Major Project Report on

PREDICTION OF PATIENT READMISSION IN HOSPITAL

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Sarthaki Gajghate (201IT155)

Lucia Americo Tembe (201IT169)

Geethanjali Manoj Kumar (201IT220)

under the guidance of

Dr. Sowmya Kamath S

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April 2024

DECLARATION

We hereby *declare* that the Major Project Work Report entitled ***"Prediction Of Patient Readmission In Hospital"*** , which is being submitted to the **National Institute of Technology Karnataka, Surathkal**, for the award of the Degree of Bachelor of Technology in Information Technology, is a *bonafide report of the work carried out by me/us.* The material contained in this Major Project Report has not been submitted to any University or Institution for the award of any degree.

Name of the Student (Registration Number) with Signature

(1) Sarthaki Gajghate (201IT155)

(2) Lucia Americo Tembe (201IT169)

(3) Geethanjali Manoj Kumar (201IT220)

Department of Information Technology
Place : NITK, Surathkal
Date :

CERTIFICATE

This is to *certify* that the Major Project Work Report entitled ***"Prediction Of Patient Readmission In Hospital"*** submitted by

Name of the Student (Registration Number)

(1) Sarthaki Gajghate (201IT155)

(2) Lucia Americo Tembe (201IT169)

(3) Geethanjali Manoj Kumar (201IT220)

as the record of the work carried out by them, is *accepted as the B.Tech. Major Project work report submission* in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Information Technology in the Department of Information Technology, NITK Surathkal.

(Dr. Sowmya Kamath S)
Associate Professor
Department of Information Technology
NITK Surathkal

Chairman, DUGC

ACKNOWLEDGEMENT

We would like to thank Dr. Sowmya Kamath S. and the IT Department of NITK, for allowing us to conduct this research project. We would like to extend our gratitude to Dr. Sowmya Kamath S. again for her continued support and guidance in completing our Major Project. We would further thank our college and all faculty for helping us meet our research and academic requirements that facilitated us to complete Major Project.

ABSTRACT

The measurement of hospital readmission rates is essential, as hospitals with high readmission rates are often associated with unsatisfactory levels of patient care effectiveness. This metric has become an increasingly important quality benchmark for healthcare systems. The concept of hospital readmission originated in 1953 in the work of Moya Woodside, who analyzed hospital readmission in psychiatric contexts. Therefore, we propose to address the important topic of "Predictive Analysis of Patient Readmission in Hospitals." This topic is of great significance due to its substantial responsibility for improving hospital care and its potential positive impact on people's lives. To tackle this complex issue, we plan to employ Machine Learning and Deep Learning models, considering that this combination can lead to better predictive performance and contribute to enhancing hospital management and patient care.

This project has implemented various forms of feature extraction, encoders, and also compares various Machine Learning and Deep Learning models for predicting hospital readmission. This paper uses machine learning models such as XGBoost, and random forest, and deep learning models such as NN, CNN, MLP, and LSTM. The best performance recorded is by MLP using one-hot which gives an accuracy score of 89%. Following MLP, utilizing label-encoder the second best models are LSTM and NN with an accuracy score of 88%.

Keywords— Machine, Deep Learning, Healthcare, patient readmission, predictive analysis

CONTENTS

LIST OF FIGURES **iv**

LIST OF TABLES **v**

1 INTRODUCTION **vi**

1.1 Overview vi

1.1.1 Challenges 1

1.2 Motivation 1

2 LITERATURE REVIEW **2**

2.1 Background and Related Works 2

2.2 Outocome of Literature Review 3

2.3 Problem Statement 5

2.4 Objectives of the Project 5

3 PROPOSED METHODOLOGY **6**

3.1 Understanding The Dataset 6

3.1.1 Outlier Analysis 8

3.1.2 Covariance 11

3.1.3 Correlation 12

3.2 Data preprocessing 14

3.2.1 Imputation Methods 15

3.3 Feature Engineering 17

3.4 Encoding 18

3.4.1 One- hot Enocoding 18

3.4.2 Label Encoder 19

3.5 Feature Selection 19

3.6 Treating Imbalanced Data 20

3.7 Methods 23

3.7.1 Random Forest 24

3.7.2 XGboost 25

3.7.3 MLP 25

3.7.4 NN 25

3.7.5 CNN 26
3.7.6 LSTM 26

4 RESULTS AND ANALYSIS **28**
4.1 Result of Analysis of Dataset 28
4.2 Performance Analysis of Imputation Methods 29
4.3 Performance of Models 30
4.4 Ablation Study 32
4.4.1 Ablating Imputation Module 32
4.4.2 Ablating Encoding Module 34
4.4.3 Ablating Feature Selection Module 35

5 CONCLUSIONS AND FUTURE WORK **38**

REFERENCES **39**

LIST OF FIGURES

3.0.1 Methodology 6
3.1.1 Gender Distribution 7
3.1.2 Race Distribution 8
3.1.3 Age vs. Number of Prescribed Medicines 9
3.1.4 Gender and Age vs. Number of Prescribed Medicines 10
3.1.5 Correlation Heat Map 10
3.1.6 Box-plot for the outliers detected 11
3.1.7 covariance analysis 12
3.1.8 Correlation analysis 13
3.4.1 Chi-Square Correlation 20
3.6.1 Distribution of Class 23

LIST OF TABLES

2.2.1 Review of Previous Studies on Hospital Readmission Prediction . . . 4

3.5.1 Best Features Selected using Boruta 21
3.5.2 Features of PC1, selected component of PCA 22

4.2.1 F1-Scores 29
4.3.1 Performance Analysis of Models 31
4.4.1 Models Without Imputation 33
4.4.2 Model Accuracy without Encoding 35
4.4.3 Models without Feature Selection 36

CHAPTER 1

INTRODUCTION

1.1 Overview

Readmission rates within the first 30 days after discharge in the mid-1980s were recorded as greater than 20 percent (Anderson and Steinberg 1984[1]), and the trends have not changed in the current times (Jencks, Williams, and Coleman,2009 [2]).In the present times, two-thirds of patients were readmitted regardless of reporting good discharge experiences. Out of the discharged patients, one-third of patients had a post-discharge doctor appointment and half of them were readmitted before the said appointment. This goes to show that the post-discharge experiences could be improved and that the continual readmission rates negatively impact cost and patient outcomes. It is important to look into the unchanged readmission rates and find a solution to the problem that will effectively revolutionize the quality of hospital care and intervention.

Predictive analysis of hospital readmission is a field of research and application that seeks to predict whether a hospitalized patient is highly likely to be readmitted in the future after being discharged from the hospital. This area holds great importance as it can contribute significantly to the improvement of healthcare quality by identifying patients at high risk of readmission, enabling doctors and healthcare professionals to provide more effective and personalized care. So, there are several notable challenges in this process. One of the primary challenges lies in the complexity of healthcare data, which is often heterogeneous, encompassing clinical, demographic, and medical history information. Effectively integrating and processing this data is a significant challenge. also, the adoption of complex AI models [3] such as neural networks is hindered by the difficulty of interpretation by doctors and healthcare professionals, which can impact the acceptance of these technologies. Technologies such as cost-sensitive deep learning [4] have been utilised currently for predicting hospital readmissions.

1.1.1 Challenges

1. Large dataset and time constraints: takes a long time to run different models and modules of the proposed architecture.

2. Large amounts of missing data: Had to employ various imputation methods to ensure that the dataset was in a usable state.

3. Great class imbalance: as this is a two class prediction model study it is vital that there is no class imbalance, but there was a huge class imbalance that had to mitigate by utilising oversampling and undersampling methods.

4. Lack of deep learning studies on the given dataset: there was a lack of references that utilizes deep learning models to predict the readmission of diabetic patients.

5. Debugging and error corrections.

1.2 Motivation

Working on these topics helps improve clinical outcomes and identify patients at risk of readmission allowing for early, personalized interventions, which can lead to better health outcomes and faster recovery rates. Allocating resources more efficiently, minimizing unnecessary readmissions, and thus reducing healthcare costs for individuals and society as a whole as a consequence will be Improvement in the Quality of Care, Patient-Centered Care The theme promotes patient-centered care, creating a healthcare system that takes individual needs and preferences into account.

CHAPTER 2

LITERATURE REVIEW

2.1 Background and Related Works

The readmission of patients has been an area of interest and has gained significant attraction in the past years leading to the development of various ideologies aimed at facilitating these processes.

Shang et al. [5] predicted the 30-days hospital readmission risk in diabetic patients using predictive modeling with machine learning classifers. The authors use three machine learning models mainly Random Forest, Naive bayes and Tree Ensembler for the prediction of the readmission of the patients. To minimize the impact of an unbalanced dataset, down- and over-sampling methods were adopted to balance the data. To minimize the impact of an unbalanced dataset, down- and over-sampling methods were adopted to balance the data.The dataset was used to train and verify the model through an 80 percent training set and a 20 percent test set. Finally, the size of the dataset was reduced to 100244 patients. The three models were compared based using AUC. It was found that Random forest performed the best. The AUC was 67 percent compared to existing models (0.649, 0.566, 0.577).

In the study, Predicting Diabetic Readmission Rates Moving Beyond HbA1c [6], the author suggests a method to build different models for the age groups (0-30), (30-70) and (70-99). An ensemble model comprising of extreme gradient-boosted trees, gradient-boosted greedy trees and extra trees classifiers designed for age group [0-30) achieved an accuracy of 84.8 percent. For the age group [30-70), an ensemble model containing random forest using Gini function, gradient and extreme gradient boosted trees with early stopping achieved an accuracy of 78.5 percent. For the age group [70-99), an ensemble model containing extra trees classifiers and extreme gradient boosted trees with early stopping achieved an average accuracy of 68.5percent. The average accuracy across three models comes out to be 77.2%.

