Predicting Hospital Patient Readmission with Medical History

Jordan Butler, Charles Cheng, Dean Choi, Joey Lieto

Kennesaw State University

DS7140: Python for Data Science

April 25, 2025

Abstract

Hospital readmissions pose a major challenge in healthcare by increasing operational costs and triggering financial penalties through programs like the Hospital Readmissions Reduction Program. This project aims to predict the likelihood of patient readmission using clinical and demographic data from the initial hospital visit. The dataset, sourced from Kaggle, includes features such as age, diagnosis codes, lab procedures, glucose levels, and medication information.

After comprehensive data cleaning and preprocessing—including missing value removal, categorical encoding, and feature scaling—exploratory data analysis was performed to identify key trends and correlations. Three classification algorithms were implemented: Logistic Regression, Decision Tree, and Random Forest. These models were evaluated using accuracy, precision, recall, and F1-score to balance overall performance and the ability to correctly identify at-risk patients.

Among the models, Random Forest showed the highest F1-score and recall, indicating better generalization and sensitivity to actual readmission cases. Logistic Regression produced the highest accuracy but slightly lower recall, while Decision Tree offered interpretability at the cost of lower performance metrics.

Although all models achieved around 60% accuracy, this level of prediction is not considered adequate for critical medical decision-making. Nevertheless, the findings demonstrate the potential of machine learning to support early identification of high-risk patients and inform preventive care strategies. Future improvements should focus on enhancing data diversity, fine-tuning model parameters, and applying imbalance correction techniques to achieve clinically viable results.

