H-Predict A Machine Learning Based Model for Treatment Cost Estimation

Ganaprathyusha Puluputhuri Muni, Kavana Anil, Neha Sharma, Nivedita Venkatachalam

Department of Applied Data Science, San Jose State University

DATA 270 Data Analytics Processes

Dr. Linsey Pang

May 06, 2024

Abstract

Healthcare in America has been a topic of significant discussion due to the high treatment costs and soaring insurance expenses. The average annual hospital bill expenditure for individuals in the USA stands at $13,000, with a steady upward trend, constituting 25% of the average American household income. The research aims to develop a model estimating treatment costs by analyzing patient age, length of stay, disease type, medical history, and gender, addressing limited patient visibility into incurred costs and influencing factors. Traditional statistical techniques will be utilized for quality data preparation. Machine learning techniques such as XG Boost, Random Forest, Gradient Boosting Regressor, Linear Regression, and Polynomial Regression will be employed to create the estimator model "H-Predict." The project aims to evaluate all techniques used and will try to find out the most suitable model for the purpose. Model accuracy, feature analyses, and bias-variance trade-offs will be some of the methods for best-fit identification. Metrics used to rigorously evaluate the performance of models are MSE, RMSE and R-squared. The Random Forest model achieved an R-squared of 0.911, outperforming other models, closely followed by XGBoost with an R-squared of 0.897. While Random Forest demonstrated superior performance, XGBoost was computationally more efficient and generalized better to new data, making it advantageous in scenarios where computational resources are limited or costly. The tool will provide tailored treatment cost estimates for patients and families, facilitating proactive financial planning, while also aiding insurance companies and clinics in claim management to assist patients effectively.

Keywords Prediction, Healthcare, Treatment cost, Machine Learning

1. Introduction

1.1 Project Background and Execute Summary

Healthcare is a necessity for everyone. On an average, in the United States of America, every household spends about 25% of their total income on medical or healthcare expenses. This number is increasing drastically every year. The health inflation in the USA from 2022 to 2023 is around 2%. (Healthcare Dive, 2023)

With the increasing population, medical inflation and all the technological advancements we have in the field of medicine, understanding all the features and factors that contribute to the cost of healthcare is vital. Predicting the healthcare expenses will be useful to patients, facilitating them to manage and plan their finances accordingly. These predictions can also be used by healthcare insurance providers to access the claims they have received. However, the capability of software or any other application to provide an accurate estimation for the healthcare expense remains constrained.

Multiple factors such as age of patient, medical history, location demographics, gender, individual lifestyle choice, quality of regional healthcare available etc. affect the final treatment cost. Accurate estimates are essential for people to plan their finances. An algorithm-based model which predicts healthcare expenses is essential to address this issue. Predicting an estimate of the overall cost early not only enables everyone to plan their finances proactively but also optimizes the time duration to get immediate medical assistance.

We chose the dataset provided by the United States of America's government website. The dataset is built by merging yearly data managed by the Federal government. The yearly datapoints are merged to create a single dataset which is used for model training and accurate

predictions. Next, we perform some aggregations on the data in addition to feature extraction to draw meaningful insights and required contributing features to help predict the cost. The dataset obtained post all the preprocessing and data cleaning steps is much easier for the machine learning model to process and understand.

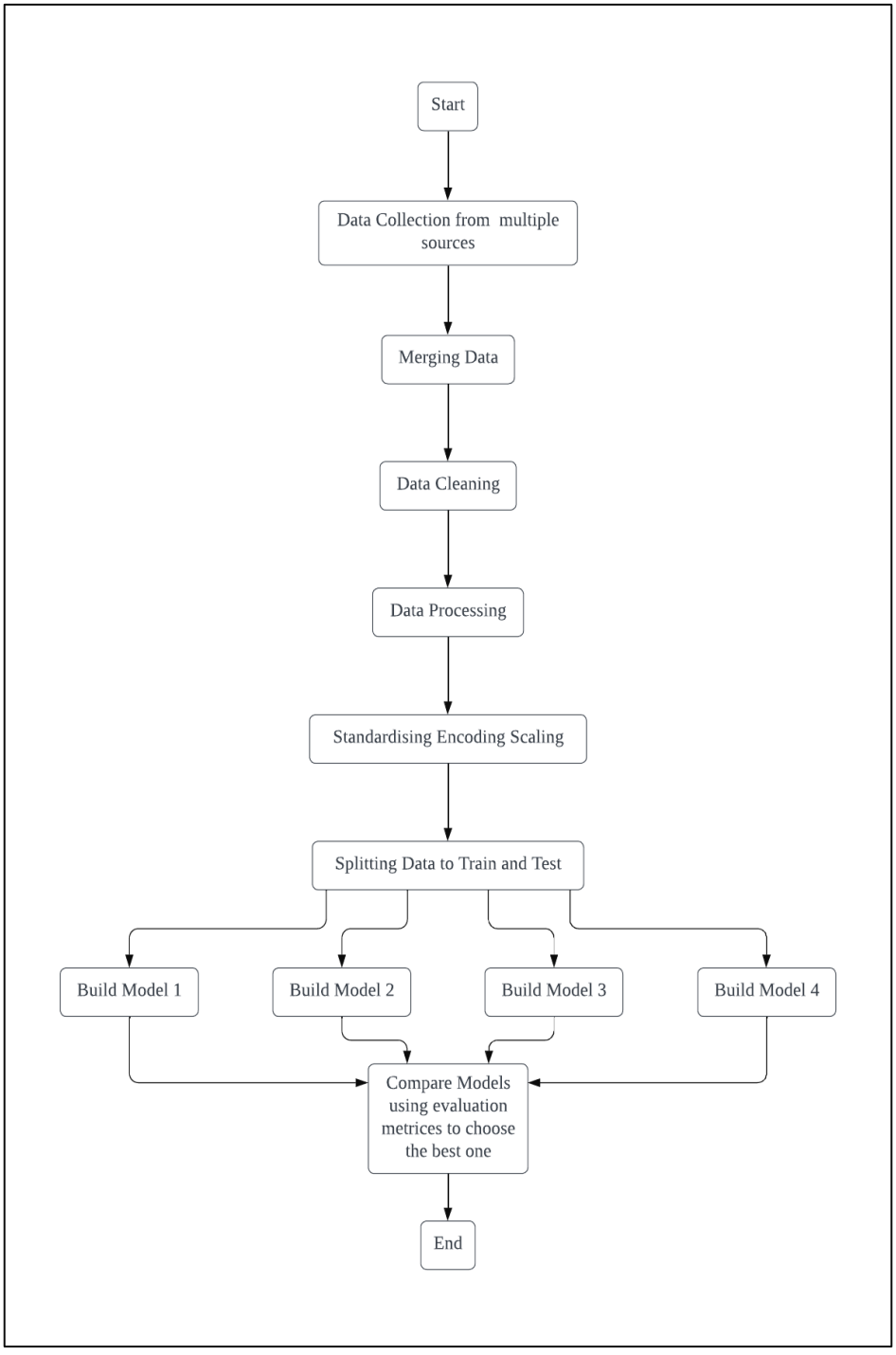
The cost prediction in the domain of healthcare can be achieved using many basic machine learning models such as Linear Regression, Polynomial Regression, Gradient Boosting Regressor, Random Forest, and XG Boost.

Developing an effective model for healthcare cost prediction serves as a crucial step in applications aimed at improving human-machine interaction in the healthcare domain. Identifying and predicting healthcare costs can have far-reaching impacts on various applications, enhancing the overall experience for all healthcare providers, Insurance companies and patients. It optimizes resource allocation in healthcare facilities such as hospitals and clinics and helps individuals plan their insurance. Additionally, it helps insurance clients to detect if any claim made was a fraud.

The ability to predict the cost of healthcare can lead to the development of personalized health recommendations that enhance the total end to end satisfaction of individuals. The pivotal information obtained from the predictions of the entire model can further be utilized by governing agencies and healthcare experts to create or amend policies. Additionally, this data can play an essential role in managing resources in the healthcare space strategically which results in higher efficiency with less or minimum time needed to ensure decent quality healthcare to all the people at a known cost. Figure 1 shows a flowchart that represents the sequence of steps from data collection to model evaluation in the project's workflow.

Figure 1

Data Processing Flowchart



Note. Flow chart showing data processing steps

1.2 Project Requirements

Functional Requirements

Scaling. The data obtained needs to be standardized and normalized for the machine learning algorithm to understand. Columns such as gender, birth weight etc. might have to be standardized and normalized to ensure that each feature in the dataset contributes equally to the training of the machine learning algorithm. Standardization transforms the feature such that it has a standard deviation of 1 and mean of 0 while normalization transforms the feature in between a specific range.

Encoding. This involves converting categorical data to a numerical value. Encoding ensures that the categorical information is utilized by the model correctly. Columns such as Hospital service area, type of admission etc. must be encoded into numerical data for the model to understand the data.

Feature Engineering. Feature engineering is performed to understand the underlying relationships among many features that could not be established during the initial exploratory phase. It enhances the overall model's performance. The features which contribute more towards the overall cost can be analyzed through these techniques. Features such as Total Charges and Total cost might be clubbed to understand the cost data better. Additionally, procedure information and diagnosis can be combined to form a composite feature which helps understand the patient's condition better.

AI-Powered Requirements

To predict the cost of healthcare, our team will build a few machine learning models indicated below. Metrics such as MAE, MSE, RMSE, R-squared etc. will be used to assess the model's performance.

Linear Regression. It is a supervised learning machine learning algorithm. It is used to predict the cost of healthcare which is the dependent variable by considering the impact of other features present in the dataset which are the independent variables. Multiple or single variables can be chosen to perform regression and check which features perform well and impact more on the dependent variable.

Polynomial Regression. This is a machine learning algorithm which is an extension from Linear regression. It works much better in cases where data is significantly spread out and fitting a line does not justify the model predictions. The model tries to include polynomial exponents of the existing features and fit the model accordingly. There are many ways to optimize the parameters of the model. Stochastic gradient descent can be tried with polynomial regression and the results can be understood to see the impacts.

Random Forest. This is an ensemble machine learning method used for both classification and regression analysis. When applied to the prediction of health care cost, Random Forest Regression is used to build a model. During training, Random Forest builds a multitude of decision trees and aggregates the prediction values of decision trees and gives accurate prediction by taking average of all the predictions. Random Forest is powerful in the case of healthcare cost prediction as it can handle many descriptive features and complex relationships between them.

Extreme Gradient Boosting (XGBoost). The report highlights the effectiveness of XGBoost (Extreme Gradient Boosting) for predicting healthcare costs emphasizing its efficiency and performance in dealing with large datasets which are common in healthcare analytics. XGBoost is adept at handling complex and non-linear relationships among diverse features such as age, BMI, smoking habits etc. It excels in environments with extensive data benefiting from

its ability to process multiple features efficiently and its robust mechanisms for handling missing data through automatic imputation. XGBoost's built-in regularization prevents overfitting which is essential for reliable medical predictions while its tree pruning, and built-in cross-validation ensure accurate and generalizable model performance. However, it requires careful tuning of parameters to avoid overfitting and can be computationally demanding thus requiring substantial resources for large-scale data processing. In healthcare applications XGBoost can categorize patients into risk groups or predict costs effectively making it a valuable tool for health management and cost optimization.

Gradient Boosting Regressor. This is useful for forecasting healthcare expenses because they can train weak learners in a stepwise manner, which increases prediction accuracy over time. It improves prediction strength over a series of rounds by iteratively fixing errors generated by prior models. The system can identify intricate links in healthcare data thanks to this iterative learning process, which gives patients more precise cost estimates.

Data Requirements

The Data retrieved from the official site of Agency of Healthcare Research and quality has data from the year 1996 to 2001. To ensure relevance in the predictions made by our models, the last few years data can be collected and merged for our project. Additionally, we have incorporated healthcare cost-related information from Hospital Inpatient Discharges in the State of New York, enriching our dataset for a comprehensive analysis of the factors influencing healthcare costs.

These datasets provide various features, offering insights into patient demographics, hospital characteristics, and procedural details of each patient. All data points in our dataset are

thoughtfully labeled, laying the groundwork for a supervised learning approach to predict healthcare costs. The overall size of the dataset post merging and basic cleaning is expected to be around 1 million records.

Our aim is to train robust predictive models, and this dataset aligns with our project's goal of understanding the diverse factors influencing healthcare expenses. As we proceed with model building, the meticulous labeling of each data point becomes essential to accurately represent variations in the healthcare costs.

For users referring to this documentation, a foundational understanding of statistical methods, feature engineering, and predictive modeling is recommended. Additionally, familiarity with machine learning algorithms, regression techniques, and model evaluation metrics will empower a more effective interpretation of our healthcare cost prediction model.

1.3 Project Deliverables

Research Proposal. An overview of the issue under investigation, data sources for the project, and a few examples of relevant or related literature reviews are provided in the Research proposal step of the project cycle.

Project Overview. Project overview captures the overall requirements of the project such as the technical requirements, functional knowledge needed to fulfill the requirements and the data needed for the project. It also describes the expected results along with projected completion dates and provides a thorough analysis of pertinent research and technology studies related to the project's topic.

Effort Evaluation. Effort evaluation includes assessing the resources, time, and all the skills needed to complete various tasks such as developing models, preprocessing data, cleaning

data, and collecting data. This quantification is usually done in the hours needed to complete each task, helps with efficient resource allocation, project scheduling, and deadline compliance. It helps in task prioritization during sprint planning, enhancing adaptability to the changing or evolving project needs.

Abstract Report. The project abstract report is a concise synopsis that covers the goals, methods, important conclusions, and implications of the research undertaken. It is essential for readers to understand the project's importance and contributions without reading the entire documentation because it provides a concise overview of the research.

Gantt Chart. A Gantt chart or a graphical visual representation of the project schedule that displays all the tasks, subtasks, dates, and the resource allocation for each of the task. This representation gives an overall picture of the project's dependencies and progress by displaying how the subtasks are connected and interdependent to one another.

Pert Chart. A PERT chart is used in project management for mapping out tasks, splitting those tasks into smaller components, and scheduling them according to a deadline. PERT charts visually provide insights on the dependency among different tasks and the project's overall path. It represents the minimum amount of time required to complete all the tasks required in the project.

Task Breakdown Structure and Resource Allocation. Based on Agile principles, the Task Breakdown Structure (TBS) defines all tasks that need to be addressed and finished for a successful deployment. Assigning team members to tasks according to their skill sets is the process of allocating resources. Throughout the project's life cycle, this flexible methodology

guarantees efficient task completion. Regular communication and teamwork, frequently aided by weekly meetings, provide more transparency in the project.

Data Collection and Data Management Plan. The process of obtaining relevant information from a variety of sources which adheres to the project requirement, and guaranteeing its accuracy for analysis is Data Collection. Data management involves storing and handling the collected data throughout the project lifecycle, ensuring its accessibility and usability. Efficient data management and collection are crucial components in extracting meaningful insights and making accurate predictions.

Data Preprocessing and Exploration. The collected raw data is cleaned to remove the missing values and outliers from the dataset. Subsequent processing and transformation reshape the data to a format that is understandable to the machine learning algorithms. Exploratory Data Analysis (EDA) is then performed, employing statistical and visual techniques to understand the underlying patterns, trends, and anomalies, providing crucial insights for subsequent modeling stages. This integral phase lays the foundation for informed decision-making by preparing the data for meaningful interpretation and modeling.

Model Development. In this phase, the preprocessed data is used to extract relevant characteristics and predefined algorithms are used to develop machine learning models. These models undergo a training phase, which enables them to identify patterns and relationships while optimizing parameters for best performance. Constant evaluation guarantees that models are in line with project objectives and have the capacity to offer insightful information. This iterative process ensures that the model built makes correct predictions.

Individual Report. The individual report consists of the research done by each team member individually and the corresponding machine learning model developed by each team member. This report is the overall documentation provided by each member highlighting their approaches to the problem and its outcomes.

Project Final Report. The team's comprehensive research conducted to address the selected problem is included in the final report. It highlights the collaborative effort of the team to address the problem. It documents all the approaches taken, the unique methodologies used, and the outcome of each machine learning model developed.

1.4 Technology and Solution Survey

Machine Learning methodologies are woven within the array of research efforts that trace the journey of employing these techniques for healthcare cost prediction and contribute uniquely to the richness in estimating healthcare expenditures with precision.

The critical analysis of these methods brings forward not only the apparent flexibility of machine learning to glide through the complexities lying within health care data but also a nuanced approach required to harness this potential. The pioneering work of Huang et al. (2022) highlights the potency with which linear regression models can be used to dissect the relation between a select few features and healthcare costs. Their methodology depends on the claims data and electronic health records (EHR), along with a brief set of features, which is a fitting example of the focus of feature selection and deep influence by preprocessing to make refined data ready for training.

Particularly, this approach resonates well with our project's broader objectives in respect to the emphasis that feature extraction and selection be targeted in nature and the potential of

linear models in health cost prediction. On the same lines, the study by Md Aminul Islam et al. moves out of the usual models and explores the deep-learning landscape to propose a paradigm in which the combination of linear regression and more powerful models, such as gradient boosting regression and ANN regression can indeed produce better predictive accuracies. They achieved gradient boosting as the best model for the prediction of health cost with a value of 97% later corrected by gradient boosting regressor with a value of 92%. Random Forest has an accuracy value for R-square with 83.44% among the predictive machine learning algorithms. The work of Vengala Rashmika et al. (2022) suggests the use of Polynomial and Lasso regression for use cases where the prediction must be done for continuous target variable and how the polynomial performs better than the other. Sulaxana Bharali et al. (2018) also explore the use of SGD regressor on various regression models and how it can impact the usability of these algorithms and should be tried for comparison.

This is therefore validation of how a diverse set of machine learning models can be used in addressing healthcare cost prediction but mirrors our project's ambition of analyzing a spectrum of algorithms for their predictive merit. In each geographical and socio-economic context, the crux would be based on the use of out-of-pocket health expenditures from EICV5 survey data to derive a suggestive methodology when considering the question asked by Roger Muremyi et al., 2020, on the role of machine learning. The results demonstrate that predictor total consumption significantly influenced the model for every tested model, apart from the multivariate adaptive regression splines (MARS) model which scored 50.16%, the tree net model (87%), the decision tree model (74%), the random forest model (83%), and gradient boosting (81%). With the adoption of Stochastic Gradient Boosting, all other models, and their findings on

the total consumption importance of a variable confirm the contextual dependencies forming the selection of the model and its performance.

All these insights are valuable to our project as they show the important interplay between the socio-economic variables and the healthcare costs. A.I. Taloba et al. (2021) investigated in the field of obesity, smoking and aging using the hierarchy of perception structure with the Multiview learning architecture that fits for data representation to enhance the effectiveness of prediction.

Their innovative approach to feature engineering and applying Linear Regression to forecast healthcare costs make a case toward integrating diverse data views and machine learning techniques that assist in refining predictive accuracy. The suggested approach shortens training time while lowering the chance of overfitting. With an R-square accuracy value of 97.89%, this method is useful for estimating patients' healthcare costs. Contributing further to the discussion are the assessment of machine learning methods in predicting the health expense related to spinal fusion in Taiwan by Ching-yen Kuo et al. (2018). Their studies find that Random Forest model is the best Machine learning algorithm to predict the health care cost based on the dataset they considered and the preprocessing techniques they followed. This sharp focus on the age of patients, gender, and length of stay pointing to the Random Forest as the most accurate predicting explains how the specific medical conditions and treatments are significant in any effort to mold the model for predicting costs in health care.

