Springboard DSC Capstone Project I

Assessing Diabetes Readmission with Machine Learning

Prepared by Alexander Olden

September 2017

Introduction to the Problem

Hyperglycemia is an excess of glucose in the bloodstream that is often associated with diabetes. Unsurprisingly, management of this condition matters greatly to hospitals and physicians. Prior to the data collection and analysis of Beata Strack et alia [1, 2], however, there was little nationwide research to serve as a baseline for tracking changes in hospitalized patients with hyperglycemia. The Strack team's analysis focused on the regularity of measurement of HbA1c (glycated hemoglobin), which forms when red blood cells join with glucose in the body. This dataset was obtained from a previous study conducted to explore a similar problem. It included about 102,000 observations and 50 variables. Measuring HbA1c allows physicians to measure blood-sugar levels in patients with hyperglycemia.

Historically, measurement of HbA1c has been performed infrequently, which Strack et al. found to be the case in their own data exploration. However, with enough data collected, the question of interest became how well one could predict hospital readmission within 30 days using HbA1c measurements – along with the other variables in this dataset. In my own analysis, I made the same inquiry while also considering other machine-learning techniques and the addition of a few new variables to the dataset.

Dataset Attributes and Limitations[1]

This dataset was obtained from the initial work done by the Strack team. [2] It included about 102,000 observations and 50 variables to start. Each row represents one patient visit to a medical facility, and each column is a variable.

Several adjustments to certain variables were needed before proceeding with the analysis. The variable for time spent in the hospital specifically captures the number of days spent in a hospital, so I changed the label accordingly. Also, because the outcome of interest is whether or not a patient was readmitted within 30 days, it is important to simplify the way that variable's column in the dataset is read to accommodate machine-learning algorithms. So it takes a binary value, with 1 indicating that a patient was readmitted within 30 days and 0 indicating that a patient was not.

The dataset also came with a few potential causes of bias, such as multiple encounters per patient (A follow-up visit depends on an initial visit, for instance.) and encounters that resulted in death or discharge to a hospice facility (as opposed to readmission or traditional discharge). I removed observations that resulted in death or discharge to a hospice, since such patients would not be eligible for readmission within 30 days.

Leaving additional encounters for a patient could skew the results toward outcomes for such patients since their data would be unfairly weighted and create observations that are not independent of each other. Maintaining independent observations is an essential step in statistical analysis to rule out underlying explanatory relationships in the data. Therefore, it makes sense to keep each patient's first visit and drop the others. I kept the first visit rather than the last one since the original researchers did the same. Before removing these additional observations, though, I captured the number of encounters (i.e., visits to a medical facility) per patient as a new variable. Patients who have had more visits to a medical facility may be more likely to be readmitted within 30 days, so I wanted to see if evidence of such a trend appeared in final analyses.

[1] For a list of features and their descriptions, see the appendix at the end of this report.

Another limitation of the dataset was missing data. The weight variable had to be dropped because 97% of data points were missing this value. Two other variables – payer code (insurer code, essentially) and specialty of attending physician – also had high percentages (more than 50%) of data missing. Before removal, I checked for associations with readmission to see if the missing data occurred at random. Payer code showed no such association, but medical specialty showed some evidence of a potential relationship. While I considered exploring medical specialty further, I decided to ultimately remove it, along with payer code, since both variables had missing values for more than half of the observations, and most conclusions would not be very reliable.[2] I did not consider imputation, either, since it would have entailed more than half of these variables' data being artificially generated. In the process, I also removed other variables that were present only for identification purposes.

The variables removed in the previous steps generally had high percentages of missing data. There are other variables with much lower percentages (less than 3%) missing, which I handled differently. Instead of removing the variable itself, I simply removed the observations with missing values for those variables. Imputation did not seem necessary since plenty of data remained after removal of affected observations. Moreover, when the percentage of missing data is low for a variable, the missing data are unlikely to affect the overall distribution of that variable. Thus, dropping observations with missing values is unlikely to change the variable's predictive power.

From there, I needed to adjust diagnosis codes to accommodate the algorithms in final analyses. Three different diagnosis codes were grouped according to the condition they pertained to and thus more easily understood. For example, code numbers 360-389 represent disorders of the sense organs; instead of keeping the assortment of numbers in that range, I re-coded them all as "sense disorder" to achieve more meaningful values.