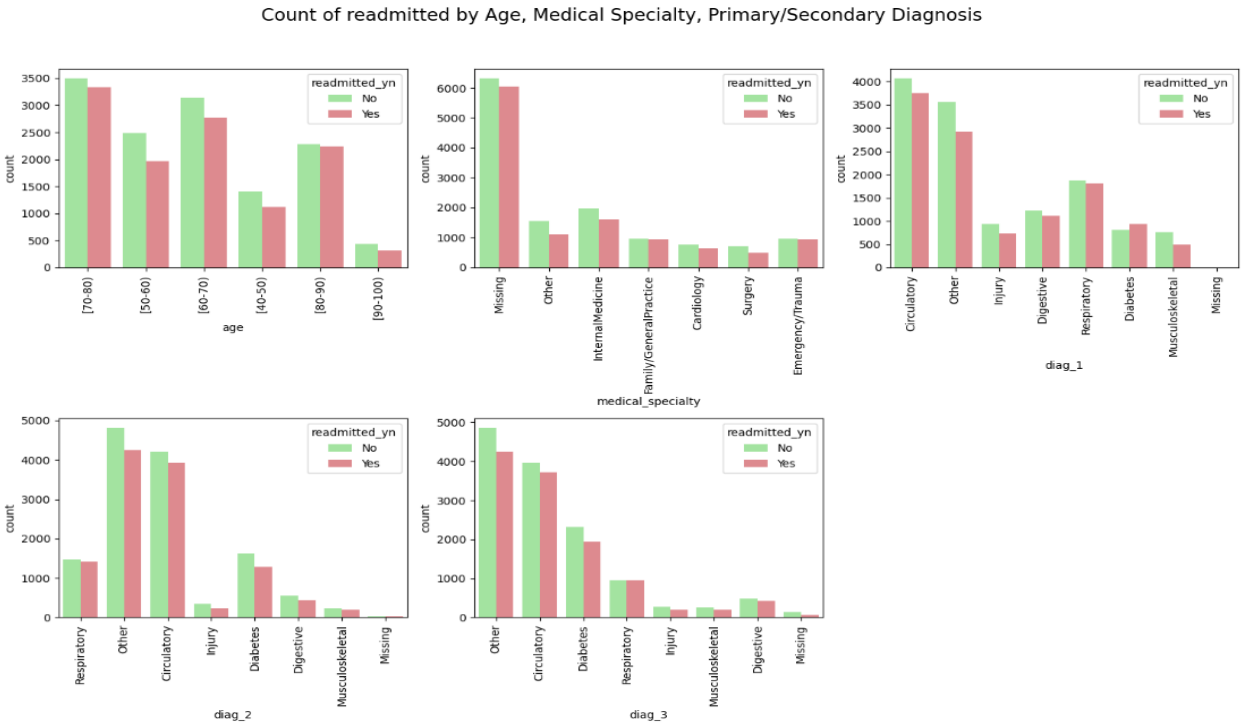
Introduction

Hospital readmissions are a significant issue in healthcare due to the resource costs from unnecessary readmissions. It's such an issue that the US has a program that penalizes hospitals for excessive avoidable readmissions called the Hospital Readmissions Reduction Program (CMS.gov). The goal of this project is to identify patients at risk of readmission in order to preemptively administer care to them so as to prevent unnecessary readmissions. Being able to catch patients at risk of readmission would decrease the pressure on healthcare resources and prevent such penalties from being levied.

Data Overview

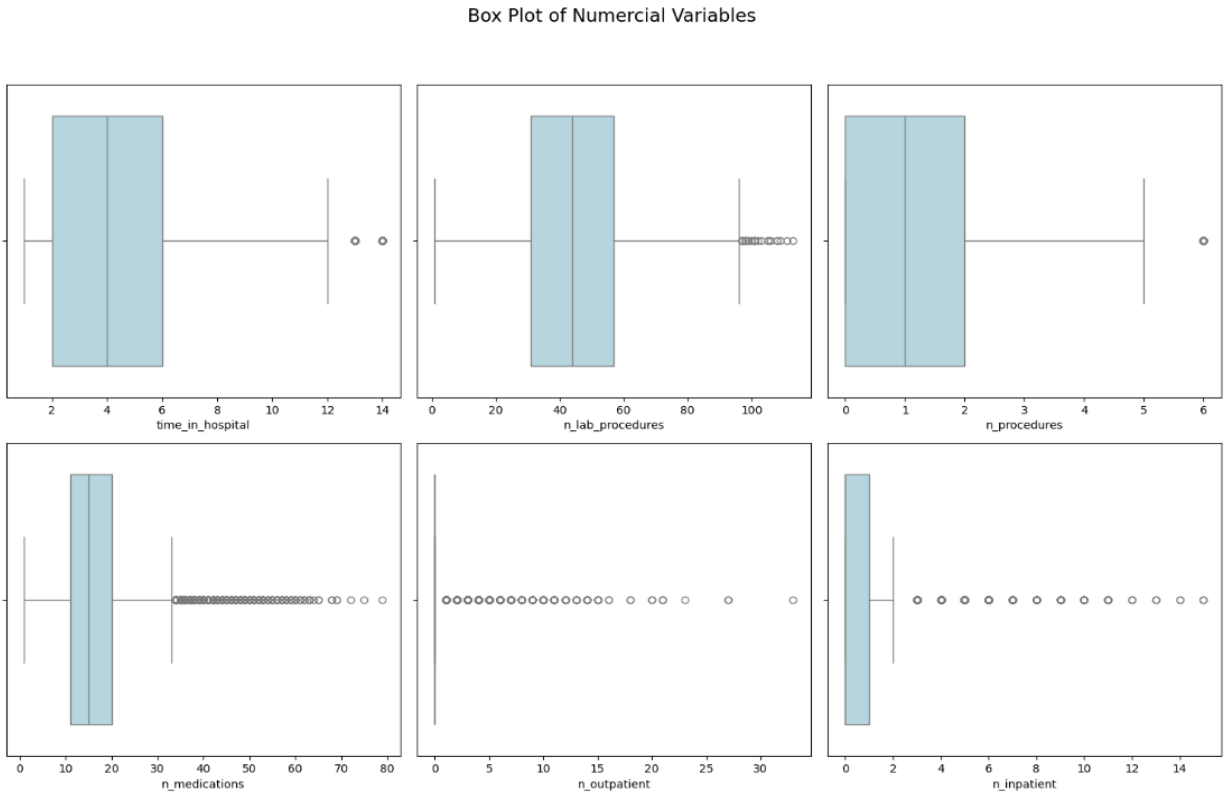
The data in this study is pulled from a Kaggle dataset containing a 10 year history of patient hospital readmission data delineated by various measures of diabetes/other diagnosis information. With the main target variable being a binary indicator on whether or not a patient was readmitted, there is various other information regarding overall patient health, patient information, and reasoning for original visit. Regarding overlying personal information, it is detailed within the dataset the patient's age. For overall patient health, the data summarizes the patient's health via hospital visits and emergency room visits in the year prior. The remainder of the data contains the results of medical tests performed during the course of the hospital stay in question, along with the number of procedures performed during the visit and both the primary/secondary diagnosis for the hospital stay. Overall, the data consists of a mix of continuous quantitative and discrete qualitative variables that contain a large quantity of information that would be required to predict whether or not a patient will need to be readmitted based on everything that transpires during their current hospital stay.

