if the decision was to keep the patient for an additional day, would there be an effect on the readmission rate? Would it end up costing the hospital more or less? Since the team was not held to any clear business objectives by a healthcare or insurance industry employer, the idea pursued was that it would help identify a patient(s) that was readmitted, which would then help drive change to lower that readmission rate. In their current state, the models developed by the team are not ready to be deployed "as is," and would require additional work before being considered for deployment.

Acknowledgements

The team wishes to acknowledge the University of California Irvine, as their Machine Learning Repository is the source of the dataset used for this project. The team also wishes to acknowledge Andrew Long. His code provided a reference point and guided the team's preliminary work through data preparation and modeling as well as sparking ideas to test regarding feature engineering. Also thank you to Dr Brett Werner for answering questions and teaching this term.

References

Bahnsen, Alejandro Correa. (2016). *Welcome to costclas's documentation!.* Github. https://albahnsen.github.io/CostSensitiveClassification/

Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

Hospital Readmissions Reduction Program (HRRP). (n.d.). Retrieved September 11, 2020, from https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/Readmissions-Reduction-Program

Leonhardt, M. (2019, November 04). Rising health-care costs stall Americans' dreams of buying homes, building families and saving for retirement. Retrieved from https://www.cnbc.com/2019/11/04/health-care-costs-are-preventing-many-americans-from-hitting-life-milestones.html

Liu, W., Stansbury, C., Singh, K., Ryan, A., Sukul, D., Mahmoudi, E., . . . Nallamothu, B.(2020, April 15). Predicting 30-day hospital readmissions using artificial neural networks with medical code embedding. Retrieved from https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0221606

Long, A. (2020, January 30). Using Machine Learning to Predict Hospital Readmission for Patients with Diabetes with Scikit-Learn. Retrieved from https://towardsdatascience.com/predicting-hospital-readmission-for-patients-with-diabetes-using-scikit-learn-a2e359b15f0

Ravi, Rakesh. (11 January 2019). *One-Hot Encoding is making your Tree-Based Ensembles worse, here's why?*. Towards Data Science. https://towardsdatascience.com/one-hot-encoding-is-making-your-tree-based-ensembles-worse-heres-why-d64b282b5769

Sklearn. (n.d.). *3.2.4.3.5. Sklearn.ensemble.GradientBoostingClassifier*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html