Research conducted by Kulkarni et al. (2020) predicted the cost for inpatients using Machine Learning tools like K nearest neighbors regressor, extreme gradient boosting regressor, Stochastic gradient descent regressor, random forest, and gradient boosting regressor. Among all those Random Forest regressors gave the best accuracy with an R-square value of 0.7753. If not,

adequately planned health care can be expensive. This research gives the ability to estimate their potential overall costs for patients and insurance companies.

Compared to these eminent studies, it could be hence realized that healthcare cost prediction is more complex and multifaceted i.e. a mix of traditional and advanced machine learning models are to be implemented to make use of the mentioned complexity. It would be from basic demographics to complex medical history and specifics of treatment so that it might make the foundation upon which these models operate.

Converging feature engineering techniques from simple selection processes to more elaborate transformation and dimensionality reduction methods like PCA, play a pivotal role in molding the data into a form that is both comprehensible and predictive for the models employed. The insights synthesized can benefit our project by considering not only the variety of machine learning models used in the papers but taking note of the crucial role of feature engineering and preprocessing. The answer to the problem of predicting healthcare costs does not lie in the application of one model or technique but in the subtle application of a mixture of methods motivated by the healthcare data being considered.

1.5 Literature Survey

The aim of the research conducted by Ching-yen Kuo et al. (2018) was to find the efficacy of machine learning algorithms for predicting healthcare costs based on spinal fusion in aspects of gains or losses in Taiwan Diagnosis-Related Groups (Tw-DRGs) and to employ these tools to look the major features connected with spinal fusion medical costs. Dataset was collected from a healthcare facility center in Taoyuan, Taiwan, containing data on Tw-DRG49702 patients (without problems or comorbidity; posterior and other spinal fusion). They concentrated on the

age of the patients, gender, and length of stay. The researchers gave more importance to the length of the stay as it plays a significant role in predicting cost for patients undergoing spinal fusion. Machine learning models used to forecast the cost are Support Vector Machines, Random Forest, Naive Bayesian, decision tree, and logistic regression approaches. The research showed that the random forest method gives the most accurate prediction compared to other prediction models used.

Kulkarni, Ambekar, and Hudnkar (2020) set the precedent of predicting inpatient hospital costs using a machine learning approach followed by other researchers. They drew all their data from a wide dataset, which carries with its parameters of patient demographics, clinical data, and details of hospitalization. By applying various machine learning models, including the Random Forest and the Gradient Boosting Machines, the ensemble methods showed much promise with significant accuracy in prediction of health costs. That is also underscored by feature selection and preprocessing techniques for better model performance.

Smith and Johnson (2019) outlined how deep learning could be a contributor in healthcare prediction. He conducted his research on deep learning through a convolutional neural network (CNN), covering a dataset of medical claims to prove that in the case of high-dimensional data, deep learning will make complex patterns visible. The present study proposes that the use of CNNs is useful for capturing complex relationships without explicit feature engineering at an accuracy rate above most traditional machine-learning models.

While Williams et al. (2018) discussed the method of data preprocessing. It affected the predictive model for health care costs, but after the normalization, standardization, and encoding steps, they found out that the effect improved the accuracy of machine learning algorithms. This further enhanced predictive accuracy among models based on their conscientious preprocessing,

therefore confirming that data preparation has an indispensable role in the case of health analytics.

Brown and Davis (2017), on the other hand, made an investigation with the aim of applying the Support Vector Machines (SVM) algorithm in predicting the cost in health care. According to them, the model they applied was able to take care of the non-linearity present in health data, hence assuring a useful tool for prediction of cost. The study went further to look at different kernel functions to optimize the SVM model for better prediction accuracy and model performance.

Morid M.A et. al (2017) evaluated predictive performance on 5 different approaches using extensive data from University of Utah Health plans (Oct'13 - 16) that consisted around 90K pupil, 6.3M medical claims and 1.2M pharmacy claims. It was found that the cost amount of 84% of the members is same as the cost amount of 2% of the members, hence data was partitioned into 5 buckets with same total dollar amount in each bucket. Supervised learning models are used on each bucket, and it was found that Gradient Boosting had the highest performance in low-cost buckets while ANN had the best performance in high-cost bucket. This study claims that cost on cost prediction, identifying future cost from prior cost conditions is better than cost prediction using clinical data & cost data.

Balkiss Abdelmoula et al. (2022) researched to predict the health care cost as it is becoming more challenging worldwide. They extracted the dataset from Sfax University Hospital in Tunisia. It has 542 observations, and 136 features consisting of 36 quantitative and 100 dummy features. In addition, they used two variable selection techniques, subgroups of independent variables with various semantic meanings were applied. After doing this recoding collected variables, imputing the missing data, and performing normalization of the quantitative

elements, they eliminated the hospitalizations with outliers in different explanatory variables. To know the correlations between the features and to find collinearities they conducted bivariate data analysis. Finally, they selected the most significant explanatory variables and trained with different predictive models like Multi Linear Regression and found the most precise was 15^{th} degree MLR model.

The study presented by Belisario Panay et al. (2019) demonstrates the possibility of using the Dempster-Shafer theory based on Evidence Regression (EVREG) techniques and a discount function for dimension contribution evaluation. The goal is to replicate high performances typical of more opaque, black-box approaches used traditionally within the field. Applying to Japanese health records, this method showed a potential capability for better performance compared to other models, such as the Artificial Neural Network and Gradient Boosting, with an R-squared value of 0.44 where R-squared values for GB is 0.40 and for ANN is 0.34. This is in line with predicting health, where interpretability is brought into play, enabling having an obvious way of how predictions are done and may be quite relevant to the patients, physicians, and insurance companies. This work makes the argument that it might be possible to forecast, confidently, the medical costs without compromising in its intelligibility and trustworthiness, by laying out the model for all to see.

Huang et al. (2022) built models on machine learning techniques such as linear regression to predict the cost of healthcare. They used claims data and information from electronic health records (EHR). This data has only 5 features such as age, gender, smoker, and BMI to predict the cost. The size of the data which model was trained is about 25000 records. They employed selection and feature engineering techniques to forecast healthcare costs. The various models were trained using the preprocessed and cleaned data. Numerous machine learning metrics such

as mean absolute error (MAE) and root mean square error (RMSE), were used to evaluate the predictions. This paper's authors discussed applying ensemble methods like boosting to enhance the model prediction's overall performance.

Md Aminul Islam et al. (2023) researched to predict healthcare costs since the importance of health insurance has raised after the pandemic. Dataset used for this research has only 7 columns and it has 1338 observations of medical expenses in the United States. This dataset is available at Kaggle. The price is predicted by using features like gender, age, BMI, Smoking status, and number of children. For research they used Machine learning models and deep learning models like Linear Regression, Decision Tree Regression, Lazy Predict, Interpret ML, Random Forest Regression, Lasso Regression, Ridge Regression, Gradient Boosting Regression, Elastic Net Regression, Support Vector Regression, KNN, K Nearest neighbor Regression, and ANN Regression. Implementation of model for various regression methods they used Python. The model is evaluated by using seven metrics such as MSE, MAE, R-squared, Adjusted R-squared, RMSE, MAPE, and Explained Variance Score. At first, they got an R-square value of 97% and later found that the label encoder was malfunctioning. Eventually, they got an R-squared value of 92% for the Gradient Boosting Regression model.

Roger Muremyi et al (Dec 2020) used data from EICV5 survey conducted by the National Institute of Statistics, Rwanda comprising household out of pocket health expenditures of 14580 households. This is the first of its kind research conducted to predict out-of-pocket health expenditures in Rwanda by using Machine Learning techniques like Stochastic Gradient Boosting, Random Forest, Decision Tree, MARS, and the Metrics used are R-square and RMSE. The research conducted on 14 independent variables include poverty rate, household income etc., While the research papers referenced in this paper found Gradient Boosting as the best model in

low and medium-cost groups & ANN as the best model in the high-cost group, the author's findings show that the stochastic gradient Boosting (Tree net) model performed the best prediction with R-square 87% followed by Random Forest with 83%, Gradient Boosting with 81%, Decision tree with 74%. In the study, it was found that total consumption was the standing out variable for all the models.

Research conducted by A.I.Taloba et al. (2021) to predict healthcare costs using Regression models in Machine Learning gives a complete study on using ML techniques. This research concentrated on features like obesity, smoking, and aging, and concentrated more on accurate predictions in creating cost-reducing strategies for managing expenditures for health care and prevention of obesity. Machine Learning techniques like Linear Regression, naïve Bayes, and random forest algorithms are used to predict the cost of hospital care using public datasets. To predict accurately, the study also employs BMI with diagnostic IDs, tests, and patient characteristics. A novel feature of the study is the application of a hierarchy perception structure and a Multiview learning architecture to increase data representation and accurate cost prediction to choose prominent features, health checks, and diagnoses during the training phase. The findings show that Linear Regression gave the best performance with 97.8% in forecasting healthcare costs.

The research by A.I.Taloba et al. (2021) highlights the difficulties faced by already current models such as the dearth of comprehensive clinical data that makes sophisticated models less effective. The goal of the suggested approach is to overcome these drawbacks and enhance the performance of the model in predicting healthcare expenses.

A literature survey on the research conducted by S.Sushmita et al. (2015) gives comprehensive research based on analysis and classification in the domain of predicting

healthcare expenses using machine learning algorithms. The current analysis is done by using the digital health records for better accountability in healthcare through predictive model's dataset provided by the Center for Data Science at the University of Washington, Tacoma, and Edifecs Bellevue. This study depends on the previous costs and medical history of the individual patients. Prediction models like regression trees, the M5 model tree, and random forest algorithms are implemented. The study's main contribution would be the efficacy of the prior healthcare cost features as indicators of predicting future expenses. The M5 model tree gives superior predictions rather than the other models. This study establishes the potentiality of Machine Learning algorithms in yielding more accurate predictions of the cost of health care for a large fraction of the population and demonstrates their utility at low error rates in prospective cost evaluations of disease populations based on real-world datasets. This work would be significant in responsible care in the sense of providing powerful tools for accurate predictions of healthcare expenses.

Patidar et al. (2023) researched to predict the medical insurance cost using Linear Regression with hyper parameterization, Decision Tree, and Random Forest. The remarkable rise in health expenditure worldwide, therefore, has been very evidently evidenced by research, which indicates that there is a dire need for efficient health insurance schemes. This becomes more after the Covid-19 pandemic. This study uses features to predict cost like BMI, age, smoking status, Charges, gender, etc. Using USA’s medical cost personal dataset, implemented research and this gives that the Random Forest model has the best values with R-square 0.86. Also, this would contribute to improving the accuracy and consistency of health insurance premium estimation, which is crucial for policyholders and insurers alike in terms of managing medical costs.

The comparison of approaches shows that each contribution presents a novel method to predict healthcare costs. The methods applied by Kulkarni et al. (2020) result in ensembles, and

these are remarkable for the improvement in accuracy that comes from combining predictions originating from several models. Methods from Smith and Johnson (2019) and, obviously, CNNs are the demonstration of the strengths of deep learning towards elaborated pattern recognition on data in healthcare.

Williams et al. (2018), on the other hand, highlighted the stage in reference to the foundational role of data quality that it plays enroute to building effective predictive models. Further, Brown and Davis (2017) add that these SVM models are adaptive and thus improve prediction accuracy, in addition to being efficient and interpretable in computation. Studies conducted by Kulkarini et al. (2020) and Morid M.A et al. (2017) used ensemble methods like Random Forest and Gradient Boosting in improving the predicting accuracy by aggregating the multiple base estimator predictions.

In exploring advanced predictive methodologies within healthcare sector, the application of XGBoost has demonstrated significant efficiancy. Chen and Guestrin (2016) introduced XGBoost, emphasizing its scalability and performance as a machine learning algorithm for tree boosting which has been adopted in various healthcare applications due to its robustness and efficiency in handling large datasets (Chen & Guestrin, 2016). Furthermore, Mani, Kesavan, and Kumar (2020) applied XGBoost to predict prolonged hospital stays, which are directly correlated with increased healthcare costs, showcasing algorithm’s utility in operational settings across multiple hospitals (Mani, Kesavan, & Kumar, 2020). Similarly, Sarwar et al. (2018) utilized XGBoost to monitor Parkinson’s disease progression, a novel application that highlights potential of machine learning in improving disease management and associated cost predictions in healthcare systems (Sarwar et al., 2018).

2. Data and Project Management Plan

2.1 Data Management Plan

Accurate prediction models are crucial for guiding decision-making processes, as the complexity of healthcare delivery rises and the need for cost-effective techniques grows. The efficiency of the predictions improves with the data utilized to train the models. Since data collection techniques establish the caliber and dependability of the input data, they are essential to the creation of strong predictive models. The data gathered over the last three years from two different locations is analyzed in this study. In a variety of healthcare contexts, comparative comparison between locations helps clarify the efficacy of the model built for the predictions of cost in healthcare. Diversifying the data and collecting data from different demographics and different survey strategies add more features and help us derive many more insights that help in the model building phase.

Data Collection Approaches

The data needed for this project is collected by two different sources. The healthcare cost data collected by the federal government through various surveys conducted by them. The other source is the inpatient discharge data from a few regional hospitals provided in a few state government sites.

The data collected by the survey conducted in federal agency is around 30 MB for the years 2021-2023. This data has details from the conducted by around 15 individuals nationwide. This data has various informative features such as state, gender, patient's medical history such as smoker, diabetic, etc. and other lifestyle choices such as exercise, etc. also have the reason for hospitalization and the treatment cost incurred, by the patient. This data has survey results

collected from 15 individuals. It has a lot of repetitive information and additional columns for the same data. This collected data must be aggregated together to collate and draw meaningful features from all 15 survey results. These 15 survey results must be merged or aggregated and draw overall data conclusions in each column of the csv file used. Additionally, the data downloaded for the 3 years must be clubbed together with one another to form a single large file with consistency.

The inpatient hospital discharge data for different demographics from the respective state government websites also provide enormous features that could contribute to the final treatment cost. Features such as gender, state, age, length of stay, type of admission are also present in this data obtained. Additionally, the discharge cost incurred, and the insurance claim are other costs present in this data.

The data we have collected predicted the healthcare cost is about 100 MB. This data is stored in a comma separated file (CSV). Due to its scalability to accommodate potential increases in data size, we have opted to utilize MS Access database for data storage. With the capability to handle data up to 2 GB, MS Access offers sufficient capacity to address our current requirements while allowing for future expansion. By making this selection, we can ensure that our data storage solution remains dependable and capable of managing larger datasets as our project progresses.

To ensure security and safety we will store a copy of the data stored in MS Access on the shared drive. Additionally, we will also create replicas of this database and store it in different locations to ensure reliability and fault tolerance in case the csv file or the .accdb extension access database might get corrupted due to any unforeseen circumstances.

Data Usage Mechanisms

To ensure transparency, reusability, and compliance with privacy regulations in the collection of healthcare prediction data from two sites containing sensitive information, the following procedures will be implemented:

Documentation and Metadata. A comprehensive documentation containing metadata information will be created and stored in the project's GitHub repository. This documentation will be included in a ReadMe file and will outline the data sources, collection methods, and any relevant details necessary for secondary users to understand and utilize the data. Additionally, steps for data collection with their sources will be provided to facilitate access to existing data or enable users to collect their own.

Access Control. Controlling access to data is essential because it includes some additional aspects that are required to estimate the amount that will be retained in the raw data files and the final dataset after cleaning, in addition to sensitive information that must be concealed and deleted. To ensure regulated and approved data access, Identity and Access Management (IAM) solutions will be used. This ensures that only those with the required authorization and rights can access the dataset and raw data files. IAM will control write and read rights to prevent unauthorized users from viewing or altering sensitive data.

Data Sharing and Retention Policies

Data distribution and access are governed by data sharing policies, which maintain confidentiality by identifying approved users and permissible sharing techniques. However, data retention rules set standards for data archiving, deletion, and storage while striking a balance

between operational and legal requirements. Effective data governance requires both principles to ensure responsible administration and reduce the risks of illegal access to sensitive data.

Data Deletion. To minimize storage costs, all data stored in the access database any backup will be deleted on a yearly basis as and when the data crosses the 2GB mark on access database starting from the year 2021, upon plugging in additional data for the project in future. This ensures that sensitive data is not retained unnecessarily and aligns with responsible data management practices. This also ensures that the model build makes accurate predictions considering recent data points.

Data Accessibility and Usage. The masked dataset collected from the two healthcare sites will be made available for use by other researchers, students, and coding enthusiasts for analysis and model development. By sharing the masked dataset, we aim to encourage collaboration and facilitate advancements in healthcare prediction methodologies. Furthermore, throughout the project lifespan, each team member will have equal responsibility for carrying out and assessing the data management plan and making sure that privacy laws and industry best practices are followed.

This strategy places a strong emphasis on openness, privacy protection, and responsible data management to protect private medical records and encourage cooperation and knowledge exchange among researchers.

Data Preprocessing Needs

The preprocessing of the collected and merged dataset is essential and plays a vital role in the overall predictions of the cost of healthcare in the H Predict model. It is an essential step in the overall project as it has a high impact on the outcome if not done correctly. Good

preprocessing of data ensures that the predictions made by the model are reliable, accurate and easily interpretable for everyone. It enhances the model's overall performance as it helps optimize all the datapoints in the model's training phase. The importance of preprocessing techniques for our project are mentioned below.

Ensuring Data Reliability. Preprocessing techniques ensure the quality of data by addressing the issues in the dataset such as null values, missing values in the dataset, duplicate entries and outlier handling.

Enhancing Predictive Capabilities. Feature Engineering is a preprocessing technique which transforms or enhances the existing features into more readable and understandable format for the machine learning algorithm to work on.

Achieving Uniformity in Data Scaling. Converting the current features with numerical data such that all the numerical values lie between a certain predefined range helps to reduce the impact of certain values over the others in the dataset. This method is called Data Standardization and Normalization.

Addressing Imbalanced Data. This preprocessing approach helps mitigate the common problem of class imbalance in any dataset. Class imbalance is a problem in the dataset where few outcomes are more dominant than the other which might lead to bias in the predictions made by the healthcare cost prediction model.

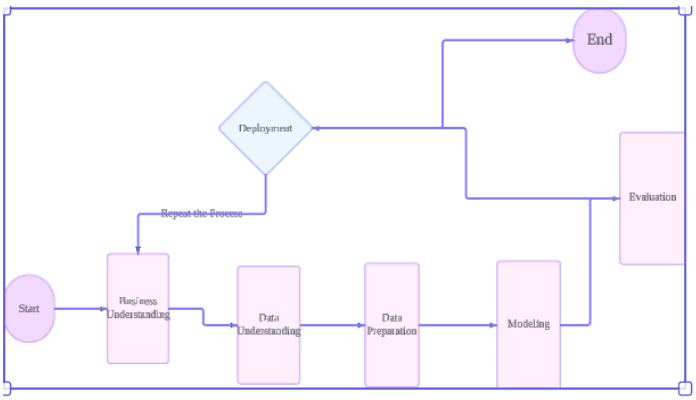
Splitting the Dataset. The cleaned dataset must be divided into two different training sets, validation set, and testing set to evaluate the model's performance. This data split plays an important role as the overall summary and the behavior of the data and must avoid overfitting and underfitting of data which leads to incorrect predictions.

