**Title: Comparative Analysis of Machine Learning Models for Predicting Patient Readmissions: A Study on Precision, Recall, and Discriminatory Performance**

Akshda Shandilya

New Jersey City University: School of Business

BUSI973: Independent Study

Dr. J. D. Jayaraman

August 18, 2023

# 

# 

# CONTENTS

1. INTRODUCTION 3
2. LITERATURE REVIEW 4
3. RESEARCH QUESTION AND HYPOTHESIS 6
4. METHODOLOGY 6
   1. DATA AQUISITION 6
   2. VARIABLE DESCRIPTION 7
   3. EXPLORATORY DATA ANALYSIS 9
   4. IDENTIFYING VARIABLE MODELS FOR ANALYSIS 24
   5. DATA MODELING 25
      1. BUILDING THE MODEL 25
      2. OLS REGRESSION 26
      3. RANDOM FOREST CLASSIFICATION 27
   6. LOGISTIC REGRESSION 28
   7. SUPPORT VECTOR MACHINE 32
   8. GRADIENT BOOSTING 35
   9. GAUSSIAN NAIVE BAYES 38
   10. K - NEAREST NEIGHBORS 40
   11. NEURAL NETWORKS 43
5. RESULTS 45
6. CONCLUSION 48
7. REFERENCES 50

# INTRODUCTION:

Reducing hospital readmissions has emerged as a critical goal for healthcare providers worldwide due to its potential to improve patient outcomes, optimize resource utilization, and minimize healthcare costs. In recent years, the application of machine learning models in healthcare has gained significant attention, offering the promise of accurately predicting patient readmission rates and enabling timely interventions. This study explores the effectiveness of multiple machine-learning models in predicting readmission for patients in a hospital setting.

Hospital readmissions occur when patients return to the hospital shortly after being discharged, often due to complications or inadequate post-discharge care. These readmissions not only impact patient well-being but also pose a considerable burden on healthcare systems. By developing robust predictive models, healthcare providers can identify patients at high risk of readmission and allocate appropriate resources and interventions to mitigate this risk.

In this research, we leverage a diverse set of machine learning algorithms, including but not limited to decision trees, random forests, support vector machines, and logistic regression. We can compare their performance by employing multiple models and identify the most accurate and reliable approach for predicting readmissions. We consider a range of patient-specific features, such as demographics, medical history, comorbidities, length of hospital stay, and post-discharge follow-up adherence, to capture a comprehensive view of the factors influencing readmission risk.

The dataset used in this study consists of anonymized patient records from a large hospital, spanning a defined period. The data includes both readmitted and non-readmitted cases, allowing us to train and validate our models effectively. Through rigorous experimentation and cross-validation techniques, we aim to achieve high predictive accuracy and generalizability.

The findings from this research have the potential to contribute to proactive patient care, enabling hospitals to develop targeted interventions and post-discharge strategies to reduce readmissions. By identifying high-risk patients, healthcare providers can focus on personalized care plans, improve care coordination, and ultimately enhance patient outcomes.

# LITERATURE REVIEW

In the study conducted by Lin et al. (2019), the goal was to develop a predictive model for unplanned readmissions in the ICU using recurrent neural networks with LSTM architecture. The research utilized ICU patient records encompassing various data such as demographics, medical history, vital signs, and lab results. By training and assessing LSTM models with different configurations, the study found that these models surpassed traditional machine learning approaches in foreseeing unplanned ICU readmissions. The incorporation of temporal data via LSTM proved pivotal in enhancing prediction accuracy. Noteworthy factors for readmission prediction included patient age, ICU stay duration, and specific lab measurements. The study showcases the potency of LSTM-based recurrent neural networks in predicting ICU readmissions, thereby potentially advancing patient care and resource allocation by identifying high-risk cases and enabling proactive interventions.

In the study by Yujuan Shang et al., a predictive model for assessing the 30-day hospital readmission risk in diabetic patients was developed using machine learning algorithms. Through analysis of a dataset from the Health Facts Database containing diabetic patient records, preprocessing, and model training, the random forest (RF) algorithm exhibited superior predictive performance compared to naive Bayes and tree ensemble models. Significant factors influencing readmission risk were identified, including admission times, age, diagnosis, emergencies, and sex. The study's findings offer valuable insights for healthcare interventions to reduce readmissions, particularly among the elderly, while acknowledging dataset imbalances and the need for further diagnostic exploration as limitations. Overall, the research contributes to the advancement of predictive modeling for hospital readmission risk in diabetic patients using machine learning techniques.

Two more studies harnessed machine learning for predicting hospital readmission risk. The first study by Riester et al. introduced the ToPP-HF tool designed to forecasted 30-day unplanned hospital readmissions in adult heart failure patients. Based on 13 carefully chosen variables, the tool classified patients into different risk categories, offering pharmacists a means to identify high-risk individuals and initiate appropriate interventions. The study's validation demonstrated good predictive performance, signifying its potential utility in clinical practice.

Another study shifted focus to the urology domain, aiming to predict 30-day unplanned readmissions through machine learning algorithms. Employing a variety of features spanning patient characteristics, medical history, lab results, and more, the study emphasized the prominence of the XGBoost model, showcasing its superior predictive ability over regression models. Notwithstanding its promising findings, the study was transparent about limitations encompassing data quality issues, the subjective feature selection process, and the need for further exploration into the real-world implications of the selected features.

All the above studies collectively underlined the promising prospects of machine learning in readmission risk prediction, offering personalized insights for patient care. However, they acknowledged the critical nature of addressing challenges such as data quality, model generalizability, and interpretability in ensuring the practicality and reliability of such predictive tools in real-world healthcare settings.

# RESEARCH QUESTION

**Research Question:** Can a comprehensive analysis of demographic, hospital stay, clinical, medical coding, treatment, and medication variables accurately predict the likelihood of readmission for patients, and which machine learning model exhibits the highest predictive accuracy and reliability in a hospital setting?

**Hypothesis**: By integrating and analyzing a wide spectrum of variables encompassing demographics, hospital stay details, clinical indicators, medical coding information, and treatment/medication data, we anticipate that machine learning models will be capable of effectively predicting the probability of patient readmissions. Specifically, we expect that among the machine learning algorithms tested, one will emerge as superior in terms of predictive accuracy and generalizability, thus offering a valuable tool for healthcare providers to proactively address patient readmission risk and enhance overall patient outcomes.

# 

# METHODOLOGY

## DATA ACQUISITION

The acquisition of medical data for research purposes is a complex endeavor due to stringent regulations like the Health Insurance Portability and Accountability Act (HIPAA), which prioritize safeguarding patient privacy and restrict the availability of medical data. Despite efforts to source data from diverse healthcare sources, freely accessible medical datasets remain scarce. In the context of predicting patient readmissions using multiple machine learning models, the study obtained its dataset from Kaggle, a reputable platform hosting various datasets, including healthcare-related ones. This particular dataset was deidentified to ensure patient privacy and included demographic, medical, and hospital-related information across 65 variables. With data from 25,000 patients, the Kaggle dataset provided a valuable resource for training and evaluating machine learning models in predicting readmissions.

However, while publicly available datasets like the one from Kaggle offer valuable insights, they may not comprehensively represent the entire patient population or the nuances of specific healthcare settings. Researchers must approach findings derived from these datasets with caution, acknowledging the limitations and potential biases inherent to the data source. It is essential to consider these factors to ensure that the research accurately reflects real-world healthcare scenarios and that the implications drawn are appropriately contextualized.

## VARIABLE DESCRIPTION

The following variables were identified from the dataset:

**Age Variables**: Binary indicators representing different age groups, such as patients in their 40s, 50s, 60s, 70s, and 80s-90s. These variables provide insight into age-related trends in health conditions and readmission rates. There were five preexisting age variables in the dataset.

**Race Variables**: Binary indicators for Caucasian and African American races. These variables help identify potential racial disparities in healthcare access, treatment outcomes, and readmission rates. There were only two preexisting race variables in the dataset.

**Hospital Variables**: Information about hospital stay and treatment, including hospital duration, lab procedures, non-lab procedures, medications, and outpatient/inpatient visits. Reveals patterns in healthcare utilization and treatment intensity. There were eight preexisting hospital variables in the dataset, namely time spent in the hospital (in days), number of lab procedures performed, number of non-lab procedures performed, number of medications the patient is on, number of patients from the OPD, number of patients from emergency, numbers of patients within the hospital and number of diagnoses given to the patient.

**Payer Variables**: Binary indicators for insurance types (e.g., Medicare, Managed Care) and payment methods. Offers insights into insurance coverage's influence on healthcare utilization and readmission. There were five preexisting payer variables in the dataset.

**Medical Specialty Variables**: Binary indicators for physician specialization (e.g., Internal Medicine, Cardiology). Indicates how medical specialties relate to treatment approaches and patient outcomes. There were five preexisting Medical Specialty variables in the dataset.

**Diagnosis Variables**: Binary indicators for specific diagnoses (e.g., heart failure, diabetes) across primary, secondary, and tertiary levels. Identifies prevalent health conditions and their potential impact on readmission. There were eleven preexisting diagnosis variables in the dataset.

**Medicine Variables**: Binary indicators for medication prescriptions (e.g., metformin, insulin). Reflects medication prevalence and its potential association with patient outcomes and readmission. There were twenty - five preexisting medicine variables in the dataset.

**Other Variables**: Binary indicators for medication changes (change\_No) and diabetes medication prescription (diabetesMed\_Yes). Reflects other factors in patient management and potential effects on readmission. There were two other preexisting variables in the dataset.

**Derived Variables**: The derived variables were created by aggregating or summing binary indicators from the original dataset. These variables consolidate information from related features to provide a broader perspective on certain aspects of patient data. For instance, the variable 'total\_age' captures the total count of age categories a patient falls into, shedding light on the distribution of patients across different age groups. Similarly, 'total\_race' amalgamates race indicators to offer insights into racial diversity within the dataset. 'total\_payer' combines payment-related indicators to explore the influence of insurance types on healthcare outcomes. 'total\_medical\_specialty' sums up medical specialty indicators, showcasing the distribution of patients across different specialties. 'total\_diagnosis' encapsulates the occurrence of specific diagnoses, aiding in understanding prevalent health conditions. 'total\_meds', 'total\_glucose\_tests', and 'total\_insulin' aggregate medication and test-related indicators, reflecting patients' medication and testing profiles. Lastly, 'total\_change\_diabetesMed' combines indicators about medication change and diabetes medication usage, providing an overview of treatment approaches. These derived variables enable a more comprehensive exploration of the data and can potentially enhance the modeling process by capturing broader patterns in patient characteristics and behaviors.

## EXPLORATORY DATA ANALYSIS

**Step 1: Initial Data Exploration**

At the outset, the dataset's structure and contents were examined. Basic information such as the number of rows and columns, as well as the column names, was gathered using the info(), head(5), shape, and column functions. This helped establish an initial understanding of the data.

**Step 2: Handling Missing Values**

The dataset was assessed for missing values represented as '?' and 'NaN'. Counts of '?' and 'NaN' occurrences in each column were computed using the isnull().sum().sum() function. Similarly, the total number of columns with '?' and 'NaN' as missing values were identified. This step laid the foundation for handling missing data in subsequent analysis.

**Step 3: Exploring Numerical Variables**

To better understand the distribution of numerical features, histograms were created for each variable. This allowed us to visually assess the spread, central tendency, and potential outliers within each variable. A correlation heatmap was generated to reveal correlations among numerical features, which is crucial for feature selection and model-building decisions. Additionally, pairplot analysis was conducted to visually explore relationships between pairs of variables and identify potential correlations within the dataset.

