Kyle Stoudt

Final Report:
Predicting Diabetic Patient Hospital Readmissions

Problem Statement:

Generally speaking, the hope of healthcare is to improve the health of a patient, many times by ‘reversing the curve’ of a patient’s declining health. This is especially appropriate in the case of illness or disease. A key indicator that this goal has *not* been achieved with a patient is when they need to be treated a second time (readmission) within a relatively short time frame, thereby reusing hospital resources and potential beds. A data-driven analysis of the primary contributors to diabetic patient readmission and fitting of a classification model could provide great value to hospitals in the form of readmission prediction and treatment plan adjustment for these individuals. This project is meant to be a proof-of-concept for a Machine Learning model that could assist Hospital staff in identifying and treating patients (in this case, *diabetic patients*) who are at-risk for readmission.

The project began with the question: “How could hospitals leverage available diabetic patient data to determine the major contributing factors to readmission, when a current patient is at risk for readmission, and if the hospital is at risk of numerous impending readmissions?” The answer to this question could provide hospitals with valuable insights into the well-being of their patients, signaling them to act in ways that would reduce readmission rate. The valuable hospital resources that would have been reused could be freed up for use with other patients which would save time and money, while improving the overall efficacy of the healthcare services provided at these hospitals. Additionally, hospitals could utilize the tool to better-forecast bed occupancy percentages for the month to come.

The Dataset:

The primary dataset was originally extracted from a large clinical database by a team of professionals while working on another article: *“Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records”.*[1] This is their description of the initial extraction:

“The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria.

(1) It is an inpatient encounter (a hospital admission).
(2) It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
(3) The length of stay was at least 1 day and at most 14 days.
(4) Laboratory tests were performed during the encounter.

(5) Medications were administered during the encounter.
The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medication, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc."

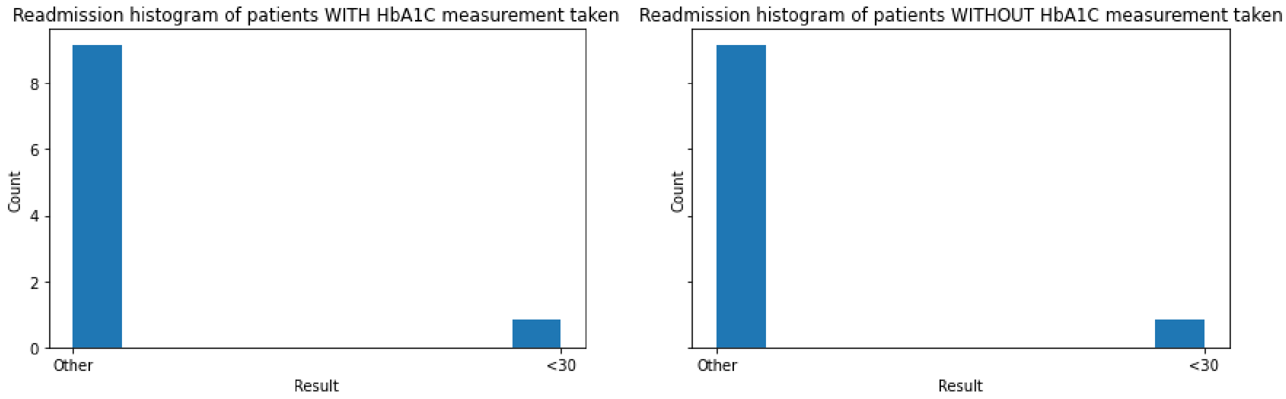
The dataset contains medical data and hospital readmission data for diabetic patients. The full table is made up of about 100,000 records and 55 fields, including a column containing NO, >30, or <30 to indicate if the patient was not readmitted, readmitted after 30 days, or readmitted within 30 days. This column was adjusted down to two categories: '<30' and 'Other', in conjunction with the academic article[1]. This was done for insurance billing purposes and represents the specific situation of interest to these hospitals.