In this research [7], Recurrent Neural Networks are used to predict whether a patient would be readmitted in the hospital and compared its accuracy with basic classifiers such as SVM, Random Forest and Simple Neural Networks. RNN showed

the highest result in all the models used. The dataset was divided 75-25 for training and validation. To deal with the missing values, Approximate Bayesian Bootstrap was used. RNN had an accuracy score of 81.12 percent and an ROC of 80 percent.

In a study [8] conducted in 2022 of multiple machine learning models for prediction of hospital readmission from acute care facilities in 2015 and 2016 in Alberta, utilizes a combination of computer-generated and manually crafted features in a machine learning model led to enhanced prediction accuracy compared to both the LACE model and a model reliant solely on manually-derived features. This refined model proves valuable in identifying patients at high risk of readmission, paving the way for targeted interventions that have the potential to mitigate such occurrences.

Another study [9] on identifying diabetic patients with high risk of readmission finds number of inpatient visits, discharge disposition, and admission type as strong indicators of readmission. Additionally, it was discovered that the combination of laboratory tests and discharge arrangements collectively determine whether a patient will experience readmission shortly after leaving the hospital (i.e., within 30 days) or after an extended period (i.e., more than 30 days).

In a paper published in 2015 [10] the researchers developed a predictive model for hospital readmissions, which allows for the management of the balance between transparency in reasoning and predictive accuracy. This was achieved by considering the distinct attributes of the database under study. They employed an ensemble approach, combining a boosted C5.0 tree as the primary classifier with a support vector machine (SVM) serving as a secondary classifier.

2.2 Outocome of Literature Review

In the 2.2.1, we summarize the studies and quantitative results of the studies performed previously. It is observed that RNN performed the best with an ROC score of 80 percent.

Table 2.2.1: Review of Previous Studies on Hospital Readmission Prediction

Author	**Methodology**	**Remarks**
Lei Wang, Zheqing Zhang, Siwei Zhou, Yun Liu, Jiancheng Dong, Yujuan Shang, Kui Jiang, and Huiqun Wu.	Compared Random Forest, Naive Bayes, and Decision Tree incorporated with Ensemble Tree. The data was downsampled.	Down Sampling of the dataset was done using the "Equal Size Sampling" node in the Konstanz Information Miner (KNIME).
Damian Mingle	Ensemble model using Extreme Gradient Boosted Trees with early stopping, SVM, RandomForest, and Extra Trees classifier.	Built three different ensemble machine learning models for three age groups: 0-30, 30-70, and 70-100.
Chahes Chopra, Shivam Sinha, Shubham Jaroli, Anupam Shukla, and Saumil Maheshwar	Recurrent Neural Network and comparing it with machine learning models like Random Forest, Naive Bayes, and Tree Ensembler.	No class balancing and RNN showed the best results
Sacha Davis, Jin Zhang, Ilbin Lee, Mostafa Rezaei, Russell Greiner, Finlay A McAlister, and Raj Padwal	Compare LACE with gradient boosting machine that utilizes both manually-engineered and machine-learned features.	Features like frequent hospital visits, chronic conditions, and older age were identified as significant predictors of readmission.
Malladihalli S Bhuvan, Ankit Kumar, Adil Zafar, and Vinith Kishore	Used Naive Bayes, Bayesian Networks, and Random Forest algorithms, conducting Associative Rule Mining (ARM), and cost analysis.	The data is preprocessed to includer diabetic patients only above 65 years of age.
Lior Turgeman, Jerrold May, Johnson Moore, and Youxu Tjader	A boosted C5.0 tree, as the base classifier, was ensembled with a support vector machine (SVM), as a secondary classifier.	The total accuracy for the ensemble ranges from 81% to 85%.

2.3 Problem Statement

To develop a predictive model combining the traditional machine learning algorithms and advanced Deep Learning techniques to predict the likelihood of readmission of patients within 30 days.

2.4 Objectives of the Project

(1) Conduct data analysis, like feature importance assessment, and appropriate visualization techniques.

(2) Perform correlation analysis to establish relationships that might indicate the likelihood of patient readmission and also perform co variance relation analysis which can help understand the distribution and central tendencies of various features in the dataset.

(3) Implement different imputation methods to handle the missing values in the data and compare them to find the best method among these models.

(4) Implement Machine Learning and Deep Learning models and compare their performance to identify and validate the most effective model for the given dataset.

CHAPTER 3

PROPOSED METHODOLOGY

The project follows a careful and methodical approach in creating and comparing various machine learning and deep learning models as shown in Fig. 3.0.1. We also utilize various methods of imputation before the deployment of the models.

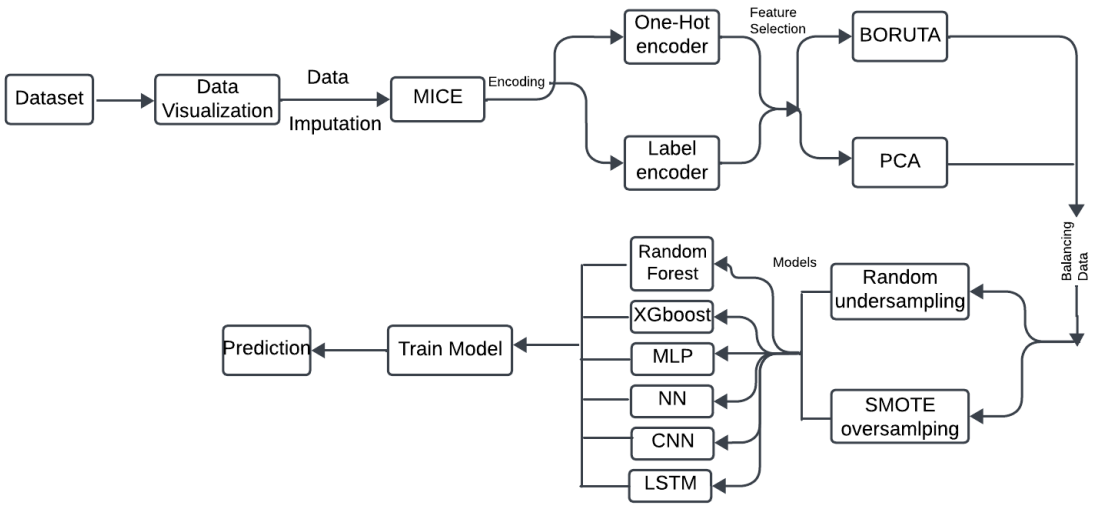

Figure 3.0.1: Methodology

3.1 Understanding The Dataset

The dataset used in this report is "Diabetes 130-US for years 1999-2008" [11]. The data set contains 50 attributes detailing the age, race, gender, number of hospitalizations, types of medicines, number of medicine changes, medical department, number of outpatient and inpatient, number of emergencies, number of procedures, etc. There are 101,766 unique patient entries. Features of the dataset:

- all data points correspond to patients who have been hospitalized at least once through the course of their treatment.
- it is a diabetic encounter, i.e. any kind of diabetes was entered into the system.

- length of hospital stay ranged from at least one day to fourteen days.
- laboratory tests and procedures were done on the patient.
- medication was administered to the patient during the hospital encounter.

The dataset was visualized using various graphs and the following are the findings. Figure 3.1.1 shows the gender distribution in the dataset, there are 53.9 percent females and 46.1 percent males, which is an almost fifty-fifty divide between the two genders and shows no gender bias.

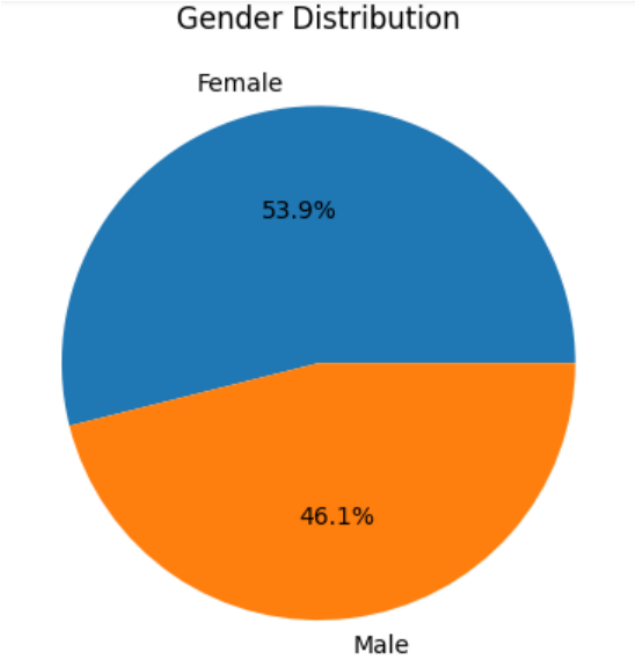

Figure 3.1.1: Gender Distribution

On the other hand, in Figure 3.1.2. there is an evidently large racial bias in the dataset as the majority of the dataset caters to Caucasian people, mostly due to the fact it is a US-based dataset.

Further, as per Figure 3.1.3 and Figure 3.1.4., we can infer that the number of prescribed medicines increases with age, with the highest number of medications prescribed to individuals in the 80-90 age group. On average it can be observed that 70-80 is the most common age in the dataset, and additionally that women are more likely to be prescribed more number of medicines or be subjected to a higher number

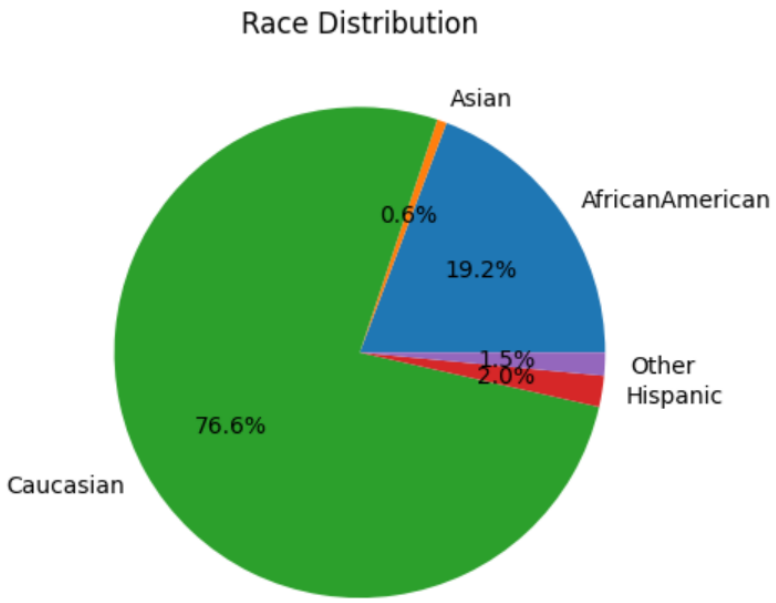

Figure 3.1.2: Race Distribution

of medicine changes. We can also infer from the dataset that the highest number of readmissions can be observed in the 70-79 age group for both women and men. It can also be noted that medication change for diabetics upon admission has been shown by previous research to be associated with lower readmission rates.