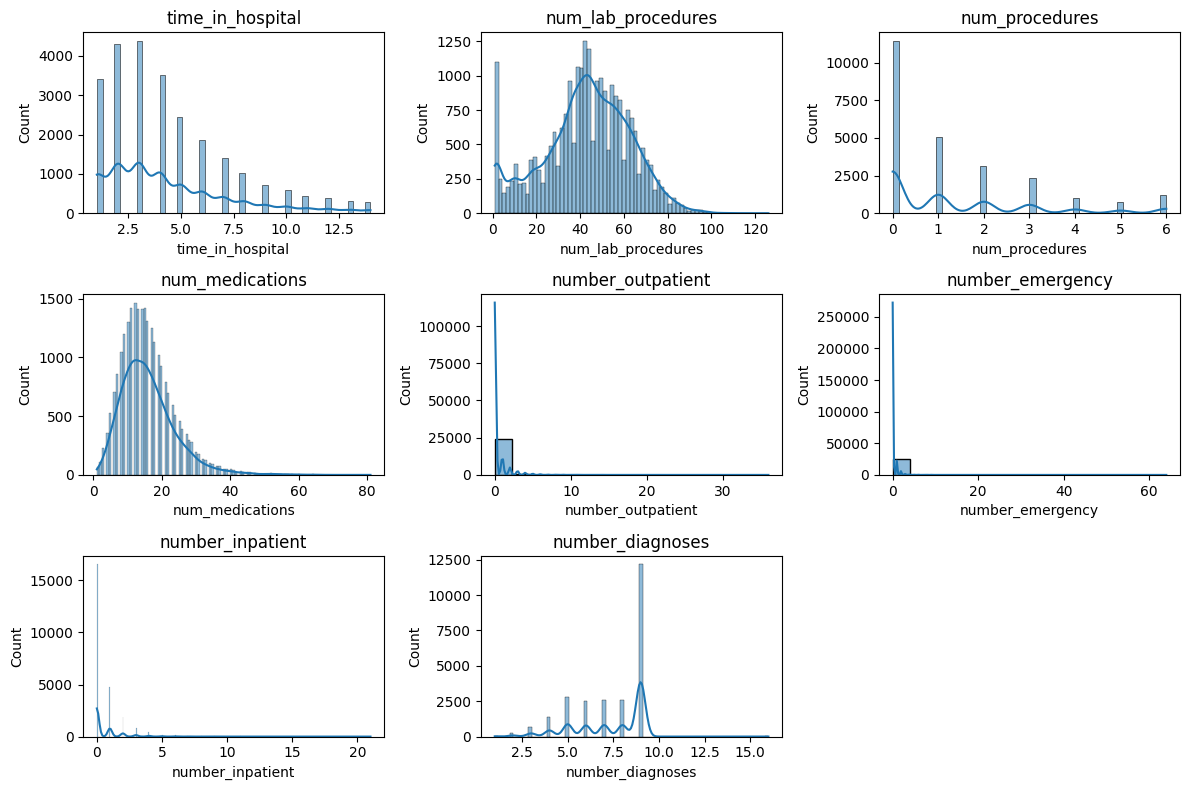


Figure 1: Histogram for all Numerical variables

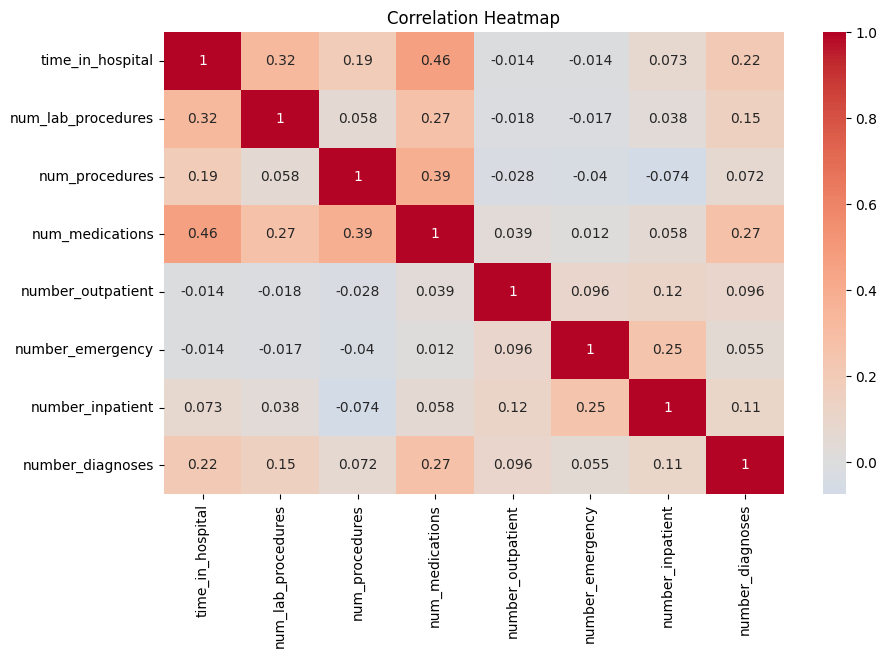


Figure 2: Heatmap for all Numerical Variables.

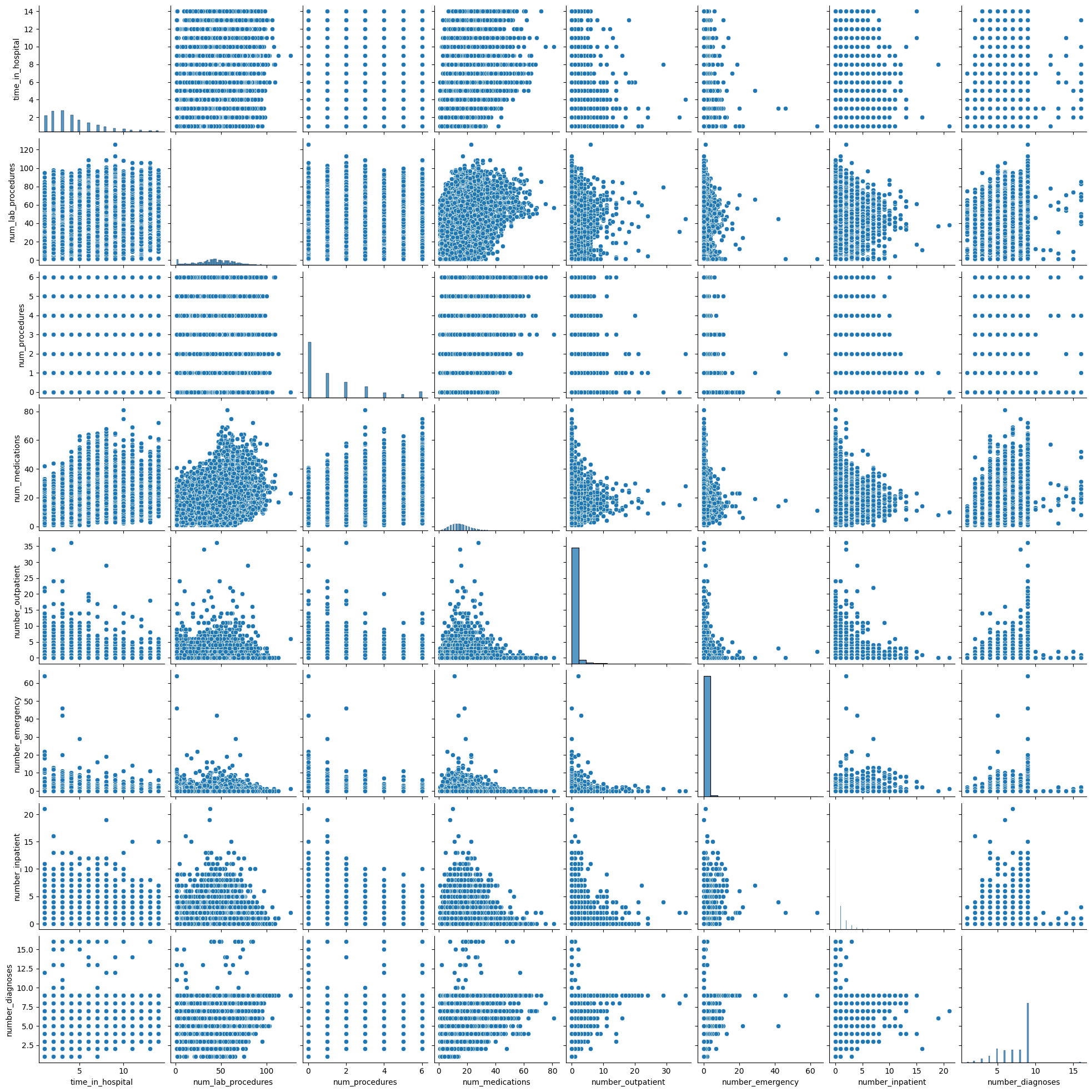


Figure 3: Pairplot for all Numerical Values

**Step 4: Categorical Variables Analysis**

Categorical variables were scrutinized using count plots. These plots depicted the frequency of each category within a variable and enabled an understanding of class imbalances, if any. Moreover, cross-tabulations and heatmaps were utilized to visualize the relationship between categorical variables and the target variable 'readmitted'. This step revealed patterns in readmission tendencies among different categories, guiding potential feature selection.