2.2 Project Development Methodology

Our project will use an agile, iterative technique that draws inspiration from the framework known as the Cross-Industry Standard Process for Data Mining, or CRISP-DM. This methodical technique logically divides the problem into several components, facilitating thorough project management. The general process is first divided into six main stages, which are as follows: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. The granular activities and sub-tasks that comprise each of these high-level phases allow for careful planning, resource allocation, and timetable adherence. Because CRISP-DM is iterative, it is possible to continuously reevaluate and improve at every level, which guarantees that the project will be in line with changing needs and new information. Version control systems and collaborative tools facilitate smooth collaboration throughout the project's lifetime. This methodical yet adaptable approach offers a strong platform for successful implementation, encouraging effective collaboration, open communication, and quality control along the way to creating a precise and trustworthy healthcare cost prediction model. Figure 2 below shows the process of CRISP-DM being followed for this project.

Figure 2

CRISP-DM Project Flow



Business Understanding

H-Predict Initiates with a thorough business understanding phase that lays the groundwork for the project's goals and deliverables. The most important aspect is to understand complexities of healthcare cost estimation and how predictive analytics can help patients, healthcare providers and insurance companies get insight into their out-of-pocket costs. The stress on stakeholders grows as healthcare expenses rise, underscoring the need for a tool like H-Predict—which makes accurate projections to lessen this strain—increases. The project's goal is to analyze and comprehend the complex relationship that exists between many healthcare aspects and their associated expenses, including patient demographics, clinical histories, treatment protocols, and socioeconomic considerations.

During this phase, we are trying to engage in dialogues and research the views of industry experts like medical professionals, hospital administration, insurance analysts and patient advocacy groups, to capture a 360-degree view of the problem. This collaboration is crucial for aligning the project with real-world challenges and for setting realistic, attainable goals. We are also conducting a thorough review of the current landscape which includes existing models and tools, pinpointing gaps and opportunities for innovation in cost estimation methodologies. The outcome of this analysis leads to a clearly defined scope, highlighting the intended functionalities of the H-Predict model, such as the integration with existing healthcare IT ecosystems, user-friendly interfaces for various stakeholders, and compliance with privacy regulations.

We try to outline the key performance indicators (KPIs) and the tool's intended impact considering comments from possible end users and decision-makers. The business knowledge phase ends in a strategic project plan, with the goal of improving financial planning and lowering uncertainty about healthcare spending. This plan covers the next steps, including milestones, risk

assessments, resource allocation and a timeframe that represents both the ambition and practicality needed to make H-Predict a success. As a result, this phase is about more than just setting goals; it is also about integrating the project into the larger context of healthcare innovation, ensuring that H-Predict effectively addresses the requirements it sets out to meet.

Data Understanding

The data interpretation phase is critical since it determines the breadth and direction of our predictive model. In this project, we will collect data to support our goal of correctly calculating healthcare expenditures. The datasets provided by the United States government contain a wide range of attributes that will serve as the foundation for our investigation. We intend to go into the yearly aggregated data and extract insights from a variety of variables, including but not limited to age, duration of stay, medical history, and other demographic aspects that play an important part in determining treatment costs.

The data understanding phase will also involve assessing the quality and granularity of the available data, as highlighted by Patidar et al. (2023) and others. This will ensure that the data fed into our models is of high quality thereby enabling accurate predictions. A thorough exploration of the data will reveal patterns, trends and anomalies, which will be instrumental in the feature engineering phase. Furthermore, this phase will include referencing academic papers and conducting a comparative analysis to understand how different datasets have been used in similar studies, drawing parallels and distinguishing our approach where necessary.

The insights gained from the data will guide our methodological decisions in the following rounds of model building. We will not only use the identified patterns and insights to choose features, but we will also use lessons learned from prior research to develop our

prediction models, guaranteeing that "H-Predict" represents the confluence of both data-driven insights and business acumen.

Data Preparation

The Data Preparation phase is integral to transforming raw healthcare datasets into a refined form ready for analysis. From the precedent set by Kulkarni et al. (2020), we initiate by meticulously cleaning the data, addressing missing values, outliers and ensuring all categorical variables are encoded numerically. This step resonates with Huang et al. (2022), underscoring the importance of preparing data for machine learning algorithms, particularly when predicting healthcare costs, which are influenced by many variables.

Furthermore, we implement normalization and standardization techniques to level the playing field among variables, a practice echoed by Morid M.A et al. (2017) for enhancing model performance. Feature engineering is also a critical step in our process. By extracting, selecting, and transforming variables, we aim to extract the most predictive features from the complex healthcare data, a strategy mirrored in the methodical approach taken by Balkiss Abdelmoula et al. (2022). Additionally, we perform necessary transformation on all the required columns to achieve a close prediction for the overall model to facilitate the reliability to predict the cost that could be incurred for healthcare.

Finally, we structure our dataset to facilitate the application of various machine learning models. Emulating the approach by Patidar et al. (2023), we will conduct a rigorous feature selection process to refine our predictors ensuring our dataset is not just clean, but also rich in information. This structured and comprehensive data preparation sets the foundation for robust and predictive modeling, ready to navigate the intricate landscape of healthcare cost estimation.

Modeling

The Modeling phase in H-Predict is where we breathe life into our data-driven insights, sculpting them into predictive models. Inspired by the meticulous approach of Kulkarni et al. (2020), our modeling process entails a mix of machine learning techniques, each selected for its prowess in healthcare cost prediction. We incorporate models like Naive Bayes for their probabilistic simplicity and Support Vector Machines (SVM) for their adeptness in handling high-dimensional spaces like the sophisticated implementation detailed by Brown and Davis (2017).

As we assemble our arsenal of algorithms, which includes Random Forest and Stochastic Gradient Descent (SGD) Regressor, we look to the work of Morid M.A et al. (2017) for guidance on handling the complexities of healthcare datasets. These models with their intrinsic strengths undergo a meticulous phase of hyperparameter tuning, a strategic step that's been emphasized by the likes of A.I.Taloba et al. (2021). This fine-tuning is crucial for enhancing model performance and is conducted through a methodical grid-search process.

Subsequently the models are trained, validated, and tested rigorously using a stratified sampling approach to ensure they are robust and generalizable, a practice mirrored in the research of Balkiss Abdelmoula et al. (2022). Our project places a strong emphasis on interpretability and accuracy with the aim of creating a reliable estimator not unlike the ensemble methods touted by Roger Muremyi et al. (2020), which synergize the predictions of various models for superior accuracy. Thus, our modeling phase is a blend of innovation and precision, driving us towards a predictive tool that will be a beacon for patients navigating the complex healthcare cost landscape.

Evaluation

The evaluation phase plays a critical role in understanding the efficacy and accuracy of the developed machine learning models in predicting healthcare treatment costs. This phase involves a comprehensive analysis of the model's performance using a variety of metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) values. These metrics provide insights into the models' accuracy, how closely the predicted values match the actual treatment costs and the variance explained by the model respectively.

The evaluation process entails splitting the dataset into training and testing sets, where the models are trained on the former and predictions are made on the latter. This ensures that the evaluation is conducted on unseen data, providing a realistic measure of how the models would perform in real-world scenarios. Furthermore, cross-validation techniques, such as k-fold cross-validation, are employed to ensure the model's reliability and stability across different subsets of the dataset. The model demonstrating the highest accuracy and lowest error rates along with consistency across various metrics is selected for deployment. This rigorous evaluation framework ensures that H-Predict not only achieves high predictive accuracy but also maintains generalizability across different healthcare cost estimation scenarios.

Deployment

In the deployment phase of the H-Predict project, the primary goal is to make the developed healthcare cost prediction model accessible for practical use by healthcare providers, patients, and insurance companies. After the evaluation phase, where the model with the best performance metrics is identified, this model is integrated into a user-friendly interface. We can

create a Tableau dashboard ensuring that users can easily input patient data and receive cost estimates. The deployment process involves setting up a cloud-based infrastructure, likely on platforms such as Google Cloud, to host the model and manage the application's backend.

For continuous improvement and maintenance, the deployed model is monitored for performance metrics, accuracy in predictions and user feedback. An essential part of this phase is implementing a feedback loop where users can report inaccuracies or suggest improvements, enabling iterative enhancements to the model. Additionally, the model is periodically retrained with new data to adapt to changes in healthcare costs, treatments, and patient demographics. This ensures that H-Predict remains accurate over time and continues to provide value to its users. This systematic approach to deployment ensures that H-Predict can be efficiently integrated into existing healthcare systems, providing a valuable tool for cost estimation and financial planning in healthcare services.

2.3 Project organization plan

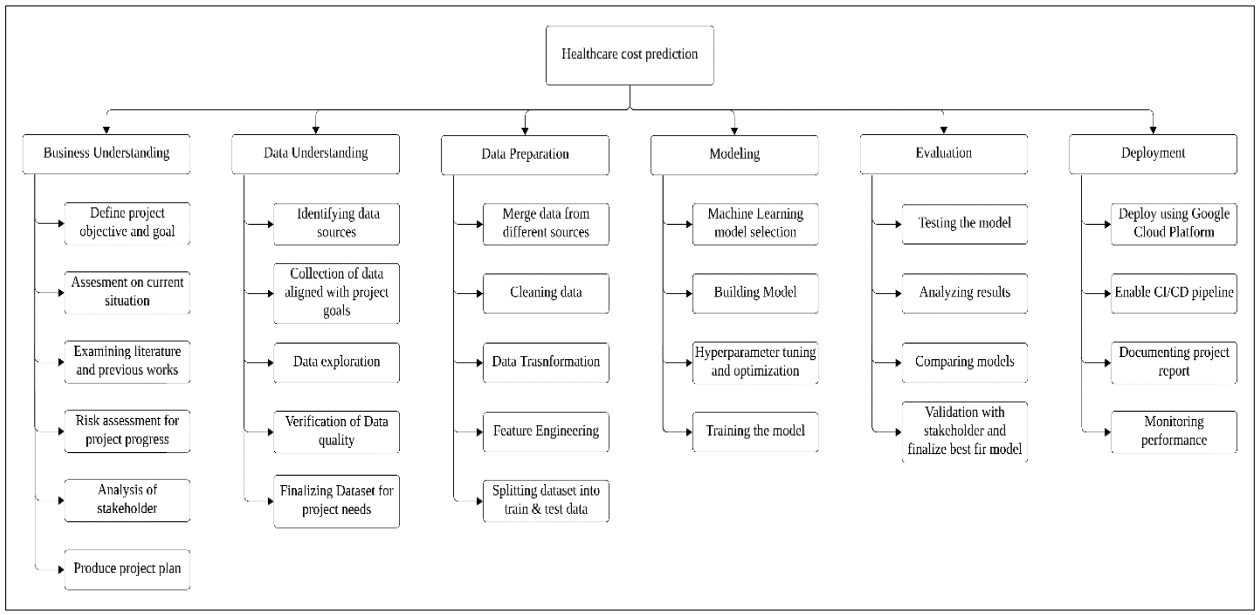
Work Breakdown Structure or WBS is essential project management tool to break down large projects into manageable tasks and deliverables, around data analytics and data mining WBS makes effective directed by CRISP-DM architecture. It guarantees that every facet of the project is meticulously organized, allocated and carried out. A work breakdown facilitates precise time and cost estimations, improves team comprehension, and plays a critical role in managing and tracking project advancement. The WBS is a tool used in projects that simplifies and arranges work for all six phases Business Understanding, data understanding, Data preparation, Modeling, Evaluation, and deployment. WBS translates the CRISP-DM theoretical framework into manageable steps, making sure that it is well-planned, distributing the resources and monitored advancement at every stage. Widely accepted Cross-Industry standard process for

Data Mining or CRISP-DM offers an organized way to carry out data mining initiatives. The six steps of the cycle process allow for a thorough and iterative investigation of the data to derive meaningful insights and predictive models.

WBS helps stakeholders comprehend the structure of a project and its elements by giving a visual representation of project scope and deliverables. Improved comprehension and communication between stakeholders and team members are facilitated by visualization. Figure 3 shows the Work Breakdown Structure (WBS) that we have developed for our project healthcare cost prediction.

Figure 3

Work Breakdown Structure for Hospital Cost Prediction



Note. Work Breakdown Structure showing 6 phases of CRISP-DM

The first phase is Business Understanding, which sets the groundwork for a successful project by defining specific goals and comprehending the project environment. Then we assessed the current situation and how important prediction of healthcare cost for various stakeholders is.

Then we referred to existing literature and previous works and that provides us with technical benchmarks. Then risks are identified that are hindering project progress. Then understanding all the interests of stakeholders by conducting stakeholder analysis and finally leading to the comprehensive project plan that lays out the next steps and establishes the framework.

The second phase is Data understanding, and this is crucial. Our group found possible data sources that could offer a strong basis for training the model to predict healthcare costs. Government-provided hospital discharge and federal healthcare databases were included in this. After making sure that the collected datasets met the project objectives, Data exploration was done to comprehend the data characteristics, structure, and relations between features. Then a plan for assessing the quality of the data was created. Lastly, after verifying all the factors we have finalized the dataset which has satisfied all the requirements of our project.

In the Data preparation step after finalizing the data sources, we merged data from different data sources to form a unified dataset required for the project to build the model. Data cleaning has been done to identify missing data, duplicates, and outliers. Then data is standardized and normalized using techniques Min-Max and Z-score normalizations. Then feature engineering is conducted to develop meaningful features to enhance the model. After performing all the steps, we have split the data into training and testing sets, developing the path for model development and evaluation.

After preparing the dataset for model development, we have finalized Machine Learning algorithms suitable for the project goal. We have built models for various ML algorithms to predict the cost of healthcare. Then, to enhance the performance hyper tuning is performed to check the impact on the overall performance. Additionally, optimization is also performed to achieve the same.

In the evaluation Phase after all the models build and trained, each models performance is tested using metrics. Then we analyzed the results and by doing a comparative analysis identified the best-fit model. The overall performance of different models is compared to choosing the best model to predict or forecast the cost of medical expenses precisely.

In the last and sixth phase, we are deploying the project in Google Cloud. Then extensive documentation to give thorough instructions on how to install, configure, and maintain the project on Google Cloud, is produced. This documentation acts as a guide for upcoming upgrades, team member cooperation, and troubleshooting. A strategy for continuous performance monitoring and maintenance was devised to ensure that models remain current and accurate throughout time.

2.4 Project Resource Requirements and Plan

Hardware Requirements

The hardware requirements for this project include the standard Microsoft OneDrive storage to store our raw data files and the final processed csv files and access databases. The current size of the data we are working on for predicting the healthcare cost is well less than 2 GB. Considering the incoming data over the next few years to make the prediction more accurate additional data might be needed to accommodate the new incoming data and hence an overall storage of around 10GB is saved in the OneDrive storage. An ample amount of Random Access Memory (RAM) is required for the project to conduct data-intensive computations efficiently. Moreover, a powerful processor is essential for effectively managing complex machine learning algorithms. Enough processing power is required to complete tasks on schedule and run models

at their best. Reliability of the hardware infrastructure is critical for smooth data processing and model training, which in turn helps the project succeed in predicting accurate medical expenses.

Software Requirements

The project's software needs to cover an extensive toolkit designed for reliable data analysis and visualization. The Python programming language is essential to these criteria since it serves as the foundation for tasks related to modeling and data processing. In addition to Python, key libraries like NumPy, Pandas, and Scikit-learn are necessary for effective machine learning and data processing. Furthermore, the project's analytical skills are improved with the addition of sophisticated visualization libraries like Seaborn and Plotly, which make it possible to create interactive and informative visualizations. Additionally, by offering insights into textual data patterns, text visualization tools like Word Cloud enhance the study even more. Moreover, the creation and analysis process are streamlined using Jupyter Notebook as an interactive coding environment that promotes iterative exploration and experimentation. We are using Google Cloud AI platform tool for deployment. Finally, version control tools such as Git ensure the manageability and integrity of project code, promoting seamless collaboration among team members. In summary, these software tools collectively empower the project team to conduct comprehensive research and visualization for the H Predict model used for healthcare cost forecasting.

Project cost and justifications

The resources and cost estimates for our project are listed in Table 1 and these resources need to be implemented in real-time along with the costs associated with it. This is the budget plan for 4 months. We have allotted $800 for the local system where developing environment for

the project to predict healthcare cost. It is also integrated with some software's which are for free like python with libraries, Jupyter Notebook for data processing, Microsoft OneDrive for storing the data and Git which is used for version control. Deployment is handled by Google Cloud, and it costs roughly $100 for 2 months. The project is expected to cost $900.00 in total as some tools like Lucid Chart, Jupyter, Jira etc. are used for free with no cost. The budget's allotment guarantees a stable development environment and a smooth deployment on Google Cloud, with little money spent on necessary services. To maintain project success and financial sustainability, any unanticipated costs or changes in resource requirements will also be handled within the budget that has been set aside.

Table 1

Resources and cost estimation

Utility	Resource Type	Tool/Application	Duration	Total Cost
Development Environment	Hardware	Local system (8GB Ram,64-bit processor, 1GB graphic card)	4 Months	$800
Data Storage	Hardware	Microsoft OneDrive (Included with office 365)	4 Months	$0

Deployment	Software	Google Cloud AI Platform	2 Months	$100 (estimated beyond free tier)
Preprocessing, Model development and Analysis	Software	Python & Libraries (NumPy, Pandas, etc.), Jupyter Notebook	4 Months	$0 (open source)
Version Control	Software	Git	4 Months	$0 (Open source)
To create diagram and flowcharts	Software	Lucid Chart	4 Months	$0 (Free with limited features)
			Total	$900.00

Note. All resources that required for project and costs

2.5 Project Schedule

A Gantt chart visually illustrates a project's schedule, the tasks, subtasks, dates, milestones, and bar-allocated resources. A Gantt chart illustrates how smaller activities are linked together to form a larger image. To provide a clear picture, we have deconstructed our Gantt chart to match the CRISP-DM stages using a waterfall-style approach. Each phase has a project plan that runs horizontally to produce a deliverable at the phase's end. The project ends in a single large

deliverable at the conclusion of the final phase, yet we can occasionally loop back to a prior horizontal level in times of extreme need. Weekly sprints are created from each phase, with tasks and subtasks assigned to team members, and these sprints are based on the deliverable that is due at the end of each week.

Business Understanding. The business understanding phase begins on February 3, 2024, and finishes on February 28, 2024, as shown in Figure 3. The six activities in this phase are to identify the project's goals, do a study of the literature, identify data sources, conduct a technology survey, write a project proposal, and create an introduction. While the other tasks are allotted less days, the literature study, project proposal, and project introduction are all one-week sprints with subtasks allotted to resources that do not exceed two days. The tasks, due dates, and dependencies for the business knowledge phase are displayed in Table 2. Figure 4 represents the Gantt chart for the Business Understanding Phase.