Exploratory Data Analysis:

For diabetic patients, the HbA1C measurement can be of great importance, as it correlates with blood sugar levels. Despite this, 83.3% of the values in the 'A1Cresult' column were 'None', indicating the measurement is rarely taken at all. A quick calculation to compare the effect this has on readmission rates produces the following:

With measurement: **8.18%** readmitted within 30 days.
Without measurement: **8.48%** readmitted within 30 days

Fig. 1 – Exploring the HbA1C measurement's impact on readmission



We can see here that the readmission rates are slightly higher among those patients who have not had an HbA1C measurement taken versus those who have had this measurement taken. This is roughly in-line with the conclusions made by the aforementioned academic research article, which state that "With respect to readmission and taken as a whole without adjusting for covariates, measurement of HbA1c was associated with a significantly reduced rate of readmission"[1].

Further analysis of the dataset provides little insight into what features may be primarily responsible for these readmissions, though other expected correlations can be observed. For instance, there are multiple features of the dataset that share some colinearity as they represent things like patient engagement or the amount of hospital interest there is in a patients' condition. A good example of this is how we observe the number of medications sharing a positive correlation with the number of lab procedures performed (Fig. 2.1). Additionally, we see an expected increase of readmission rate along with age (Fig. 2.2).

Fig. 2.1 – Correlation in the dataset between num_medications and num_lab_procedures

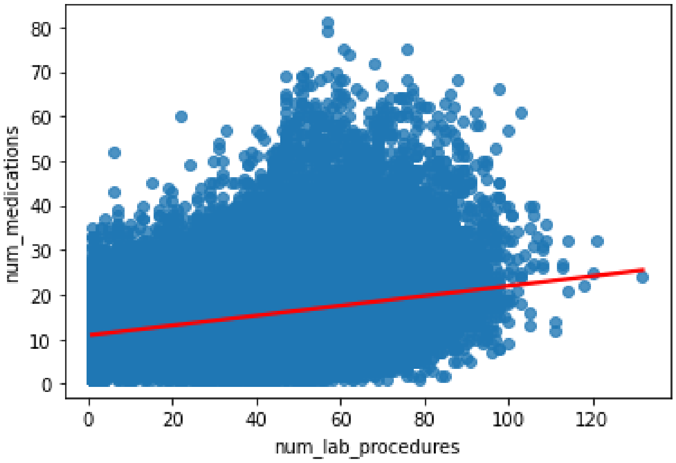
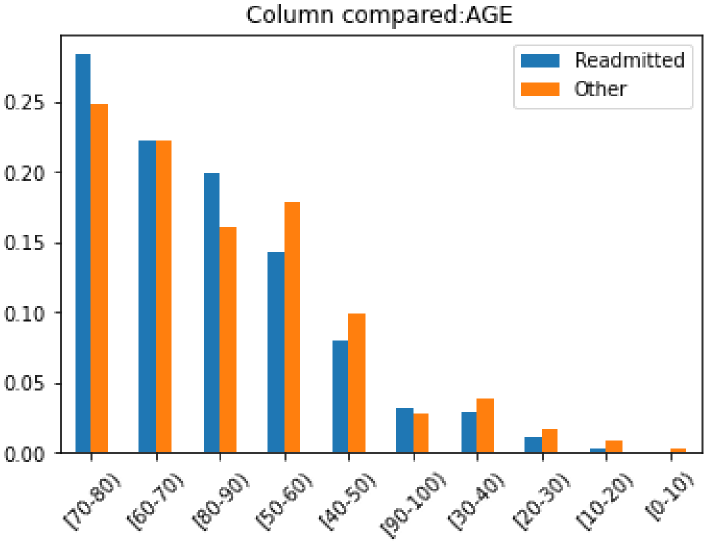


Fig. 2.2 – Higher readmission rates seen among higher age groups



As a result of the EDA stage as a whole, we can see that the dataset contains *a lot* of information, none of which seems to show a clear problem with *treatment*. That is, the dataset doesn't clearly tell us *what to do.* Because numerous factors contribute to this and equate to a small difference in a patient's odds of readmission. It would be impossible to manually identify and appropriately-treat at-risk patients. The application of a binary classification algorithm could be very helpful with this problem.

Data Preparation and Modeling:

For simplicity, the dataset was treated as primarily categorical, using a StandardScaler() and One Hot Encoding across all columns of the dataset. This could easily be improved in the future (see Future Scope). A train-test split was then performed with a 25:75 test-train ratio.

In total, 7 different algorithms were tested with relatively 'out-of-the-box' hyperparameter settings. 5-fold cross-validation was performed on the training data, using the weighted F1-score as the scoring metric (Table 1).