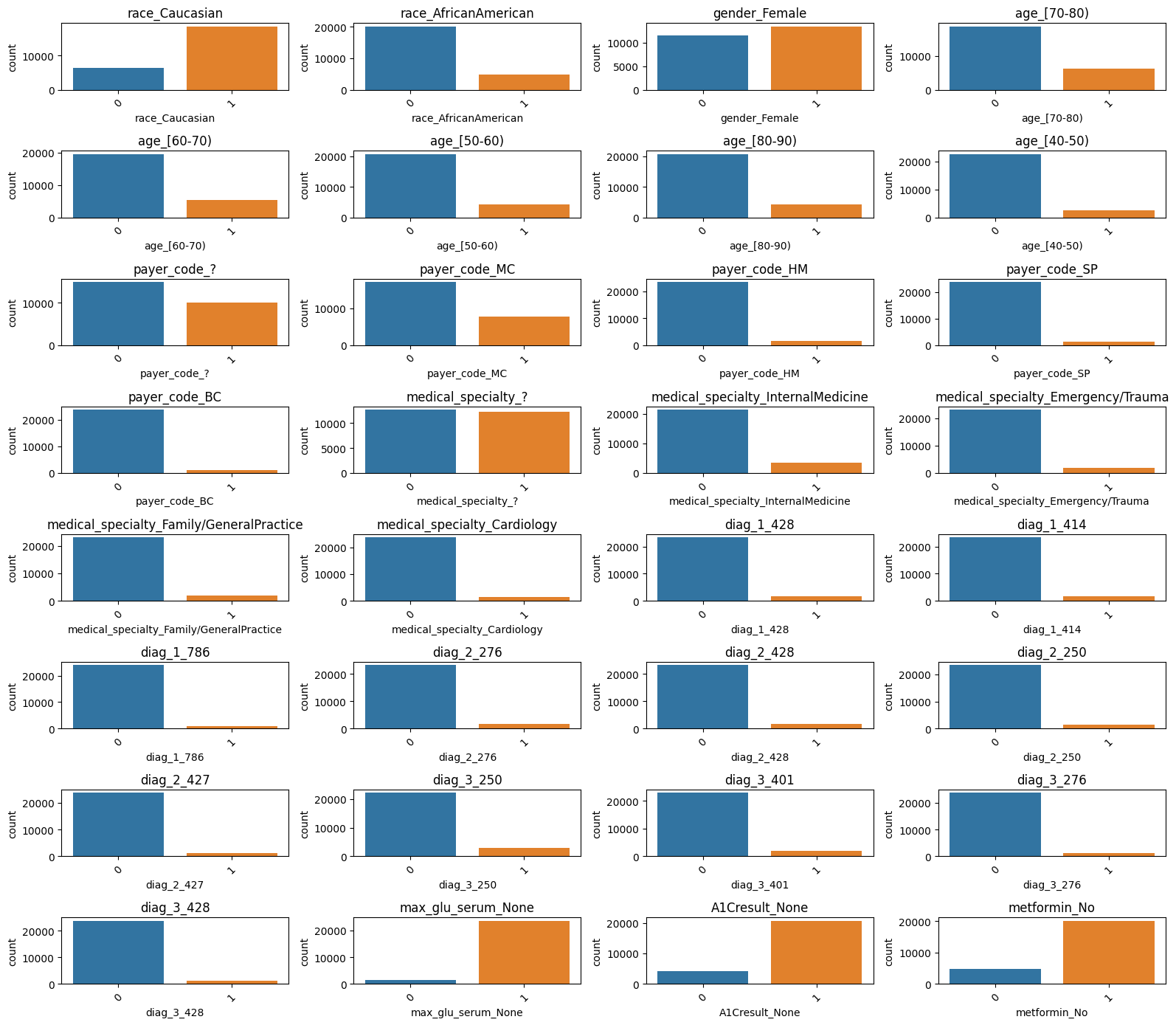


Figure 4: Countplot for all the Categorical Variables



Figure 5(a): Cross-tabulation and Heatmap for all the Categorical Variables



Figure 5(b): Cross-tabulation and Heatmap for all the Categorical Variables



Figure 5(c): Cross-tabulation and Heatmap for all the Categorical Variables



Figure 5(d): Cross-tabulation and Heatmap for all the Categorical Variables



Figure 5(e): Cross-tabulation and Heatmap for all the Categorical Variables

**Step 5: Analyzing the Target Variable**

The distribution of the target variable 'readmitted' was visualized using a count plot. This step provided insight into class imbalance and the proportion of patients who were readmitted or not. Such an understanding is crucial for modeling, as it helps to determine the need for techniques like oversampling or undersampling to balance classes.

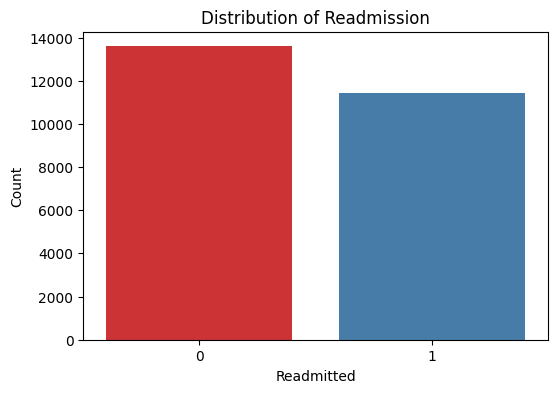


Figure 6(a): Countplot to display target variable distribution across the dataset

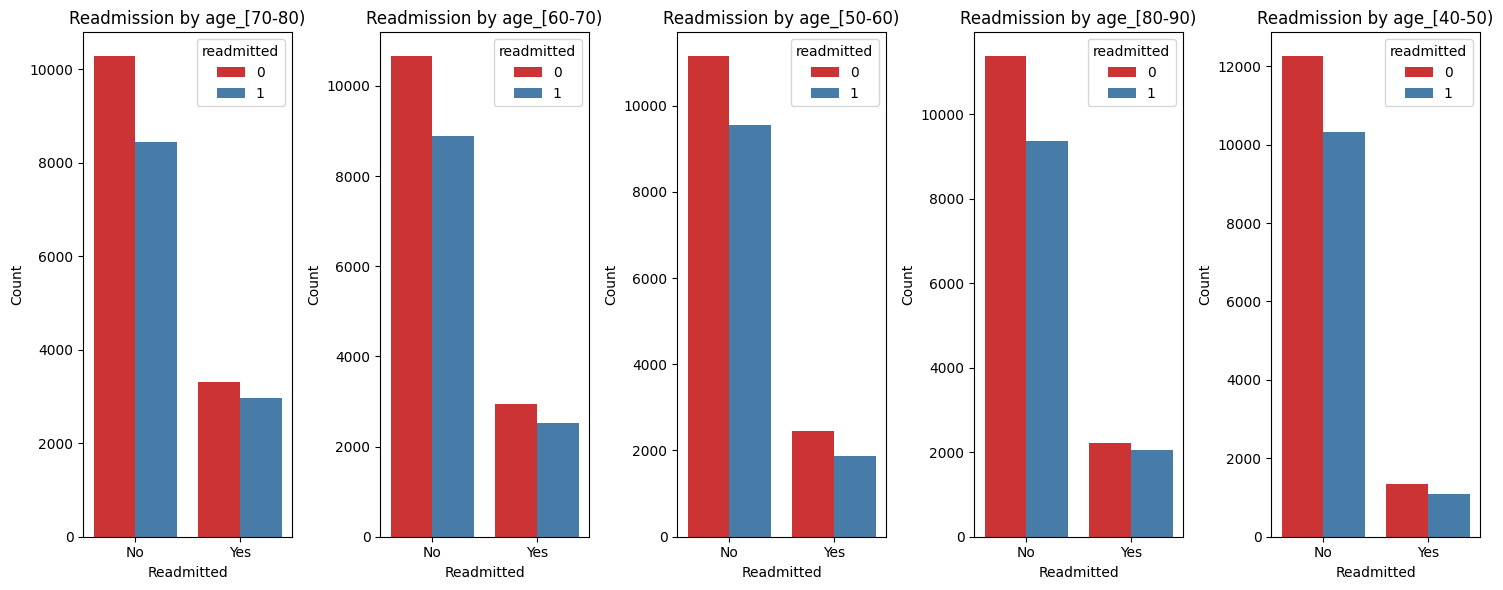


Figure 6(b): Countplot to display target variable distribution across the age variable

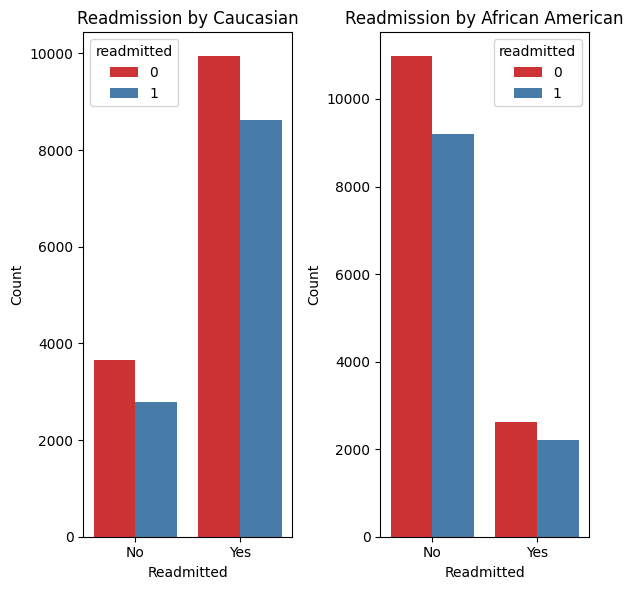
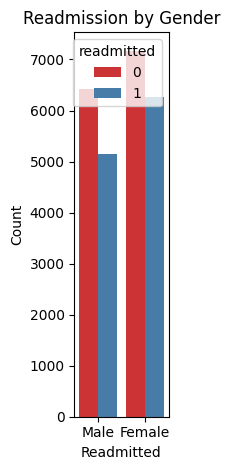
 

Figure 6(c): Countplot to display target variable distribution across the race and gender variables

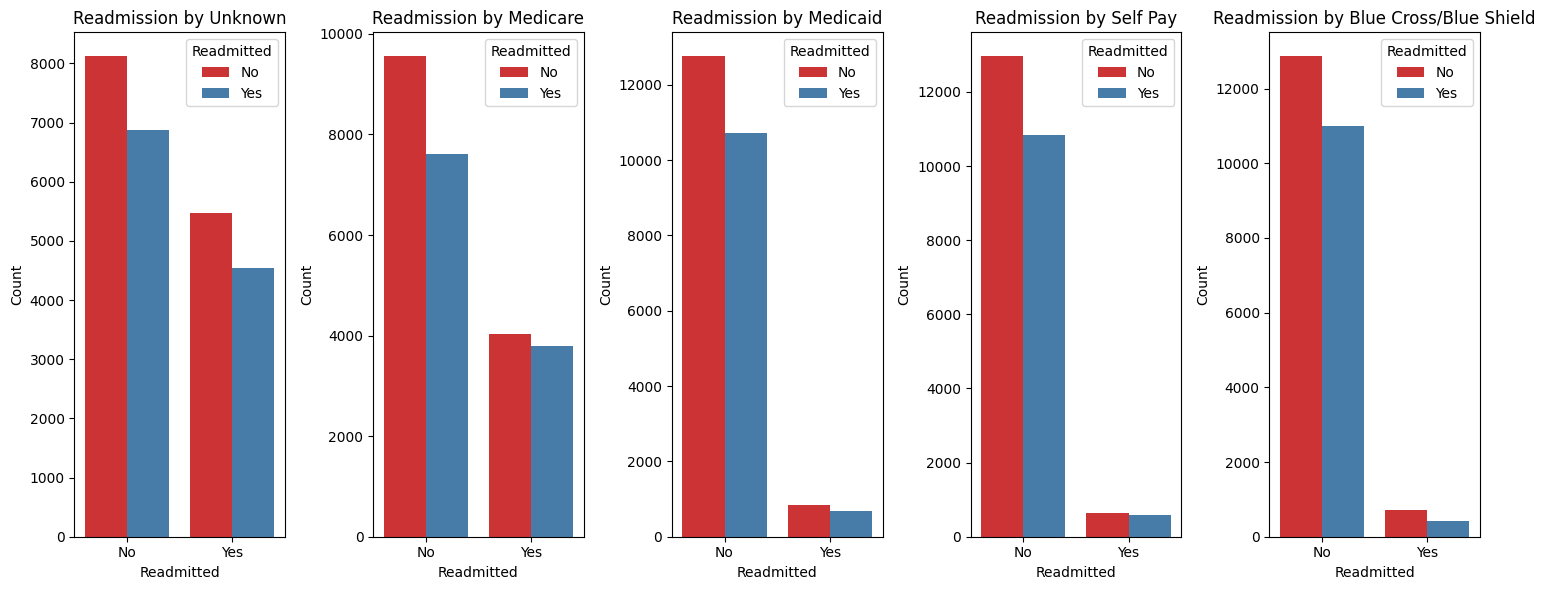


Figure 6(d): Countplot to display target variable distribution across the payer code variables