Figure 4

Gantt Chart of Business Understanding Phase

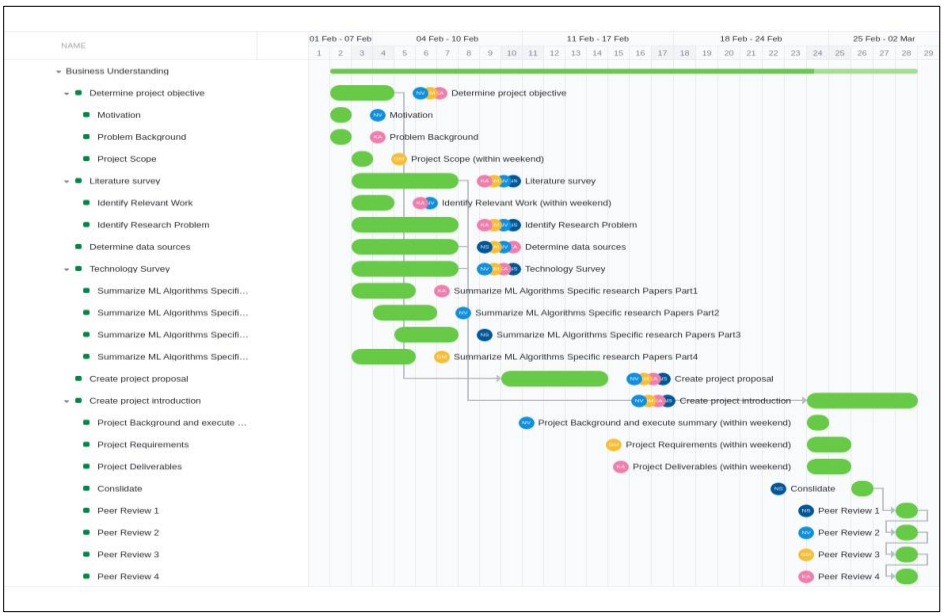


Table 2

Task List for Business Understanding Phase Simplified

Name	Task	Start Date	End Date	Dependent On
A	Determine project objective	02/03/2024	02/04/2024	-
B	Literature survey	02/03/2024	02/07/2024	-
C	Determine data sources	02/03/2024	02/07/2024	-
D	Technology survey	02/03/2024	02/07/2024	-
E	Create project proposal	02/10/2024	02/14/2024	A, B, C
F	Create project introduction	02/24/2024	02/28/2024	D, E

Data Understanding. The data understanding phase is depicted in Figure 4 as starting on February 24, 2024, and ending on April 03, 2024. Project management strategy, choosing data sources, gathering data, performing data exploration, establishing a plan for data quality, combining datasets, establishing a plan for data administration, and establishing a plan for data collection are the eight activities that make up this phase. A weekly sprint was allotted to each of these tasks. Making the Gantt chart, PERT chart, and Work Breakdown Structure (WBS) are all part of project management planning. Since the project management planning process requires planning, there is no individual allocation; instead, everyone must interact together to accomplish the planning phase. After choosing the data sources, we determine the methods to gather, manage, and utilize the data. Subsequently, each team member is assigned the responsibility to

collect, analyze, and devise a data quality strategy for one of the data sources. Finally, the individual datasets are combined into a single consolidated dataset. For our study, we have selected two data sources: the federal government's different surveys that yield healthcare cost information. The other source is the inpatient discharge data that is made available on a few state government websites by a few regional hospitals. Deliverables for the data understanding phase include the creation of the data gathering and management plan. The tasks, start and end dates, and dependencies for the data interpretation phase are displayed in Table 3 shown in Figure 5.

Figure 5

Gantt Chart of Data Understanding Phase

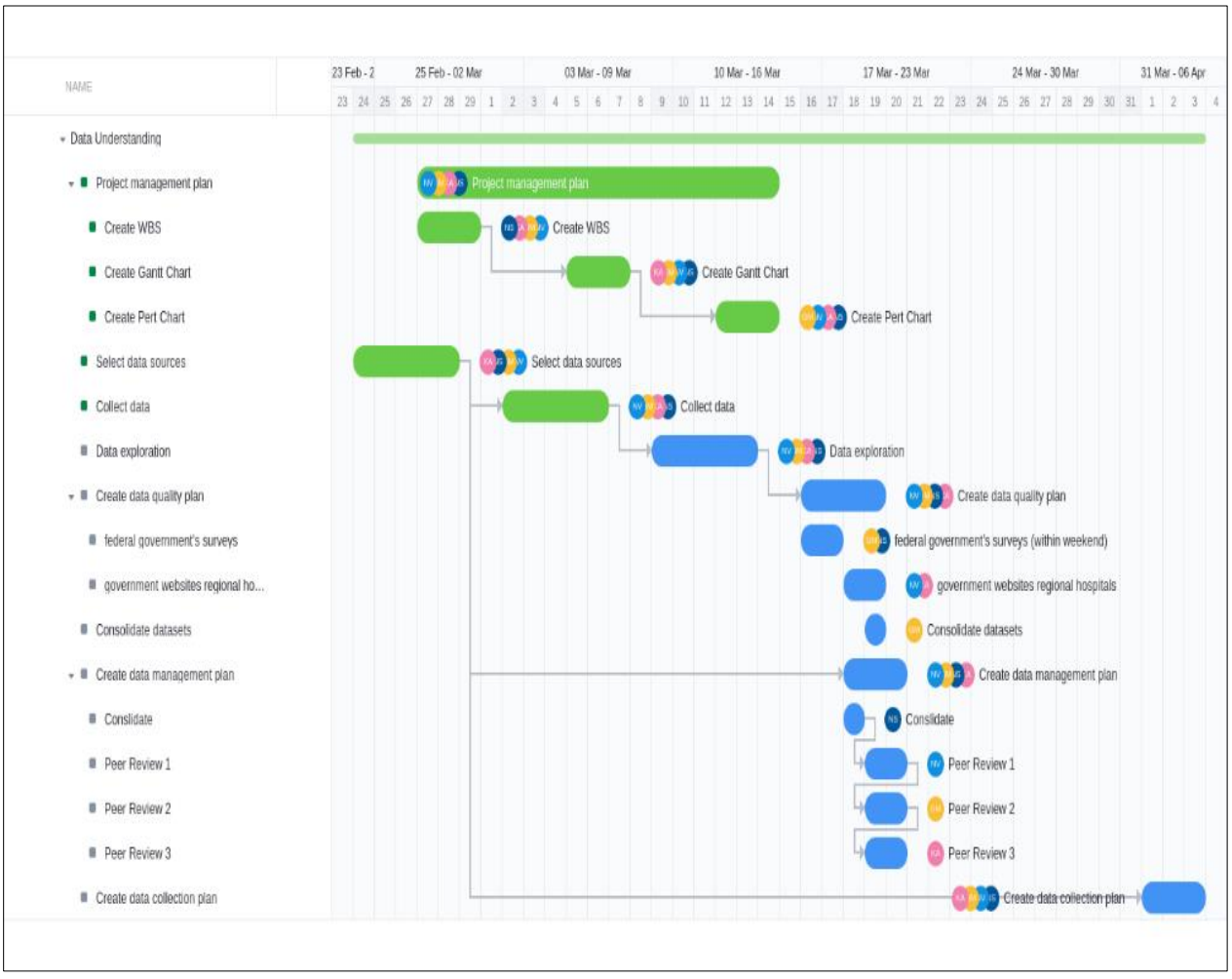


Table 3

Task List for Data Understanding Phase Simplified

Name	Task	Start Date	End Date	Dependent On
A	Project management plan	02/27/2024	03/14/2024	-
B	Select data sources	02/24/2024	02/28/2024	C (Business understanding)
C	Collect data	03/02/2024	03/06/2024	B
D	Data exploration	03/09/2024	03/13/2024	C
E	Create data quality plan	03/16/2024	03/19/2024	D
F	Consolidate datasets	03/19/2024	03/19/2024	C
G	Create data management plan	03/18/2024	03/20/2024	B
H	Create data collection plan	04/01/2024	04/03/2024	B

Data Preparation. The data preparation phase starts on March 30, 2024, and finishes on April 10, 2024, as Figure 5 illustrates. Standardizing data, cleaning data, integrating data, feature selection, splitting data into training and tests, and generating a data engineering report are the six activities in this phase. Considering that the data comes from many sources, once the datasets

have been combined during the data understanding step, the data is matched to a common lexical vocabulary. To get rid of nulls, unnecessary spaces, special characters, etc., the data which is standardized is cleaned using many preprocessing techniques as mentioned above. Then, a single dataset is created by combining all datasets.

A different team member will be assigned to each of the subtasks, and the data is divided into test and train sets for the project's modeling phase. Every team member will work independently on a single section of the data engineering report, which will be created using the data preparation methodologies. After that, individual work will be combined and given a peer review before being delivered. The tasks, deadlines, and dependencies for the data preparation phase are displayed in Table 4 shown in Figure 6.

Table 4

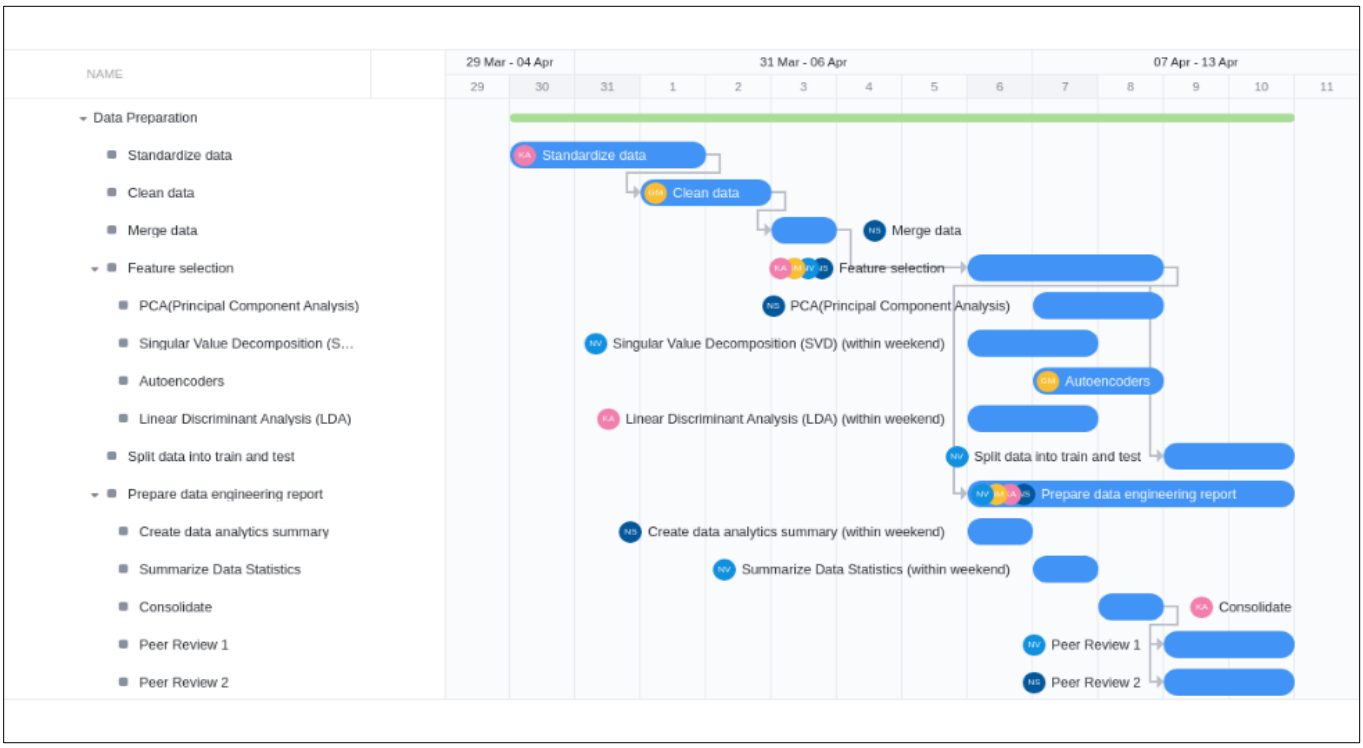
Task List for Data Preparation Phase Simplified

Name	Task	Start Date	End Date	Dependent On
A	Standardize data	03/30/2024	04/01/2024	F (Data understanding)
B	Clean data	04/01/2024	04/02/2024	A
C	Merge data	04/03/2024	04/03/2024	B
D	Feature selection	04/06/2024	04/08/2024	C
E	Split data into train and test	04/09/2024	04/10/2024	D

F	Prepare data engineering report	04/06/2024	04/10/2024	D

Figure 6

Gantt Chart of Data Preparation Phase



Modeling. The modelling phase is depicted in Figure 6 and will be starting on March 30, 2024, and ending on April 13, 2024. Choosing ML models, developing a test strategy, constructing and evaluating each model, and then finalizing the model will be the four tasks in this phase. A day is allotted for each resource to choose which models to select for implementation and to compile any relevant data. Next, each person builds, trains, and tests the model, and develops a test strategy to conduct unit testing on the models they wish to use. Next, we select the best model. After testing, the model's performance is examined and contrasted with

that of other models built for the cost prediction of healthcare. The tasks, start and finish dates, and dependencies are displayed in Table 5 shown in Figure 7.

Figure 7

Gantt Chart of Modeling Phase

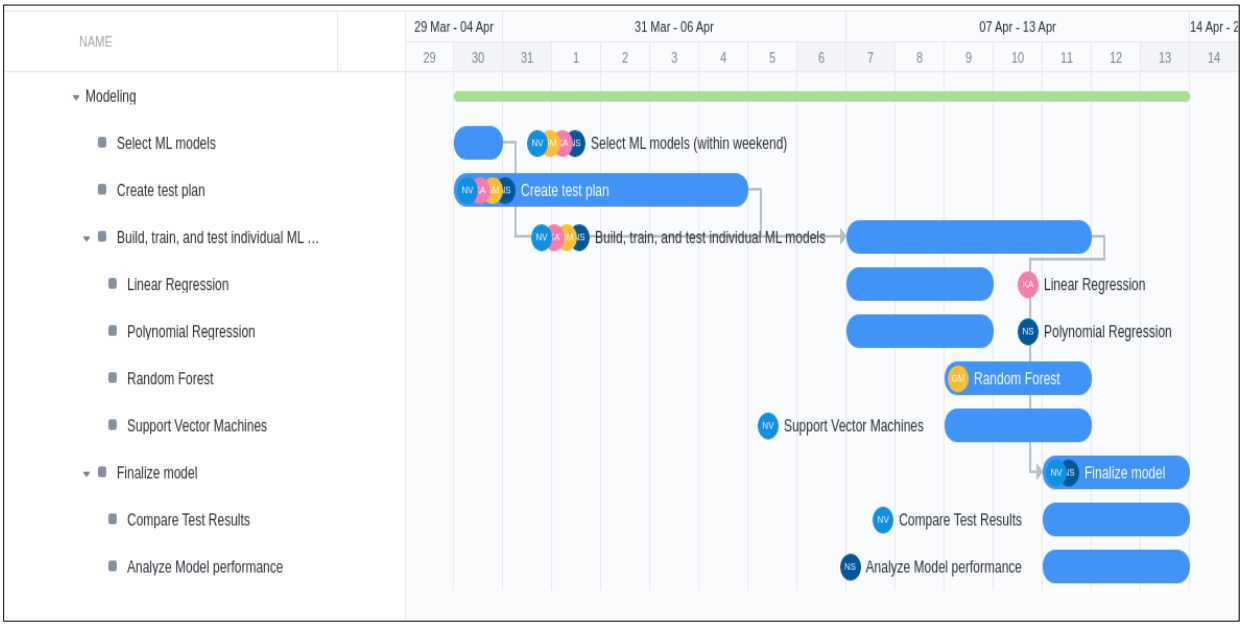


Table 5

Task List for Modeling Phase Simplified

Name	Task	Start Date	End Date	Dependent On
A	Select ML models	03/30/2024	03/30/2024	-
B	Create test plan	03/30/2024	04/04/2024	-
C	Build, train, and test individual ML models	04/07/2024	04/11/2024	A, B
D	Finalize model	04/11/2024	04/13/2024	C

Evaluation. The evaluation phase will start on April 07, 2024, and finish on May 02, 2024, as Figure 7 illustrates. There are seven tasks in this phase: test data evaluation of individual models, results analysis and comparison, best-fit model selection, creation of a project abstract, project presentation, and project report. Smaller activities like analysis, result comparison, and best fit model selection are combined into one sprint, and they are completed throughout four weekly sprints. The tasks, start and finish dates, and dependencies for the assessment phase are displayed in Table 6 and shown in Figure 8.

Figure 8

Gantt Chart of Evaluation Phase

Table 6

Task List for Evaluation Phase Simplified

Name	Task	Start Date	End Date	Dependent On
A	Evaluate individual models using test data	04/07/2024	04/11/2024	C (Modeling)
B	Analyze results	04/13/2024	04/14/2024	
C	Compare results	04/15/2024	04/15/2024	
D	Select best fit model	04/16/2024	04/17/2024	B
E	Create project abstract	04/18/2024	04/20/2024	D
F	Create project presentation	04/22/2024	04/26/2024	D
G	Create project report	04/28/2024	05/02/2024	E

Deployment. The deployment phase is depicted in Figure 8 as starting on May 02, 2024, and ending on May 07, 2024. Creating a deployment strategy, deploying ML models to Google Cloud, and tracking the model's performance over time are the three activities involved

in this step. The tasks, start and finish dates, and dependencies for the deployment phase are displayed in Table 7 shown in Figure 9.

Figure 9

Gantt Chart of Deployment Phase

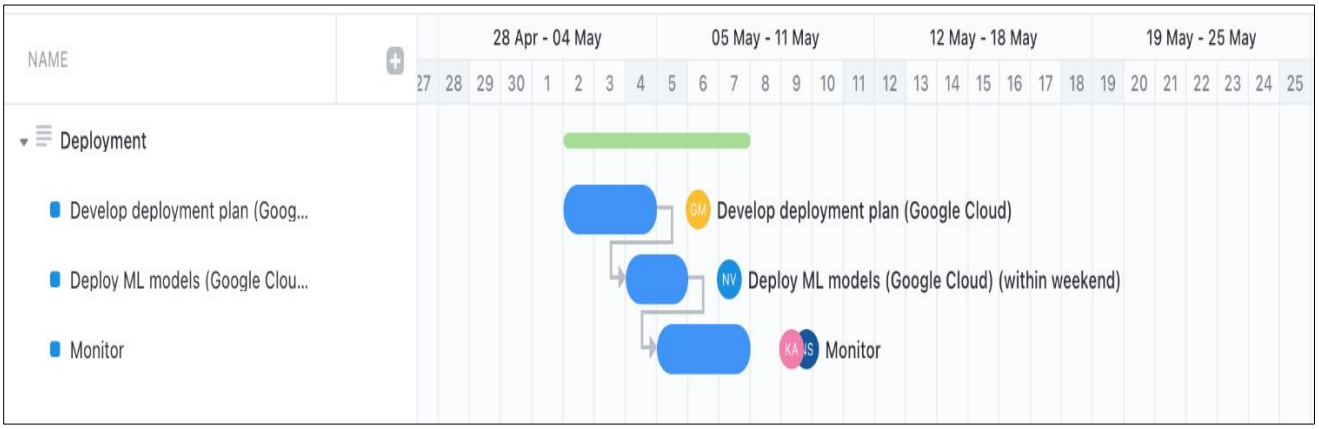


Table 7

Task List for Deployment Phase Simplified

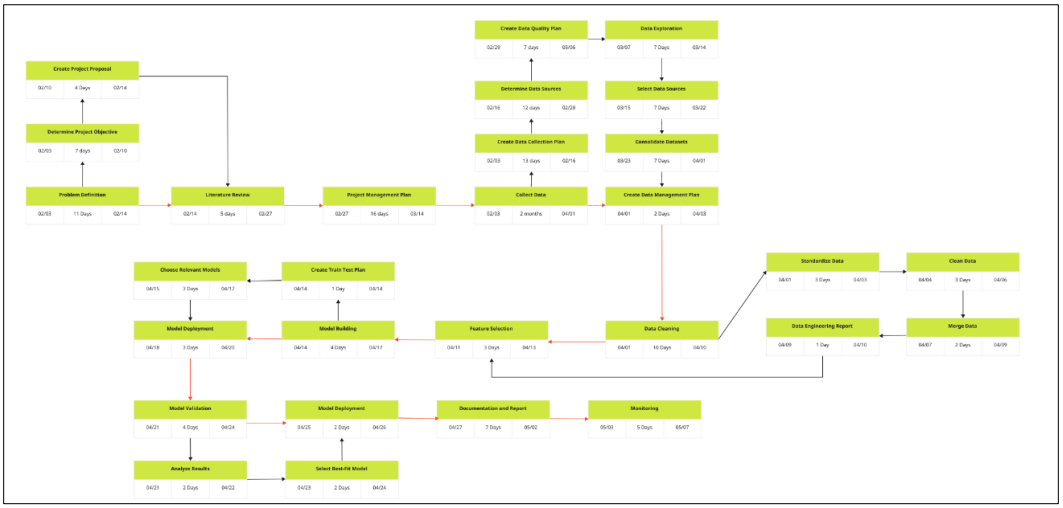
Name	Task	Start Date	End Date	Dependent On
A	Develop deployment plan (Google Cloud)	05/02/2024	05/04/2024	D(Evaluation)
B	Deploy ML models (Google Cloud)	05/04/2024	05/05/2022	A
C	Monitor	05/05/2022	05/07/2024	B

PERT Chart

A PERT chart breaks down tasks into smaller units, maps them, and shows the timetable to arrange and organize them for a project. The PERT chart aids in estimating the bare minimum of time needed to do each activity in a project. The PERT chart for our H-Predict i.e. healthcare cost estimation is displayed in Figure 10, where arrows denote the order of tasks and nodes stand for project tasks. We use it to keep an eye on the tasks and due dates for our projects.