The variables are represented on the x and y axis, and the correlation between them is represented by the color of the square. The darker the color, the stronger the correlation. In this case, the variables are different types of patient data, such as encounter ID, patient number, admission type, discharge disposition, etc. The strongest correlations are between number of diagnoses and number of procedures, number of diagnoses and time in hospital, and number of procedures and time in hospital. The correlation heat map in Figure 3.1.5. suggests that patients who have more diagnoses and procedures tend to spend more time in the hospital.

3.1.1 Outlier Analysis

One of the most important tasks during the handling of such a large dataset is to deal with the outliers as they can significantly change the outcome of the data. The

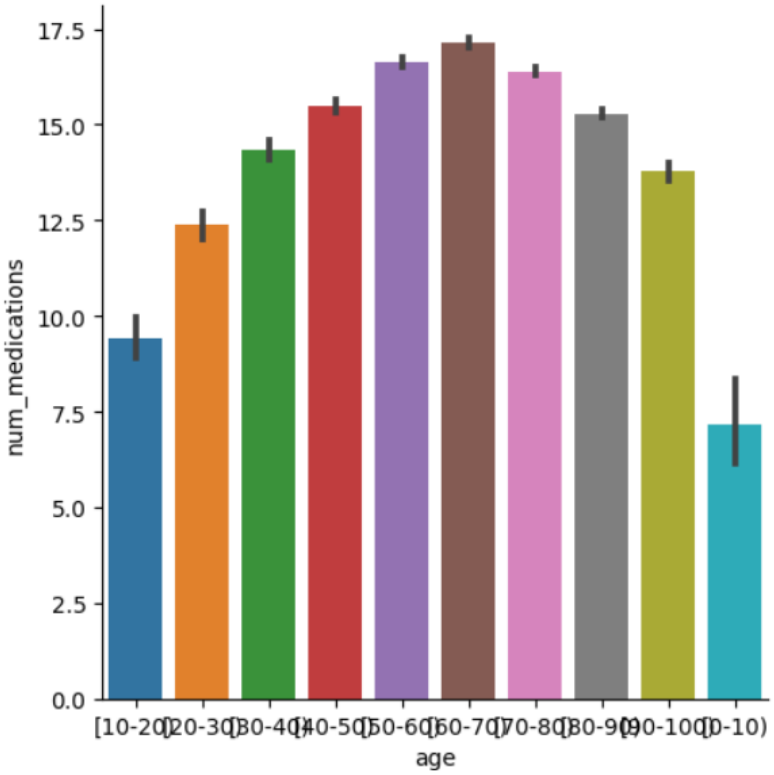

Figure 3.1.3: Age vs. Number of Prescribed Medicines

method used by us is F-Scores. Calculating the Z-score provides a way to understand how far the data points are from the mean. Below is the formula 3.1.

$$Z = \frac{X - \mu}{\sigma} \tag{3.1}$$

These outliers are detected on different variables like time in hospital, number of lab procedures, number of procedures, number of medications, number of outpatients, number of patients in emergency, number of inpatient. Figure 3.2.1 depicts the boxplots of the outliers detected.

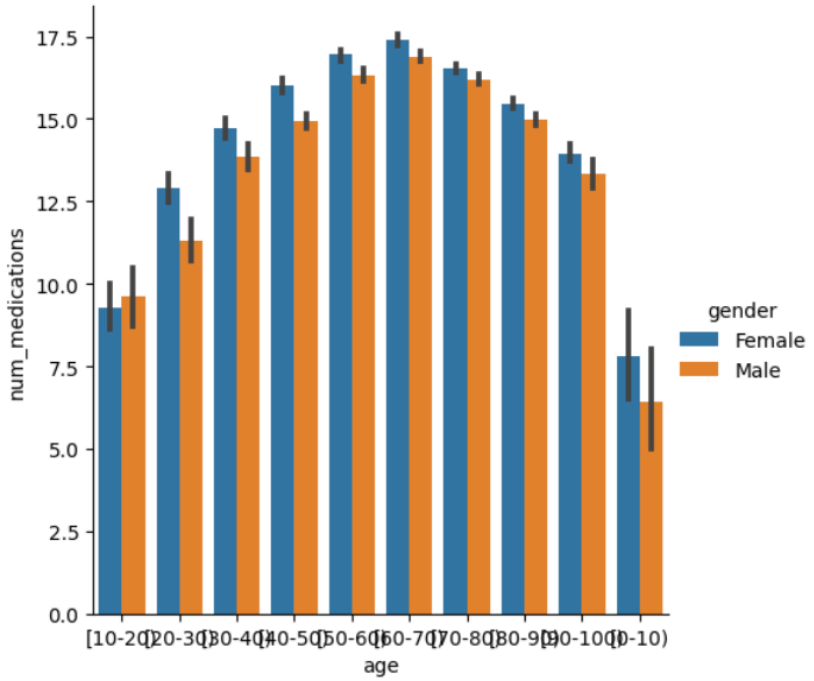

Figure 3.1.4: Gender and Age vs. Number of Prescribed Medicines

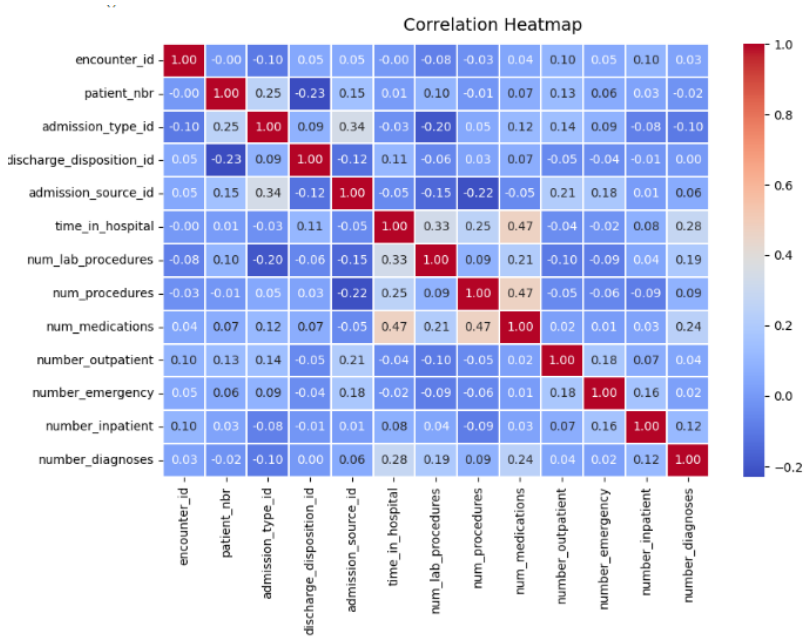

Figure 3.1.5: Correlation Heat Map

Figure 3.2.1 depicts the boxplots of the outliers detected.

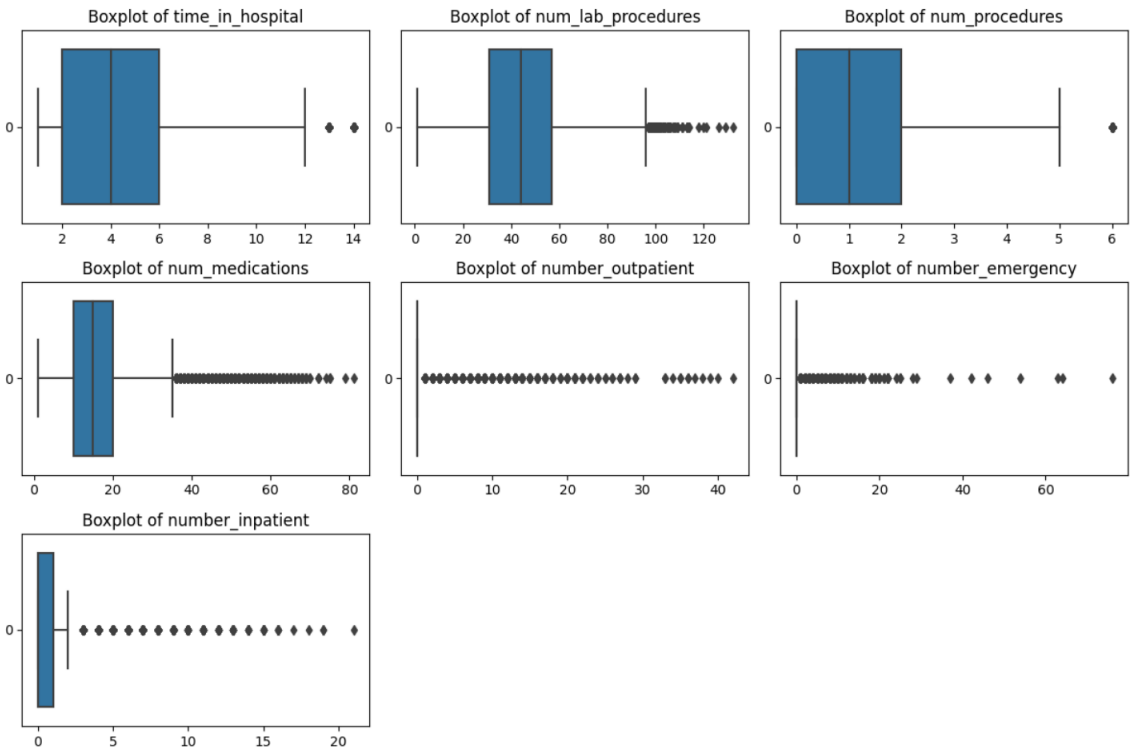

Figure 3.1.6: Box-plot for the outliers detected

3.1.2 Covariance

Covariance is a fundamental concept in statistical analysis, providing invaluable insight into the relationships between variables. Its importance is particularly apparent in a variety of fields, including data analysis, where it facilitates the assessment of associations between features. We use the covariance formula 3.2 to grasp.

$$\mathrm{Cov}(X, Y) = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \tag{3.2}$$

The covariance matrix illustrates the relationships between variables, It is evident that:

1. encounter ID and patient number, suggesting a strong linear relationship. This may indicate that certain patients have a consistent pattern of encounters.

