Case Study 2 | Diabetes & Hospital Readmission

Joey Hernandez

Daniel Chang

DS 7333 | Quantifying the World

2023

**Introduction**

**Background**

Diabetes, a chronic metabolic disorder characterized by elevated blood sugar levels, is a significant global health concern. The management of diabetes often involves hospitalization, especially for individuals experiencing acute complications or undergoing significant treatment adjustments. One critical aspect of diabetes care is the prevention of readmission into a hospital within a short period after the initial discharge. Readmissions can indicate various issues, such as inadequate treatment during the initial hospitalization, complications arising post-discharge, or a lack of effective outpatient care.

## Objective and Scope

The primary objective of this report is to develop a predictive model to anticipate whether a patient with diabetes will be readmitted to a hospital within 30 days following their initial discharge. Achieving this objective can offer several advantages, including improved patient care, reduced healthcare costs, and enhanced resource allocation within healthcare facilities.

To accomplish our goal, we will employ logistic regression, a widely used statistical technique for binary classification. Logistic regression is well-suited for this task as it allows us to model the probability of readmission based on a set of relevant predictor variables. By analyzing these predictors, we can gain insights into the factors contributing to readmission risk among diabetic patients. Additionally, we will utilize imputation techniques to handle missing data, ensuring that our analysis is robust and representative of the patient population.

## Data Source

This case study will utilize two datasets, “diabetic\_data” and “IDs\_mapping”. The data was provided to us in the Case 2 Study Module and is in the form of two separate csv files. When combined the data contains 101766 observations and 53 features.

## Data Inspection

Before creating any models or analysis with the data our first step was to inspect our data to better understand data types (such as “int”, “cat”, “object”, etc.), distributions of values, identification of missing values, duplicated data, and outliers. This step is vital in understanding how we should approach any types of transformations or adjustments to the modeling and analysis process of our data.

## Target Variable Inspection

To gain insight into the distribution of our classification target variable, we performed a thorough examination of the data. The target variable in this analysis “readmitted” represents the status of no readmission, readmission within 30 days or greater than 30 days of the initial hospitalization for patients. It is a critical factor in our predictive model, as it helps determine whether a patient falls into one of two categories: "no readmittance" or "less than 30 days readmittance." Which is the objective classification problem for our case study.

Initially, we visualized the distribution of the target variable in its original form. This initial examination revealed a substantial class imbalance between the three categories. Most instances were labeled as "no readmittance," while a considerably smaller portion represented "less than 30 days readmittance", and “greater than 30 days readmittance”. This significant imbalance posed a challenge for our predictive modeling, as models may tend to be biased toward the majority class, potentially leading to suboptimal performance.

***Figure 1:*** *Count Plot Illustrating the Distribution of the Target Variable across the classes “No” “>30,” “<30”*

*A graph of a number of different colored bars

Description automatically generated*

The first amendment to our data and thus the scope of our project began with removal of the ‘greater than 30 days readmittance’. The objective of our data as provided to us by the client specifically mentioned the desire to predict patients that would be readmitted to the hospital within 30 days, thus this change in the data allows us to remove information from our data that may otherwise bias our prediction of what type of readmittance is occurring.

***Figure 2:*** *Count Plot Illustrating the new Distribution of the Target Variable across the binary values of “No Readmittance” and “Readmittance”, where Readmittance is only occurrences where the patient was readmitted <30 days.*

*A blue and orange rectangular object with white text

Description automatically generated*

## Patient Data Privacy And Information Inclusion/Exclusion

Respecting patient data privacy is of paramount importance in healthcare research, and this study is no exception. We recognize that certain demographic and clinical variables, such as race, gender, and age, can be invaluable in identifying potential contributors to the prediction of patient readmission. However, it is essential to emphasize that in our data analysis and modeling process, stringent measures have been implemented to safeguard patient privacy.

All patient identifiers, including but not limited to names, addresses, and specific identification numbers, have been rigorously excluded from our dataset. Additionally, any data attributes that could be used to directly infer individual patient identities or sensitive personal information have been carefully removed or anonymized to ensure the utmost protection of patient privacy.