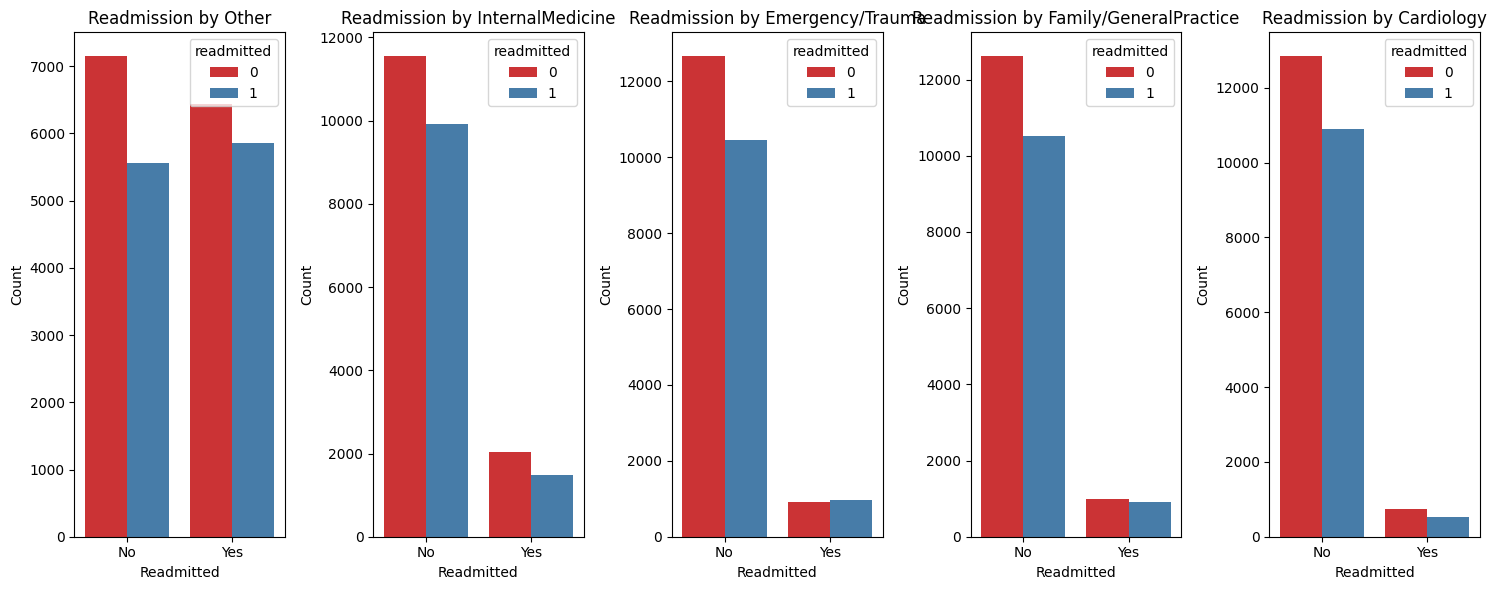


Figure 6(e): Countplot to display target variable distribution across the medical specialty

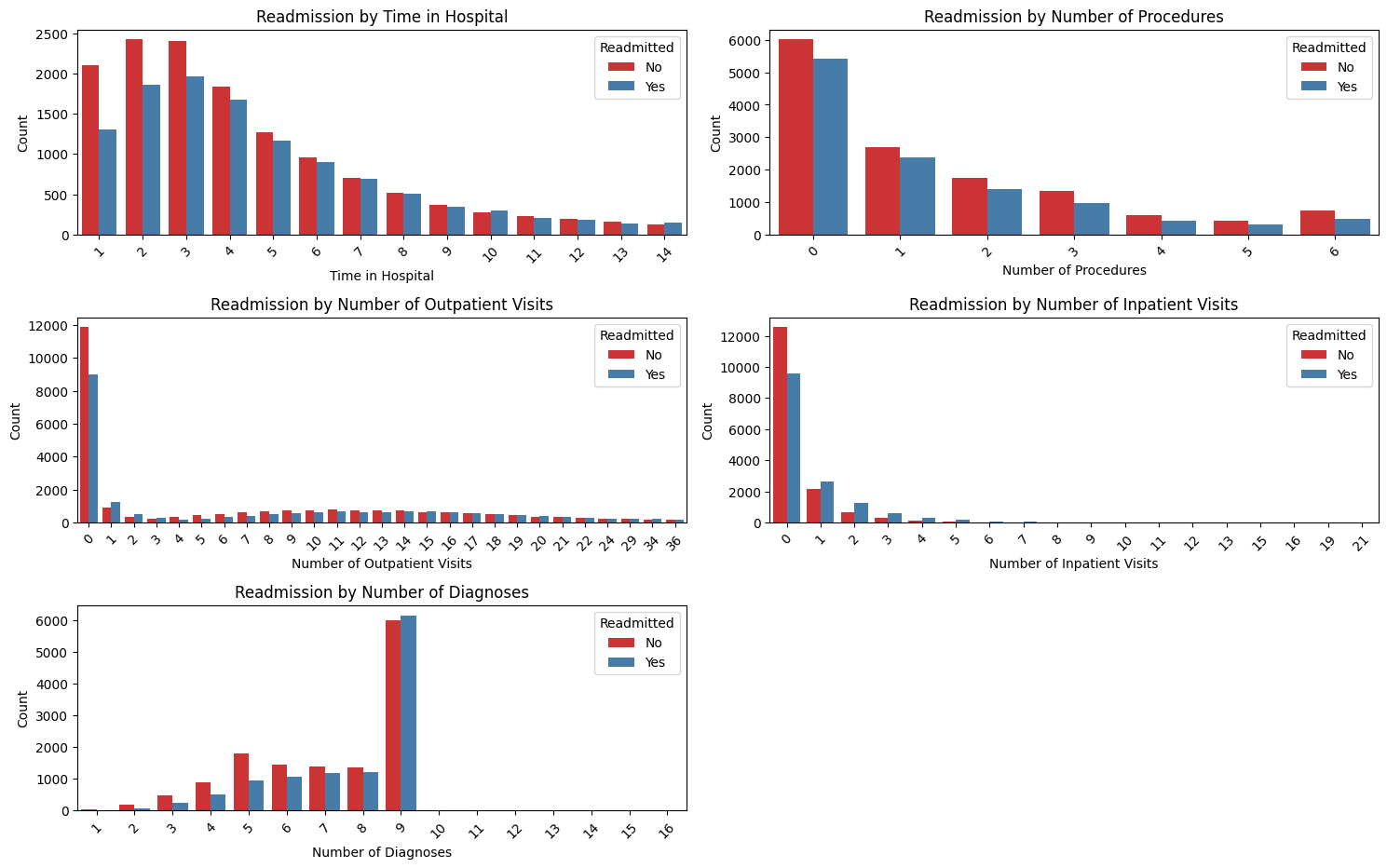


Figure 6(f): Countplot to display target variable distribution across the hospital variables

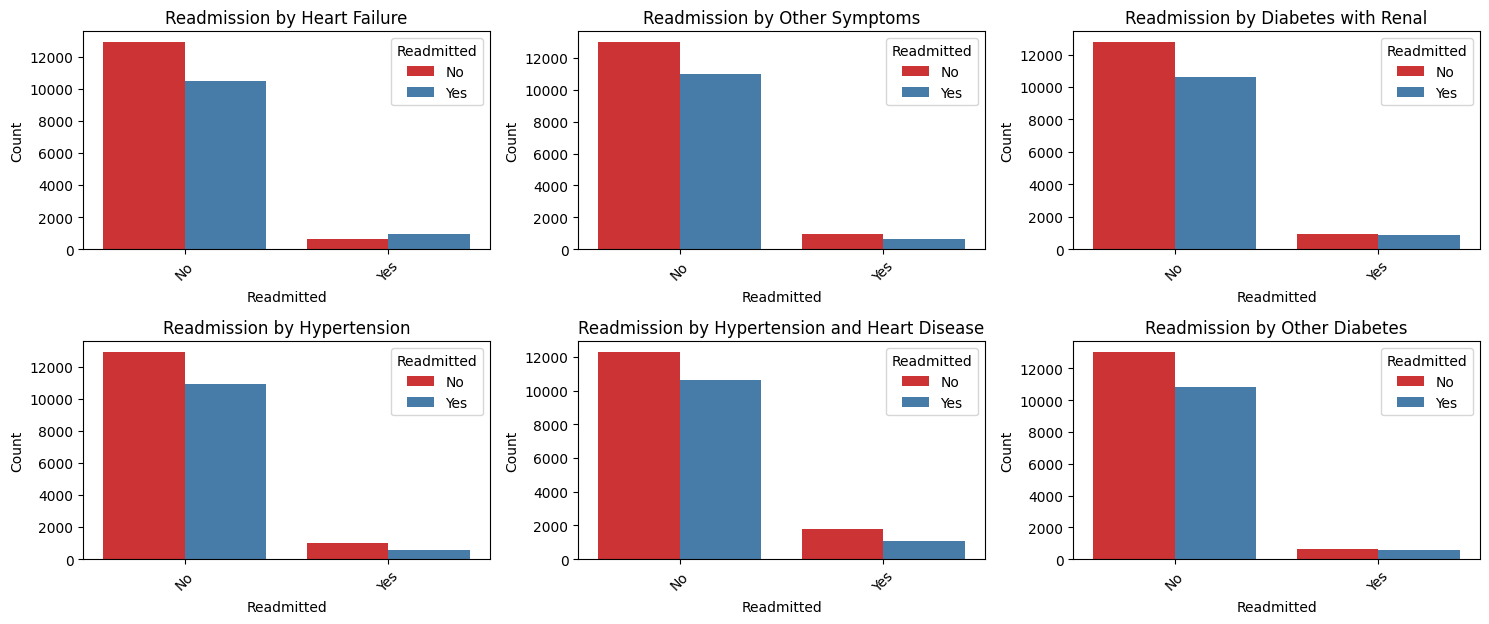


Figure 6(f): Countplot to display target variable distribution across diagnoses



Figure 6(g): Countplot to display target variable distribution across the dataset

**Step 6: In-depth Variable Analysis**

A granular analysis of individual variables was performed. For each variable, detailed visualizations were generated. For categorical variables, count plots showcased the distribution of categories by readmission status. For numerical variables, histograms or stacked histograms depicted how the variable's distribution changed based on readmission status. Furthermore, text representations provided unique values, value counts, and cross-tabulations for categorical variables, while summary statistics and distributions by readmission status were presented for numerical variables.

**Step 7: Creating Derivative Variables**

To capture broader patterns, derivative variables were generated by combining similar variables. For each derived variable, visualizations were produced to illustrate their distribution by readmission status. This step provided an opportunity to identify trends that might not have been evident when analyzing individual variables.