2. number of diagnoses and the number of medications, This implies that patients with a higher number of diagnoses may also require a greater number of medications.

3. number of emergency visits and the number of inpatient admissions is positive. This suggests that patients with more emergency visits may also have a higher likelihood of subsequent inpatient admissions.

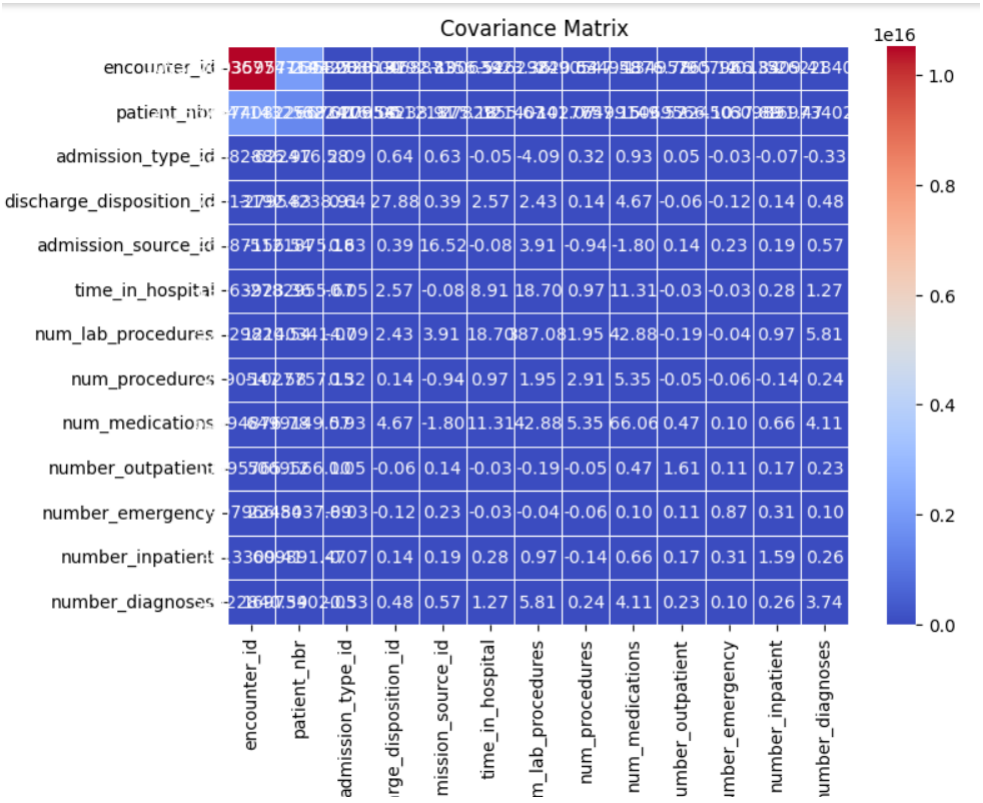

Figure 3.1.7: covariance analysis

3.1.3 Correlation

Correlations are useful for describing simple relationships among data. The degree of correlation between two or more variables can be determined using correlation. However, it does not consider the cause-and-effect relationship between variables. The correlation is determined using Pearson's formula 3.3.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{3.3}$$

Figure 4.2.1 depicts the correlation between the numerical columns of the data. It is observed that:

1. time in hospsital is highly correlated with discharge disposition id
2. encounter id and patient id are merely a unique identity so we do not care about that.
3. admission source id has some correlation with admission type id
4. num lab procedures has good correlation with num medication.
5. num procedures is highly correlated with num medication
6. number emergency is a little bit correlated with number inpatient

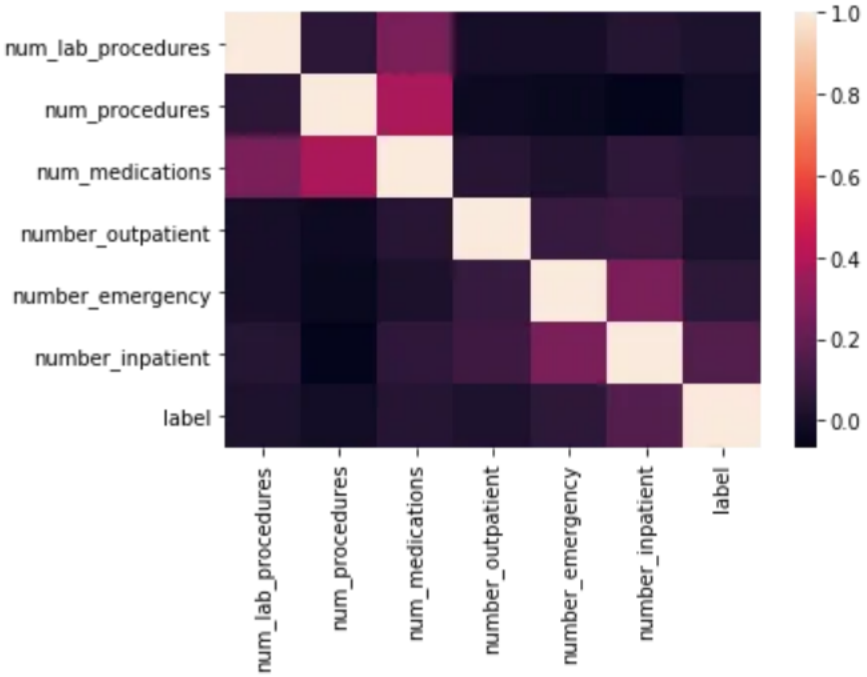

Figure 3.1.8: Correlation analysis

3.2 Data preprocessing

During the analysis of the data, it was found that there are several problems with the data like, redundancy, missing values and certain non-relevant attributes. We preprocess the dataset to deal with these problems and make the data reproducible. The following steps were taken:

(1) The redundant data like the *patient number* and *encounter id* were dropped. Also the columns having more than 45% of missing data were dropped.

(2) The prediction problem can be considered as a classification problem, in the dataset the output classes are of mainly three types: readmitted in less than 30 days, readmitted after 30 days, not readmitted. For ease of implementation, this paper has condensed it into two classes: readmitted and not readmitted.

(3) Additionally, in the IDS mapping, there are 848 three-digit codes mapped to various types of diagnosis, we have condensed that into nine broad categories, namely: circulatory, respiratory, digestive, diabetes, injury, musculoskeletal, genitourinary, neoplasm, and other. It can be observed in the dataset there are features diag1, diag2, and diag3 which contain codes for each time the patient was diagnosed, these three codes have been mapped to the nine diagnosis categories.

(4) There are 23 columns of various medications administered to the patients which is collapsed into a medicaments column where the name of the medicine is listed for each of the instances.

(5) The change of medication column has four categorical values: up, down, steady, and no corresponding to increasing, decreasing, steady, and no change of dosage of medicines. The feature has been changed to a binary yes or no for a change in medication or not.

(6) Age is given in ordinal values and it has been converted to nominal values such as: [0-10] as 0, [1-10] as 1, and finally [90-100] as 9.

(7) Implement different imputation methods to handle the missing values in the data and compare them to find the best method among these models.

(8) We also perform encoding methods such as One-hot encoding and Label-encoding to make the nominal attributes into numerical one.

Also, after performing the outlier analysis, it was found that no extreme outliers were found that needed to be removed.

3.2.1 Imputation Methods

Imputation methods are used to deal with the missing data in the dataset. These methods are used to fill the missing data to create a complete data matrix that can be analyzed using standard methods. We try three methods: Mean and Mode, KNN and MICE.

Empirical Methods: Mean and Mode

Mean and mode imputation are simple methods for dealing with missing data in a dataset. They are significant due to their simplicity and ease of use. Mean imputation replaces missing values with the mean of observed values, preserving a variable's overall average. This is beneficial for continuous data. Mode imputation, on the other hand, is appropriate for categorical data or situations where the most common value is obvious. Both methods are quick fixes that work well when missing data is dispersed randomly. While they may not capture all nuances, their simplicity makes them useful for quick initial data cleanup, ensuring that key patterns are preserved before applying more sophisticated analyses.

KNN

K-Nearest Neighbors (KNN) stands out as a significant technique for dealing with missing data, and its significance is emphasized by several key characteristics. It is easy to use and understand. One of its distinguishing features is its non-parametric approach, which avoids making assumptions about the underlying data distribution. This adaptability allows KNN to be applied to

a wide range of datasets, adding to its versatility. The absence of a formal training phase distinguishes KNN. This property not only improves its adaptability to changes in dynamic datasets, but it also simplifies its application in scenarios where data is constantly changing. Furthermore, KNN's ability to impute missing values by taking into account the values of neighboring observations is a useful asset in data preprocessing. KNN's user-friendly interface, non-parametric nature, adaptability to changes, and ability to imput missing values all contribute to its importance as a dependable technique for handling missing data in a variety of analytical contexts.