Table 1 – 5-fold CV Scores with initial models – 3 best models highlighted in bold

Model	**Weighted F1-Score**
Logistic Regression	0.8746337107535922
K-Nearest Neighbors	**0.8750672328312273**
Decision Tree	0.8549024876952227
Random Forest	**0.8758155785678916**
Support Vector Machine	0.8673007422358227
Gradient Boosting	0.874575455730425
XGBoost Classification	**0.8765311717848258**

K-Nearest Neighbors, Random Forest, and XGBoost's Classification algorithms were all chosen to be further developed due to the results seen in Table 1. It should be noted here that these algorithms were performed with little to no specification beyond the default settings and the data itself is very imbalanced, with a DummyClassifier achieving a weighted f1-score of ~0.875 while predicting 'Other' exclusively. Therefore it is possible that some of these other algorithms could perform better with more imbalance-focused pre-processing and hyperparameter tuning. See Future Scope for further discussion.

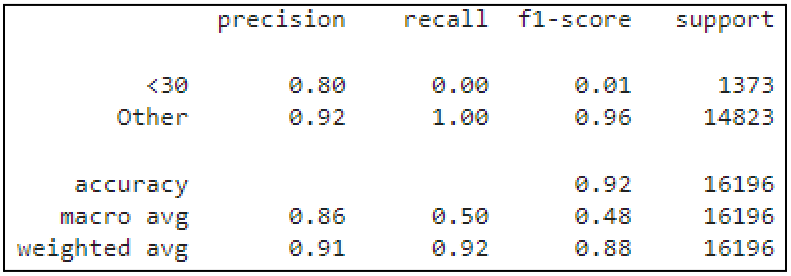
These top three models were used in 5-fold RandomizedSearchCV to tune hyperparameters. The models were then evaluated on accuracy with the 'best parameters' in place. The results of this survey can be found in Table 2. The argument 'scale_pos_weight' in the XGBoost algorithm handles the class imbalance of the data by scaling the weight of the target class '<30' to compensate for the lower frequency. I should also note that the 1.0 training AUC for the Random Forest model indicates that the model is overfitting and would poorly generalize to new data. This could be remedied with further Hyperparameter tuning, though it seems XGBoost is the clear favorite with much better prediction power on the positive class than the other algorithms.

Table 2 – Top 3 models compared after Hyperparameter Tuning

Model	Accuracy Score	Best Hyperparameters	Training AUC
K-Nearest Neighbors	0.915982299063497	{'p': 1, 'n_neighbors': 28, 'leaf_size': 41}	0.721088731825609
Random Forest	0.915982299063497	{'n_estimators': 1800, 'max_depth': 90}	1.0
XGBoost	**0.916002881547803**	**{'scale_pos_weight': 10.0, 'n_estimators': 600, 'max_depth': 7, 'learning_rate': 0.1}**	**0.882231034570797**

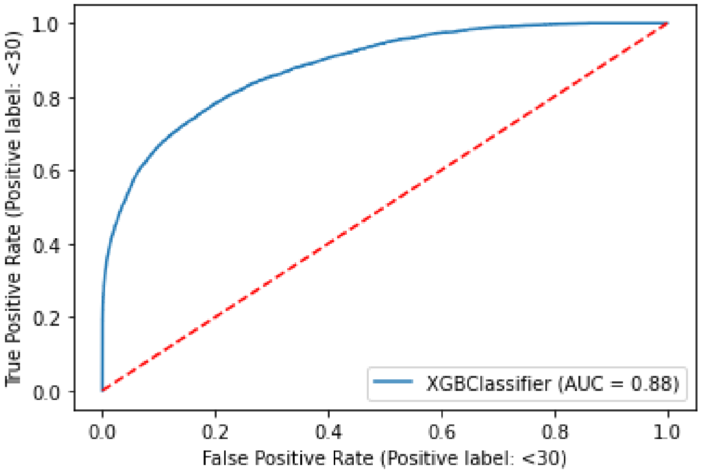
With XGBoost emerging as the favorite, the ROC curve and the detailed classification report were produced to see how the model could be tuned to be a more-accurate predictor (Fig. 3.1 & 3.2).