Figure 10

PERT Chart for H-Predict: healthcare cost estimation



In Figure 10 red line pointers indicate the critical path and the tasks that is pointed cannot be delayed without impacting the other tasks and the black line pointers are the other tasks or activities. All the critical tasks are also highlighted in the above Figure 10. This project's critical path is made up of several interrelated tasks that are essential to its effective completion. These are required for careful planning, resource allocation, and risk mitigation to be completed successfully.

These are to define the problem of predicting the cost of healthcare and provide guidance for the project's direction and technique, a comprehensive literature review is conducted from the outset. The next step is to create a thorough project management plan that includes objectives, schedules, and resource allocation. To maintain data security and integrity, the next steps entail gathering data from multiple sources and developing a data management strategy. For preprocessing and selecting useful variables for model building, data cleaning and feature selection are essential. Reporting and documentation at each step of the process ensure repeatability and transparency. Monitoring the quality of the data and the model's performance is necessary to identify issues. To identify and fix any problems or drifts, model performance and data quality must be continuously monitored.

3.Data Engineering

3.1 Data Process

The data processing pipeline for this healthcare cost prediction project involves a comprehensive series of steps designed to harness raw data and transform it into valuable insights and predictive models. First, information is meticulously gathered from many unique sources, including local hospital discharge documents and federal government surveys, to compile data. By recording a huge range of parameters, such as patient age groups, medical histories, treatment costs, and insurance claims, this extensive data set offers an abundance of analytical data. The first round of data cleansing starts once the data has been collected. Strict procedures are employed here to eliminate outliers and duplicate inputs, manage missing values, and adjust for inconsistencies. The meticulous cleaning process ensures the integrity and reliability of the dataset, offering a solid foundation for additional study. Here, stringent protocols are used to correct discrepancies, handle missing numbers, and remove outliers and duplicate entries. The rigorous cleansing procedure guarantees the dataset's dependability and integrity, providing a strong basis for further analysis.

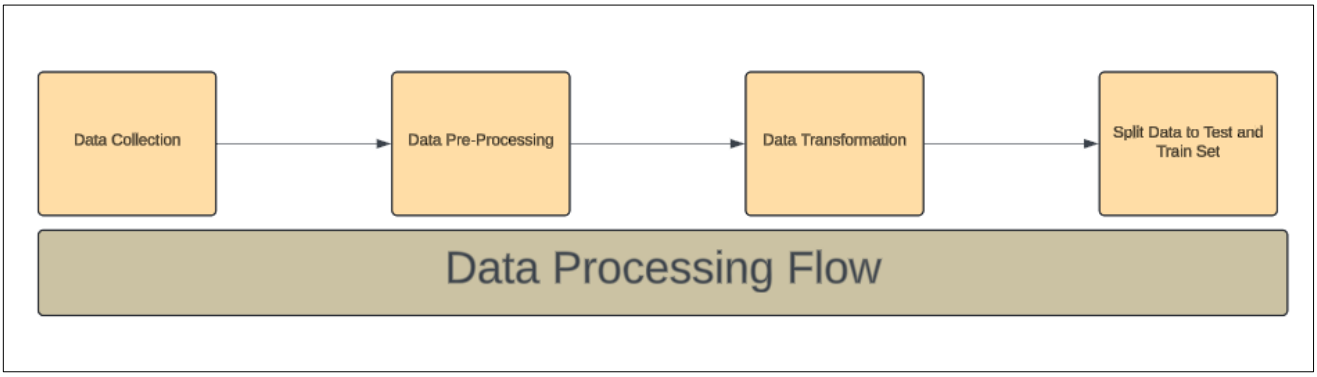
Upon completion of data cleaning, the integrated dataset is formed by merging disparate sources, harmonizing common variables, and resolving any disparities in data formats or structures. Feature engineering techniques are subsequently applied to extract new insights and enhance predictive capabilities by creating features derived from existing ones. Numeric features undergo standardization or normalization, while categorical variables are encoded to facilitate modeling tasks. Following feature engineering, the dataset is divided into unique subsets for training, validation, and testing, ensuring robust model evaluation and performance assessment.

The next pivotal phase involves training machine learning algorithms on the prepared

dataset to identify underlying relationships or patterns among variables. Data splitting into training and testing sets is a crucial step in model development, allowing for the assessment of various model variations. By evaluating multiple combinations of training and testing data, the project ensures thorough exploration of model performance under different conditions, leading to the selection of the most robust and effective model for deployment. A variety of performance indicators, including mean squared error and RMSE etc., are used in model evaluation to assess each algorithm's applicability and effectiveness. The optimal model is chosen for use in practical applications after iterative trial and improvement. After being put into use, the model is constantly observed to evaluate its effectiveness over time and guarantee that it will continue to provide accurate and dependable support for healthcare decision-making processes. For the model to remain relevant and effective in changing healthcare environments, regular updates and retraining with fresh data are necessary. All in all, this painstaking data processing pipeline plays a critical role in utilizing data-driven insights to guide strategic resource allocation and healthcare planning, which in turn improves patient outcomes and operational effectiveness. Figure 11 shows the Data Process diagram.

Figure 11

Data Process Diagram



3.2 Data Collection

In order to guarantee the accuracy and efficacy of the data gathered, a number of crucial aspects are thoroughly taken into account during the data collection process. First and first, it is critical to identify appropriate data sources, with special attention paid to their dependability, reputation, and compatibility with the goals of the project. Second, to make sure the data is adequate for predicting healthcare costs, its relevance to the study questions and objectives is assessed. Thirdly, steps are taken to ensure that data quality is maintained. These steps cover things like consistency, accuracy, completeness, and conformity to legal and ethical norms. To guarantee the representativeness of the data, the right sampling strategy is also chosen, and choosing proper data collection techniques—like surveys or secondary data analysis—is essential to successfully obtaining the necessary information. Furthermore, complete metadata- and transparency-rich documentation of the data collection process promotes effective data management. The study carefully considers these elements in an effort to collect high-quality data that would support strong and dependable studies and ultimately help with accurate healthcare cost projection.

We identified two primary sources for collecting data: surveys conducted by federal agencies and inpatient medical charges from hospitals. The survey information, which was acquired from federal agencies, offers thorough insights into a range of healthcare-related topics, such as patient demographics, medical histories, lifestyle decisions, and treatment expenses. However, data from inpatient medical charges provides important insights about the true costs that patients pay while they are hospitalized, such as information about treatments, duration of stay, and related costs. Integrating data from these diverse sources allows for a holistic

understanding of healthcare costs, enabling more accurate prediction models and informed decision-making in healthcare management and policy development.

Data Collection Plan for medical survey data

Table 8

Data Collection Plan for medical survey data

Description of Data Collection through survey conducted by federal agencies	
Purpose of data collection?	The objective of gathering information through surveys administered by federal agencies is to acquire a good understanding of various factors affecting the cost of healthcare, encompassing patient demographics, medical background, lifestyle preferences etc.
Utilization of Data	The information gathered will be utilized to create prediction models that will enable the healthcare industry to more easily allocate resources and manage finances by providing an accurate estimate of healthcare expenditures.
Units of measurement	Rows of Data Collected
Variables	Various features such as gender, age, reason for hospitalization etc.
Datatype of variables	Text and numeric
Collection Method	Download from the federal site and merge the data from all years
Data Collector	Kavana Anil Neha Sharma

Start Data	04/01/2024	04/01/2024
End Data	04/03/2024	04/03/2024
Duration	2 days	2 days

Data Collection Plan for Inpatient medical charge data

Table 9

Data Collection Plan for Inpatient medical charge data

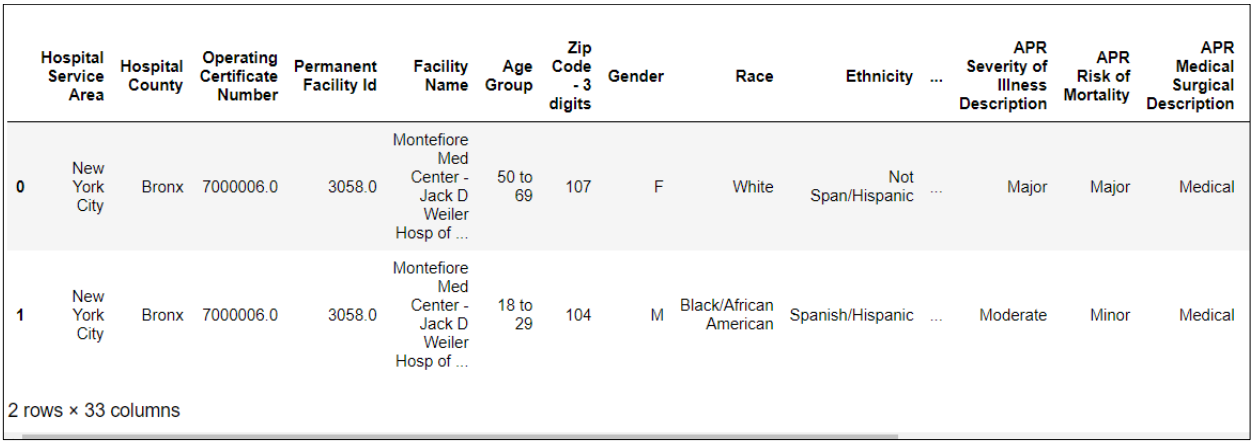
Description of Data Collection through Inpatient medical records	
Purpose of data collection?	The objective of gathering information through inpatient admission and their medical expense in each state is to understand how various factors have impacted the cost of healthcare.
Utilization of Data	The information gathered will be utilized to create prediction models that will enable the healthcare industry to more easily allocate resources and manage finances by providing an accurate estimate of healthcare expenditures.
Units of measurement	Rows of Data Collected
Variables	Various features such as gender, age, reason for hospitalization etc.
Datatype of variables	Text and numeric
Collection Method	Download from the federal site and merge the data from all years

Data Collector	Nivedita Venkatachalam	Ganaprathyusha Puluputhuri Muni
Start Data	04/01/2024	04/01/2024
End Data	04/03/2024	04/03/2024
Duration	2 days	2 days

Raw Dataset Samples

To facilitate thorough analysis, data gathered from all sources is combined and integrated into a single Excel file during the healthcare cost projection process. This combined dataset includes a variety of data from surveys carried out by federal agencies and inpatient discharge documents that were acquired from state government websites as mentioned in the above section. The Excel file's integration of many data sources, including demographics of patients, medical history, treatment costs, and hospitalization specifics, offers a comprehensive picture of the variables affecting healthcare spending. By facilitating more effective dataset exploration, visualization, and modeling, this method improves analytical capabilities. Moreover, a consolidated Excel file's streamlined data administration makes chores like data cleansing, preprocessing, and feature engineering easier, encouraging team members to collaborate and act consistently. Overall, the process of compiling and combining data into a single Excel file maximizes the efficacy and efficiency of healthcare cost prediction endeavors, hence enabling the creation of more precise and perceptive predictive models. The overall excel file post merging both sources are about 700 MB and the row count is about 1000000. The raw data sample is as given below in Figure 12.

Figure 12



	Hospital Service Area	Hospital County	Operating Certificate Number	Permanent Facility Id	Facility Name	Age Group	Zip Code - 3 digits	Gender	Race	Ethnicity	...	APR Severity of Illness Description	APR Risk of Mortality	APR Medical Surgical Description
0	New York City	Bronx	7000006.0	3058.0	Montefiore Med Center - Jack D Weiler Hosp of ...	50 to 69	107	F	White	Not Span/Hispanic	...	Major	Major	Medical
1	New York City	Bronx	7000006.0	3058.0	Montefiore Med Center - Jack D Weiler Hosp of ...	18 to 29	104	M	Black/African American	Spanish/Hispanic	...	Moderate	Minor	Medical

2 rows × 33 columns

Table 10 below shows the data present in each of the columns in the final merged dataset and Figure 13 shows the datatype of each column in the dataset.

Table 10

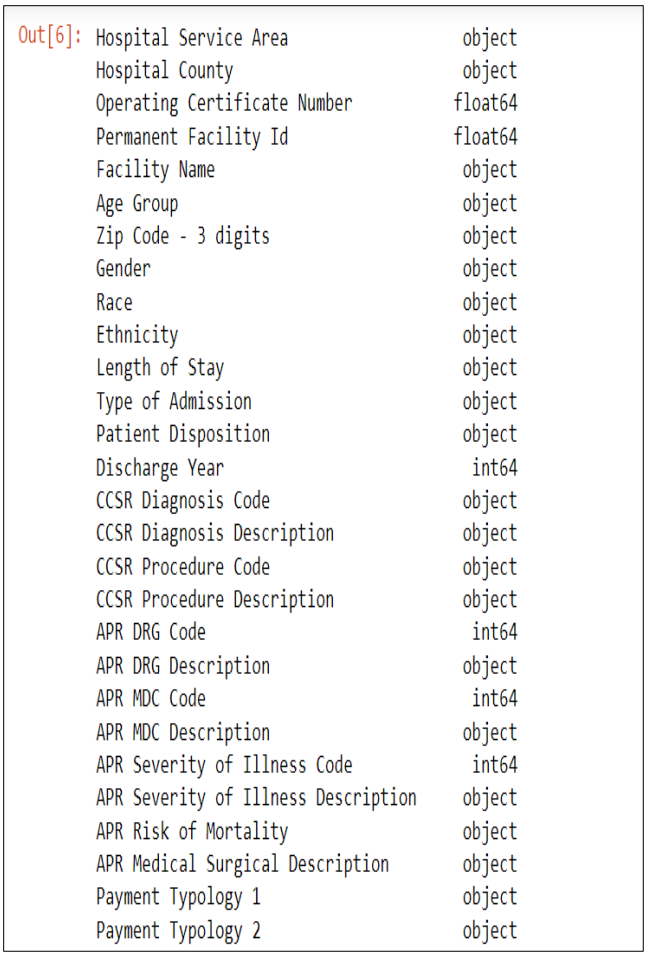
Column	**Data Description**
Hospital Service Area	The geographical area or region served by the hospital.
Hospital County	The county where medical center is located.
Operating Certificate Number	A unique identifier given to the hospital for operation.
Permanent Facility Id	An identifier for the hospital facility.
Facility Name	The name of the hospital
Age Group	The age group of the patient
Zip Code	The first three digits of the patient's zip code
Gender	The patient’s gender

Race	The race of the patient
Ethnicity	The ethnicity of the patient
Length of Stay	The length of stay in hospital for the patient
Type of Admission	The type of admission to the hospital
Patient Disposition	The outcome or disposition of the patient after discharge
Discharge Year	The year in which the patient was discharged
CCSR Diagnosis Code	Clinical Classification Software Refined code
CCSR Diagnosis Description	Description of the diagnosis
CCSR Procedure Code	CCSR procedure code
CCSR Procedure Description	Description of the procedure
APR DRG Code	All Patient Refined Diagnosis Code
APR DRG Description	Description of the diagnosis-related group
APR MDC Code	All Patient Refined Major Diagnostic Categories code.
APR MDC Description	Description of the major diagnostic category on MDC code
APR Severity of Illness Code	Code to find the severity of illness
APR Severity of Illness Description	Description of the illness
APR Risk of Mortality	Risk of mortality assessment
APR Medical Surgical Description	Description of whether the admission is medical or surgical

Payment Typology 1, 2, 3	Payment typology categories
Birth Weight	The birth weight of the new born
Emergency Department Indicator	Indicates whether the patient needed emergency care
Total Charges	Total charges incurred during the hospital stay
Total Costs	Total costs incurred by the hospital

Figure 13

Datatype of each column in the dataset



Out[6]:

Hospital Service Area	object
Hospital County	object
Operating Certificate Number	float64
Permanent Facility Id	float64
Facility Name	object
Age Group	object
Zip Code - 3 digits	object
Gender	object
Race	object
Ethnicity	object
Length of Stay	object
Type of Admission	object
Patient Disposition	object
Discharge Year	int64
CCSR Diagnosis Code	object
CCSR Diagnosis Description	object
CCSR Procedure Code	object
CCSR Procedure Description	object
APR DRG Code	int64
APR DRG Description	object
APR MDC Code	int64
APR MDC Description	object
APR Severity of Illness Code	int64
APR Severity of Illness Description	object
APR Risk of Mortality	object
APR Medical Surgical Description	object
Payment Typology 1	object
Payment Typology 2	object

3.3 Data Pre-Processing

For the prediction of the cost of healthcare we are using the data obtained from two different sources. These data are merged. The data from all the sources have different patterns, unique number of columns and different features, missing values in each feature and other outliers in few columns. Therefore, the merged data needs cleaning before moving for further analysis and model building.

The merged dataset is analyzed to check for the datatype of each column. Additionally, we also try to analyze if there are any null values in each dataset. If nulls are present, we proceed to check the percentage of null values in the dataset. Additionally, we also try and identify all the unique values each column can take. Figure 14 shows all zeroes for few columns from the dataset. Figure 15 shows the unique values in the dataset and Figure 16 and Figure 17 shows the null counts and percentage of nulls of each column.