MICE

Multivariate Imputation by Chained Equations (MICE) is an important technique in handling missing data because it addresses the issue of missing data in a comprehensive and systematic way. It works by iteratively imputing missing values using regression models based on the observed data. The algorithm for MICE algorithm is given below:

- The dataset with the missing value is taken.
- It fills in missing values iteratively, considering all columns as features and using regression techniques to predict missing values based on the other columns. The regressor used here is Bayesian Ridge Regressor.
- the number of iterations are set to 10.
- It repeats this process for a specified number (10 in this case) of iterations wherein the imputed values stabilize.

Comparison of The Various Imputation Methods

The process of comparing imputation methods is critical in ensuring the selection of an appropriate technique for dealing with missing data in a specific context. This assists data analysts and researchers in making well-informed decisions about which imputation strategy to use, carefully balancing accuracy and computational considerations. Our overarching goal was to determine which

imputation method performs best in terms of accuracy, reliability, and suitability for the data at hand as well as the intended analysis. This comparative assessment ensures that the chosen approach is in sync with the complexities of the dataset and the specific requirements of the analysis, contributing to robust and reliable data imputation practices.

Out of the three imputation methods we have discovered that MICE is the best imputation method for our dataset and project objectives. The reasons why MICE was selected over the other imputation methods are:

(1) MICE can handle a mix of continuous, categorical, and binary variables, whereas KNN is primarily suitable for continuous variables. Empirical methods might not handle different data types gracefully.

(2) MICE imputation uses chained equations to impute missing values, taking into account the relationships between variables. This can preserve the underlying structure and correlations in the data, which is especially beneficial in complex datasets. KNN imputation, while considering neighboring data points, might not capture these relationships as effectively.

(3) MICE generally produces more accurate imputations compared to KNN, particularly when dealing with high-dimensional data or datasets with complex patterns. KNN's performance can degrade in high-dimensional spaces due to the curse of dimensionality.

(4) MICE can handle various missing data patterns, including missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) scenarios, by specifying appropriate imputation models for each variable. KNN might not handle missing data mechanisms as effectively.

3.3 Feature Engineering

Feature engineering is a vital part of preprocessing utilized for reducing dimensionality, handling missing data, and such. In this paper, we have condensed

certain features down to singular features that are a comprehensive amalgamation of the listed features. This helps reduce dimensionality and also makes model implementation easier in the later part of this paper. A new feature named visits-column was created by combining the following features: number-emergency, number-outpatient, number-inpatient. Subsequently, two columns medicaments-column and diagnoses-column were created. Medicaments-column is a compilation of all the 23 individual medicine columns (medicine columns relate to the various medications administered), and the diagnoses-column is a combination of diag1, diag2, and diag3.

3.4 Encoding

Machine learning models require numerical inputs. Categorical variables, such as gender, race, or medication type, are represented as strings or labels, which algorithms cannot directly process. Hence, encoding the data becomes necessary. We explore two encoding methods as the data has different types of data types in it. E.g. ordinal, nominal, numerical, categorical, etc. One-hot encoding is useful for algorithms that do not assume any ordinal relationship between the categories, such as decision trees, random forests, and neural networks. Each category is represented as one-hot vector. On the other hand, label encoding assigns a unique integer to each category. The integers are typically assigned in increasing order based on the alphabetical order of the categories or their frequency.

During the encoding process, we utilized two distinct encoding techniques: one-hot encoding and label encoding.

3.4.1 One- hot Enocoding

As the data contains both, numerical and categorical values, it is better to convert these categorical values into numerical values for better results for the machine learning models. One such method that can be used in such a case is one-hot encoding. This allows giving a numerical input to machine learning

models without creating any biases. It can also improve the performance of the model by providing additional information about the categorical variabe according to its importance.

3.4.2 Label Encoder

In the data pre-processing phase, we applied Label Encoder in detail to encode categorical variables, transforming them into a suitable numeric format. This involved identifying categorical features in the dataset and systematically applying the Label Encoder to each of them. For ordinal variables, Label Encoder assigned labels based on their inherent order, while for nominal variables, arbitrary numeric labels were assigned. This process ensures that the algorithm can interpret and analyze categorical data effectively.

In the exploratory data analysis phase, we strategically employ feature selection techniques to increase the predictive power of our machine learning models. Specifically, we leverage the Chi-Square test to assess the statistical significance of each feature in relation to the target variable, also we conduct a thorough analysis of feature correlations using visualizations such as correlation 3.4.1. This allowed us to gain insights into the relationships between different features, identifying potential multicollinearity and guiding our decisions about feature inclusion or exclusion. By combining the statistical rigor of the Chi-Square test with the visual interpretation of correlation plots, we were assured of a well-informed, data-driven approach to feature selection.

3.5 Feature Selection

Our methodology includes data encoding with Label and One-Hot Encoders, followed by feature selection using Principal Component Analysis (PCA) for the Label Encoded dataset and Boruta for the One-Hot Encoded dataset. Specifically, PCA produced 2 principal components with the variance of [0.05713081, 0.04805385] respectively. Since, the variance of PC1 is more than that of PC2, we selected the features as shown in Table 3.5.2 from the Label Encoded data,

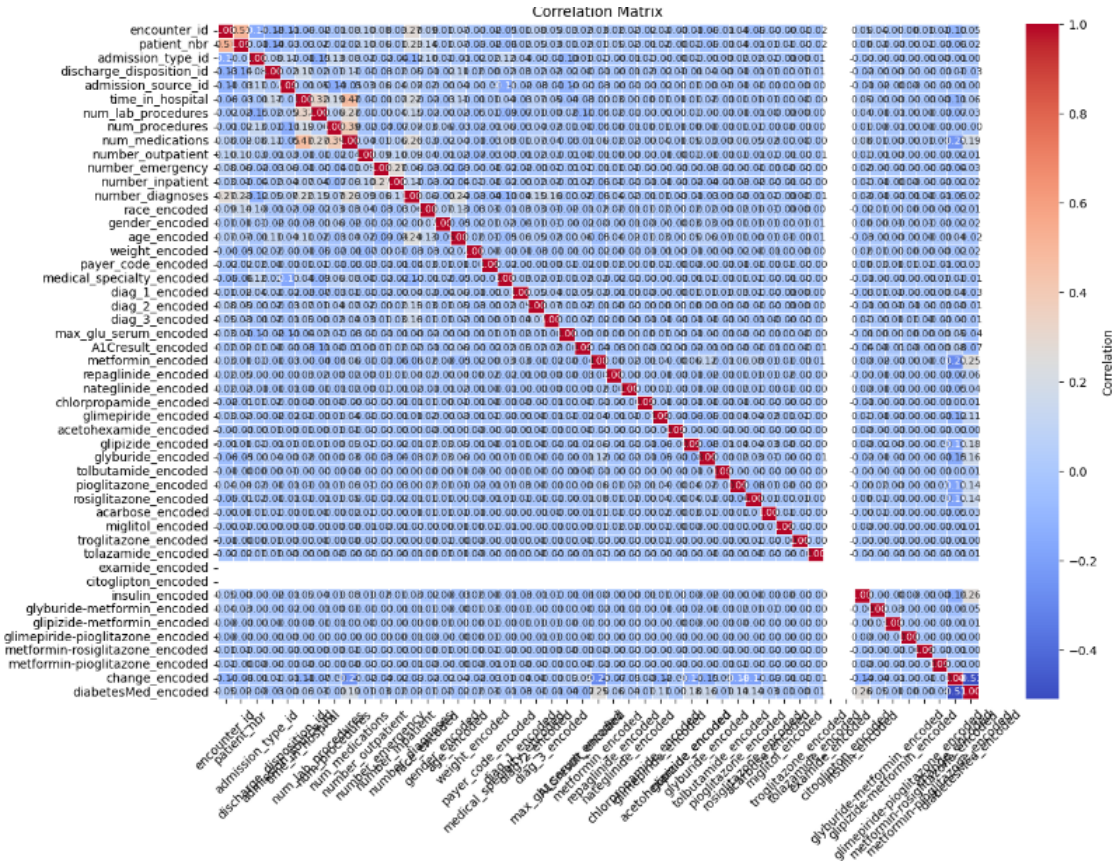

Figure 3.4.1: Chi-Square Correlation

effectively reducing dimensionality while capturing essential variance, whereas Boruta identified 20 critical features table 3.5.1 from the One-Hot Encoded dataset. booktabs

3.6 Treating Imbalanced Data

An imbalanced dataset occurs when the distribution of classes within the dataset is skewed, implying that one class significantly outnumbers the other(s). The "readmitted" output class shows an imbalance in it, with 11.2% of the data representing instances of readmission, while the remaining 88.8% does not, as seen in the 3.6.1 figure below. To address this issue, we used two different techniques: undersampling and oversampling.