The information used in our predictive model is limited to objective and de-identified data points relevant to the healthcare context. This means that our analysis is focused solely on variables that contribute to the accurate prediction of patient readmission risk, without compromising the confidentiality and privacy of individual patients. Our commitment to data privacy aligns with the highest ethical standards and legal requirements, ensuring that the insights derived from this study are both valuable and ethically sound.

## Missing Data

In the process of preparing and cleaning the dataset for our analysis, we encountered various missing data points across different attributes. Our approach to handling these missing values is guided by thorough investigations into the nature and potential implications of the missing data.

1. **Race:** In the 'race' attribute, we identified missing values denoted by "?". After further investigation, including visual analysis and cross-tabulations, we determined that these missing values were Missing Completely at Random (MCAR). As a result, we chose to proceed with imputation by replacing the missing values with the most frequent category. This approach allows us to maintain the integrity of the dataset while addressing the missing data issue.
2. **Weight:** The 'weight' attribute exhibited a significant percentage of missing values, exceeding 90%. Similar to the 'race' attribute, we investigated the nature of this missing data and confirmed that it was MCAR. However, due to the substantial extent of missing values and the limited potential usefulness of this feature, we made the decision to remove the 'weight' attribute from our dataset.
3. **Payer Code:** Missing values represented by "?" in the 'payer\_code' attribute were also identified as MCAR upon thorough investigation. In this case, instead of imputation or removal, we chose to re-label the missing values and include them as a separate category within the 'payer\_code' feature. This approach ensures that we retain valuable information while handling the missing data appropriately.
4. **Medical Specialty:** For the 'medical\_specialty' attribute, there was no discernible relationship between the values in our dataset and the missingness of medical specialty values. To address this, we proceeded with inputting the missing values by assigning them a new category labeled as "other." This imputation strategy helps preserve the overall structure of the data while accounting for the missing information.
5. **Diagnosis Codes (diag\_1, diag\_2, diag\_3):** While these attributes did contain missing values, it is worth noting that some of these gaps may be attributed to patients not having a specific diagnosis to report. Given the inconsequential amount of missing data in the diagnosis codes, we opted for a conservative approach by removing the null values from these attributes.

Our data preprocessing efforts regarding missing data aim to ensure that the resulting dataset is as informative and representative as possible while mitigating the potential bias introduced by the missing values. These carefully considered strategies allow us to maintain the integrity of the data and facilitate meaningful analyses for our predictive model of patient readmission.

***Figure 3:*** *A heatmap representing the distribution of missing values across the dataset. Darker areas indicate the absence of data points, while lighter regions denote complete information. Understanding the pattern of missingness is crucial for effective data preprocessing and predictive modeling.* A screen shot of a computer

Description automatically generated

## Correlation Plot (Original Target Variable)

Prioritizing the preprocessing steps for the target variable in the step prior to this was important because it allowed us to assess integrity and accuracy of subsequent analyses. By addressing the target variable's distribution and more narrow scope, the resulting correlation values can be better trusted to either accurately reflect the underlying relationships between variables or understand what limitations may arise from the less than desirable target variable distribution. Failure to preprocess and identify data discrepancies in the target variable could lead to misinterpretations, as correlations might be influenced by skewedness, outliers, or nonlinearities within the target data.

Performing a correlation heatmap provides a visually informative representation of the relationships between variables within a dataset. By illustrating the strength and direction of linear associations, the heatmap becomes an indispensable tool for uncovering patterns and dependencies that might not be immediately apparent from individual variable analyses. Each cell in the heatmap corresponds to a pair of variables, with the color gradient indicating the magnitude of correlation. This enables the rapid identification of high and low correlation values, highlighting potential areas of interest for further investigation.

***Figure 4 (next page):***  *A heatmap illustrating the correlation among integer-type features in the dataset. The color intensity reflects the strength and direction of associations between variables. Analyzing feature correlations is essential for identifying potential patterns and dependencies that can inform our predictive modeling efforts.*

A screenshot of a computer

Description automatically generated

## Modeling

## Lasso (L1 Regularization) Logistic Regression Model

To build an effective predictive model for patient readmission within 30 days of initial hospitalization, we began with a Lasso Logistic Regression approach. Lasso (Least Absolute Shrinkage and Selection Operator) is a regularization technique that helps prevent overfitting by penalizing the absolute values of the regression coefficients. In our analysis, we utilized Lasso Logistic Regression as a starting point due to its ability to perform feature selection by driving some coefficient estimates to zero, thereby simplifying the model and enhancing its interpretability.