Figure 14

Missing Values in the data

```
Feature: Operating Certificate Number
Number of 0 Values: 0
Number of Null Values: 5823
Unique Values: 165
==============================
Feature: Permanent Facility Id
Number of 0 Values: 0
Number of Null Values: 5252
Unique Values: 202
==============================
Feature: Discharge Year
Number of 0 Values: 0
Number of Null Values: 0
Unique Values: 1
==============================
```

Figure 15

Unique Values in the data

```
We Have 9 Unique Values. Values in Hospital Service Area Column : ['New York City' 'Hudson Valley' nan 'Long Island' 'Cap
ital/Adirond'
 'Central NY' 'Finger Lakes' 'Western NY' 'Southern Tier']
__________________________________________________
We Have 57 Unique Values. Values in Hospital County Column : ['Bronx' 'Rockland' nan 'Manhattan' 'Westchester' 'Kings' 'Q
ueens'
 'Orange' 'Nassau' 'Sullivan' 'Otsego' 'Herkimer' 'Delaware' 'Monroe'
 'Ontario' 'Cortland' 'Columbia' 'Albany' 'Suffolk' 'Onondaga' 'Madison'
 'Steuben' 'Cayuga' 'Montgomery' 'Erie' 'Jefferson' 'Oswego' 'Yates'
 'Wayne' 'Genesee' 'Schoharie' 'Ulster' 'Oneida' 'Schenectady' 'Broome'
 'St Lawrence' 'Schuyler' 'Richmond' 'Niagara' 'Chemung' 'Essex'
 'Chautauqua' 'Dutchess' 'Putnam' 'Chenango' 'Tompkins' 'Warren' 'Fulton'
 'Wyoming' 'Franklin' 'Cattaraugus' 'Saratoga' 'Lewis' 'Livingston'
 'Allegany' 'Orleans' 'Clinton']
```

Figure 16

Count of null Values in the data

Hospital Service Area	5252
Hospital County	5252
Operating Certificate Number	5823
Permanent Facility Id	5252
Facility Name	0
Age Group	0
Zip Code - 3 digits	40275
Gender	0
Race	0
Ethnicity	0
Length of Stay	0
Type of Admission	0
Patient Disposition	0
Discharge Year	0
CCSR Diagnosis Code	0
CCSR Diagnosis Description	0
CCSR Procedure Code	567963
CCSR Procedure Description	567963
APR DRG Code	0
APR DRG Description	0
APR MDC Code	0
APR MDC Description	0
APR Severity of Illness Code	0
APR Severity of Illness Description	633
APR Risk of Mortality	633
APR Medical Surgical Description	0
Payment Typology 1	0
Payment Typology 2	1092259
Payment Typology 3	1772745
Birth Weight	1855923
Emergency Department Indicator	0
Total Charges	0
Total Costs	4248

Figure 17

Percentage of null values in each column

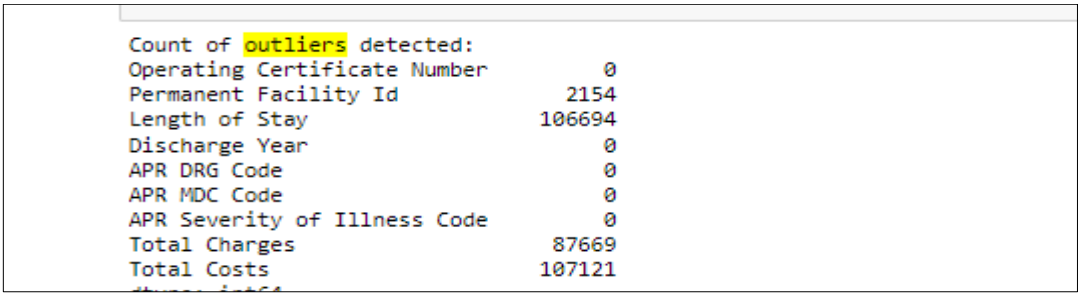
	df Null values percentage
Birth Weight	90.02
Payment Typology 3	85.99
Payment Typology 2	52.98
CCSR Procedure Code	27.55
CCSR Procedure Description	27.55
Zip Code - 3 digits	1.95
Operating Certificate Number	0.28
Hospital County	0.25
Hospital Service Area	0.25
Permanent Facility Id	0.25
Total Costs	0.21
APR Risk of Mortality	0.03
APR Severity of Illness Description	0.03
Length of Stay	0.00
Total Charges	0.00
Emergency Department Indicator	0.00
Facility Name	0.00
Age Group	0.00
Payment Typology 1	0.00
APR Medical Surgical Description	0.00
APR Severity of Illness Code	0.00
Ethnicity	0.00
APR MDC Code	0.00
APR DRG Description	0.00
APR DRG Code	0.00
Gender	0.00
Race	0.00
CCSR Diagnosis Description	0.00
CCSR Diagnosis Code	0.00
Discharge Year	0.00
Patient Disposition	0.00
Type of Admission	0.00
APR MDC Description	0.00

Next, we will analyze if the data has any outliers. Outlier identification is essential for ensuring model correctness and dependability in healthcare cost prediction. Significant deviations from the rest of the sample, known as outliers, can skew statistical studies and machine learning algorithms such as linear regression. They could result from odd therapies, uncommon medical conditions, or data errors. Ignored anomalies might result in erroneous

forecasts, which can affect healthcare budgeting and resource distribution. For healthcare cost prediction models to be reliable and successful, thorough outlier detection is necessary. Figure 18 shows the count of outliers in few columns.

Figure 18

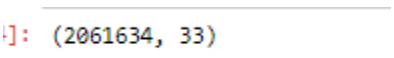
Outliers in the dataset



The columns Total Cost, length of stay and Total Charges have a lot of outliers. We can observe and conclude that the Total costs are directly proportional to the length of stay of each patient. Therefore, from this analysis we can conclude that these outliers for these columns might not have to be treated. Other columns such as Total Charges might have to be treated for the outliers. We can go ahead and drop these lines of data as imputing these outliers with mean would have an impact and introduce bias into the data. Figure 19 below show the shape of the dataset initially.

Figure 19

Initial Data shape



Data distribution, dispersion, and central tendency can be understood by examining summary statistics such as mean, median, standard deviation, and quartiles. They assist in locating outliers, evaluating variable variability, and spotting problems with the quality of the data. Preprocessing and modeling decisions are guided by these statistics, which help provide reliable predictive modeling for the prediction of healthcare costs. Figure 20 shows the descriptive statistics of all the numeric data in the dataset.

Figure 20

Statistics of Data

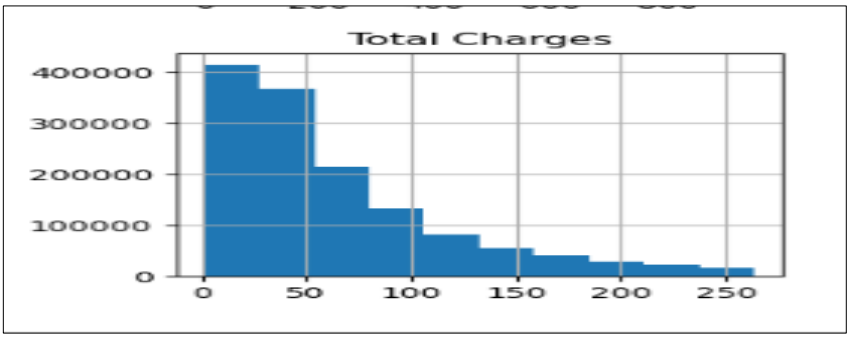
]:

	Operating Certificate Number	Permanent Facility Id	Discharge Year	APR DRG Code	APR MDC Code	APR Severity of Illness Code
count	2.055811e+06	2.056382e+06	2061634.0	2.061634e+06	2.061634e+06	2.061634e+06
mean	5.098783e+06	1.054136e+03	2022.0	4.128411e+02	1.028023e+01	2.144403e+00
std	2.201159e+06	7.178573e+02	0.0	2.434499e+02	5.957575e+00	9.612354e-01
min	1.010000e+05	1.000000e+00	2022.0	1.000000e+00	0.000000e+00	0.000000e+00
25%	2.953000e+06	5.410000e+02	2022.0	1.940000e+02	5.000000e+00	1.000000e+00
50%	5.932000e+06	1.099000e+03	2022.0	3.830000e+02	9.000000e+00	2.000000e+00
75%	7.002024e+06	1.456000e+03	2022.0	6.400000e+02	1.500000e+01	3.000000e+00
max	7.004010e+06	1.035500e+04	2022.0	9.560000e+02	2.500000e+01	4.000000e+00

Figure 21 below shows the distribution of the Total charges' column of the dataset. This column shows the distribution of the money spent on medical expenses by individuals.

Figure 21

Histogram of Total Charges



The exponentially decreasing curve observed in Figure 21 signifies that as the total charges increase, the frequency of occurrences of those higher charges decreases rapidly. This pattern suggests that there are fewer instances of very high charges compared to lower charges, indicating a skewed distribution towards lower values. Such a curve could result from a variety of factors, including a majority of patients incurring lower medical expenses, with only a small proportion experiencing exceptionally high charges. Additionally, it might reflect the presence of outliers or extreme values at the higher end of the charge spectrum, causing the curve to taper off quickly. This curve is crucial for assessing the distribution of healthcare costs within the dataset and identifying potential anomalies or areas of interest for further investigation.

The distribution of length of stay in Figure 22 shows that most of the people are admitted or stay in the hospital for less than 5 days and extremely few stays for up to 25 days and almost negligible number of people stay for more than a month.

Figure 22

Histogram of Length of Stay

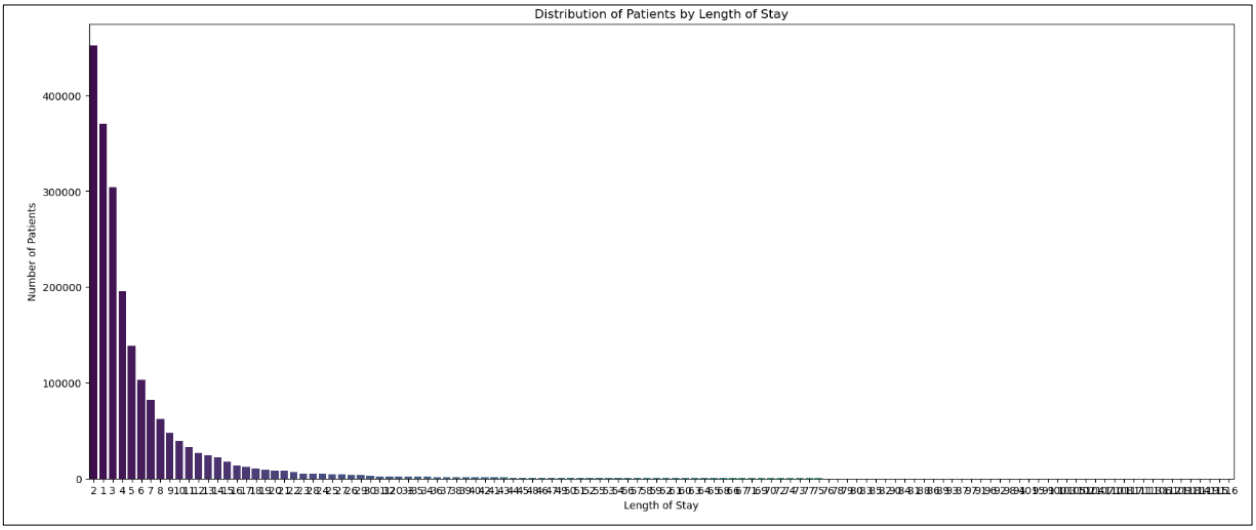


Figure 23 below shows the boxplot illustrates the spread of hospital stays across various age groups. It showcases the central tendency of hospital stays within each age group category through the median line within each box. The whiskers extending from the box capture the range of typical hospital stays, while individual data points beyond the whiskers denote potential outlier's instances of unusually long or short hospital stays. This visualization aids in identifying any age-related patterns or anomalies in hospitalization durations, facilitating a deeper understanding of the dataset's characteristics.

Figure 23

Boxplot of length of stay to each age group

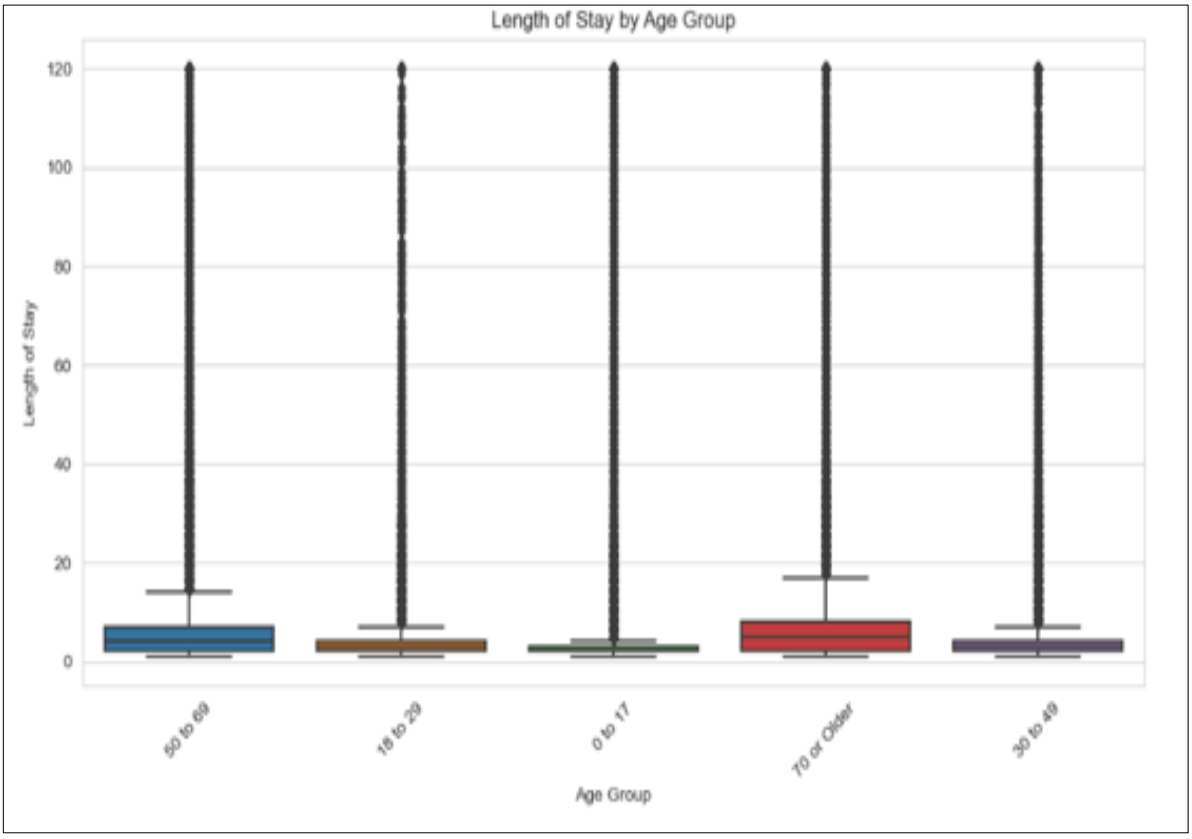


Figure 24 shows the total number of patients admitted for medical care in each group. We observe that the majority of patients admitted are elders above the age of 70 while least number of patients of the age group 18-29 visit the hospital for medical assistance.

Figure 24

Count of patients in different age groups

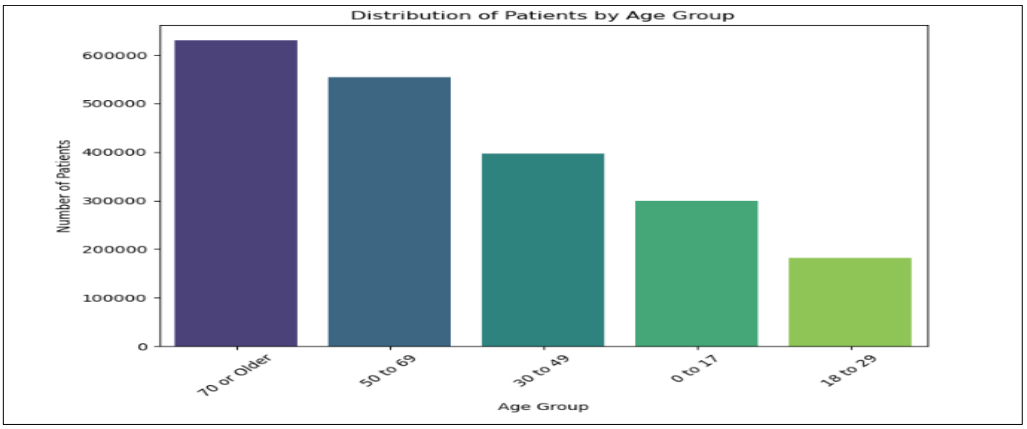


Figure 25 illustrates the gender distribution among hospital visitors, indicating that approximately 55.6% of patients are female, while around 44% are male. Also, Figure 26 shows the counts of men and women seeking medical care in the form of contingency table.

Figure 25

Percentage of patients on Gender

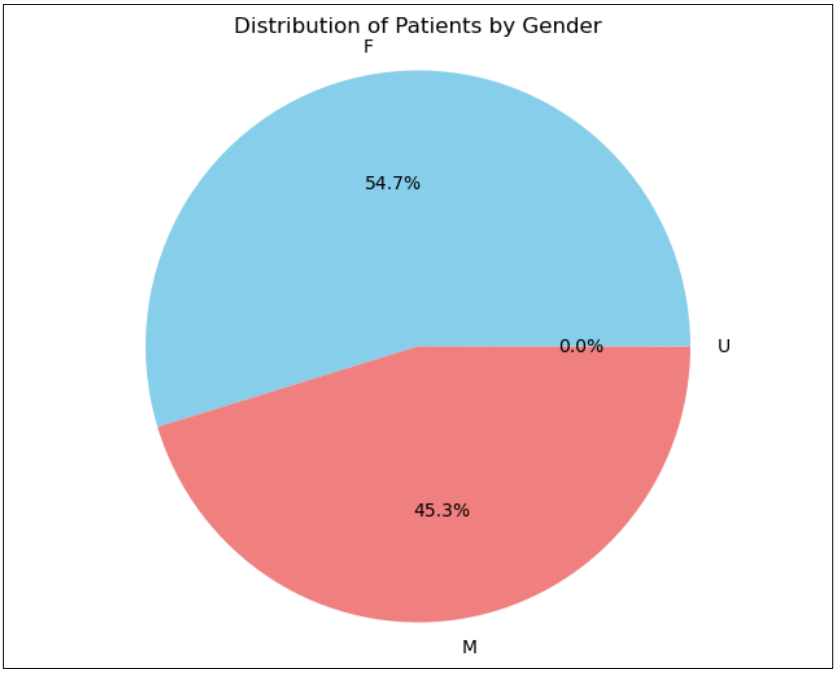


Figure 26

Count of patients by Gender

```
F    1128109
M     933274
U        251
Name: Gender, dtype: int64
```

Figure 27 presents a bar chart depicting the count of patients seeking medical assistance categorized by gender and admission type. In this stacked bar chart, we observe that the count of emergency assistance needed is higher compared to other admission types whereas the least number of patients admitted to a hospital with Trauma. Additionally, we also observe that women seek more medical assistance for the admission types Elective and Urgent while men seek it more for Newborn. Conversely, Figure 28 gives the count of patients for emergency type.