Finally, there are two variables not in the original dataset that I will use in my final approach. The first is the use of HbA1c testing not only with four different, categorical outcomes but as a binary variable, too. The original researchers found predictive value

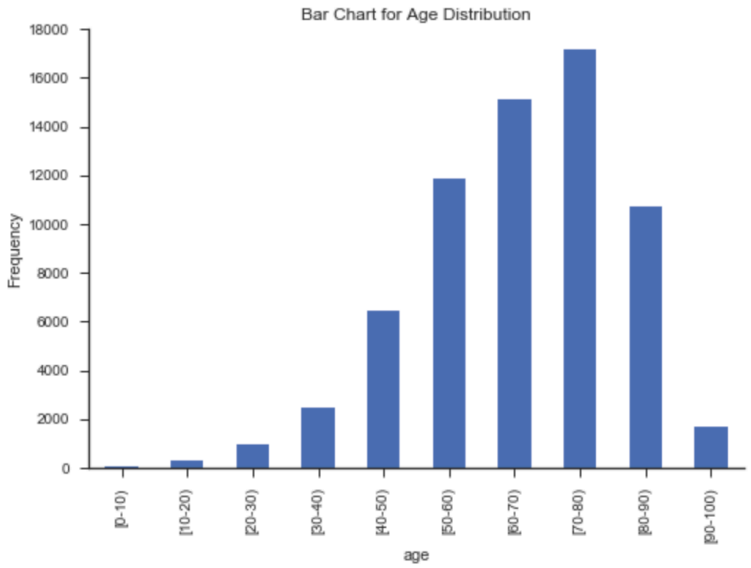
[2] The original researchers kept medical specialty, perhaps due to the potential relationship with readmission.

in HbA1c as a four-category variable; I wanted to know if it remained useful in binary form as well. The second new variable is the aforementioned number of patient visits to a medical facility.

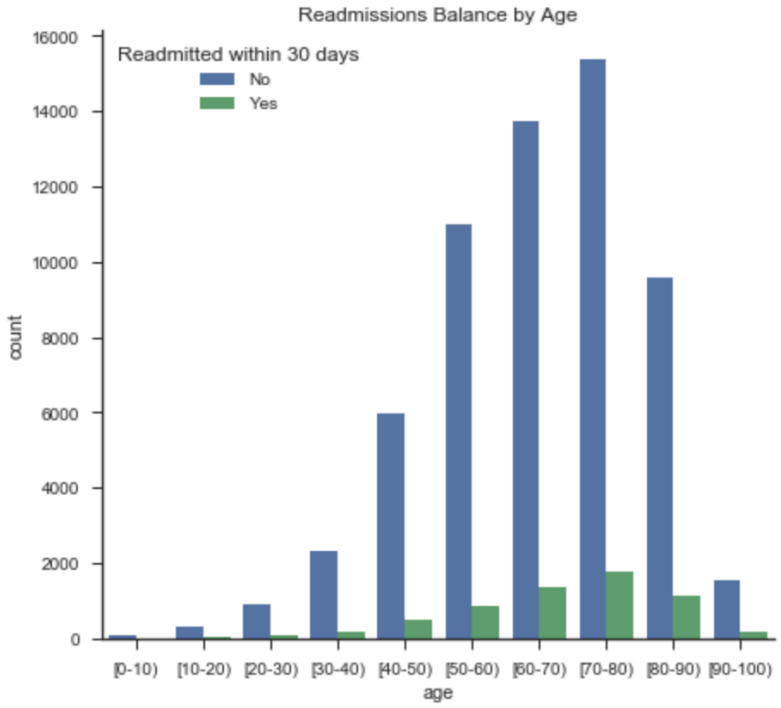
Exploratory Analysis

After looking at independent variables alone, I checked key variables for potential relationships with the outcome. Before doing so, however, I checked the balance of the outcome variable. Importantly, the ratio of negative (not readmitted within 30 days) to positive (readmitted within 30 days) outcomes was roughly 10:1. This ratio illustrates the overall likelihood of being readmitted within 30 days. It is useful context in looking at the independent variables because if the balance is much different from 10:1 within a particular variable, that variable may have significant predictive value.

To start looking at independent variables, I considered the demographic variables that were not dropped, starting with patient age. As the bar chart below shows, patient age skews toward older ages; because many other medical variables could correlate with age, it may be instructive to look at it in isolation before checking scatter plots.