Our model-building process included several crucial steps:

**1. Grid Search for Optimal Alpha Value:** We performed a grid search to determine the best alpha value for the Lasso Logistic Regression model. Alpha controls the strength of the L1 regularization penalty, and finding the optimal value is essential for achieving the right balance between model complexity and performance.

**2. Train-Test Split:** To assess the model's performance effectively, we divided the dataset into training and testing subsets. The training data were used to train the model, while the testing data served as an independent dataset for evaluating its generalization performance.

**3. Feature Scaling and Dummitizing:** Proper scaling of the features is crucial for logistic regression models. We standardized the numerical features to have zero mean and unit variance to ensure that all features contributed equally to the model. Additionally, we employed one-hot encoding (dummitizing) for categorical variables to convert them into a format suitable for the logistic regression model.

**4. Applying Threshold to Classification:** In binary classification problems like ours, a probability threshold is applied to determine the predicted class labels. We experimented with various threshold values to optimize the trade-off between sensitivity and specificity, tailoring the model to prioritize either minimizing false positives or false negatives, depending on the clinical context.

By implementing these steps, we aimed to develop a robust Lasso Logistic Regression model capable of predicting patient readmission within 30 days. This model not only provides predictive accuracy but also offers interpretability, allowing us to identify the most influential factors contributing to the likelihood of readmission. In the subsequent sections, we will detail the results of our model evaluation and discuss its implications for improving patient care and healthcare resource allocation.

**Results:**

**Optimal Alpha (Lambda):** The grid search procedure identified the best alpha value for Lasso regularization as 0.0100. This parameter controls the strength of regularization, striking a balance between model complexity and performance.

**Test Set Accuracy:** The Lasso Logistic Regression model achieved an accuracy of 0.6104 on the test dataset. This metric reflects the proportion of correctly classified instances, indicating that the model correctly predicted the readmission outcomes for approximately 61.04% of the test cases.

**User-Defined Threshold:** In binary classification, the choice of threshold for converting predicted probabilities into class labels can significantly impact the model's performance. In this case, a user-defined threshold of 0.65 was applied to the model's predicted probabilities to determine the classification labels.

**Classification Report with Custom Threshold:**

* **Precision:** The precision for class "0" (no readmittance) was notably high at 0.94. This indicates that when the model predicted "no readmittance," it was accurate 94% of the time. Intuitively this makes sense given that the majority class of our data was “no readmittance” so it would be expected that the model would be bias to predicting “no readmittance”.
* **Recall:** The recall for class "0" was 0.62, indicating that the model captured 62% of the actual "no readmittance" cases. For class "1" (less than 30 days readmittance), the recall was 0.49, suggesting that the model identified 49% of the actual cases.
* **F1-Score:** The F1-score, which balances precision and recall, was 0.75 for class "0" and 0.16 for class "1."
* **Support:** The support represents the number of instances in each class within the test dataset. In this case, class "0" had significantly more instances (12,966) than class "1" (1,086).

**Macro and Weighted Averages:** The macro-average F1-score was 0.46, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, which accounts for class imbalances, was 0.70. These averages provide a comprehensive view of the model's performance across all classes.

In summary, the Lasso Logistic Regression model, with an optimized alpha value and a user-defined threshold, demonstrated reasonable accuracy in predicting patient readmission within 30 days. It exhibited strong precision for the "no readmittance" class, indicating a high degree of confidence in negative predictions. However, there is room for improvement in recall and F1-score, particularly for the "less than 30 days readmittance" class, which represents cases where early readmission is crucial to identify. These results provide a baseline for our predictive modeling efforts, and further analysis and model refinements will be discussed in subsequent sections.