Table 3.5.1: Best Features Selected using Boruta

Features	**Description**
admission_type_id	Admission type ID
discharge_disposition_id	Discharge disposition ID
admission_source_id	Admission source ID
time_in_hospital	Time spent in hospital
num_lab_procedures	Number of laboratory procedures
num_procedures	Number of procedures
num_medications	Number of medications
number_outpatient	Number of outpatient visits
number_emergency	Number of emergency visits
number_inpatient	Number of inpatient visits
number_diagnoses	Number of diagnoses
readmitted_binary	Readmitted (Binary)
visits_sum	Sum of visits
age	Age
diabetesMed_No	Diabetes medication (No)
diabetesMed_Yes	Diabetes medication (Yes)
diag_1_category_diabetes	Diagnosis 1 category (Diabetes)
diag_1_category_respiratory	Diagnosis 1 category (Respiratory)
number_medicaments_changes	Number of medication changes
number_medicaments	Number of medications

Table 3.5.2: Features of PC1, selected component of PCA

Features	**Description**
num_medications	Number of medications
metformin-pioglitazone	Metformin-pioglitazone
time_in_hospital	Time spent in hospital
change	Change
num_lab_procedures	Number of laboratory procedures
number_diagnoses	Number of diagnoses
num_procedures	Number of procedures
A1Cresult	A1C result
citoglipton	Citoglipton
admission_type_id	Admission type ID
tolbutamide	Tolbutamide
acetohexamide	Acetohexamide
pioglitazone	Pioglitazone
max_glu_serum	Max glucose serum
glipizide	Glipizide
chlorpropamide	Chlorpropamide
age	Age
metformin	Metformin
diag_1	Diagnosis 1
number_inpatient	Number of inpatient visits
number_outpatient	Number of outpatient visits
gender	Gender
diag_3	Diagnosis 3
repaglinide	Repaglinide
readmitted	Readmitted
rosiglitazone	Rosiglitazone
number_emergency	Number of emergency visits
admission_source_id	Admission source ID
insulin	Insulin
discharge_disposition_id	Discharge disposition ID
acarbose	Acarbose
glimepiride	Glimepiride
glimepiride-pioglitazone	Glimepiride-pioglitazone
miglitol	Miglitol
glyburide-metformin	Glyburide-metformin
glipizide-metformin	Glipizide-metformin
glyburide	Glyburide
race	Race
troglitazone	Troglitazone
metformin-rosiglitazone	Metformin-rosiglitazone
nateglinide	Nateglinide
diag_2	Diagnosis 2
examide	Examide
tolazamide	Tolazamide

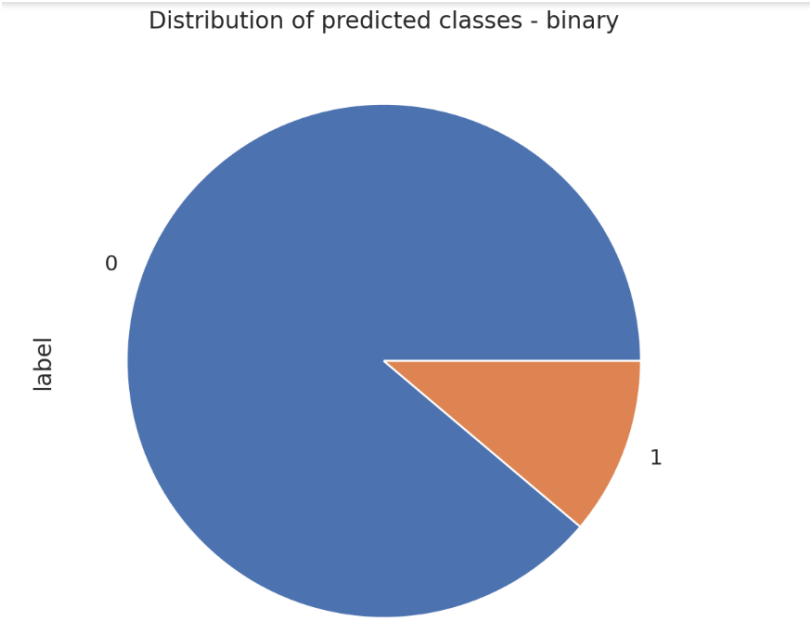

Figure 3.6.1: Distribution of Class

Undersampling reduces the number of instances in the majority class to balance it with the minority class, whereas oversampling increases the number of minority class instances. After random undersampling, we had 18,188 instances of one-hot encoding and 18,208 instances of label encoding. This process helped to alleviate the class imbalance by reducing the number of instances in the majority class. In contrast, using SMOTE oversampling resulted in a significant increase in instances, with 144,630 for one-hot encoding and 144,612 for label encoding. SMOTE created synthetic instances for the minority class to balance the dataset, ensuring better representation and improving model performance.

3.7 Methods

Our approach to tackling the task of predicting early readmission is based on machine-learning and deep-learning methods. After adequate data preprocessing, we conducted initial experiments with two distinct machine learning models: Random Forest (RF) and XGBoost, and four deep learning models: Multilayer Perceptron (MLP), Neural Network (NN), Convolutional Neural Network

(CNN), and Long short-term memory (LSTM) network. Both Random Forest and XGBoost are tree-based algorithms, sharing similarities but differing in construction and combination of results. While Random Forest builds each tree independently, XGBoost increases the gradient one tree at a time, correcting errors made by previous trees. The MLP, in turn, represents a class of feed-forward artificial neural networks, composed of at least three layers: an input layer, an output layer, and multiple hidden layers with several neurons. Each neuron in a layer is fully connected to neurons in the previous and next layers.

NN can grasp complex non-linear relationships, extract meaningful patterns, and handle large datasets, which makes it apt for the task at hand. CNNs have several advantages in predictive modeling, such as their ability to learn hierarchical representations, handle complex data structures, and automatically extract relevant features from raw data. In this paper, the CNN model comprises a 1D convolutional layer with 32 filters and a kernel of size 3, followed by a ReLu activation function.

LSTM has proven to be highly useful for prediction due to its many features. LSTMs can decrease dimensionality by capturing relevant features and discarding redundant features, which comes in handy in our scenario with 50 features all of which are not productive.

As seen during the analysis of data, it is very evident that only 11% of the patients have been re-admitted. Thus, another problem that we try to deal with is the problem of class imbalance of the data. We propose two methods to deal with this: undersampling using Random and oversampling using SMOTE algorithm. The data was divided into the ratio of 80% to 20% for training and testing respectively. Only the training set was employed with the balancing techniques. We performed experiments on the original data, the undersampled data and the oversampled data on the three models.

3.7.1 Random Forest

Random forest is a popular machine learning model for predictive modeling and classification, which is necessary for the task at hand, due to its robustness and

versatility. It operates by constructing multiple decision trees during the training phase and then outputs the mode of the classes for classification problems. This ensemble approach helps mitigate overfitting and improves generalization performance. For the original train data, oversampled, and undersampled train data this paper creates 10 trees to implement random forest.

3.7.2 XGboost

In our study, we employed XGBoost which is a powerful gradient boosting algorithm. XGBoost iteratively combines the individual predictions from each model (usually decision trees) and gives a final prediction. As it is an ensemble model, it uses multiple learners to combine the predictive power from different models. In this study, the model was initialized with a learning rate of 0.001, a maximum depth of 6, and a scale positive weight of 9.

3.7.3 MLP

MLP or Multi Layer Perceptron is the simplest of all the Neural Network models. It is composed of an input layer, hidden layer and finally an output layer. The number of hidden layers may vary with the increase in complexity. The proposed MLP model consists of two hidden layers followed by an output layer. The activation function ReLu is utilized which introduces non-linearity in the hidden layer. The 'adam' solver is employed to optimize the weights of the network during training. The maximum number of iterations were set to 30.

3.7.4 NN

NN can grasp complex non-linear relationships, extract meaningful patterns, and handle large datasets, which makes it apt for the task at hand. The first layer of the NN used has 64 neurons and a ReLu activation function. then, a dropout layer with a dropout rate of 0.2 is added to mitigate overfitting. 20 percent of the neurons' outputs from the previous layer during training is

randomly dropped to help mitigate overfitting. The second dense layer has 32 neurons and a ReLu function, and another dropout layer with a 0.2 dropout rate is added. Finally, the output layer is a Dense layer with a single neuron and a sigmoid activation function, suitable for binary classification tasks, such as predicting readmission of diabetic patients.

3.7.5 CNN

CNNs have several advantages in predictive modeling, such as their ability to learn hierarchical representations, handle complex data structures, and automatically extract relevant features from raw data. In this paper, the CNN model comprises a 1D convolutional layer with 32 filters and a kernel of size 3, followed by a ReLu activation function. A Flatten layer converts the output into a one-dimensional vector. Two dense layers, utilizing ReLu activation functions, are then added. The output layer consists of a single neuron with a sigmoid activation function for binary classification. The model is compiled using the Adam optimizer and binary cross-entropy loss function, with accuracy as the evaluation metric.

3.7.6 LSTM

LSTM has proven to be highly useful for prediction due to its many features. LSTMs can decrease dimensionality by capturing relevant features and discarding redundant features, which comes in handy in our scenario with 50 features all of which are not productive. The paper uses an LSTM and CNN hybrid model architecture. 32 kernels are applied in the convolutional layer and the size of the convolutional kernel is 3, with a pool size of 2. Additionally, there are 100 neurons in the LSTM layer. There are 4 layers in this architecture, namely: convolutional layer, max pooling layer, LSTM layer, and output layer. The convolutional layer applies convolutional operation and uses a ReLu activation function to introduce non-linearity to the output. The max pooling layer reduces the dimensionality of the feature maps obtained from the convolutional layer. The LSTM layer utilizes LSTM units to process sequential data

and capture long-term dependencies in the input sequences. Lastly, the output layer consists of a single neuron with a sigmoid activation function, suitable for binary classification tasks.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Result of Analysis of Dataset

The dataset used in this paper is "Diabetes 130-US for years 1999-2008". The dataset contains 50 attributes detailing the age, race, gender, number of hospitalizations, types of medicines, number of medicine changes, medical department, number of outpatient and inpatient, number of emergencies, number of procedures, etc. There are 101,766 unique patient entries.

It can be observed from figure 3.6.1 that there is a great class imbalance in the dataset. The "readmitted" output class is only 11.2% whereas the "not readmitted" class takes up the remaining 88.8%. As seen previously in the paper this has been compacted by random undersampling and oversampling using SMOTE.

Further study of the dataset reveals the following results:

1. Patients with a higher number of diagnoses may require a greater number of medications.
2. Patients with more emergency visits may also have a higher likelihood of subsequent inpatient admission.
3. Figure 3.1.1 shows us there is an almost fifty-fifty split among the two genders with 53.9% females and 46.1% males.
4. There is a racial bias as the majority of the dataset caters to Caucasians, mostly due to the fact it is a US-based dataset as seen in figure 3.1.5
5. 70-80 is the most common age in the dataset and the highest number of readmissions can be observed in the 70-79 age group for both men and women.
6. It can also be noted that medication change for diabetics upon admission is associated with lower readmission rates.
7. After performing the outlier analysis, it was found that no extreme outliers were found that needed to be removed.