Fig. 3.1 – Classification Report for XGBoost



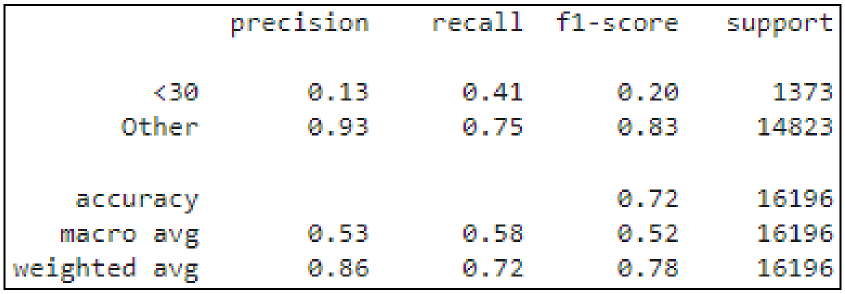
```
              precision    recall  f1-score   support

         <30       0.80      0.00      0.01      1373
       Other       0.92      1.00      0.96     14823

    accuracy                           0.92     16196
   macro avg       0.86      0.50      0.48     16196
weighted avg       0.91      0.92      0.88     16196
```

Fig. 3.2 – ROC Curve for XGBoost



The ideal threshold for classification was determined to be ~0.012020 by optimization of the geometric mean between true-positive-rate and false-positive-rate. The application of this new threshold to the model produced a much better f1-score for the target class (Fig. 3.3).

Fig. 3.3 – Classification Report for XGBoost after Thresholding



```
              precision    recall  f1-score   support

         <30       0.13      0.41      0.20      1373
       Other       0.93      0.75      0.83     14823

    accuracy                           0.72     16196
   macro avg       0.53      0.58      0.52     16196
weighted avg       0.86      0.72      0.78     16196
```

In regard to feature importances in the model without thresholding, there are upwards of 2,000 to evaluate and so the visualizations don't tell us too much (Fig. 4.1). However, it makes sense to see the diagnosis codes playing a larger role, as well as 'surgery' and 'discharge_disposition_id' as they indicate the intensity of the visit and the patients' nature of departure, respectively.

Fig. 4.1 – The Spread of feature importances doesn't tell us much

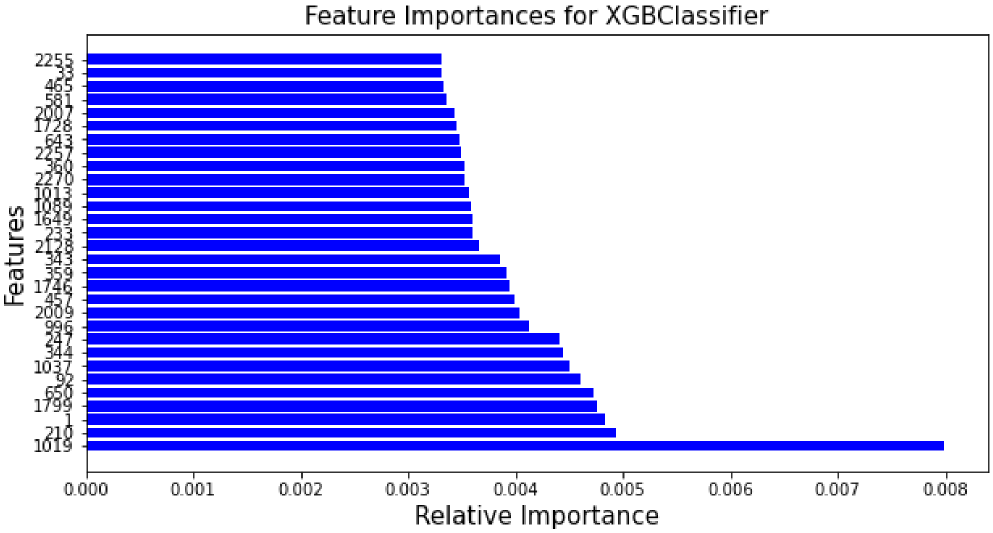
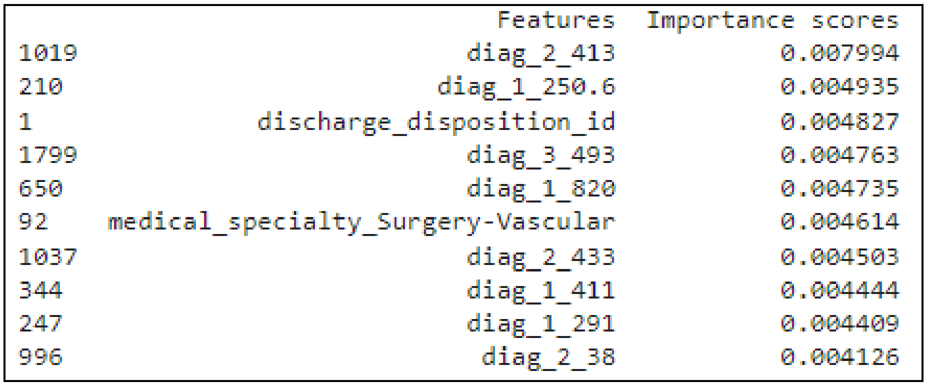