A blue and white chart

Description automatically generated

## A graph of a positive rate Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Report** | | | | |
|  | **Precision** | **Recall** | **F1- Score** | **Support** |
| **Not Readmitted** | 0.94 | 0.62 | 0.75 | 12966 |
| **Readmitted** | 0.10 | 0.49 | 0.16 | 1086 |
| **Accuracy** |  |  | 0.61 | 14052 |
| **Macro Average** |  | 0.56 | 0.46 | 14052 |
| **Weighted Average** |  | 0.61 | 0.70 | 14052 |

## Lasso Logistic Regression Model – Under Sampling

For our second Lasso Logistic Regression model, we followed a similar procedure to the previous model but introduced a key modification: under sampling. Under sampling is a technique used to address class imbalance, which is particularly relevant in our binary classification problem due to the significant disproportion between the "no readmittance" and "less than 30 days readmittance" classes.

The steps in building this model included:

**1. Grid Search for Optimal Alpha Value:** We conducted a grid search to identify the optimal alpha value for Lasso Logistic Regression, maintaining the importance of regularization in feature selection and model simplification.

**2. Train-Test Split:** We again divided the dataset into training and testing subsets, ensuring an independent evaluation of model performance.

**3. Feature Scaling and Dummitizing:** Consistent with the previous model, we standardized numerical features and applied one-hot encoding to categorical variables for compatibility with logistic regression.

**4. Under sampling:** To mitigate the effects of class imbalance, we employed under sampling. This technique involved randomly selecting a subset of the majority class (in this case, "no readmittance") to balance the class distribution. By reducing the number of instances in the majority class, we aimed to ensure that the model did not disproportionately favor this class during training.

The introduction of under sampling in this Lasso Logistic Regression model allowed us to address the class imbalance issue more effectively. It enabled the model to learn from a more balanced dataset, potentially leading to better generalization and predictive performance, especially for the minority class ("less than 30 days readmittance"). In the subsequent sections, we will present the results of this model, including its evaluation and impact on predictive accuracy for patient readmission within 30 days.

**Results**

**Optimal Alpha (Lambda):** The grid search determined the best alpha value for Lasso regularization to be 0.0100, consistent with our first model.

**Test Set Accuracy:** The Lasso Logistic Regression model with under sampling achieved an accuracy of 0.6600 on the test dataset. This metric indicates that the model correctly predicted the readmission outcomes for approximately 66% of the test cases.

**User-Defined Threshold:** To classify predictions into classes, a user-defined threshold of 0.98 was applied to the model's predicted probabilities.

**Classification Report with Custom Threshold:**

* **Precision:** The precision for class "0" (no readmittance) remained high at 0.93, indicating that when the model predicted "no readmittance," it was accurate 93% of the time. For class "1" (less than 30 days readmittance), the precision was 0.10, reflecting that the model's positive predictions for this class had a lower precision rate.
* **Recall:** The recall for class "0" was 0.68, suggesting that the model captured 68% of the actual "no readmittance" cases. The recall for class "1" was 0.41, indicating that the model identified 41% of the actual cases of "less than 30 days readmittance."
* **F1-Score:** The F1-score for class "0" improved to 0.79, demonstrating a better balance between precision and recall compared to our first model. However, the F1-score for class "1" remained at 0.16.
* **Support:** As in the previous model, class "0" had a substantially larger number of instances (12,966) compared to class "1" (1,086).

**Macro and Weighted Averages:** The macro-average F1-score was 0.47, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, accounting for class imbalances, improved to 0.74.

In summary, the Lasso Logistic Regression model with undersampling showed an improvement in accuracy compared to our first model. While precision remained high for class "0," the model's ability to identify cases of "less than 30 days readmittance" improved, as reflected in the higher recall for this class and a slightly improved F1-score. The undersampling technique helped address class imbalance, leading to a more balanced model. However, there is still room for further enhancement, especially in improving the model's ability to identify cases of early readmission. These results provide valuable insights for our ongoing analysis and model refinement efforts.