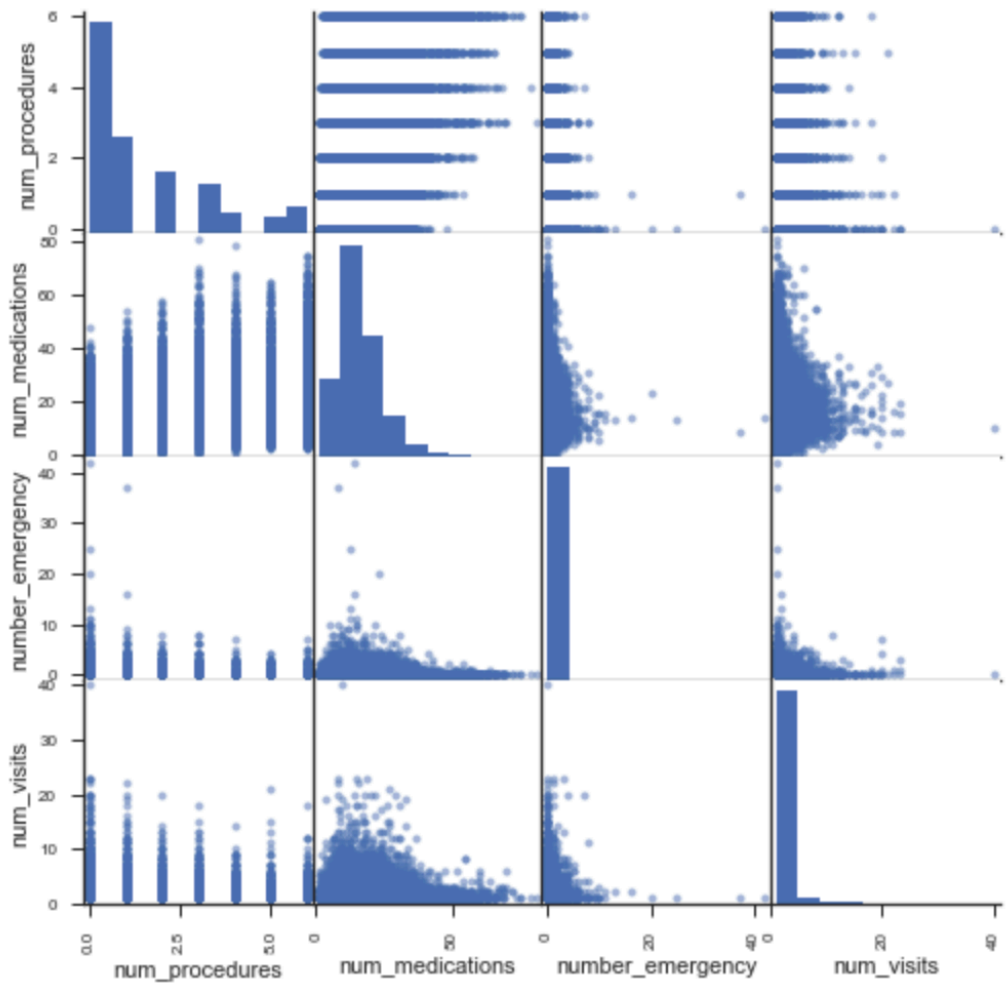


From there, I explored possible association between age and the dependent variable with the plot on the next page.



In general, the number of readmitted patients increases with the number of non-readmitted patients for each age group, so percentages of readmitted patients look similar across age groups. The rates are not perfectly constant, however: the 80-90 age bracket clearly has a higher percentage of readmitted patients. Thus, there may be some predictive value in higher age brackets. It is also worth mentioning that I checked for correlations between age and other independent variables, including number of medications, and found no evident trends. Multicollinearity did not appear to be an issue with age or the variables pertaining to patient stays in a medical facility.

My next step was producing a correlation matrix, which shows scatter plots for several pairs of variables at once. The one on the next page captures four potentially significant variables: the number of procedures, number of medications, number of emergencies, and number of visits to a medical facility.