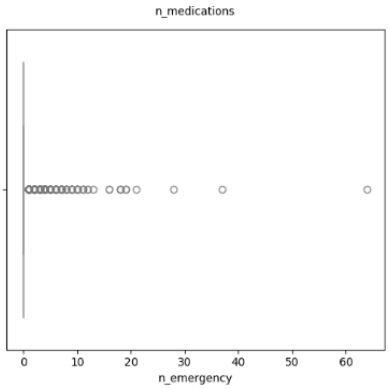
We can first take a look at how each of the categorical variables relate to readmission status as shown in the below histogram charts.



There seems to be a direct correlation between age and readmission status due to the difference in no readmit vs yes readmit closing as age increases. While not statistically backed just yet, it is pretty safe to say that as patient age increases, the % chance of readmission increases as well which isn't entirely crazy to predict. Secondly, while not necessarily being able to quantify the seriousness of certain diagnoses or medical specialties, there seems to be a clear correlation between the hospital visit being for a more serious condition (pertaining to heart and other major organs) and the % chance of readmission. Once again, while not quite being able to quantify seriousness, there is certainly a higher chance a patient will need to be readmitted when dealing with more serious diagnoses pertaining to circulatory and respiratory vs other diagnoses such as digestive or generic injury.

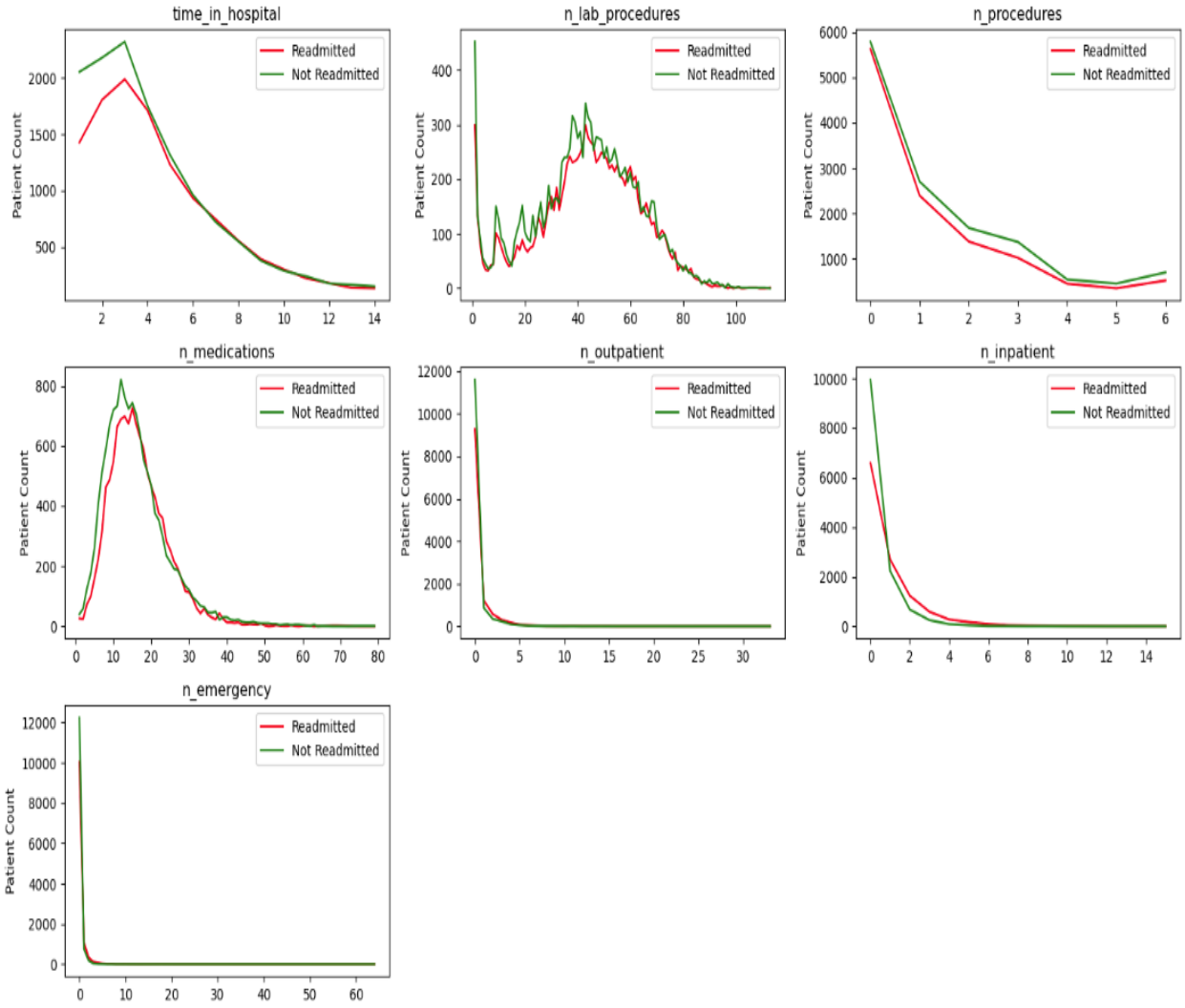
When looking at the overall spread of the numerical data, certain patterns and trends emerge about some of the variables in particular as shown by the box and whiskers plots below.