4.2 Performance Analysis of Imputation Methods

Imputation methods are used to deal with missing data in the dataset. These methods are used to fill in the missing data to create a complete data matrix that can be analyzed using standard methods. This paper has employed and compared three imputation methods: Empirical methods (mean and mode), KNN, and MICE.

Table 4.2.1: F1-Scores

Imputation Method	**F1-Score**
Empirical Method	0.30461
KNN	0.33520
MICE	0.35202

When we train a model on an imputed dataset, the effectiveness of the imputation would be reflected on the model's performance. We have used logistic regression as a model and trained and evaluated it on all three imputation methods. The evaluation metric used is F1 evaluation metric.

It can be observed from table 4.2.1 that MICE has outperformed KNN, and the Empirical method (mean and mode) with an F1-score of 0.35202. Subsequently, this paper has selected MICE as the main imputation method in the architecture.

4.3 Performance of Models

Here, we examine the classification performance of different deep learning models and machine learning methods on two different datasets encoded using One Hot Encoder (OH) and Label Encoder (LE). Accurately classifying the datasets was the aim, and each model's performance was assessed by independent tests. To assess the performance of the models, we use a wide range of evaluation metrics, such as recall, accuracy, precision, F1 score, and ROC-AUC score. Random Forest (RF) and XGBoost consistently showed strong performance on both datasets, obtaining high F1 scores and accuracy. The accuracy and performance metrics of all the used models are represented in table 4.3.1, Convolutional Neural Network (CNN) demonstrated competitive performance,articularly on the dataset with LE encoding. Promising outcomes have been observed with deep learning models such asLong Short Term Memory (LSTM) and Multi-layer Perceptron (MLP) , particularly on the OH.

Table 4.3.1: Performance Analysis of Models

Models	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Random Forest					
Original (LE)	88.92	0.84	0.89	0.84	0.65
Oversampled (LE)	88.72	0.83	0.89	0.84	65
Undersampled (LE)	61.84	0.84	0.62	0.69	50
Original (OH)	67.42	0.67	0.53	0.67	0.65
Oversampled (OH)	46.38	0.40	0.75	0.46	0.64
Undersampled (OH)	60.56	0.60	0.63	0.60	0.65
XGBoost					
Original (LE)	88.74	0.36	0.22	0.41	0.55
Oversampled (LE)	88.78	0.40	0.26	0.49	0.55
Undersampled (LE)	60.76	0.16	0.59	0.25	0.50
Original (OH)	12.01	0.12	01.0	0.12	0.60
Oversampled (OH)	11.22	0.11	1.00	0.11	0.62
Undersampled (OH)	58.74	0.58	0.63	0.58	0.61
MLP					
Original (LE)	88.87	0.83	0.63	0.70	0.61
Oversampled (LE)	74.32	0.83	0.61	0.69	0.61
Undersampled (LE)	63.08	0.70	0.56	0.61	0.55
Original (OH)	89.90	0.89	0.02	0.89	0.69
Oversampled (OH)	89.10	0.57	0.69	057	0.64
Undersampled (OH)	57.42	0.89	0.01	0.89	0.69
LSTM					
Original (LE)	88.88	0.61	0.34	0.68	0.57
Oversampled (LE)	81.12	0.59	0.28	0.67	0.55
Undersampled (LE)	61.24	0.46	0.2	0.56	0.54
Original (OH)	89.99	0.48	0.01	0.89	0.63
Oversampled (OH)	88.88	0.56	0	0.47	0.62
Undersampled (OH)	76.34	0.19	0.36	0.55	0.63
CNN					
Original (LE)	88.93	0.83	0.85	0.83	0.65
Oversampled (LE)	76.67	78	0.68	0.69	0.61
Undersampled (LE)	84.63	0.80	0.81	0.78	0.65
Original (OH)	89.86	0.48	0.01	0.89	-0.65
Oversampled (OH)	54.47	0.15	0.66	0.54	0.63
Undersampled (OH)	54.73	0.15	0.66	0.46	0.63
NN					
Original (LE)	88.88	0.85	0.69	0.75	0.64
Oversampled (LE)	73.08	0.70	0.70	0.70	0.69
Undersampled (LE)	67.28	0.68	0.67	0.67	0.67
Original (OH)	88.88	0.55	0.60	0.47	0.50
Oversampled (OH)	88.88	0.56	0.47	0.47	0.50
Undersampled (OH)	88.88	0.65	0.60	0.47	0.50

4.4 Ablation Study

An ablation study was conducted to identify the impact of different parts of the methodology as seen in figure 3.0.1 on the overall performance of the individual models. This helps us to gauge the necessity of each step and the requirement of the same for the proper working of the proposed architecture.

4.4.1 Ablating Imputation Module

The dataset contains several missing information in several data instances. Imputation has been done to tackle this problem so that the models can perform their best and we can utilize the most out of each model and obtain accurate predictions. Missing data can lead to biased analyses and inaccurate conclusions if not properly handled. Imputation helps preserve the integrity of the dataset by replacing missing values with estimated values, allowing for a more complete analysis. Complete datasets generally provide greater statistical power compared to incomplete datasets. Imputation enables the utilization of more data points, thus increasing statistical power and potentially improving the reliability of the analysis. Imputation allows for the retention of cases with missing values, thereby maintaining the sample size and maximizing the utilization of available data. Complete case analysis can introduce bias, particularly if missingness is related to the outcome or other variables of interest (known as missing not at random or MNAR). Imputation methods aim to reduce bias by estimating missing values based on observed data, potentially leading to more accurate parameter estimates and inferential results. To conduct this ablation analysis, we have removed all instances that have a null value and then run the architecture and measured the accuracy of the various models used as shown in Table 4.4.1. The total size of the dataset was reduced to 1043 instances losing most of the data. Downsampling of data only gave us 144 instances and oversampling gave 1524 instances

Table 4.4.1: Models Without Imputation

Models	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Random Forest					
Original (LE)	89.92	0.86	0.89	0.84	0.67
Oversampled (LE)	88.34	0.84	0.89	0.83	0.65
Undersampled (LE)	36.78	0.46	0.38	0.69	0.50
Original (OH)	87.43	0.78	0.53	0.67	0.65
Oversampled (OH)	84.72	0.59	0.68	0.46	0.64
Undersampled (OH)	84.24	0.67	0.67	0.58	0.65
XGBoost					
Original (LE)	63.52	0.34	0.22	0.41	0.65
Oversampled (LE)	76.46	0.59	0.35	0.49	0.45
Undersampled (LE)	57.60	0.65	0.59	0.35	0.50
Original (OH)	14.01	0.12	01.0	0.12	0.60
Oversampled (OH)	12.22	0.11	1.00	0.11	0.62
Undersampled (OH)	58.74	0.58	0.63	0.58	0.61
MLP					
Original (LE)	88.87	0.83	0.63	0.70	0.61
Oversampled (LE)	74.32	0.83	0.61	0.69	0.61
Undersampled (LE)	63.08	0.70	0.56	0.61	0.55
Original (OH)	87.6	0.86	0.02	0.89	0.69
Oversampled (OH)	86.0	0.57	0.64	0.57	0.64
Undersampled (OH)	20.06	0.49	0.01	0.8 34	0.38
LSTM					
Original (LE)	88.88	0.61	0.34	0.68	0.57
Oversampled (LE)	81.12	0.59	0.28	0.67	0.55
Undersampled (LE)	61.24	0.46	0.2	0.56	0.54
Original (OH)	87.0	0.48	0.01	0.89	0.63
Oversampled (OH)	14.35	0.56	0	0.47	0.62
Undersampled (OH)	86.60	0.19	0.36	0.55	0.63
CNN					
Original (LE)	88.93	0.83	0.85	0.83	0.65
Oversampled (LE)	76.67	78	0.68	0.69	0.61
Undersampled (LE)	84.63	0.80	0.81	0.78	0.65
Original (OH)	87.56	0.48	0.01	0.89	-0.65
Oversampled (OH)	78.74	0.15	0.66	0.54	0.63
Undersampled (OH)	58.59	0.15	0.66	0.46	0.63
NN					
Original (LE)	88.88	0.85	0.69	0.75	0.64
Oversampled (LE)	73.08	0.70	0.70	0.70	0.69
Undersampled (LE)	67.28	0.68	0.67	0.67	0.67
Original (OH)	71.7	0.55	0.60	0.47	0.50
Oversampled (OH)	71.7	0.56	0.47	0.47	0.50
Undersampled (OH)	71.7	0.65	0.60	0.47	0.50

4.4.2 Ablating Encoding Module

Machine learning models require numerical inputs. Categorical variables, such as gender, race, or medication type, are represented as strings or labels, which algorithms cannot directly process. Hence, encoding the data becomes necessary. Many machine learning algorithms require numerical input. Encoding categorical attributes into numerical representations allows these algorithms to process the data effectively. Categorical attributes often contain valuable information, such as class labels or categorical distinctions. Encoding preserves this information in a format that can be understood and utilized by machine learning models. Without proper encoding, categorical attributes may introduce bias into the model. For example, if categorical attributes are treated as ordinal without encoding, the model might incorrectly infer an order or hierarchy that doesn't exist. Encoding ensures that categorical attributes are represented appropriately, minimizing the risk of bias. Many machine learning algorithms, such as regression and neural networks, require numerical input. Encoding categorical attributes allows these algorithms to be applied to datasets containing a mix of numerical and categorical data. Encoding categorical attributes can facilitate feature engineering by creating new features based on categorical variables. For example, one-hot encoding can create binary variables for each category, which can then be used as input features in machine learning models.