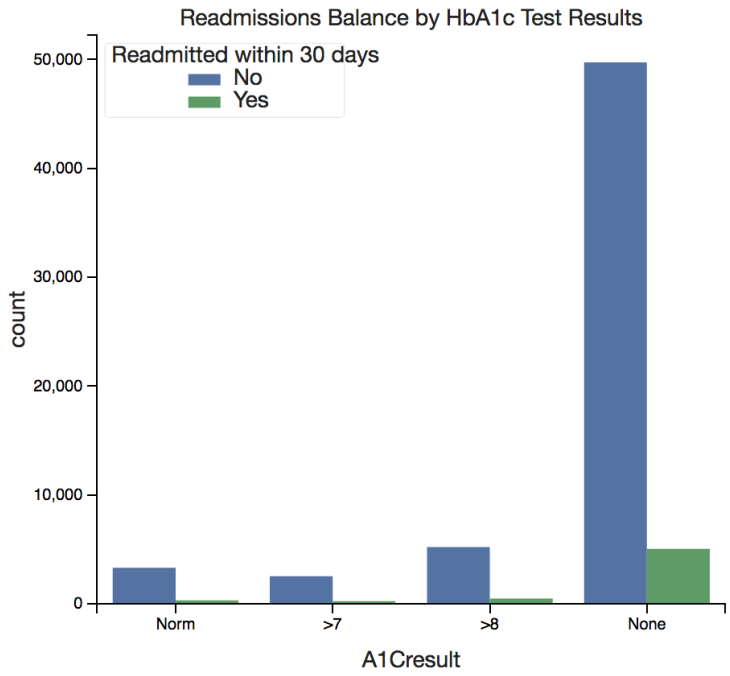


These plots show heavy clustering around low values, but little else in the way of convincing trends.[3] It is worth noting that number of emergency admissions and number of medications show a curved, downward relationship, which indicates that these two variables may be nonlinearly related. They appear to have a relationship, but it is not one that could indicate multicollinearity. Additionally, I checked for multicollinearity between a patient's number of days in the hospital and number of procedures and found no issues. (The plot is somewhat bulky and appears in my Python notebooks rather than here.)

The initial researchers were interested in determining if patients who had HbA1c levels tested were significantly less likely to be readmitted within 30 days. As such, I

[3] Note that the plots resembling histograms (on the diagonal running from top left to bottom right) are for variables plotted against themselves, hence their unique appearances.

proceeded to check for correlation between HbA1c testing and readmission. Per the Strack team, HbA1c result has four categories: 3 different levels and None, indicating that the test was not performed.



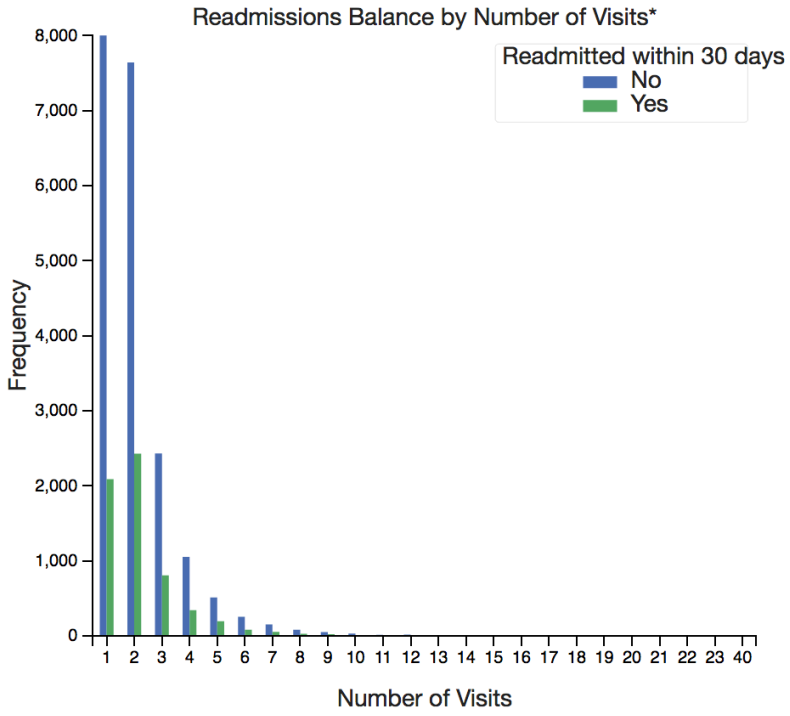
The plot shows a somewhat steady balance in readmissions across the levels of HbA1c tests, so there may be evidence of correlation. For more information, I looked at the data in binary format using a 2x2 table and a significance test for independence between the two variables.

	Readmission Within 30 Days		
HbA1c	**No**	**Yes**	**All**
No	49718	5033	54751
Yes	11052	1041	12093
All	60770	6074	66844

After seeing a possible relationship between having HbA1c levels tested and readmission, I performed a Chi-square test for independence and found that patients

with HbA1c levels tested were less likely to be readmitted within 30 days than were patients who were not tested, and that the relationship was not due to chance.[4]

Finally, I checked for a relationship between a patient's number of hospital visits and readmission. The resultant graph below shows a nonlinear, negative trend between readmissions frequency and number of hospital visits, so the newly created variable could have some predictive power.[5]



Preprocessing

To make the data ready for the analysis, I made certain adjustments to the format of the dataset. The primary task was mapping categorical variables into separate, binary variables for each of their categories. Not all variables were immediately ready for mapping, though. The number of lab procedures ranged from 1 to

[4] Another limitation is that some conventionally continuous variables appear as categorical ones. This restriction makes further hypothesis tests difficult in exploratory analysis.
[5] The graph produced in my Jupyter notebook is interactive and allows for zooming to help address the scale issues encountered here.