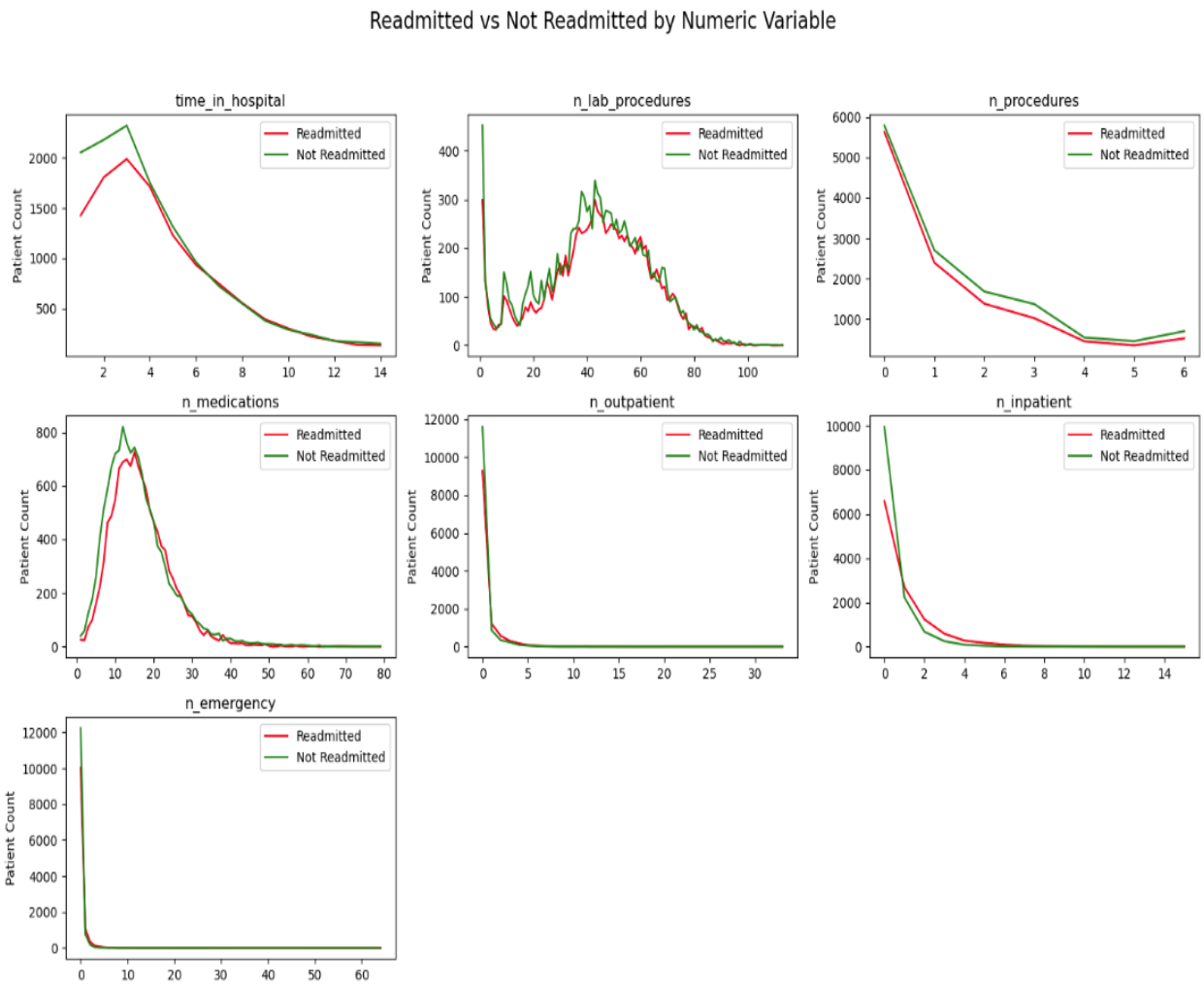




Variables such as time_in_hospital and number of procedures seem to follow a more traditional spread while other variables such as number of medications prescribed and number of hospital visits in the past year are much more spread out and contain a seemingly endless amount of outliers. This may help explain volatility in the data later on when looking at the predictive model and variable importance testing as it will be much harder to detail how important a variable is when there is effectively no consistency like with n_outpatient and n_emergency, especially when almost all data points for that variable are at the same number which in both of those cases is 0. Other variables like time_in_hospital and n_procedures will be much easier to tell how exactly they factor into readmission prediction due to their. standard distribution status. While once again not statistically backed, a brief look at the relationships between continuous variables and readmission status can be seen below.

Readmitted vs Not Readmitted by Numeric Variable





While not consistent across every numerical variable, there is a clear trend of the number of readmissions overtaking the number of no readmissions at some point as the x-axis units increase. For the number of medications, it would certainly stand to reason that more medications would signify a more complex treatment meaning the chance of readmission is higher, and this is confirmed by the line chart which shows the overtake at approximately 5 medications remaining side by side with no readmission for the remainder of the variable range. This same pattern can be observed with other continuous variables such as time in hospital and n_inpatient. Overall, visualizing the relationship between both the categorical and quantitative variables present in the data and the target variable, hospital readmission status, effectively confirms what theories one might have without being able to actually dive into the data. The question still remains though, is it possible to conclude at a statistically significant level if any of these variables, by themselves or along with others, will be able to consistently and accurately predict whether or not a patient will need to be

readmitted at some point in the future allowing hospitals to be able to potentially nip these sorts of unfortunate medical situations in the bud.

Methodology

1. Data Extraction

The dataset used in this project, titled hospital_readmissions.csv, was sourced from a publicly available database focused on hospital patient outcomes.

The data was accessed by directly loading the CSV file into the Python programming environment using the pandas library.

This dataset is particularly relevant to the research objective because it contains a wide range of patient-level features such as demographics, hospitalization details, and clinical attributes, which are known factors influencing hospital readmission rates.

By analyzing this dataset, we aim to identify patterns associated with readmissions and develop predictive models that can assist healthcare institutions in reducing unnecessary readmission rates, thereby improving patient care and lowering costs.

2. Data Preparation

Data preprocessing was an essential step to ensure that the dataset was clean, consistent, and suitable for model training.

First, missing values were addressed. Using the isnull() function in Python, missing entries across all columns were identified. All rows containing any missing values were subsequently dropped to maintain data integrity and prevent bias during model building.

Unnecessary columns were also removed to streamline the feature set. Specifically, the columns diag_1 and diag_2 were dropped if they existed. These columns were either redundant or not directly relevant to the primary prediction objective.

Handling of categorical variables was performed next. Object-type columns, representing categorical data, were identified automatically. Each categorical variable was then encoded into numerical values using the .astype('category').cat.codes method from the pandas library. This transformation ensured that the dataset was fully numeric and compatible with machine learning algorithms.

Normalization was applied to all numerical features to bring the data into a consistent range. A MinMaxScaler from the sklearn.preprocessing module was employed to scale all numeric variables to a [0, 1] range. Normalization is particularly important when working with regularized models like Ridge and Lasso, which are sensitive to feature scales.