A blue and white chart

Description automatically generated

A graph of a positive rate

Description automatically generated with medium confidence

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Report** | | | | |
|  | **Precision** | **Recall** | **F1- Score** | **Support** |
| **Not Readmitted** | 0.93 | 0.68 | 0.79 | 12966 |
| **Readmitted** | 0.10 | 0.41 | 0.16 | 1086 |
| **Accuracy** |  |  | 0.66 | 14052 |
| **Macro Average** | 0.51 | 0.54 | 0.47 | 14052 |
| **Weighted Average** | 0.87 | 0.66 | 0.74 | 14052 |

## Random Forest

In addition to Lasso Logistic Regression, we explored the application of a Random Forest classifier as our third predictive model for patient readmission within 30 days of initial hospitalization. The Random Forest algorithm is an ensemble learning method that leverages the collective decision-making of multiple decision trees. This approach often yields robust and accurate predictions, making it a valuable tool for complex classification tasks like ours.

Our strategy for implementing the Random Forest model consisted of the following steps:

**1. Hyperparameter Tuning:** To optimize the performance of the Random Forest model, we conducted hyperparameter tuning. This involved exploring various settings, such as the number of trees in the forest and the maximum depth of each tree. We aimed to identify the combination of hyperparameters that would result in the most effective model.

**2. Train-Test Split:** Similar to our previous models, we divided the dataset into training and testing subsets. This separation allowed us to train the Random Forest model on one portion of the data and evaluate its performance on an independent dataset, ensuring that the model generalizes well to new, unseen instances.

**3. Feature Importance Analysis:** One of the advantages of Random Forest is its ability to provide insight into feature importance. We conducted a feature importance analysis to identify which variables had the most significant influence on predicting patient readmission. This analysis can be valuable for healthcare providers and decision-makers to prioritize interventions and allocate resources effectively.

**4. Model Evaluation:** We rigorously assessed the performance of the Random Forest model using a range of evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. These metrics allowed us to gauge the model's predictive accuracy and its ability to balance sensitivity and specificity in predicting readmission outcomes.

**5. Class Weight:** In contrast to the under sampling technique utilized in the logistic regression model Random Forest Classifier has a parameter” class\_weight” where we can set class weight equal to 'balanced'. This is a more convenient option compared to under/overs sampling because it doesn't require modifying the training dataset's size. Instead, it automatically calculates class weights inversely proportional to the class frequencies in the training data. In other words, it assigns higher weights to the minority class and lower weights to the majority class.

By incorporating the Random Forest model into our analysis, we aimed to leverage its ensemble capabilities to capture complex relationships in the data and provide a more accurate and robust prediction of patient readmission within 30 days. The results of our Random Forest model will be discussed in subsequent sections, shedding light on its effectiveness in enhancing patient care and healthcare resource allocation.

**Results:**

Our third model employed the Random Forest classifier with class weights set to 'balanced' to address the class imbalance issue. Here are the results obtained from this model:

**Test Set Accuracy:** The Random Forest model with balanced class weights achieved an accuracy of 0.6050 on the test dataset. This metric indicates that the model correctly predicted the readmission outcomes for approximately 60.50% of the test cases.

**Classification Report:**

* **Precision:** The precision for class "0" (no readmittance) remained high at 0.93, indicating that when the model predicted "no readmittance," it was accurate 93% of the time. However, the precision for class "1" (less than 30 days readmittance) was lower at 0.08, reflecting a lower precision rate for positive predictions.
* **Recall:** The recall for class "0" was 0.62, suggesting that the model captured 62% of the actual "no readmittance" cases. For class "1," the recall was 0.41, indicating that the model identified 41% of the actual cases of "less than 30 days readmittance."
* **F1-Score:** The F1-score for class "0" was 0.74, demonstrating a good balance between precision and recall. However, the F1-score for class "1" remained low at 0.14.
* **Support:** As in the previous models, class "0" had a significantly larger number of instances (12,966) compared to class "1" (1,086).

**Macro and Weighted Averages:** The macro-average F1-score was 0.44, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, accounting for class imbalances, was 0.70.

In summary, the Random Forest Classifier with balanced class weights exhibited a moderate accuracy in predicting patient readmission within 30 days. The model demonstrated high precision for class "0" (no readmittance), but the ability to identify cases of "less than 30 days readmittance" remained limited, as reflected in the lower recall and F1-score for class "1." While balanced class weights helped mitigate class imbalance, further model refinement may be needed to improve its performance in identifying early readmission cases. These results provide insights for ongoing analysis and potential model enhancements.