132, with very few patients at certain numbers. To keep a desirable ratio (10:1, or higher) of new binary variables to actual observations, I binned the number of procedures so that there are only 13 new variables. There were also a few ID variables (e.g., admission type ID) that are functionally categorical but have numeric values, so I changed their variable types to categorical, too. These steps expedited the one-hot encoding process, which I next employed to turn all categorical variables into n-1 binary variables (where n is the number of unique values a variable can take).

Logistic Regression and Results

Logistic regression is frequently used in classification problems. It comes with the benefit of variable coefficients that indicate the log odds, which are useful during interpretation. Additionally, since the Strack team used logistic regression, my doing so here will enable direct comparison of results. I started model construction by splitting the dataset into training and test components: 80 percent became training, and 20 percent became test. The model was built using training data and then checked on test data to see if it generalized well to new datasets.

In choosing a logistic model, I implemented certain settings to improve performance and relevance. First was the consideration of possible values for the regularization parameter, which is used when data cannot be linearly separated into two categories. A regularization parameter helps compensate for this problem by introducing an amount of bias to reduce variance so that the model that is not too specific to training data while still generalizing well to new data. I employed an L2 regularization penalty to further control over-fitting, too. Finally, to help address the imbalance of outcomes within readmissions, I added weights the positive (.9) and negative (.1) cases.

The initial results for logistic regression reported accuracy scores of 80.85 percent and 82.44 percent for the training and test datasets, respectively. These results were a promising start in terms of performance and generalization. However, accuracy scores do not tell the entire story because they do not offer detailed information about

performance with negative and positive cases separately. To that end, I looked at precision and recall scores as well.

For reference, precision reflects the percentage of predictions that are correct. Recall captures the percent of true positives that are correctly classified. Among positive cases (readmitted within 30 days) in my initial logistic model, the results were not very good: precision was .26 and recall was .5. The results for negative cases were much better, but we are more concerned with positive cases in this analysis. And a strong performance on negative data indicated that the model was struggling with class imbalance. Such performance is well summarized in a confusion matrix like the one below for the model's performance on training data:[6]

Actual	Model Prediction: Training Data		
	Not Readmitted	Readmitted	Total
Not Readmitted	40694	7922	48616
Readmitted	2317	2542	4859
Total	43011	10464	53475

This table shows the raw numbers behind the precision and recall rates and corroborates the model's poor performance. Of note, the confusion matrix for test data – shown on the next page – was similarly discouraging.

[6] A quick reference guide for confusion matrices:
- In the top-left quadrant is the number of observations classified as not readmitted within 30 days that were in fact not readmitted within 30 days. This is the true negative count.
- In the top-right quadrant is the number of observations classified as readmitted within 30 days that were in fact not readmitted within 30 days. This is the false positive count.
- In the lower left quadrant is the number of observations classified as not readmitted within 30 days that were in fact readmitted within 30 days. This is the false negative count.
- In the lower right quadrant is the number of observations classified as readmitted within 30 days that were in fact readmitted within 30 days. This is the true positive count.

Actual	Model Prediction: Test Data		
	Not Readmitted	Readmitted	Total
Not Readmitted	10408	1746	12154
Readmitted	602	613	1215
Total	11010	2359	13369

In order to address the issues caused by imbalanced data, I considered random undersampling of negative cases as well as oversampling of positive cases, which are implemented in the “imbalanced-learn” package [3]. Both approaches improved the model, though SMOTE was better overall, which makes sense because it ultimately generates more data on which to train a model. Model accuracy dropped for both test and training data – 72.5 and 72.8 percent, respectively. However, this drop is a worthwhile tradeoff in exchange for improved precision and recall. Here are the key metrics (in percentages) for the training and test datasets using SMOTE:

	Training	Test
Precision	75.2	75.5
Recall	67.5	67.5

The closeness between test and training metrics likely comes from the fact that SMOTE, by definition, builds new data points by combining near neighbors. This process can effectively produce a test dataset that aligns better with the model than the training set does. With an improved model in place, I proceeded to generate coefficients for the variables in the logistic model. Each coefficient represents the log odds of readmission within 30 days.

Variable	Coefficient
num_visits	0.589403
number_inpatient	0.284375
discharge_disposition_id_3	0.180511
discharge_disposition_id_22	0.138675
first_diag_injury	0.082984

My results for logistic regression differ from those of the Strack team in a few ways. For one, my added variable of number of hospital visits showed the most predictive value for readmission, along with number of inpatient visits. In contrast, the original researchers found a medical specialty of “none” to have the highest coefficient (.463). They also found discharge disposition ID of “other” to have a coefficient of .302, which may well pertain to my third and fourth coefficients. The Strack team categorized discharge IDs into “home” and “other” but did not offer more information on categorizing ID numbers from the original data, so I was unable to follow suit.