No advanced feature engineering techniques, such as creation of new variables or aggregation, were applied in this project. Instead, emphasis was placed on encoding categorical variables properly and standardizing numerical features, ensuring that the dataset was optimized for regression analysis.

Through these systematic preprocessing steps, a clean, consistent, and machine-learning-ready dataset was prepared for modeling and evaluation.

3. Data Exploration

Exploratory Data Analysis (EDA) was conducted to understand the distributions, relationships, and potential issues present in the dataset.

The first step involved correlation analysis. A correlation matrix was computed for all numerical variables and visualized using a heatmap created with the seaborn library. This allowed for the identification of strong linear relationships between variables and potential multicollinearity concerns. Highly correlated variables were noted, informing later model selection and regularization strategies.

Distribution analysis was also performed for each numerical variable. Histograms combined with Kernel Density Estimates (KDE) were plotted using seaborn.histplot to visualize the spread and skewness of each feature. This helped in identifying variables with non-normal distributions and extreme outliers.

A focused analysis was conducted on the readmission variable, which is the target outcome. A count plot was generated to observe the distribution of readmission occurrences. Additionally, a boxplot analysis was performed to investigate the relationship between age and readmission status.

These visualizations revealed that certain age groups may be more susceptible to readmissions, suggesting age as a potentially significant predictor.

The main visualization tools employed in this project were Matplotlib and Seaborn, along with pandas for data summaries and quick previews.

Key insights gained from the EDA included:

- Moderate correlations among clinical and demographic features.

- Non-normal distributions across several variables, reinforcing the need for normalization.
- A noticeable relationship between age and readmission, supporting its inclusion in predictive models.

These exploratory findings helped shape the modeling strategy by confirming the relevance of specific features and guiding preprocessing choices, ultimately ensuring that the models built would address the primary research question effectively.

For predictive modeling, Logistic Regression, Decision Tree and Random Forest classifiers are implemented. Decision Tree is chosen for its interpretability in handling non-linear relationships, while Random Forest provides robustness through ensemble learning, mitigating overfitting. Logistic regression is simpler than both Decision Tree and Random Forest but is reliant on a linear relationship between variables.

The models were evaluated using accuracy, precision, recall, and F1-score. Accuracy measures how many of our predictions were correct. Precision measures how well the model predicts true positives. Recall measures the proportion of true positives out of all positive predictions. F1-Score provides a balance between precision and recall score. These are all important measurements because we need to know how well our model catches people in need of readmission as looking at too many people that are not at risk or missing too many that are at risk will result in unnecessary healthcare resource usage all the same.

Experiments

The experimental dataset used in this study originates from patient records containing medical, procedural, and hospital visit data. The dataset includes key features such as medical specialty, glucose test results, A1C levels, medications, number of inpatient/outpatient visits, emergency visits, and diagnosis codes. It is structured with categorical variables transformed via one-hot encoding and numerical variables standardized using StandardScaler. The total dataset size is significant, ensuring generalizability for predicting patient readmission probabilities. Its relevance lies in assisting healthcare providers in identifying high risk patients, allowing for proactive intervention strategies.

To optimize model performance, hyperparameter tuning was conducted using GridSearchCV, an exhaustive search technique that evaluates multiple combinations of parameters across a defined grid. For both models, Decision Tree and Random Forest this approach refined parameters such as tree depth, feature selection methods, minimum samples for splitting, and classification criteria. This approach ensures that the models are optimized for performance while avoiding overfitting.

Decision Tree Hyperparameter Selection

For the Decision Tree model, key hyperparameters were tuned to balance complexity and accuracy:

- **Criterion (gini, entropy)** – Determines how splits are made; **entropy** was chosen, ensuring more information gain at each step.
- **Max Depth (None, 10, 20, 30)** – Controls tree growth; **10** was optimal, preventing excessive branching and overfitting.
- **Min Samples Split (2, 5, 10)** – Defines the minimum samples required for splitting; the best value was **2**, allowing flexibility while ensuring meaningful divisions.
- **Min Samples Leaf (1, 2, 4)** – Ensures a minimum number of samples in terminal nodes; **1** was optimal, preserving all patterns.
- **Max Features (None, sqrt, log2)** – Determines how many features to consider per split; **None** was best, allowing full dataset access.
- **Splitter (best, random)** – Controls how splits are chosen; random works best, ensuring diverse partitions across samples.