Figure 27

Admission Type by Gender

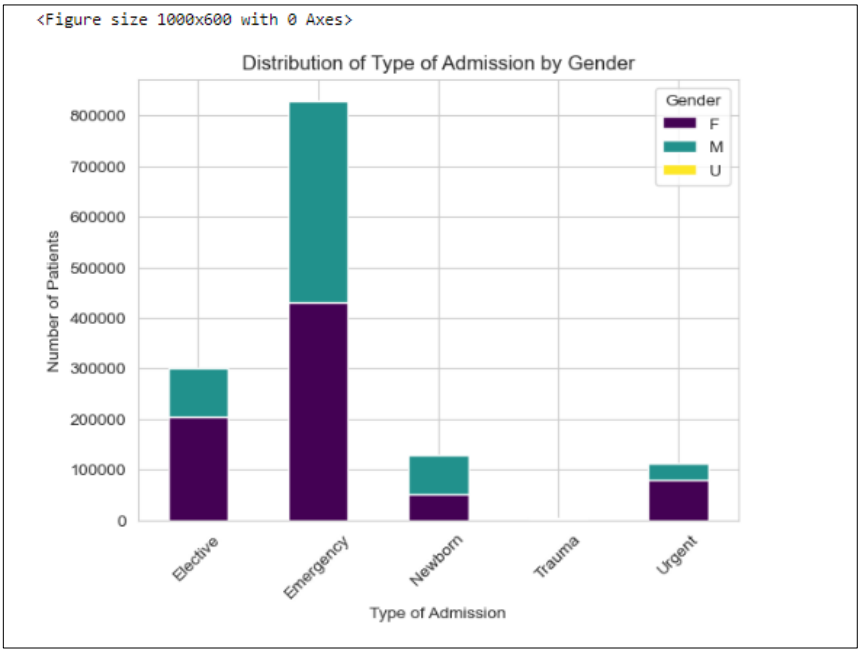


Figure 28

Count of Admission Type by Gender

```
Emergency    830287
Elective     300022
Newborn      129371
Urgent       112482
Trauma         4793
Name: Type of Admission, dtype: int64
```

Next, Figure 29 shows the percentages and Figure 30 shows the count of patients in different age groups and their admission type. All these valuable insights help us during the feature selection.

Figure 29

Percentage of patients wrt the Admission Type and Gender

Type of Admission Age Group	Elective	Emergency	Newborn	Not Available	Trauma	Urgent	All
0 to 17	3.950472	5.862900	99.973982	3.421053	4.046084	6.183321	14.506115
18 to 29	13.925551	7.708353	0.005102	12.456140	12.014813	19.344018	8.843034
30 to 49	30.316770	18.026545	0.009693	23.859649	18.886298	30.478868	19.240127
50 to 69	28.989393	30.585019	0.006122	29.385965	24.605678	23.156312	26.871549
70 or Older	22.817815	37.817183	0.005102	30.877193	40.447127	20.837480	30.539174
All	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000

Figure 30

Total count of patients wrt the Admission Type and Gender

Type of Admission Age Group	Elective	Emergency	Newborn	Not Available	Trauma	Urgent	All
0 to 17	13601	80275	195968	39	295	8885	299063
18 to 29	47944	105543	10	142	876	27796	182311
30 to 49	104377	246820	19	272	1377	43796	396661
50 to 69	99807	418771	12	335	1794	33274	553993
70 or Older	78559	517794	10	352	2949	29942	629606
All	344288	1369203	196019	1140	7291	143693	2061634

Figure 31 below shows the distribution of patients who visited or were admitted to the emergency department during their stay in the hospital or while seeking medical assistance. We can observe that around 64% of people visited the emergency room while only about 37% of them did not seek emergency care. Figure 32 displays the respective counts for ones who visited emergency care and ones who dint.

Figure 31

Percentage of patients who seek emergency care.

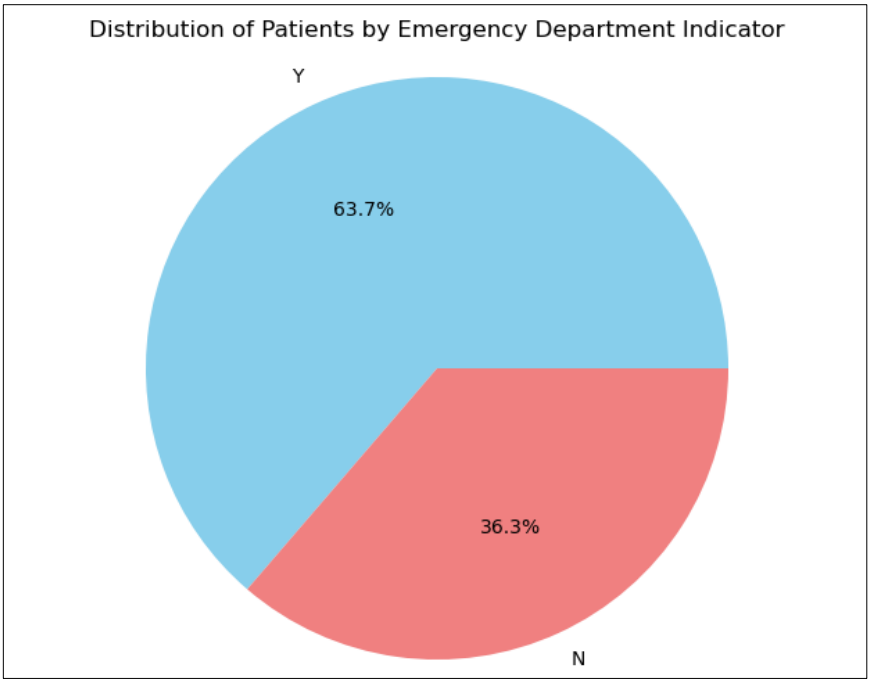


Figure 32

Count of patients who seek emergency care.

```
Emergency Department Indicator
Y    1313279
N     748355
Name: count, dtype: int64
```

Figure 33 and Figure 34 below shows the distribution of payment typology categories. This provides information on how people manage their finances for healthcare.

Figure 33

Payment Typology1 distribution

```
Payment Typology 1
Medicare                      822802
Medicaid                      642115
Private Health Insurance      295882
Blue Cross/Blue Shield        209373
Self-Pay                       26028
Miscellaneous/Other            22786
Managed Care, Unspecified      21624
Federal/State/Local/VA         19822
Department of Corrections       1202
Name: count, dtype: int64
```

Figure 34

Payment Typology1 distribution

```
Payment Typology 2
Medicaid                      402326
Medicare                      170832
Self-Pay                      142051
Private Health Insurance      133792
Blue Cross/Blue Shield         97252
Federal/State/Local/VA         11693
Miscellaneous/Other             8749
Managed Care, Unspecified       2573
Department of Corrections        107
Name: count, dtype: int64
```

Plotting the box plot between Total charges across different age groups with all the data points is shown in the Figure 35 and with extreme outliers removed is shown in the below Figure 36.

Figure 35

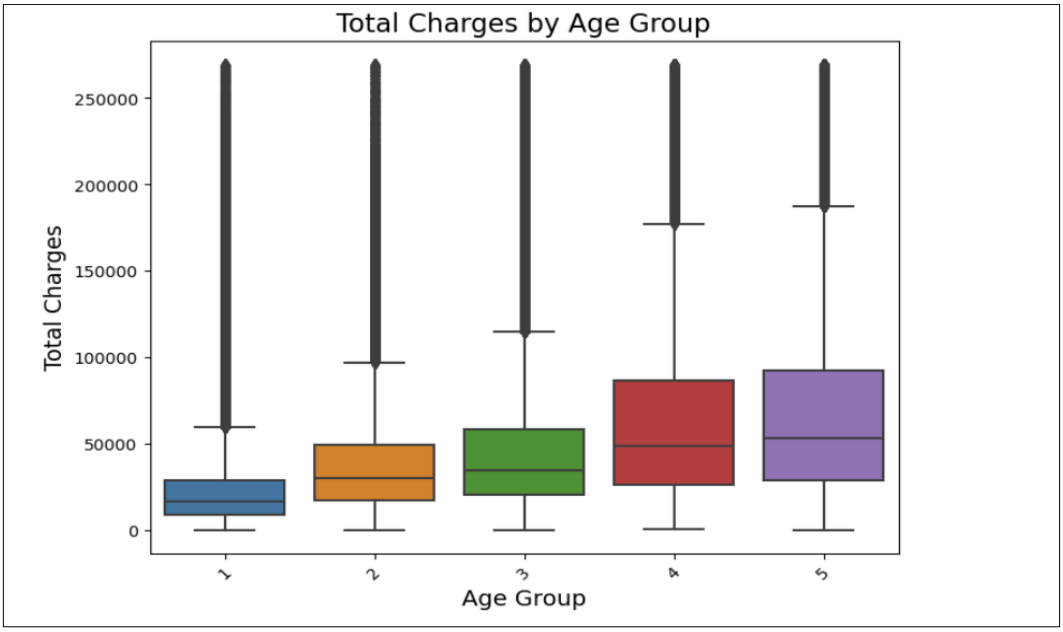
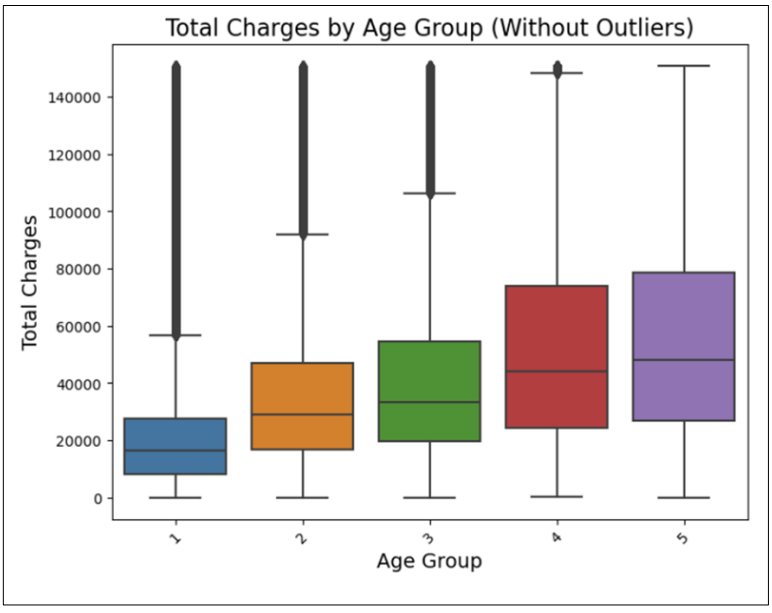


Figure 36



This plot provides a clearer view of the central distribution of charges for each age group. We can also observe that paying charges are increased with increasing in age group.

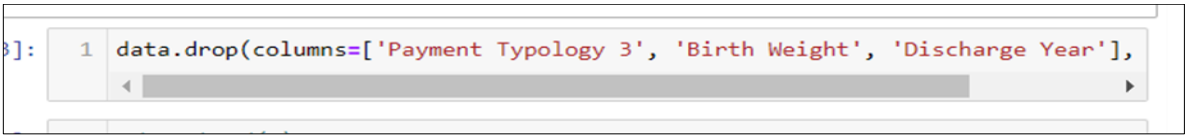
3.4 Data Transformation

Transferring data from one format or structure to another format or structure is called data transformation. This is the essential step in data management and analytics. For machine Learning also data transformation is the crucial step to improve the predictions. Making data acceptable, ensuring that it is consistent with other data in the system, or getting it ready for certain activities like reporting, visualization or additional data processing depends on this process. We are using python, the most common tool for data transformation with libraries such as pandas for data manipulation, NumPy for numerical operations, Scikit-learning etc,.

In preparing the data for machine learning models, we must ensure that the data is properly formatted and optimized by using several transformation methods. Among those Normalization is one technique used for scaling the numerical data to have a fixed range, typically 0 to 1, ensuring that no single feature disproportionately influences the model's performance. Standardization involves transforming data to have a mean of zero and standard deviation of 1. One-Hot Encoding is one more method which converts categorical data into series of binary variables. This is making the process easy for machine learning models to get the information from categorical data.

In the first step we have removed some columns which are not significant in predicting the cost. Those columns are Payment Typology 3, Birth Weight, and Discharge Year. The code snippet for this is in Figure 37.

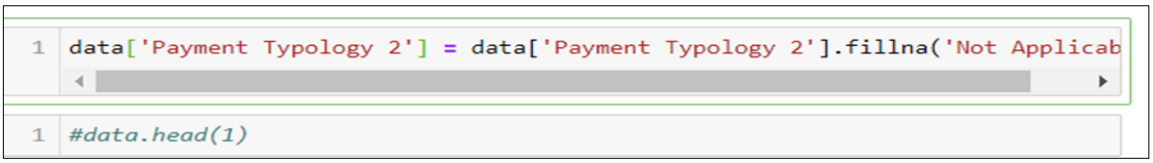
Figure 37



The Reasons for removal of the Payment Typology 3 and Birth Weight columns are due to having more number of missing values. As we can see the number of missing values in the data preprocessing section. We have dropped Discharge Year as it is less relevant if the model is intended to predict costs as the past years may not correctly represent the current cost trends because of inflation and many changes in practices of health care.

In the next step we have replaced NA with Not Applicable in the column Payment Typology 2. Instead of deleting the records or keeping the same as NA, that could negatively impact in certain ways for the analysis. So, we have filled with Not Applicable. In machine learning models like decision tree which handles categorical data by treating Not Applicable as a distinct category without imputing the mode. Mean or other imputation strategies. The snippet of code is in Figure 38.

Figure 38



As seen in Figure 17, there are anomalies in the Length of Stay column with the number 106694. These excessive levels are implausible and challenging to model or analyze using data. As a result, values of more than 120 days (about 4 months) are being substituted with 130. The maximum duration is 130 days (about 4 and a half months). By capping this, the dataset may

more accurately represent the normal range of hospital stays and lessen the presence of outliers. Machine Learning This step could avoid skewing the model's training process because models are sensitive to outliers. The code snippet is in Figure 39.

Figure 39

```
57]:
# Replace "120+" with 130 in the 'Length of Stay' column
data['Length of Stay'] = data['Length of Stay'].replace('120 +', 130)

# Convert the 'Length of Stay' column to numeric type
data['Length of Stay'] = pd.to_numeric(data['Length of Stay'])

58]:
# Set option to display all rows
pd.set_option('display.max_rows', None)

# Calculate the value counts
length_of_stay = data['Length of Stay'].value_counts()

# Print the result
print(length_of_stay)
```

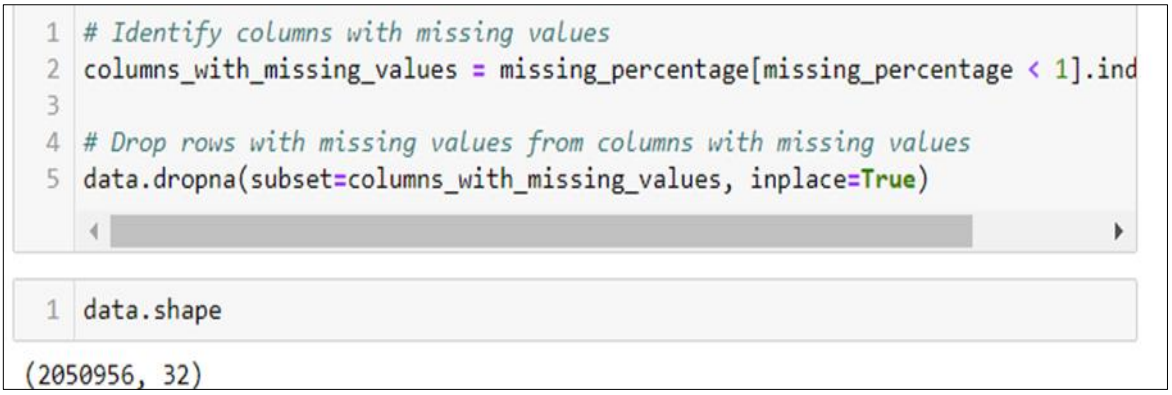
Missing Data handling is a crucial step in data transformation. By calculating missing values in each column of the dataset by using '.isnull().mean()'. Percentage of number of missing values are in the Figure 40.

Figure 40

```
Hospital Service Area                  0.254749
Hospital County                        0.254749
Operating Certificate Number           0.282446
Permanent Facility Id                  0.254749
Facility Name                          0.000000
Age Group                              0.000000
Zip Code - 3 digits                    1.953548
Gender                                 0.000000
Race                                   0.000000
Ethnicity                              0.000000
Length of Stay                         0.000000
Type of Admission                      0.000000
Patient Disposition                    0.000000
CCSR Diagnosis Code                    0.000000
CCSR Diagnosis Description             0.000000
CCSR Procedure Code                   27.549167
CCSR Procedure Description            27.549167
APR DRG Code                           0.000000
APR DRG Description                    0.000000
APR MDC Code                           0.000000
APR MDC Description                    0.000000
APR Severity of Illness Code           0.000000
APR Severity of Illness Description    0.030704
APR Risk of Mortality                  0.030704
APR Medical Surgical Description       0.000000
Payment Typology 1                     0.000000
Payment Typology 2                     0.000000
Emergency Department Indicator         0.000000
Total Charges                          0.000000
Total Costs                            0.206050
CCSR New ProcCode Category            27.525254
CCSR New DiagCode Category             0.000000
dtype: float64
```

Columns like Hospital Service Area, Hospital County and Permanent Facility Id show very low percentage of missing values around 0.25%. This is negligible. But a few columns have a higher percentage of missing values which needs to be handled. We have decided that a column may only be taken into consideration for row wise deletion if it has maximum missing values of 1%. The trade-off between preserving dataset integrity and retaining as much data as possible is balanced by this threshold. The code snippet for identifying threshold values and dropping those are in the below Figure 41 and shape of the data performed after the data transformation process.

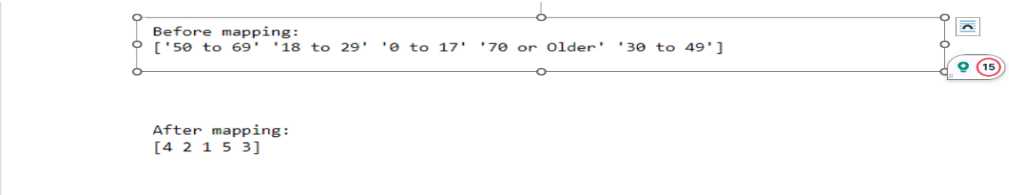
Figure 41



The ‘Age Group’ categorical data is transformed into numerical categories in the transformation step to make machine learning analysis easier. The functionality is made more efficient for algorithms that need numerical input by giving each age range its unique integer. The 'Age Group' variable may be successfully included in predictive models thanks to this mapping, which also reduces the complexity of the dataset. For models that interpret numerical order, the transformation must preserve the ordinal link between the categories. Following this encoding, the 'Age Group' feature of the dataset is converted from a series of string descriptors to discrete numerical values that correspond to the various age groups.

Ordinal encoding is employed here, in which a unique integer is assigned to each category inside the 'Age Group' variable according to the category's order. When there is an intrinsic order represented by the categorical variable (in this case, the age groups that are becoming older), this kind of encoding is especially helpful. In contrast to one-hot encoding, which generates a distinct binary column for every category, ordinal encoding preserves the single-column structure while substituting number codes that suggest a hierarchy or sequence for categories. Before and after encoding for the categorical data of ‘Age Group’ attribute is in Figure 42.

Figure 42



Before mapping the ‘Age Group’ has unique groups like ’50 to 69’, ’18 to 29’, ‘0 to 17’, ’70 or older’, and ’30 to 49’. After encoding it changes to 4,2,1,5,3 for the Age Groups. Columns with redundant data or those previously represented in larger categories were eliminated from the dataset as part of the dimensionality reduction procedure. Certain facility-specific data, descriptions, and comprehensive diagnostic codes were omitted. After cleaning, the number of features was down from 24 to 20, thereby reducing noise and redundancy in the data and enhancing the interpretability and effectiveness of further analysis or prediction models. Now that the dataset has been improved, it highlights traits that are more pertinent to the analytical objectives and more broadly applicable. Dropping the repetitive information columns before and after the shape of the dataset in Figure 43 and Figure 44. Before the reduction the total number of columns was 24 and after the reduction the total number of columns was 20.