*Table: Analysis of Derivative Variables*

| **Variable** | **Readmitted=0** | **Readmitted=1** | **Total** | **Percent\_Readmitted=1** |
| --- | --- | --- | --- | --- |
| Total\_age | | | | |
| 0 | 1335 | 916 | 2251 | 40.693025 |
| 1 | 12255 | 10494 | 22749 | 46.129500 |
| Total\_race | | | | |
| 0 | 1036 | 594 | 1630 | 36.441718 |
| 1 | 12554 | 10816 | 23370 | 46.281558 |
| Total\_payer | | | | |
| 0 | 1889 | 1406 | 3295 | 42.670713 |
| 1 | 11701 | 10004 | 21705 | 46.090762 |
| Total\_medical\_specialty | | | | |
| 0 | 2444 | 1694 | 4138 | 40.937651 |
| 1 | 11146 | 9716 | 20862 | 46.572716 |
| Total\_diagnosis | | | | |
| 0 | 5842 | 5220 | 11062 | 47.188573 |
| 1 | 5807 | 4747 | 10554 | 44.978207 |
| 2 | 1766 | 1343 | 3109 | 43.197170 |
| 3 | 175 | 100 | 275 | 36.363636 |
| Total\_meds | | | | |
| 17 | 0 | 2 | 2 | 100.000000 |
| 18 | 9 | 6 | 15 | 40.000000 |
| 19 | 182 | 153 | 335 | 45.671642 |
| 20 | 1061 | 843 | 1904 | 44.275210 |
| 21 | 2857 | 2436 | 5293 | 46.023049 |
| 22 | 6005 | 5631 | 11636 | 48.392919 |
| 23 | 3476 | 2339 | 5815 | 40.223560 |
| Total\_glucose\_tests | | | | |
| 0 | 35 | 38 | 73 | 52.054795 |
| 1 | 2986 | 2504 | 5490 | 45.610200 |
| 2 | 10569 | 8868 | 19437 | 45.624325 |
| Total\_insulin | | | | |
| 3 | 0 | 1 | 1 | 100.000000 |
| 4 | 44 | 51 | 95 | 53.684211 |
| 5 | 6902 | 6352 | 13254 | 47.925155 |
| 6 | 6644 | 5006 | 11650 | 42.969957 |
| total\_change\_diabetesMed | | | | |
| 1 | 9381 | 7869 | 17250 | 45.617391 |
| 2 | 4209 | 3541 | 7750 | 45.690323 |

**Step 8: EDA Findings and Modeling Choices**

The EDA process unearthed key insights that informed modeling choices. Patterns in the data, such as relationships between certain variables and readmission, guided the selection of features for modeling. For example, variables that exhibited significant differences in readmission rates across categories were considered essential for predictive modeling. Additionally, understanding correlations and distributions helped identify potential multicollinearity and outliers, informing preprocessing steps before feeding the data into machine learning models.

In conclusion, the EDA process was a comprehensive exploration of the dataset, involving a mix of visualizations, statistical summaries, and derived variables. The findings directly influenced modeling decisions by highlighting relevant features, relationships, and data preprocessing steps necessary for building accurate and effective predictive models.

## IDENTIFYING VARIABLE MODELS FOR ANALYSIS

During the phase of identifying influential variables, a systematic evaluation was undertaken involving diverse models, each targeting different subsets of features, to assess their predictive accuracy. The analysis commenced by isolating numerical variables, leading to the attainment of an accuracy of 58.75%. Subsequently, the focus shifted to the evaluation of categorical variables in isolation, resulting in an accuracy of 54.93%. Subsequent stages involved the exploration of distinct feature sets encompassing age, race, attributes related to hospitalization, payer information, medical specialties, diagnostic codes, medication details, and factors pertinent to diabetes.

Of particular note, the model constructed using hospital-related attributes exhibited a predictive accuracy of 58.75%. Similarly, models emphasizing diagnostic codes showcased accuracies spanning the range of 54.42% to 56.24%. Beyond conventional features, the investigation extended to derived variables, incorporating cumulative age, race, medical specialty, count of diagnoses, medication usage, glucose tests, insulin administration, and changes in diabetes medication. This exploration yielded a predictive accuracy of 62.8%. Moreover, upon amalgamating pre-existing and derived variables, the resultant model achieved an accuracy of 61.87%.

The thorough analysis conducted during this stage provided insights into the distinct contributions of each variable group, serving as a guiding factor in the selection and refinement of features to enhance model performance. Lastly, the ranking of features based on their predictive capabilities not only offers a valuable reference point for subsequent efforts in feature prioritization but also contributes to a more nuanced understanding of the dynamics governing predictive accuracy in the context of the study. Finally, the model with only independent variables was selected because of the accuracy of this model.

## DATA MODELING

### *BUILDING THE MODEL*

To construct a predictive model with high accuracy, a systematic approach was taken to split the dataset into training and testing subsets, using a 75-25 ratio. This division ensured an unbiased evaluation of the model's predictive abilities. To enhance the model's effectiveness, a comprehensive preprocessing strategy was employed. Numerical attributes were standardized using the StandardScaler, while categorical features were encoded via the OneHotEncoder with an 'ignore' strategy. A pivotal element was the creation of a preprocessor, seamlessly integrating numerical and categorical transformations.

This orchestrated pipeline, merging preprocessing with a RandomForestClassifier, underwent a rigorous fitting process using the training data. Subsequently, predictions were made on unseen testing data, with accuracy computed to quantify the model's performance. The achieved accuracy of 0.628 indicated that the model correctly predicted outcomes about 62.8% of the time. This rigorous approach not only demonstrated the model's predictive prowess but also underscored the significance of careful preprocessing and thoughtful integration of features for optimal predictive performance.

### *ORDINARY LEAST SQUARES (OLS) REGRESSION*

An Ordinary Least Squares (OLS) regression analysis was conducted to investigate the associations between various independent variables and the dependent variable 'readmitted.' The results summary from the OLS regression provided valuable insights into the model's performance. The R-squared value, approximately 0.08, indicated that the model could explain around 8% of the variability in the 'readmitted' outcome. The adjusted R-squared value, approximately 0.077, considered the model's complexity and suggested that a modest portion of the variance was captured. The F-statistic, with a value of 36.53 and a corresponding p-value, indicated that at least some coefficients within the model were statistically significant. Nevertheless, specific variables had p-values exceeding the conventional threshold of 0.05, hinting at a lack of statistical significance for those features.

The coefficients associated with each independent variable illuminated the direction and strength of their impact on 'readmitted.' Positive coefficients implied a positive relationship, while negative coefficients indicated a negative relationship. However, careful consideration of these coefficients alongside their respective p-values was essential to comprehensively assess the statistical significance of these relationships. Overall, the OLS regression analysis provided insights into the predictive capacity of the selected independent variables for the 'readmitted' outcome, as indicated by the R-squared value and p-values.

*Table: OLS Regression Results*

| **Dep. Variable** | readmitted | **R-squared** | 0.080 |
| --- | --- | --- | --- |
| **Model** | OLS | **Adj. R-squared** | 0.077 |
| **Method** | Least Squares | **F-statistic** | 36.53 |
| **Date** | Tue, 22 Aug 2023 | **Prob (F-statistic)** | 0.00 |
| **Time** | 03:24:24 | **Log-Likelihood** | -17013 |
| **No. Observations** | 25000 | **AIC** | 3.415e+04 |
| **Df Residuals** | 24940 | **BIC** | 3.463e+04 |
| **Df Model** | 59 |  |  |
| **Covariance Type** | nonrobust |  |  |
| **Omnibus** | 109088.534 | **Durbin-Watson** | 1.998 |
| **Prob(Omnibus)** | 0.000 | **Jarque-Bera (JB)** | 3116.151 |
| **Skew** | 0.182 | **Prob(JB)** | 0.00 |
| **Kurtosis** | 1.309 | **Cond. No.** | 1.19e+16 |

### *RANDOM FOREST CLASSIFIER MODEL AND TESTING ACCURACIES*

The RandomForestClassifier model was instantiated with 100 estimators and a random state of 42, and subsequently trained on the provided training dataset. After training, the model's predictive capabilities were assessed by making predictions on the test set. The resulting accuracy of the model was determined to be 63%, demonstrating its ability to correctly classify instances.

Further insights into the model's performance were obtained through a comprehensive set of evaluation metrics. The confusion matrix offered a breakdown of true positive, true negative, false positive, and false negative predictions. Notably, the model correctly identified 2521 true negatives and 1406 true positives but also produced 880 false positives and 1443 false negatives. Subsequent metrics such as training and validation accuracy, precision, recall, F1 score, and balanced accuracy were computed. The model achieved perfect training accuracy, suggesting an overfitting to the training data, while the validation accuracy stood at 62.83%. Likewise, the model demonstrated high precision and recall on the training data, with slightly lower values observed on the validation data. These metrics collectively provide a comprehensive assessment of the model's strengths and areas for improvement, serving as valuable guidance for refining its performance.

## LOGISTIC REGRESSION

In the pursuit of addressing class imbalance within the training data, the KMeansSMOTE oversampling method was employed. This technique sought to create a more balanced representation of the classes by adjusting the cluster balance threshold, choosing a neighbor count of 12, and aiming for a sampling strategy of 99%. The outcome of this resampling was a modified dataset, X\_resampled and y\_resampled, which was subsequently utilized to train a Logistic Regression model. The model's efficacy was then evaluated on the test set, culminating in a comprehensive classification report detailing precision, recall, F1-score, and support metrics for each class.

The training process was similarly carried out for the Logistic Regression model on the resampled data, further illuminating the model's ability to generalize its predictions within the training context. Additionally, a more intricate approach was adopted by implementing Polynomial features into the Logistic Regression model. This transformation, achieved through polynomial expansion of degree 2, resulted in a model of greater complexity. Training the polynomial Logistic Regression model on the resampled data enabled its evaluation on the test set, producing a classification report that furnished detailed insights into its predictive efficacy. Correspondingly, the polynomial Logistic Regression model was evaluated on the training set, showcasing its performance on the resampled training data. These comprehensive classification reports served as valuable tools for assessing and refining the predictive capacities of both models, providing essential insights into precision, recall, F1-score, and overall predictive capabilities.

Furthermore, the evaluation of the Linear Regression model also offered a holistic perspective on its predictive performance. The model demonstrated an accuracy of around 60% on the test dataset, along with associated metrics like precision, recall, F1-score, and support for each class. These findings were echoed in its performance on the training set, reaffirming its generalization capabilities. Similarly, the Polynomial Logistic Regression model exhibited a stable AUC score of 0.63, indicating its consistent discriminatory power.

*Table: Logistic Regression Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.62 | 0.68 | 0.65 | 3401 | 0.62 | 0.68 | 0.65 | 10189 |
| Class 1 | 0.57 | 0.50 | 0.53 | 2849 | 0.64 | 0.57 | 0.60 | 10093 |
| Accuracy |  |  | 0.60 | 6250 |  |  | 0.63 | 20282 |
| Macro  Avg | 0.59 | 0.59 | 0.59 | 6250 | 0.63 | 0.63 | 0.63 | 20282 |
| Weighted  Avg | 0.60 | 0.60 | 0.59 | 6250 | 0.63 | 0.63 | 0.63 | 20282 |

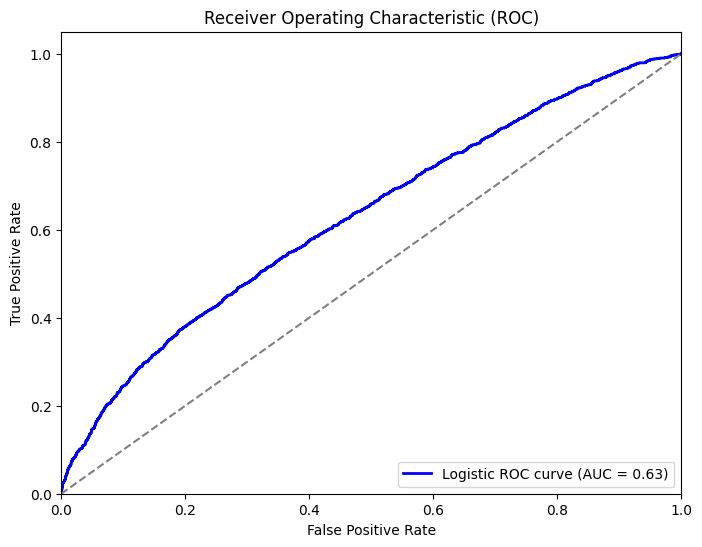


Figure 7: ROC Curve for Logistic Regression Model.