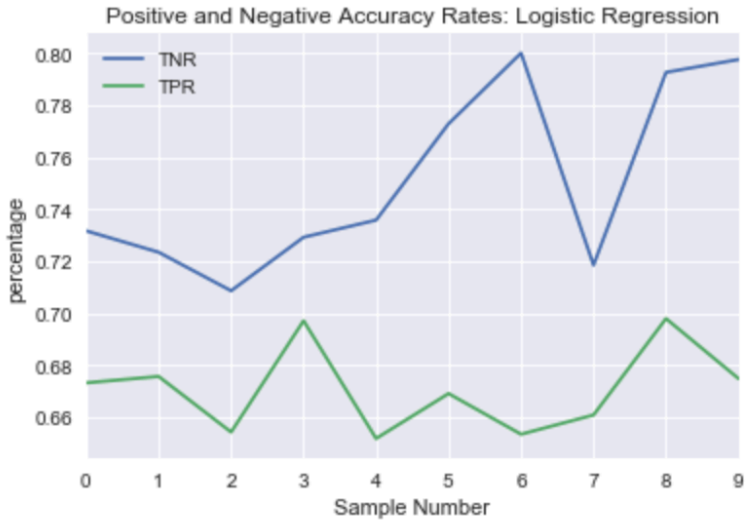
The original research also focused on HbA1c monitoring, which was associated with patients not being readmitted. For reference, the variables most predictive of not being readmitted in my model are reproduced here:

Variable	Coefficient
first_diag_respiratory	-0.418719
first_diag_digestive	-0.306081
discharge_disposition_id_25	-0.293866
insulin_No	-0.278398
second_diag_digestive	-0.275262

These results do not square with those of the original researchers, who found a diagnosis of musculoskeletal problems (coefficient of -.627) and high levels of HbA1c (-.398 and -.579 for two different measured levels) to have high coefficients. These different results may well have to do with my accounting for imbalanced data as well as the fact that I did not categorize numerical variables as the Strack team did. The latter difference may have created right-skewed variables instead of normally distributed ones, which can improve stability in logistic regression.

To ensure that the SMOTE and random undersampling methods are consistently reliable, I also performed the procedures on loops with repeated samples. The plot on the next page summarizes the results for logistic regression using SMOTE, which performed better than random undersampling did. TNR stands for True Negative Rate, and it is defined as the number of correctly classified negative cases, divided by the

total number of negative cases. TPR stands for True Positive Rate, and it is defined similarly for positive cases. By definition, TPR is the same as Recall.



These plots capture the results of ten trials with SMOTE. There range of results looks good for positive predictions, staying within a range of 5 percentage points. But there is a little more variation (9 percentage points) for the true negative rate, although the peaks and valleys seem to align. There appears to be some “consistent inconsistency,” which perhaps stems from extra noise in predicting the negative case.

Tree-based Methods and Results

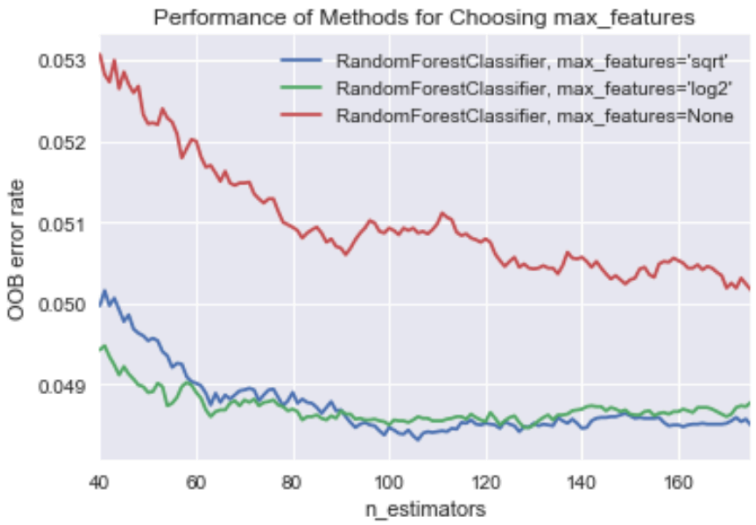
After applying logistic regression, I explored tree-based approaches – Random Forests in particular. Algorithms relying on decision trees often perform well with classification problems with many categorical variables, and I used random forest because it offers the benefits of tree-based analysis with the additional support of an ensemble approach (multiple, randomly created trees). Decision trees work kind of like a game of 20 Questions. They consider each variable in a dataset and find its optimal "split point," where the values in a given variable can be most accurately split into each category of the dependent variable. Variables with more predictive power are used

earlier in the process. But individual trees are weak learners, meaning that their accuracy is limited (often not much higher than 50%). Thus, ensemble approaches, like random forest, are often used. Random forests combine multiple trees, combining the "knowledge" of many weak learners to create a stronger learner, which is much more accurate.