We can see that encoding is an integral part of running the various machine learning models due to the aforementioned reasons. As a part of ablating the encoding module, we have chosen to remove all categorical attributes run the architecture as is, and observe the impact on the accuracy of the models. We can also understand that ablating this module can cause inaccurate predictions due to biases, inaccurate representation of data, and such. We dropped all the categorical features for the ablation study of encoding features. We were left with 8 features which is not a good representation of the dataset. Ablating the encoder we can see that there is a drastic drop in accuracy among all models and this can be observed in Table 4.4.2 which gives a complete overview of the drop in accuracy while removing the encoding modules.

Table 4.4.2: Model Accuracy without Encoding

Sampling	Models					
	Random Forest	**XGBoost**	**MLP**	**NN**	**CNN**	**LSTM**
Original	69.4	12.06	89.7	89.9	67.41	87.08
Undersampled	65.3	66.2	63.0	88.8	73.21	74.42
Oversampled	28.02	11.32	72.61	47.50	67.41	11.34

4.4.3 Ablating Feature Selection Module

Feature selection is a vital part of meticulously designed architecture. By selecting only the most relevant features, feature selection can enhance model performance by reducing overfitting and improving generalization. Irrelevant or redundant features can introduce noise into the model, leading to decreased predictive accuracy. Removing such features through feature selection can result in simpler and more interpretable models that generalize better to unseen data. Including irrelevant features in the model can lead to overfitting, where the model captures noise in the training data rather than underlying patterns. Feature selection helps mitigate overfitting by focusing the model on the most informative features, preventing it from memorizing noise present in irrelevant or redundant features. Models trained on datasets with fewer features typically require less computational resources and time for both training and inference. Feature selection reduces the dimensionality of the dataset, resulting in faster model training and more efficient predictions, especially important in scenarios with large datasets and limited computational resources. In real-world applications, collecting and storing data can be expensive, especially in scenarios where features are obtained through complex measurements or experiments. Feature selection can help reduce costs by focusing resources on collecting and maintaining only the most informative features, eliminating the need for unnecessary data collection and storage. The effect of ablating feature selection can be observed in table 4.4.3.

Table 4.4.3: Models without Feature Selection

Models	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
Random Forest					
Original (LE)	63.00	0.67	0.63	0.64	0.69
Oversampled (LE)	62.00	0.67	0.62	0.63	0.69
Undersampled (LE)	62.00	0.67	0.62	0.63	0.69
Original (OH)	61.00	0.67	0.56	0.63	0.63
Oversampled (OH)	43.29	0.40	0.40	0.60	0.55
Undersampled (OH)	79.04	0.60	0.28	0.56	0.66
XGBoost					
Original (LE)	66.92	0.64	0.67	0.62	0.69
Oversampled (LE)	66.21	0.64	0.66	0.64	0.67
Undersampled (LE)	62.35	0.67	0.62	0.63	0.69
Original (OH)	52.00	0.52	0.15	0.69	0.64
Oversampled (OH)	58.36	0.16	0.63	0.42	0.64
Undersampled (OH)	42.35	0.12	0.42	0.38	0.51
MLP					
Original (LE)	65.03	0.00	0.00	0.00	0.50
Oversampled (LE)	34.96	0.34	1.00	0.51	0.50
Undersampled (LE)	64.72	0.43	0.26	0.50	0.50
Original (OH)	67.07	0.74	0.79	0.68	0.55
Oversampled (OH)	64.45	0.72	0.47	0.57	0.58
Undersampled (OH)	43.00	0.77	0.39	0.46	0.61
LSTM					
Original (LE)	66.32	0.58	0.59	0.60	0.62
Oversampled (LE)	60.56	0.62	0.59	0.59	0.62
Undersampled (LE)	58.85	0.63	0.59	0.60	0.62
Original (OH)	74.33	0.65	0.06	0.52	0.76
Oversampled (OH)	64.83	0.34	0.14	0.56	0.60
Undersampled (OH)	58.00	0.27	0.16	0.54	0.61
CNN					
Original (LE)	66.62	0.64	0.60	0.62	0.65
Oversampled (LE)	60.54	0.65	0.61	0.62	0.65
Undersampled (LE)	61.30	0.65	0.61	0.62	0.65
Original (OH)	60.05	0.60	0.49	0.16	0.65
Oversampled (OH)	62.56	0.62	0.50	0.17	0.65
Undersampled (OH)	60.38	0.62	0.50	0.16	0.65
NN					
Original (LE)	61.24	0.45	0.57	0.51	0.64
Oversampled (LE)	60.05	0.44	0.58	0.50	0.63
Undersampled (LE)	60.04	0.44	0.58	0.50	0.63
Original (OH)	48.7	0.45	0.56	0.17	0.52
Oversampled (OH)	48.7	0.56	0.47	0.47	0.50
Undersampled (OH)	48.7	0.65	0.60	0.47	0.50

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this project, we aimed to develop a predictive model using a combination of traditional machine learning algorithms and advanced Deep Learning techniques to predict the likelihood of patient readmission within 30 days. We conducted comprehensive data analysis, feature importance assessment, correlation analysis, and implemented various imputation methods to handle missing values. By comparing the performance of Machine Learning and Deep Learning models, we identified the most effective model for predicting patient readmission, contributing to enhanced healthcare quality and patient outcomes.

Observations based on the work reveal that while deep learning models like Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) showed promise, Random Forest (RF) and XGBoost consistently demonstrated solid performance across different encoding strategies. By focusing on the entire dataset rather than specific age groups, the study achieved superior accuracy rates by leveraging a combination of traditional and advanced machine learning techniques.

Future improvements in this project could involve incorporating larger and more recent datasets to ensure the predictive models are aligned with current healthcare scenarios. By utilizing newer datasets spanning a broader timeframe, the predictive accuracy and applicability of the models can be further enhanced, leading to more reliable predictions of patient readmission in hospitals. Additionally, exploring emerging machine learning and deep learning techniques could offer opportunities to refine the predictive models and improve their performance in real-world healthcare settings.

REFERENCES

[1] Gerard F Anderson and Earl P Steinberg. Hospital readmissions in the medicare population. *New England Journal of Medicine*, 311(21):1349–1353, 1984.

[2] Stephen F Jencks, Mark V Williams, and Eric A Coleman. Rehospitalizations among patients in the medicare fee-for-service program. *New England Journal of Medicine*, 360(14):1418–1428, 2009.

[3] Santiago Romero-Brufau, Kirk D Wyatt, Patricia Boyum, Mindy Mickelson, Matthew Moore, and Cheristi Cognetta-Rieke. Implementation of artificial intelligence-based clinical decision support to reduce hospital readmissions at a regional hospital. *Applied clinical informatics*, 11(04):570–577, 2020.

[4] Haishuai Wang, Zhicheng Cui, Yixin Chen, Michael Avidan, Arbi Ben Abdallah, and Alexander Kronzer. Predicting hospital readmission via cost-sensitive deep learning. *International Conference on Health Big Data and Artifcial Intelligence 2020 Guangzhou, China*, 15(6):1968–1978, 2018.

[5] Lei Wang Zheqing Zhang Siwei Zhou1 Yun Liu3 Jiancheng Dong Yujuan Shang1, Kui Jiang and Huiqun Wu. The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifers. *BMC Medical Informatics and Decision Making Volume 21, Supplement 2 2021: Health Big Data and Artificial Intelligence*, 2020.

[6] Damian Mingle. Predicting diabetic readmission rates: Moving beyond hba1c. *Current Trends in Biomedical Engineering Biosciences*, 2017.

[7] Chahes Chopra, Shivam Sinha, Shubham Jaroli, Anupam Shukla, and Saumil Maheshwari. Recurrent neural networks with non-sequential data to predict hospital readmission of diabetic patients. *Association for Computing Machinery.*

[8] Sacha Davis, Jin Zhang, Ilbin Lee, Mostafa Rezaei, Russell Greiner, Finlay A McAlister, and Raj Padwal. Effective hospital readmission prediction models using machine-learned features. *BMC Health Services Research*, 22(1):1415, 2022.

[9] Malladihalli S Bhuvan, Ankit Kumar, Adil Zafar, and Vinith Kishore. Identifying diabetic patients with high risk of readmission. *arXiv preprint arXiv:1602.04257*, 2016.

[10] Lior Turgeman, Jerrold May, Johnson Moore, and Youxu Tjader. A mixed-ensemble predictive model for hospital readmission. 11 2015.

[11] Cios Krzysztof DeShazo Jon Clore, John and Beata Strack. Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5230J.

APPENDIX A

Timeline

Month and Year	Expected Work Done
August'23	Study and Survey of techniques that can be used for solving the problem, along with the understanding of the dataset and Problem Statement. Literature Survey.
September'23	Dataset visualization, preprocessing, conversion to binary class, Remove Columns
October'23	Implement imputation techniques such as Empirical Method, MICE, and KNN, evaluating their respective performances on the dataset. Subsequently, we analyze the augmented dataset to ascertain the most effective imputation method
November'23	Execute encoding techniques employing both one-hot and label encoders. That allowed us to efficiently transform categorical variables into numerical representations
January'24 - February'24	Evaluate the performance of diverse models LSTM, CNN, XGBoost, MLP, Random Forest, and NN utilizing metrics including accuracy, precision, recall, F1 score, and AUC-ROC.
March'24 - April'24	Performance feature selection techniques such as PCA and Boruta. Subsequently, we evaluate the model using the selected features.

APPENDIX B

APPENDIX C

Biodata

Student Name	**Roll No.**	**Email ID**	**Phone Number**	**Address**
Sarthaki Gajghate	201IT155	sarthakigajghate .201it155@nitk.edu.in	9423069690	S5, Nalini 1, Shivam Kunj, Barde Layout, Boregaon, Katol Rd, Nagpur, 440013
Geethanjali Manoj Kumar	201IT220	geethanjalimanojkumar .201it220@nitk.edu.in	8138960286	Pallathoor House, Kadikkad Punnayurkulam, Thrissur, Kerala, 679561
Lucia Americo Tembe	201IT169	luciaamericotembe .201it169@nitk.edu.in	9035264132	Home number 9 Tsalala Q8 Maputo,Matola