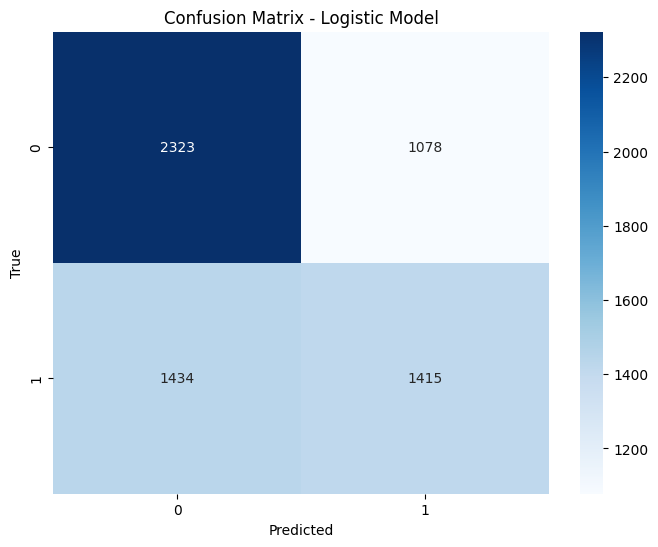


Figure 8: Heatmap for Logistic Regression Model

*Table: Polynomial Logistic Regression Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.62 | 0.69 | 0.66 | 3401 | 0.62 | 0.69 | 0.65 | 10189 |
| Class 1 | 0.58 | 0.50 | 0.54 | 2849 | 0.65 | 0.57 | 0.61 | 10093 |
| Accuracy |  |  | 0.60 | 6250 |  |  | 0.63 | 20282 |
| Macro  Avg | 0.60 | 0.60 | 0.60 | 6250 | 0.63 | 0.63 | 0.63 | 20282 |
| Weighted  Avg | 0.60 | 0.60 | 0.60 | 6250 | 0.63 | 0.63 | 0.63 | 20282 |

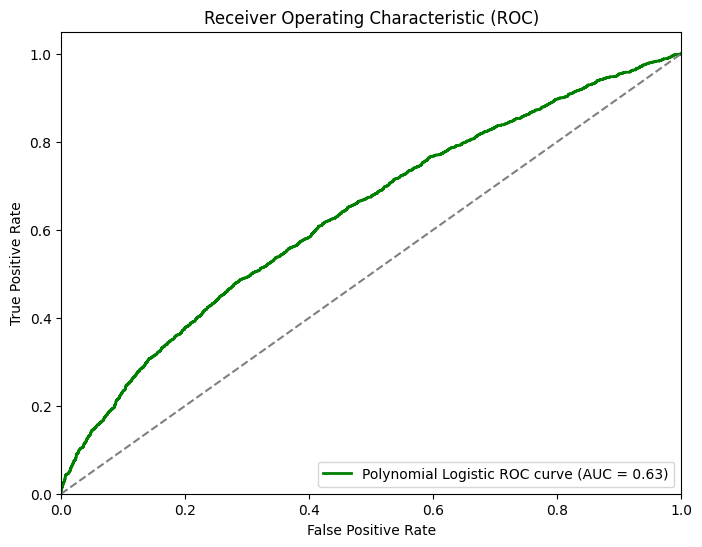


Figure 9: ROC Curve for Polynomial Logistic Regression Model

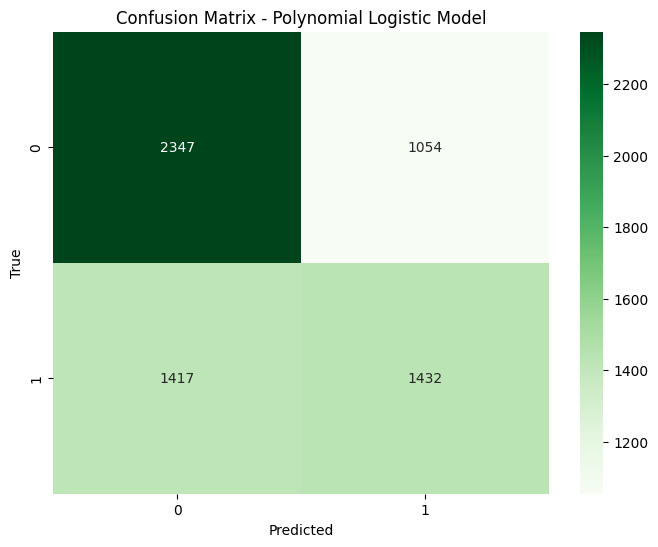


Figure 10: Heatmap for Polynomial Logistic Regression Model

## SUPPORT VECTOR MACHINE (SVM)

The Support Vector Machine (SVM) model underwent training with a linear kernel and regularization parameter (C) set at 1.0. Its predictive capabilities were then evaluated using the test dataset, yielding an accuracy of around 0.60, indicative of its overall prediction precision. The resultant confusion matrix presented a comprehensive breakdown of accurate and erroneous predictions, revealing the distribution of true positives, true negatives, false positives, and false negatives. A detailed classification report furnished precision, recall, F1-score, and support metrics for both class labels (0 and 1), accompanied by macro and weighted averages.

For class 0, the SVM model showcased a precision of 0.60, a recall of 0.88, and an F1-score of 0.71, while class 1 exhibited a precision of 0.67, a recall of 0.29, and an F1-score of 0.40. Macro and weighted average F1-scores were recorded at 0.56 and 0.57, respectively. The model's efficacy in prediction was further evaluated on the training dataset, with similar results observed as in the test set. The SVM model achieved a training accuracy of 0.61, akin to its test accuracy, with class-specific metrics echoing those from the test set.

In addition, the Receiver Operating Characteristic (ROC) curve was plotted to assess the model's discriminatory capability. This curve illuminated the equilibrium between true positive and false positive rates at varying probability thresholds. The area under the ROC curve (AUC), quantifying the model's discrimination prowess, computed to 0.66. Lastly, a heatmap of the confusion matrix was visually represented, offering insights into the distribution of predicted versus actual class labels and aiding in the identification of patterns and disparities in the model's predictions.

*Table: Support Vector Machine Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.60 | 0.88 | 0.71 | 3401 | 0.60 | 0.88 | 0.71 | 10189 |
| Class 1 | 0.67 | 0.29 | 0.40 | 2849 | 0.67 | 0.29 | 0.41 | 8561 |
| Accuracy |  |  | 0.61 | 6250 |  |  | 0.61 | 18750 |
| Macro  Avg | 0.63 | 0.58 | 0.56 | 6250 | 0.64 | 0.59 | 0.56 | 18750 |
| Weighted  Avg | 0.63 | 0.61 | 0.57 | 6250 | 0.63 | 0.61 | 0.57 | 18750 |

## 

Figure 11: ROC Curve for SVM Model

## 

Figure 12: Heatmap for SVM Model

## GRADIENT BOOSTING

The exploration commenced by initializing and training a Gradient Boosting model on the provided dataset, configuring hyperparameters such as the number of estimators (100), maximum depth (3), and learning rate (0.80). Subsequently, a thorough evaluation of the model's performance on the training data ensued. The resulting classification report encapsulated key metrics for class 0 and class 1 within the training dataset. Notably, for class 0, the report indicated a precision of 0.70, recall of 0.80, and F1-score of 0.75, backed by a support count of 10189. Similarly, for class 1, precision stood at 0.72, recall at 0.60, and F1-score at 0.65, with a support of 8561. This comprehensive assessment illuminated the model's adeptness in distinguishing between the two classes.

To ensure data uniformity, an essential preprocessing step involved replacing invalid characters in column names with underscores, thereby facilitating seamless data processing. This preparatory phase was instrumental in preserving data accuracy and integrity. Moving forward, an XGBoost model was instantiated, parameterized with 150 estimators and a maximum depth of 4, and subsequently trained. Both the training and testing datasets underwent meticulous evaluation, yielding insightful classification reports. In the case of the testing data, the XGBoost model exhibited a precision of 0.64 for class 0, a recall of 0.72, and an F1-score of 0.68, with a support of 3401. For class 1, precision was 0.60, recall was 0.51, and F1-score was 0.55, supported by 2849 instances. The calculated accuracy for the XGBoost model on the testing data was 0.62, with a macro average F1-score of 0.61 and a weighted average F1-score of 0.62, derived from a dataset size of 6250.

The analysis culminated with the visualization of an ROC curve for the XGBoost model, portraying its ability to discriminate between classes across varying probability thresholds. The area under the ROC curve (AUC) emerged as 0.66, quantifying the model's discrimination capabilities and indicating moderate performance. To enrich the interpretive aspect of the model's predictions, a confusion matrix was generated, translating prediction outcomes into a graphical representation that highlighted true positives, true negatives, false positives, and false negatives. This visualization mechanism facilitated a more intuitive grasp of the model's predictive dynamics and patterns.

*Table: XG Boosting Model*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.70 | 0.80 | 0.75 | 10189 | 0.64 | 0.72 | 0.68 | 3401 |
| Class 1 | 0.72 | 0.60 | 0.65 | 8561 | 0.60 | 0.51 | 0.55 | 2849 |
| Accuracy |  |  | 0.71 | 18750 |  |  | 0.62 | 6250 |
| Macro  Avg | 0.71 | 0.70 | 0.70 | 18750 | 0.62 | 0.61 | 0.61 | 6250 |
| Weighted  Avg | 0.71 | 0.71 | 0.71 | 18750 | 0.62 | 0.62 | 0.62 | 6250 |

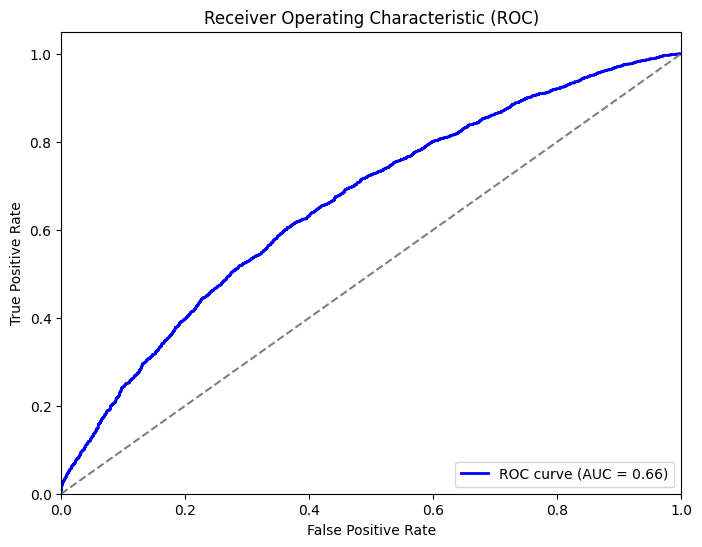


Figure 13: ROC Curve for XG Boosting Model

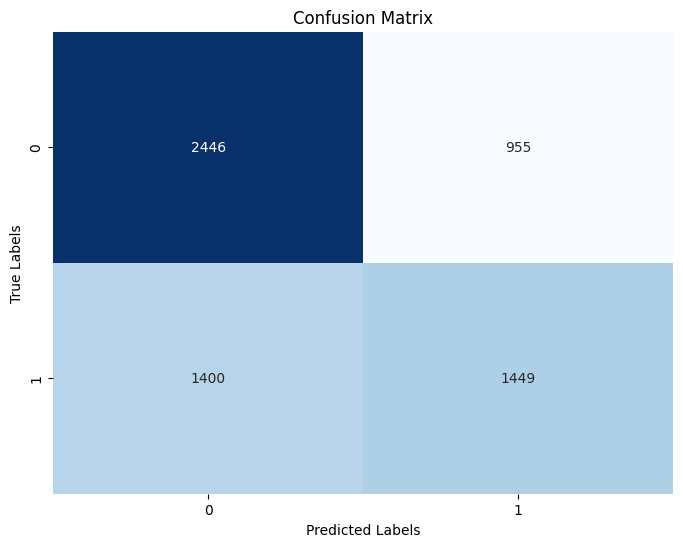


Figure 14: Heatmap for XG Boosting Model

## GAUSSIAN NAIVE BAYES

The analysis began by initializing and training a Gaussian Naive Bayes model using the dataset. Once trained, the model's performance was assessed on both the training and testing datasets, generating comprehensive classification reports for each. These reports provided a detailed breakdown of precision, recall, F1-score, and support metrics for both class 0 and class 1.