Fig. 4.2 – A glance at the top 10 features of XGBClassifier



```
                                 Features  Importance scores
1019                           diag_2_413           0.007994
210                          diag_1_250.6           0.004935
1                    discharge_disposition_id           0.004827
1799                           diag_3_493           0.004763
650                            diag_1_820           0.004735
92     medical_specialty_Surgery-Vascular           0.004614
1037                           diag_2_433           0.004503
344                            diag_1_411           0.004444
247                            diag_1_291           0.004409
996                             diag_2_38           0.004126
```

Conclusions:

XGBoost is definitely the most powerful when it comes to classifying the target class. It is also the easiest to tune as ‘scale_pos_weight’ is an easy hyperparameter to calculate and substantially improves the model. While the performance at this stage isn’t necessarily production-ready, it certainly has room to improve due to the concrete improvements seen with thresholding and the potential of oversampling the minority class to create a less-skewed class ratio.

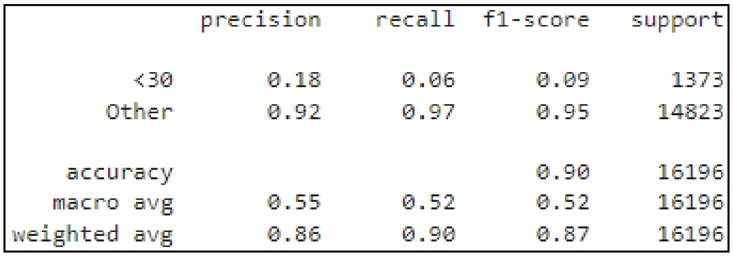
This model would be best-used as a tool that flags individuals who are at a high-risk for readmission and can show the features that correspond to this increased risk. The information provided would be supplemental to the treatment plan of the individual and may provide insight into what causes their readmission risk to increase, as well as how many beds the hospital can expect to be filled in the following month due to readmissions.

Future Scope:

In the data preparation stage, the categorical and numerical data could be separated and later rejoined so the One Hot Encoding is performed exclusively on the categorical data and the numerical data is then scaled to a standard. While ‘pd.get_dummies(drop_first=True)’ *should* automatically disregard numerical data, it would be wise to do this manually to be sure the data is properly encoded. Though it was deemed outside the scope of this project, more intensive feature encoding could lead to better prediction power.

In order to explore other ways in which the model could be improved by accounting for class imbalance, the method of oversampling the minority (or *target*) class to a 1:1 ratio with the majority class was briefly explored. The classification report for this XGBoost Classifier with oversampling shows a slight improvement over the base model (Fig. 5). The improvement seen by this simple oversampling hints that an even better model could be produced with the combination of multiple methods. The best model would most likely be an XGBClassifier with the use of thresholding *and* oversampling of the minority class. A balance could also be struck between ‘scale_pos_weight’ and the oversampling in order to reduce the impact of each transformation independently.

Fig. 5 – Classification Report for XGBClassifier with oversampling of the minority class



```
              precision    recall  f1-score   support

         <30       0.18      0.06      0.09      1373
       Other       0.92      0.97      0.95     14823

    accuracy                           0.90     16196
   macro avg       0.55      0.52      0.52     16196
weighted avg       0.86      0.90      0.87     16196
```

Finally, the initial decision-making process that led to our top 3 models could be further-refined, as both decision-tree-based models (Decision Tree and Random Forest) overfit and were thrown out accordingly. Some finer hyperparameter tuning could yield better results with these models, but it would not be surprising to still see XGBoost as the most robust of the group.

Sources:

[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. *[https://www.hindawi.com/journals/bmri/2014/781670/]*