Random Forest Hyperparameter Selection

For the Random Forest model, hyperparameter tuning focused on ensemble diversity:

- **Number of Estimators (100, 200, 300, 500)** – Controls the number of trees in the forest; **300** was ideal, providing a strong ensemble effect without excessive computation.
- **Max Depth (None, 15, 20, 30)** – Limits tree depth; **15** was optimal, reducing overfitting.
- **Min Samples Split (2, 5, 10)** – Ensures meaningful splits; **10** was chosen, avoiding overly specific branches.

- **Max Features (sqrt, log2)** – Determines feature selection per tree; **log2** was optimal, balancing accuracy and diversity.
- **Bootstrap (True, False)** – Controls sample resampling; **False** provided the best results by using the entire dataset.

Class Weight (None, balanced, balanced_subsample) – Adjusts for imbalanced data; **balanced** ensured fair weight distribution.

After training, the evaluation metrics, accuracy, precision, recall, and f1-score, were used to assess model effectiveness. Below is the summary of results

	Model	Accuracy	Precision	Recall	F1-score
0	Random Forest	0.5994	0.6112	0.6112	0.6112
1	Decision Tree	0.5994	0.5994	0.5994	0.5994

As you can see the Random Forest model outperformed the Decision Tree model, with a slightly higher precision and recall. This would indicate that ensemble methods provide better generalization, capturing more patterns from the patient records. The decision tree, while interpretable, lacks robustness, leading to marginally lower performance.

The results highlight how predictive modeling can improve patient management strategies. Hospitals can use such models to identify high risk patients before discharge, enabling better post-treatment care. Additionally, automated risk assessment systems could integrate machine learning to optimize healthcare resource allocation, reducing avoidable readmissions.

Logistic Regression Hyperparameter Selection

Compared to the random forest and decision tree models, logistic regression only had one hyperparameter to select for, C. The C value controls the regularization in the model. A higher C value indicates weaker regularization and a lower C value indicates stronger regularization. In order to find the optimal C value for use in the linear regression function, we used LogisticRegressionCV() with a cv of 5 and a max iteration of 1000. This function found that the best C value was around 21.54 which was then plugged into the LogisticRegression function. The results of the logistic regression function are found below:

```
Accuracy Score: 0.608
Precision Score: 0.608
Recall Score: 0.608
f1 Score: 0.608
```

These values show that the logistic regression model will predict whether or not a patient has a readmission correctly around 60.8% of the time which is a similar number to those found in the random forest and decision tree models. This result isn't usable for the medical field as a doctor needs to be mostly certain that what they are recommending to their patients is the best course of action.

Conclusion

The key outcome of this project is the development and evaluation of three machine learning models—Logistic Regression, Decision Tree, and Random Forest—for predicting hospital readmissions based on structured clinical data. Among the models, Random Forest demonstrated the best overall performance with the highest precision, recall, and F1-score, indicating its superior ability to detect high-risk patients. Logistic Regression achieved the highest accuracy, while Decision Tree provided interpretability with limited predictive strength. However, all models yielded accuracies around 60%, highlighting the challenges of predicting readmissions from available features.

For healthcare stakeholders, these findings emphasize the potential value of predictive modeling in guiding early interventions and reducing avoidable readmissions. Hospitals can leverage such models as part of clinical decision support systems to flag at-risk patients before discharge, improving patient outcomes and optimizing resource allocation. Moreover, predictive insights can help mitigate financial penalties under performance-based programs like the HRRP.

Future research should focus on expanding the dataset with more detailed clinical variables, including longitudinal patient histories, medication adherence, and social determinants of health. Enhancing model performance could also involve advanced techniques such as ensemble stacking, cost-sensitive learning, and class imbalance correction (e.g., SMOTE). By refining both data inputs and model architectures, subsequent work may yield tools capable of supporting real-world deployment in clinical environments.

References

“Hospital Readmissions Reduction Program (HRRP).” *CMS.Gov*, 10 Sept. 2024, www.cms.gov/medicare/payment/prospective-payment-systems/acute-inpatient-pps/hospital-readmissions-reduction-program-hrrp.

Data source

https://www.kaggle.com/datasets/dubradave/hospital-readmissions