In the context of the training set, the Gaussian Naive Bayes model showcased a precision of 0.62 for class 0 and 0.59 for class 1. Accompanying these figures were recall values of 0.72 for class 0 and 0.47 for class 1, with corresponding F1-scores of 0.67 and 0.52, respectively. The model exhibited an overall training accuracy of 0.61. Additionally, the macro average F1-score was determined as 0.59, while the weighted average F1-score stood at 0.60.

Similarly, the evaluation of the testing set revealed the model's precision to be 0.63 for class 0 and 0.60 for class 1. Notably, the recall values were 0.73 for class 0 and 0.48 for class 1, accompanied by F1-scores of 0.67 and 0.53, respectively. The overall testing accuracy of the model reached 0.62. Furthermore, the macro average F1-score was 0.60, with a weighted average F1-score of 0.61.

In pursuit of a comprehensive assessment, an ROC curve was generated to depict the Gaussian Naive Bayes model's capacity for discrimination. The resulting area under the ROC curve (AUC) was quantified at 0.64, offering a measurable gauge of the model's ability to distinguish between the two classes. To enhance the interpretability of prediction outcomes, a visual representation in the form of a confusion matrix was crafted. This matrix provided a heatmapped depiction of true positives, true negatives, false positives, and false negatives, enabling a more intuitive grasp of the model's predictive performance.

*Table: Naive Bayes Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.63 | 0.73 | 0.67 | 3401 | 0.62 | 0.72 | 0.67 | 10189 |
| Class 1 | 0.60 | 0.48 | 0.53 | 2849 | 0.59 | 0.47 | 0.52 | 8561 |
| Accuracy |  |  | 0.62 | 6250 |  |  | 0.61 | 18750 |
| Macro  Avg | 0.61 | 0.60 | 0.60 | 6250 | 0.60 | 0.60 | 0.59 | 18750 |
| Weighted  Avg | 0.61 | 0.62 | 0.61 | 6250 | 0.60 | 0.61 | 0.60 | 18750 |

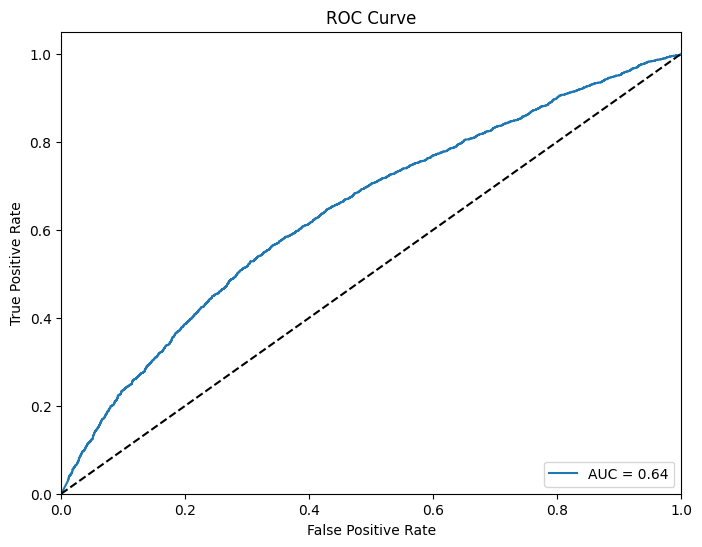


Figure 15: ROC Curve for Naive Bayes Model

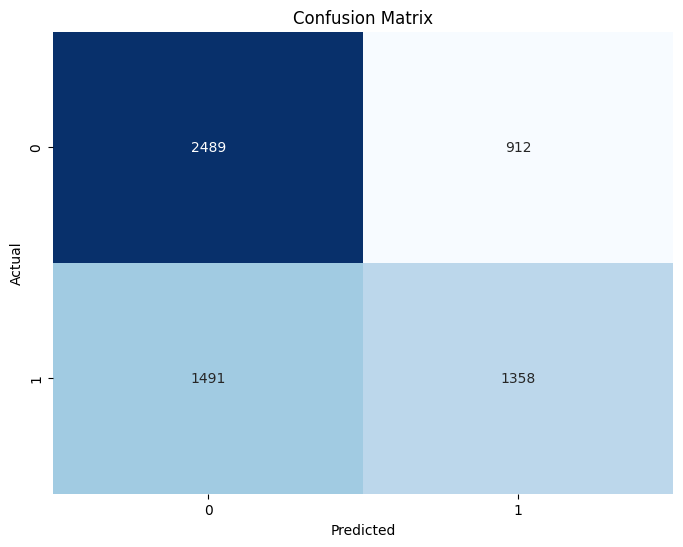


Figure 16: Heatmap for Naive Bayes Model

## K - NEAREST NEIGHBORS

K-Nearest Neighbors (KNN) model was trained using the scaled training data, considering the five nearest neighbors for predictions. The model's performance was subsequently assessed on both the training and testing datasets. The classification reports offered a comprehensive overview of its precision, recall, F1-score, and accuracy for each class (0 and 1). On the training set, the KNN model achieved an accuracy of around 0.72, showcasing a good balance between precision and recall.

Upon transitioning to the testing set, the model maintained an accuracy of approximately 0.56, demonstrating its performance on previously unseen data. The provided precision, recall, and F1-score metrics allowed for a deeper understanding of its predictive capabilities.

The Receiver Operating Characteristic (ROC) curve further enriched the analysis by illustrating the trade-off between true positive rate and false positive rate across different probability thresholds. With an Area Under the Curve (AUC) score of 0.58, the model's ability to distinguish between classes was quantified. Additionally, the confusion matrix visually depicted the breakdown of predicted and actual class labels, providing insights into prediction patterns.

*Table: K - Nearest Neighbor Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.59 | 0.64 | 0.61 | 3401 | 0.72 | 0.78 | 0.75 | 10189 |
| Class 1 | 0.52 | 0.46 | 0.49 | 2849 | 0.71 | 0.64 | 0.67 | 8561 |
| Accuracy |  |  | 0.56 | 6250 |  |  | 0.72 | 18750 |
| Macro  Avg | 0.55 | 0.55 | 0.55 | 6250 | 0.71 | 0.71 | 0.71 | 18750 |
| Weighted  Avg | 0.56 | 0.56 | 0.56 | 6250 | 0.72 | 0.72 | 0.71 | 18750 |

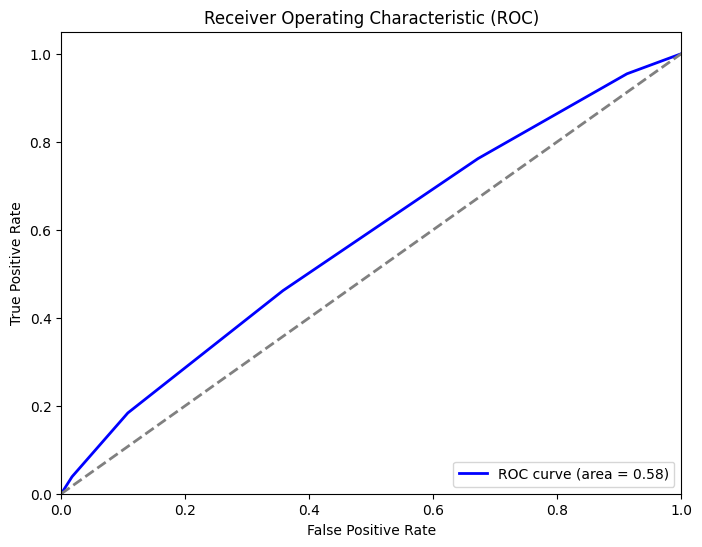


Figure 17: ROC Curve for KNN Model

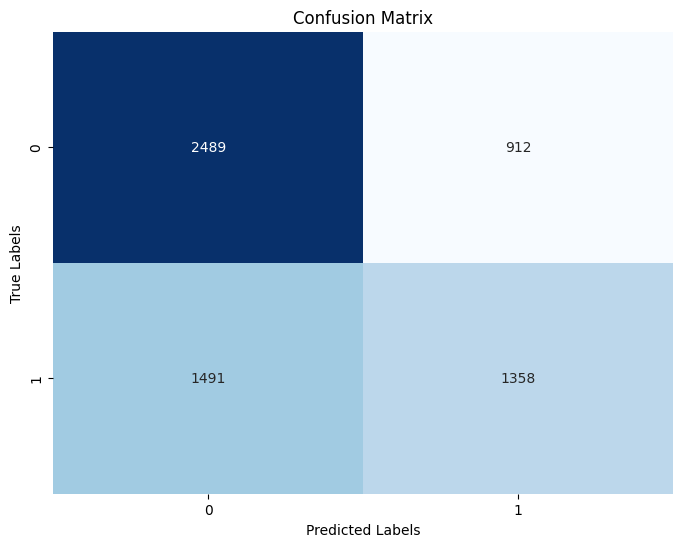


Figure 18: Heatmap for KNN Model

## NEURAL NETWORKS

A neural network model was constructed using the Keras library with TensorFlow backend. The model architecture consisted of an input layer, followed by two hidden layers with ReLU activation functions, and finally an output layer with a sigmoid activation function. The model was compiled with the 'adam' optimizer and the 'binary\_crossentropy' loss function, along with accuracy as the evaluation metric.

The training process commenced with the model being trained on the scaled training data for 10 epochs with a batch size of 32. A validation split of 0.1 was employed for monitoring training progress. Upon completion of training, the model was evaluated on the testing set. Predicted probabilities were obtained, and a threshold of 0.5 was applied to classify instances as either class 0 or 1. The classification report for the testing set was generated, encompassing precision, recall, F1-score, and accuracy metrics. The model's performance on the training set was also evaluated, and a corresponding classification report was produced. These reports provided insights into the model's performance across different metrics for both the testing and training datasets.

To visualize the model's ability to distinguish between classes, an ROC curve was created using the predicted probabilities. The Area Under the Curve (AUC) score was calculated to quantify the curve's discriminatory power, yielding a value of 0.63. Finally, a confusion matrix was generated to offer a visual representation of the model's predictions against actual class labels. The matrix provided insights into the distribution of true positives, true negatives, false positives, and false negatives.