A graph with a line

Description automatically generated

A blue and white chart

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Report** | | | | |
|  | **Precision** | **Recall** | **F1- Score** | **Support** |
| **Not Readmitted** | 0.93 | 0.62 | 0.74 | 12966 |
| **Readmitted** | 0.08 | 0.41 | 0.14 | 1086 |
| **Accuracy** |  |  | 0.61 | 14052 |
| **Macro Average** | 0.51 | 0.52 | 0.44 | 14052 |
| **Weighted Average** | 0.86 | 0.61 | 0.70 | 14052 |

**Conclusion**

In this comprehensive analysis, we delved into the critical task of predicting patient readmission within 30 days of their initial hospitalization, a significant concern in the management of diabetes and other chronic conditions. Our study revolved around the evaluation of three distinct predictive models: Lasso Logistic Regression, Lasso Logistic Regression with Under sampling, and Random Forest Classifier with Balanced Class Weights. Each model brought its unique strengths and considerations to the table, contributing to our understanding of this complex problem.

The Lasso Logistic Regression model served as a foundational step, showcasing the importance of regularization in feature selection and model simplification. Despite reasonable accuracy, this initial model revealed the challenges of striking a balance between precision and recall, particularly for cases of early readmission. Feature engineering and custom thresholding were essential steps to fine-tune its performance.

The introduction of under sampling in our second model marked an effective attempt to mitigate class imbalance, providing a more balanced dataset for training. This adjustment yielded improvements in overall accuracy and recall for cases of early readmission, although there is still room for enhancement in precision and F1-score.

Our third model, the Random Forest Classifier with balanced class weights, showcased the robustness of ensemble learning in addressing complex classification tasks. While achieving a moderate level of accuracy, this model upheld high precision for class "0" predictions but struggled to effectively identify cases of "less than 30 days readmittance." Balancing class weights helped, yet further optimization is needed to bridge the gap between precision and recall for class "1."

In conclusion, the prediction of patient readmission within a 30-day window is a multifaceted challenge with significant clinical implications. Our models provide a foundation for understanding this problem, and their results shed light on the intricacies of balancing accuracy, precision, and recall, especially concerning early readmission cases.

**Recommendations**

Based on our analysis and findings, we propose several recommendations to enhance the predictive modeling of patient readmission within 30 days of initial hospitalization, with a focus on improving model performance and the practical utility of the predictions:

**1. Feature Engineering and Selection:**

* **Feature Engineering:** Continue to explore feature engineering techniques to create meaningful and predictive variables. Consider interactions between variables, time-based features, and domain-specific variables that may impact readmission risk.
* **Dimensionality Reduction:** Given the dimensionality of the dataset, consider employing dimensionality reduction techniques, or feature selection algorithms, to reduce noise and enhance model interpretability.

**2. Dual Threshold Model:**

* Develop a dual-threshold model that assigns probabilities or likelihood scores to the likelihood of readmission. This model can provide more nuanced insights by distinguishing between patients with a higher risk of early readmission and those with a lower risk. Implementing two thresholds allows healthcare providers to prioritize interventions more effectively.

**3. Handling Class Imbalance:**

* Continue experimenting with techniques to address class imbalance. Undersampling and oversampling methods, such as Synthetic Minority Over-sampling Technique (SMOTE), may further improve the model's ability to identify early readmission cases.

**4. SME Collaboration:**

* Collaborate closely with healthcare professionals to clinically validate the model's predictions. Incorporate domain expertise to refine the model's features, thresholds, and evaluation metrics to align better with the practical needs of healthcare providers.

**5. Interpretability and Explainability:**

* Prioritize the development of model interpretability and explainability techniques. Transparent models are essential for healthcare practitioners to understand and trust model predictions.

**6. Data Quality and Collection:**

* Invest in improving data quality by addressing missing data issues and conducting regular data audits. Collect additional relevant variables that might not be part of the current dataset but could have significant predictive power.

By implementing these recommendations, healthcare providers and data scientists can work together to develop more accurate, reliable, and clinically relevant models for predicting patient readmission. These efforts can ultimately contribute to improved patient care, better allocation of healthcare resources, and enhanced outcomes in the management of chronic conditions such as diabetes.

**Appendix**