In building the initial model on training data, I implemented weights for positive and negative cases to address the imbalance in numbers of observations readmitted within 30 days. The model looked good in its initial performance with the test data, yielding a global accuracy of 90.84%. As with logistic regression, I then dug deeper by looking at precision and recall values for the positive case, which came out to 25.3% and 1.6%, respectively. Also like logistic regression, the initial random-forest model clearly needed improvement.

With the assumption that the model's problems stemmed from the class imbalance, I next used random undersampling and SMOTE oversampling in an effort to make improvement. Again, the SMOTE model was superior, and likely for the same reasons. Model accuracy came out to 95.12%, and precision and recall were 99.2% and 90.9%, respectively. These numbers were of course much better, but more work was needed to verify the model's usefulness. Before finalizing the choice of model, I still needed to explore a few parameters of the random forest algorithm, including number of estimators (number of trees in the forest), maximum depth (of a tree), and maximum number of variables to consider when looking for the optimal split. These steps helped protect against overfitting, which a nearly perfect precision score may signal.

To start, I looked at the maximum number of variables, using out-of-bag (OOB) error as a metric to compare different methods of choosing the number of variables. [4] OOB error reflects the average error for each classifier calculated using predictions from the trees that do not contain the given classifier for their respective bootstrap sample.



The log2 approach appears to be the best choice for the maximum number of variables, as it has a low error rate and stabilizes early on. Additionally, that stabilization occurs with a value of 65 for the number of estimators, so I proceeded with 65 for n_estimators. Although OOB error rate is lower for higher numbers of estimators, it is preferable to sacrifice that small in gain in favor of a much lower number of estimators. This tradeoff prevents overfitting and helps the model generalize.

Next, using the number of variables, I calculated the maximum depth by taking log base 2 (since our outcome is binary) of the number of variables. This step produced a maximum depth of 7. With these three parameters set and the model trained, I ran the final model for random forest on the test data. It yielded an accuracy score of 91.06% (vs. 72.8 for the logistic regression) with respective precision and recall scores of 91.4% and 90.7%.

The ten variables with the highest predictive values for readmission within 30 days are reproduced on the next page for each model. (In the interest of brevity, I did not reproduce full models, but they are readily accessible in my Python notebooks.) In random forest, importance is measured by the Gini score, which reflects the average gain in purity by splitting on a given variable, with respect to readmission within 30 days. That is, Gini importance measures how well a variable minimizes the probability of misclassification. (For reference, its maximal value approaches 1, and its minimal value is 0.)

Logistic Regression	
Variable	Coeffi-cient
num_visits	0.5894
number_inpatient	0.2844
discharge_disposition_id_3	0.1805
discharge_disposition_id_22	0.1387
first_diag_injury	0.0830
discharge_disposition_id_5	0.0765
age_[70-80)	0.0728
num_lab_procs_[61-70]	0.0698
age_[60-70)	0.0648
glipizide_Steady	0.0593

Random Forest	
Variable	Import-ance
num_visits	0.2438
race_Caucasian	0.0452
admission_source_id_7	0.0444
number_inpatient	0.0378
third_diag_other	0.0339
discharge_disposition_id_3	0.0305
discharge_disposition_id_22	0.0274
age_[50-60)	0.0270
age_[70-80)	0.0236
first_diag_neoplasms	0.0211

Looking at these two sets of outputs together, one can see that both models favor a few variables: number of previous hospital visits on a patient's record, number of inpatient visits for the previous year, age being in the 70s, and certain discharge disposition IDs. Neither model chose HbA1c results as a significant predictor of readmission. However, it does have some predictive value, as its coefficient was greater than zero.

Given the choice between these two models, I would choose the random forest. Its performance metrics were better than those of the logistic regression model, and I suspect that such a dynamic has to do with random forest being better able to handle continuous variables, which I did not transform to make normal in the logistic regression initially. Additionally, because it is an ensemble method, random forest offers the benefits of more randomization in model construction, which is not exploited in the logistic regression method.

Since successful adjustments for the imbalanced dependent variable have been made, it is helpful to repeat those processes a few times to observe the variability of random undersampling and SMOTE. By repeating the process a few times on a loop and plotting results, we can confirm the validity of using SMOTE in this case.