*Table: Neural Network Model - Classification Report*

|  | Test Set | | | | Training Set | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Precision | Recall | F1-Score | Support | Precision | Recall | F1-Score | Support |
| Class 0 | 0.62 | 0.64 | 0.63 | 3401 | 0.75 | 0.77 | 0.76 | 10189 |
| Class 1 | 0.56 | 0.54 | 0.55 | 2849 | 0.71 | 0.70 | 0.71 | 8561 |
| Accuracy |  |  | 0.59 | 6250 |  |  | 0.73 | 18750 |
| Macro  Avg | 0.59 | 0.59 | 0.59 | 6250 | 0.73 | 0.73 | 0.73 | 18750 |
| Weighted  Avg | 0.59 | 0.59 | 0.59 | 6250 | 0.73 | 0.73 | 0.73 | 18750 |

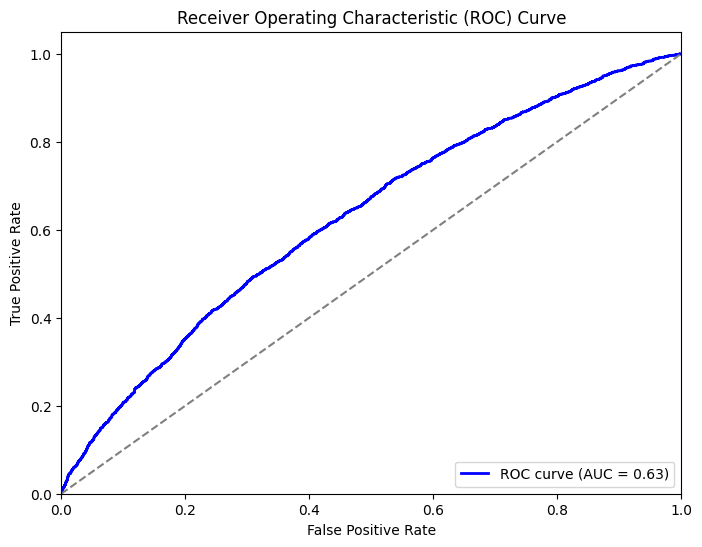


Figure 19: ROC Curve for Neural Networks Model

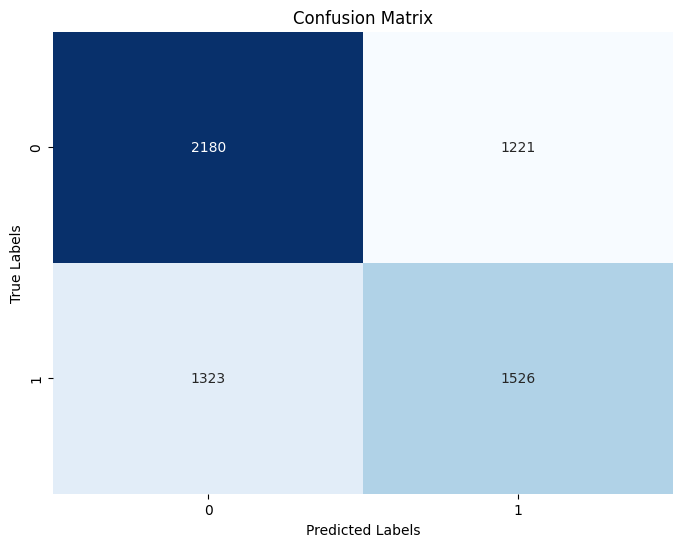


Figure 20: Heatmap for Neural Networks Model

# RESULTS

The Linear Regression model, despite being a regression algorithm, was also evaluated on this classification problem. It achieved an accuracy of around 0.60 on the test set, indicating that it made correct predictions for about 60% of the instances. The model's AUC of 0.63 demonstrated its capability to discriminate between the classes, although it's important to note that AUC values are often interpreted in the context of binary classifiers.

The Support Vector Machine (SVM) displayed a test set accuracy of approximately 0.60, suggesting that the model correctly predicted the class labels for about 60% of the instances. The Area Under the ROC Curve (AUC) score of 0.66 indicated that the SVM had a reasonably good ability to distinguish between the two classes. However, its performance was somewhat constrained by a recall of 0.29, indicating that it struggled to identify instances of the positive class.

Moving on to the Gradient Boosting model, it showcased notable performance across various metrics. On the training set, the model achieved a precision of 0.70, indicating that when it predicted a positive class instance, it was correct about 70% of the time. The high recall value of 0.80 suggested that the model effectively captured a significant portion of the actual positive class instances. The F1-score, which considers both precision and recall, stood at 0.75, reflecting a balance between the two. The overall accuracy on the training set was 0.71, indicating the proportion of correctly predicted instances. The model maintained a similar trend of performance on the testing set, with a precision of 0.64, recall of 0.72, F1-score of 0.68, and accuracy of 0.62. The AUC score of 0.66 further underscored its ability to discriminate between the classes.

The Gaussian Naive Bayes model, a probabilistic classifier, demonstrated a moderate level of performance. On the training set, it achieved a precision of 0.62, signifying its accuracy in predicting the positive class. A recall of 0.72 indicated that it was effective in identifying a substantial portion of actual positive class instances. The corresponding F1-score was 0.67, striking a balance between precision and recall. The model's accuracy on the training set was 0.61, indicating its overall correctness. On the testing set, it maintained its effectiveness with a precision of 0.63, a recall of 0.73, an F1-score of 0.67, and an accuracy of 0.62. The AUC value of 0.64 further supported its performance.

The K-Nearest Neighbors (KNN) model, known for its simplicity, showcased strong precision and recall on the training set, with a precision of 0.72 and a recall of 0.78. These metrics indicated that the KNN model was adept at accurately classifying instances of the positive class and identifying a high proportion of actual positive class instances. However, its performance experienced a decline on the testing set, with a precision of 0.59 and a recall of 0.64. The AUC value of 0.58 indicated its ability to discriminate between the two classes, albeit with some limitations.

Lastly, the Neural Network model, a powerful and flexible algorithm, exhibited competitive performance on the training set. With a precision of 0.75 and a recall of 0.77, it demonstrated its capability to accurately classify positive class instances while identifying a significant portion of actual positives. The corresponding F1-score and accuracy were 0.76 and 0.73, respectively. On the testing set, its precision of 0.62, recall of 0.64, F1-score of 0.63, and accuracy of 0.59 reflected a slight decrease in performance. The AUC value of 0.63 showcased its ability to discriminate between the classes.

In conclusion, the Gradient Boosting model emerged as the best-performing model, demonstrating a balanced combination of precision, recall, and AUC on both training and testing sets. The Linear Regression model also exhibited competitive performance, particularly in terms of accuracy and AUC. These models displayed their respective strengths in addressing the classification task within the given dataset, offering valuable insights for model selection and deployment.

*Table: Comparing all models*

| **Model** | **Accuracy** | **Precision (Class 1)** | **Recall (Class 1)** | **F1-Score (Class 1)** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| Linear Regression | 0.60 | 0.57 | 0.50 | 0.53 | 0.63 |
| Support Vector Machine | 0.60 | 0.67 | 0.29 | 0.40 | 0.66 |
| Gradient Boosting | 0.71 | 0.72 | 0.60 | 0.65 | 0.66 |
| XGBoost | 0.62 | 0.60 | 0.51 | 0.54 | 0.66 |
| Gaussian Naive Bayes | 0.62 | 0.60 | 0.48 | 0.53 | 0.64 |
| K-Nearest Neighbors | 0.56 | 0.52 | 0.46 | 0.49 | 0.58 |
| Neural Network | 0.59 | 0.56 | 0.54 | 0.55 | 0.63 |

# CONCLUSION

In conclusion, this study sought to determine the effectiveness of various machine learning models in predicting the likelihood of patient readmission based on a comprehensive analysis of a diverse range of variables encompassing demographics, hospital stay characteristics, clinical indicators, medical coding details, treatment records, and medication information. The investigation revealed both promising outcomes and certain limitations that provide insights into the predictive capabilities of the models and the applicability of the dataset.

The results demonstrated that the Gradient Boosting model exhibited the highest predictive accuracy among the models evaluated, achieving balanced precision and recall scores on both training and testing datasets. This model successfully harnessed the amalgamation of extensive variables to forecast patient readmissions. Additionally, the Neural Network model showcased competitive performance, further emphasizing the potential of deep learning techniques for predictive tasks in healthcare.

However, the study also encountered limitations that warrant consideration. While the dataset was substantial, comprising diverse variables, the absence of certain demographic and hospital-specific features might have influenced the models' predictive power. A more extensive inclusion of hospital-related variables could potentially enhance the predictive accuracy, as the study mainly focused on a disease-oriented dataset. Moreover, the dataset's variability across different hospital settings could have introduced noise into the predictive models, affecting their reliability. Additionally, the current dataset was already acquired encoded and deidentified. If more raw data is available to the researchers, there might be greater possibility to run a more comprehensive assessment and determine which factors actually contribute towards readmissions.

In essence, the study provides valuable insights into the feasibility of employing machine learning techniques for predicting patient readmissions based on a comprehensive range of factors. The findings underscore the significance of considering a broader spectrum of hospital-related variables to further bolster the models' predictive capabilities. Despite the limitations, the superior performance of the Gradient Boosting model emphasizes its potential as a valuable tool for healthcare providers to identify patients at risk of readmission, enabling more proactive and tailored interventions to improve patient outcomes and resource allocation.

# 

# 

# 

# 

# 

# 

# 

# REFERENCES

1. Cox, M., Panagides, J. C., Tabari, A., Kalva, S., Kalpathy-Cramer, J., & Daye, D. (2022). Risk stratification with explainable machine learning for 30-day procedure-related mortality and 30-day unplanned readmission in patients with peripheral arterial disease. *PLoS ONE*, *17*(11), 1–18.
2. *KWETCZER, L. A.,*(2020). *Hospital Readmission Risk Prediction Using Machine Learning.*
3. Lin, Y.-W., Zhou, Y., Faghri, F., Shawl, M. J., & Campbell, R. H. (2019). Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long shortterm memory. *PLOS ONE*, *14*(7), e0218942.
4. *Matheny, ME, Ricket, I, Goodrich, CA, Shah, RU, Stabler, ME, Perkins, AM, Dorn, C, Denton, J, Bray, BE, Gouripeddi, R, Higgins, J, Chapman, WW, MacKenzie, TA,* *Brown, JR*. (2021).  *Development of Electronic Health Record-Based Prediction Models for 30-Day Readmission Risk Among Patients Hospitalized for Acute Myocardial Infarction*.
5. Riester, M. R., McAuliffe, L., Collins, C., & Zullo, A. R. (2021). Development and validation of the Tool for Pharmacists to Predict 30-day hospital readmission in patients with Heart Failure (ToPP-HF). *American Journal of Health-System Pharmacy*, *78*(18), 1691–1700.
6. Saha, P., Sircar, R., & Bose, A. (2021). Using hospital Admission, Discharge & Transfer (ADT) data for predicting readmissions. *Machine Learning with Applications*.
7. Welvaars, K., van den Bekerom, M. P. J., Doornberg, J. N., van Haarst, E. P., OLVG Urology Consortium, van der Zee, J. A., van Andel, G. A., Lagerveld, B. W., Hovius, M. C., Kauer, P. C., & Boevé, L. M. S. (2023). Evaluating machine learning algorithms to Predict 30-day Unplanned REadmission (PURE) in Urology patients. *BMC Medical Informatics & Decision Making*, *23*(1), 1–13.
8. Yujuan Shang, Kui Jiang, Lei Wang, Zheqing Zhang, Siwei Zhou, Yun Liu, Jiancheng Dong, & Huiqun Wu. (2021). The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifiers. *BMC Medical Informatics and Decision Making*, *21*(S2), 1–11.