The percentage on the vertical axis reflects the relevant rate (true positive or true negative). Overall, the plot shows that rates do not vary too much; the range for true negative rate is about 3%, and the true positive rate varies by maybe 1%. The SMOTE process appears to be consistent and reliable.

Summary and Future Work

In future work, I would transform the continuous variables so that their distributions would be closer to normal, as such distributions can stabilize results in logistic models. The original researchers chose logistic regression for their model, and perhaps it is fair to assume they prepared the independent variables accordingly. Indeed, certain numerical variables are binned for logistic regression in the Strack team's work. Relatedly, I would like to develop an ensemble model that combines the logistic regression and random forest by averaging predicted probabilities from each model.

However, my own analysis favors random forest, as mentioned. My models selected primary diagnoses of neoplasm issues (random forest) and injury (logistic) as

important predictors. These outcomes align with those of the original research, which found primary diagnosis to be an important contributor to classification.

There is also the topic of higher scores in the SMOTE models than in the ones that used random undersampling. I hypothesized that such results may stem from a higher amount of overall data, but it also pertains to the fact that the test set in that situation has many synthetically generated positive cases spawned from the original ones. Training a model on this many “original” positive cases leads to computing performance metrics on the test data that look very similar to the data on which the model was trained. (This may also factor into similar test and training metrics I observed.) This issue could be avoided by splitting the data before applying methods for imbalanced data, but that step comes with the cost of having fewer original positive cases for learning. Ultimately, one wants to strike a balance between these two goals, so further exploration could help.

To return to the work of the original researchers, there are a few other related items I would hope to explore in future research. One is the use of interaction terms in the model, particularly with HbA1c testing and primary diagnosis, which proved to be important in my models as well as those of the Strack team. Additionally, I would like to try some of the approaches used in the original research that I did not look into this time through, including the isolation of readmission probability according to different values within categorical variables such as HbA1c result and the binned age variable. Finally, I would want to determine significance levels of coefficients in the logistic regression for additional comparison and model construction. If a coefficient is high but has great variance and low significance, it is likely not as valuable as advertised.

Recommendations

Clients in this project are stakeholders associated with healthcare institutions interested in studying and reducing the readmission rates for diabetes patients. The results of my analysis can potentially help them determine whether monitoring HbA1c

levels should be a significant part of treatment. Ultimately, that particular variable did show as much importance in my analysis.

Said stakeholders might also be interested in monitoring the new variables that I considered, such as the previous number of patient visits. While it may not seem surprising that older patients or those with many previous visits are likely to be readmitted, it is important to know if a variable like that one has more predictive value than other ones being studied do. On a related note, I suggest getting more information about the discharge disposition IDs, as those variables proved important in both models.

References

1. 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.

2. This dataset was obtained from a previous study done to explore a similar problem. It included about 102,000 observations and 50 variables. More information is available here: http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008.

3. Guillaume Lema, Fernando Nogueira, and Christos K. Aridas. “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.” Journal of Machine Learning Research, vol. 18, no. 17, pages 1-5, 2017. http://jmlr.org/papers/v18/16-365.html.

4. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python.” Journal of Machine Learning Research, vol. 12, pages 2825-2830, 2011.

Appendix: Variables and Their Descriptions

Variable	Description
encounter ID	unique identifier of a patient visit
patient number	unique identifier of a patient
race	values: Caucasian, Asian, African American, Hispanic, and other
gender	values: male, female, and unknown/invalid
age	grouped in 10-year intervals up to 100
weight	patient weight in pounds
admission type	emergency, urgent, elective, newborn, and not available
discharge disposition	discharged to home, expired, and not available
admission source	physician referral, emergency room, and transfer from a hospital
time in hospital	number of days between admission and discharge
payer code	Blue Cross/Blue Shield, Medicare, and self-pay
medical specialty	specialty of the admitting physician
number of lab procedures	
number of procedures	number of procedures (other than lab tests) performed
number of medications	number of medications administered during the encounter
number of outpatient visits	for year preceding the encounter
number of emergency visits	for year preceding the encounter
number of inpatient visits	for year preceding the encounter
diagnosis 1	primary diagnosis, 848 distinct values/codes
diagnosis 2	secondary diagnosis, 923 distinct values/codes
diagnosis 3	additional secondary diagnosis, 954 distinct values/codes
number of diagnoses	
glucose serum test result	indicates the range of the result or if the test was not taken
A1c test result	indicates the range of the result or if the test was not taken
change of	dichotomous indicator of change in diabetic medications

medications	
diabetes medications	dichotomous indicator of diabetic medication being prescribed
24 features for medications	one variable exists for each medication; values indicate increased, decreased, held, or no usage
readmitted	days to inpatient readmission; "no" for no readmission