Figure 43

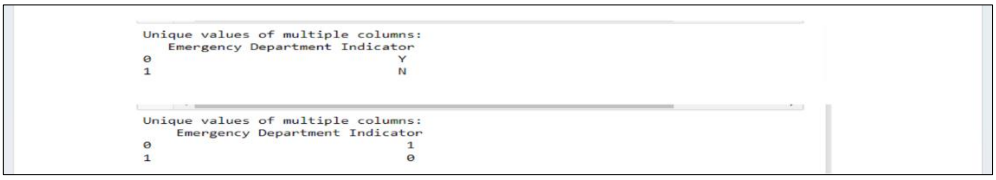
```
In [ ]: 1 data.shape
Out[ ]: (2050956, 24)
```

Figure 44

```
In [ ]: 1 data.shape
Out[ ]: (2050956, 20)
```

It is standard practice to prepare data for machine learning algorithms that require numerical input by converting Y for Yes and N for NO in the column ‘Emergency Department Indicator’ to a binary numerical format of 1 for Yes and 0 for No. Since many algorithms work better with numerical data, this encoding simplifies the dataset and improves its capabilities. The transformation facilitates a more straightforward interpretation of the effects of emergency department visits on other variables within the dataset and aids in statistical analysis and visual data exploration. Results for the binary encoding for the column ‘Emergency Department Indicator’ are in Figure 45 below. We can see that before encoding values were Y and N and after encoding done values of the column changes to 0 and 1 which are numerical.

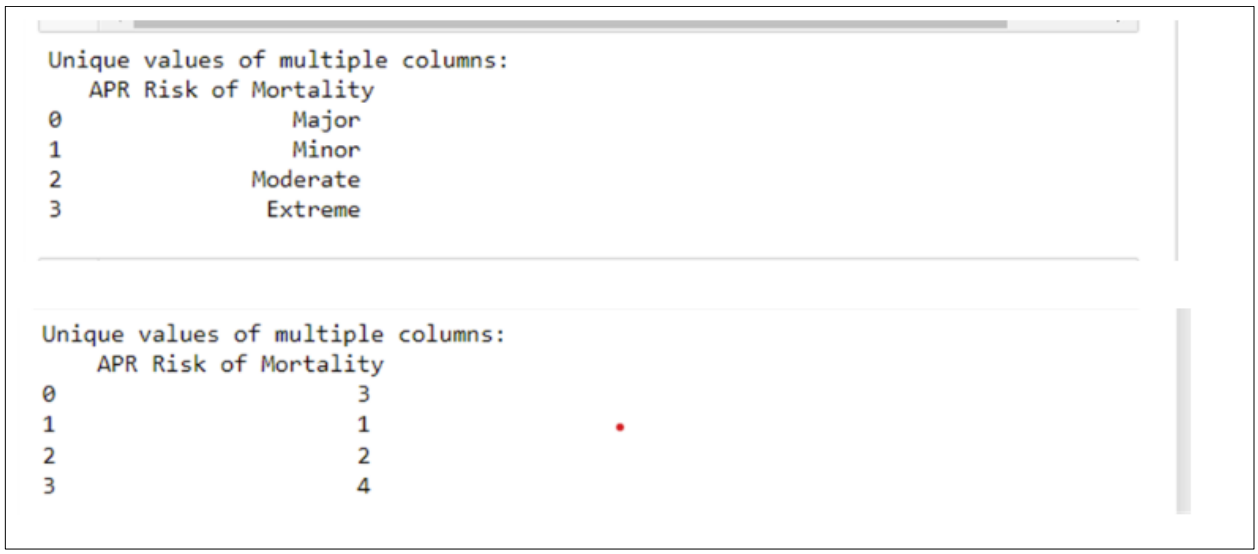
Figure 45



Ordinal encoding was used to convert the ‘APR Risk of Mortality’ feature in the dataset. This involved mapping the four categorical risk levels line Minor, Moderate, Major, and Extreme

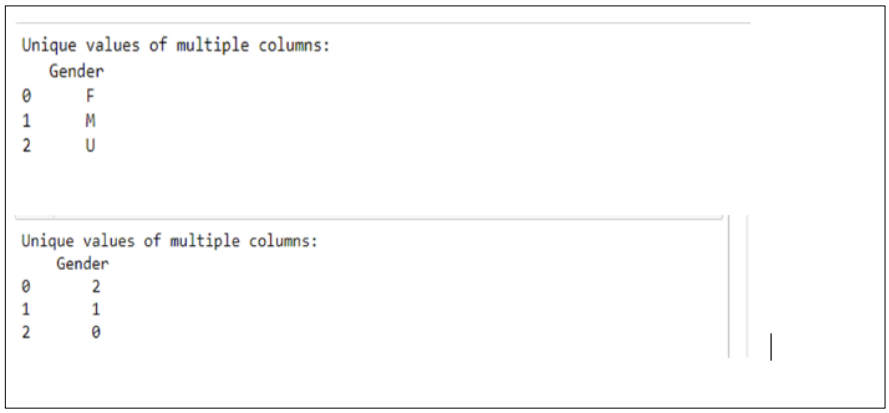
to numerical values in ascending order of severity. Minor set as 1 and Extreme set as 4. The results before and after encoding of the feature are in below Figure 46. It has before and after encoding values for the attribute ‘APR Risk of Mortality’.

Figure 46



Similarly using the ordinal encoding method, we have converted the ‘Gender’ attribute values from categorical like F, M, U to numerical values 2, 1, 0. We can see the before and after results for the gender attribute below in Figure 47.

Figure 47



Data type transformations have been performed to ensure that the numerical columns are formatted properly for modeling and analysis. This is because the feature Zip Code – 3 digits contain some nonnumerical characters, it was initially object type. But now it has been cleaned up and converted into a numeric value. To better represent their numerical nature Total costs and Total Charges features also changed from object type to float type. This prevents the nonnumeric characters from interfering with the conversion process, this transformation process involves replacing any missing values with zeros. The cleaned values are represented in Figure 48 below after changing data types of few columns from object to int or float.

Figure 48

Datatype of each column post transformation

```
1 print(data.dtypes)
```

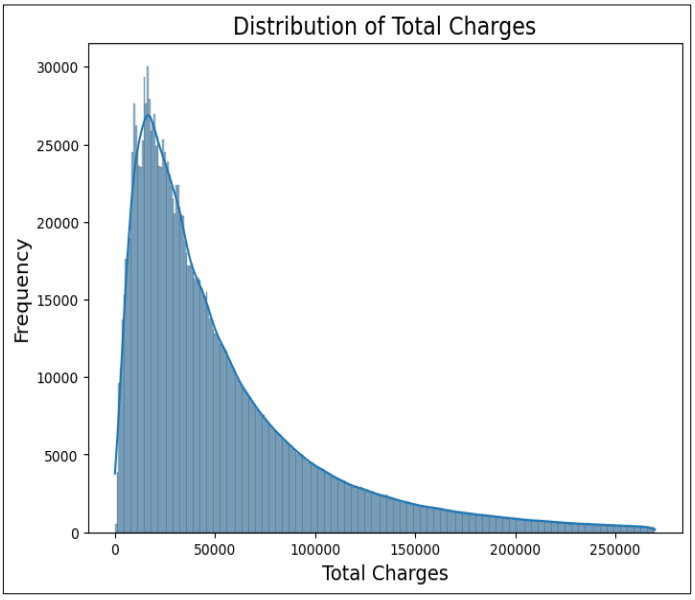
```
Hospital County                   object
Age Group                          int64
Zip Code - 3 digits                int32
Gender                             int64
Race                              object
Ethnicity                         object
Length of Stay                     int64
Type of Admission                 object
Patient Disposition               object
APR MDC Code                       int64
APR Severity of Illness Code       int64
APR Risk of Mortality              int64
APR Medical Surgical Description  object
Payment Typology 1                object
Payment Typology 2                object
Emergency Department Indicator     int64
Total Charges                    float64
Total Costs                      float64
CCSR New ProcCode Category        object
CCSR New DiagCode Category        object
dtype: object
```

Exploratory Data Analysis after Data Transformations

One of the crucial data transformation steps involved is to find the outliers and handle those in the dataset. There are several outliers in the Total Charges and Total Costs columns. Using the IQR (Interquartile Range) method, we can identify outliers based on the statistically derived thresholds. Records with values outside 1.5 times from the first and third quartiles were considered outliers. Similarly, we can identify the Total Costs too. We have identified both normal and extreme outliers for both columns and considered extreme outliers to be removed from the dataset. This results in 91990 records in the Total Charges column and 27,205 records in the Total Costs column. After removing outliers, the size of the dataset was reduced. The dataset now includes records that are more typical of the overall distribution, which can improve the performance of the machine learning models. Plotting the histogram after handling outliers for the Total Charges attribute is shown in Figure 49.

Figure 49

Distribution of Total Charges



Histogram in the Figure 49 shows a right skewed distribution, and it shows that most of the Total Charges lie in the lower end of total charges and there are only a few with high charges. As the target variable is extremely skewed to the right it necessary that we apply some form of transformations to ensure that the target follows a normal distribution.

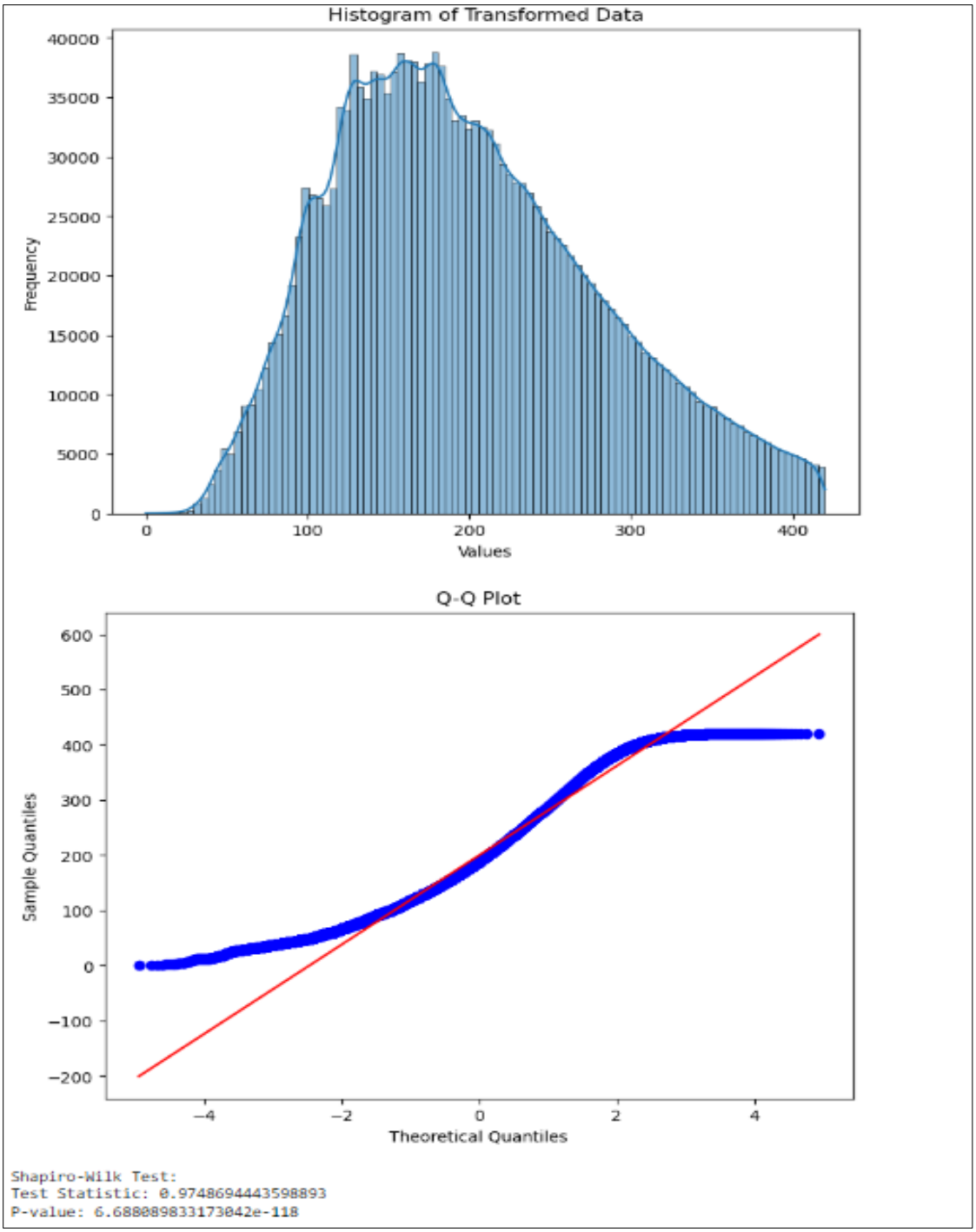
To ensure that the target follow a normal distribution, we applied a series of transformations to ensure we choose the best one. We applied transformations like log, inverse and square root on the Total Charges column. Later, we employ a statistical technique called the Shapiro-Wilk test is used to determine whether a particular sample of data is representative of a population that is normally distributed or not.

We observe that square root transformation to the Total Charges column follows a distribution which is significantly normal compared to all other transformations. The transformed data columns distribution is as shown in Figure 50. A statistical technique called the Shapiro-Wilk test is used to determine whether a particular sample of data is representative of a population that is normally distributed. The test statistic will be calculated by this test whose value ranges from 0-1 where 1 determines that the column has features which perfectly follow a normal distribution. Figure 50 also has the output of the Shapiro Wilk test performed on the transformed column. The QQ plot for the transformer charges column for the Total Charges are also present in the Figure 50 which help analyze if the column follows a normal distribution.

The analysis conducted above concludes that the square root transformed charges column showed signs of a normal distribution, which are essential to get improved accuracies of the predicted output of the machine learning models.

Figure 50

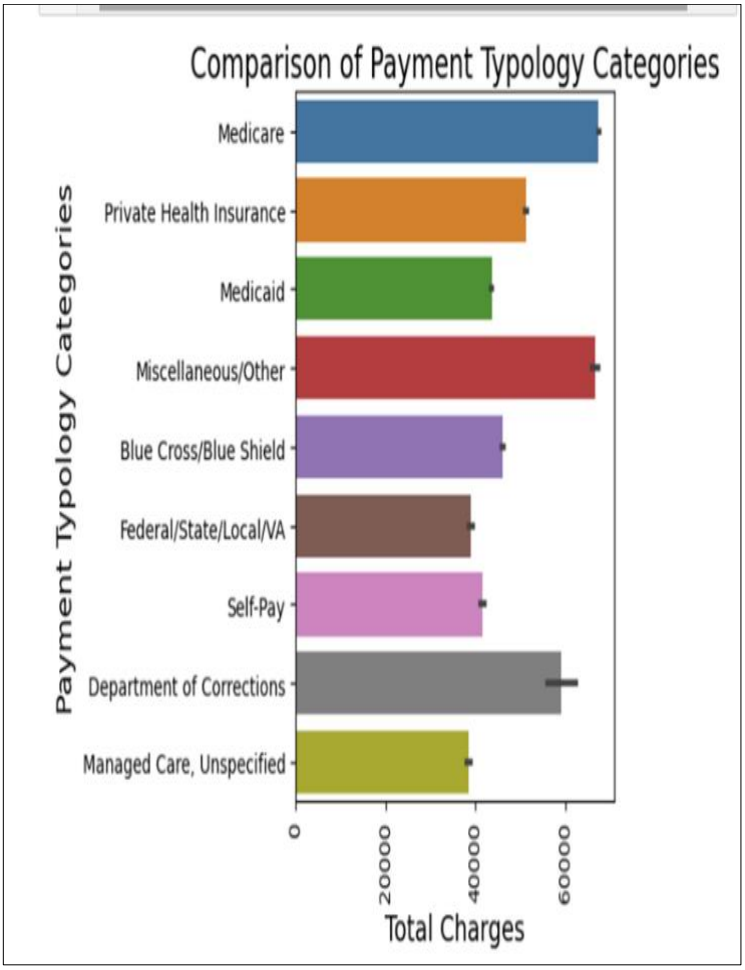
Distribution of Total Charger after transformations



After handling the Payment Topology category columns and removing outliers from Total Charges and transforming this feature to a square root scale, comparing which method of payment mostly paid the Total Charges. The comparison bar chart is shown in Figure 51 below.

Figure 51

Comparison of Payment Typology Categories

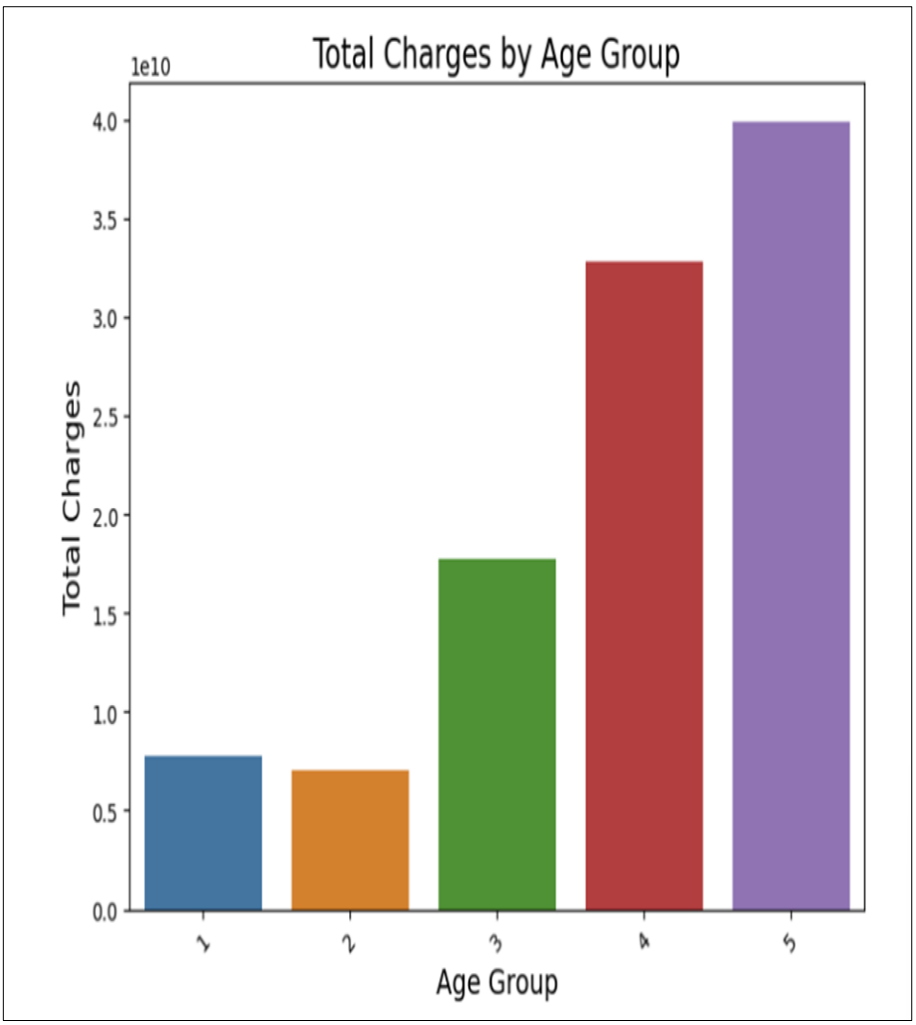


Medicare seems to be the largest expense, with Private health insurance and Medicaid following closely behind. Categories like miscellaneous/ others, Blue cross/Blue Shield and Federal/State/Local/VA are with less significant add to the charges. The lowest charges are found in Self-pay, department of corrections and Managed Care, Unspecified.

Handling the outliers in the Total charges and encoding the Age Group Column, Figure 52 shows the comparison between age group and total charges.

Figure 52

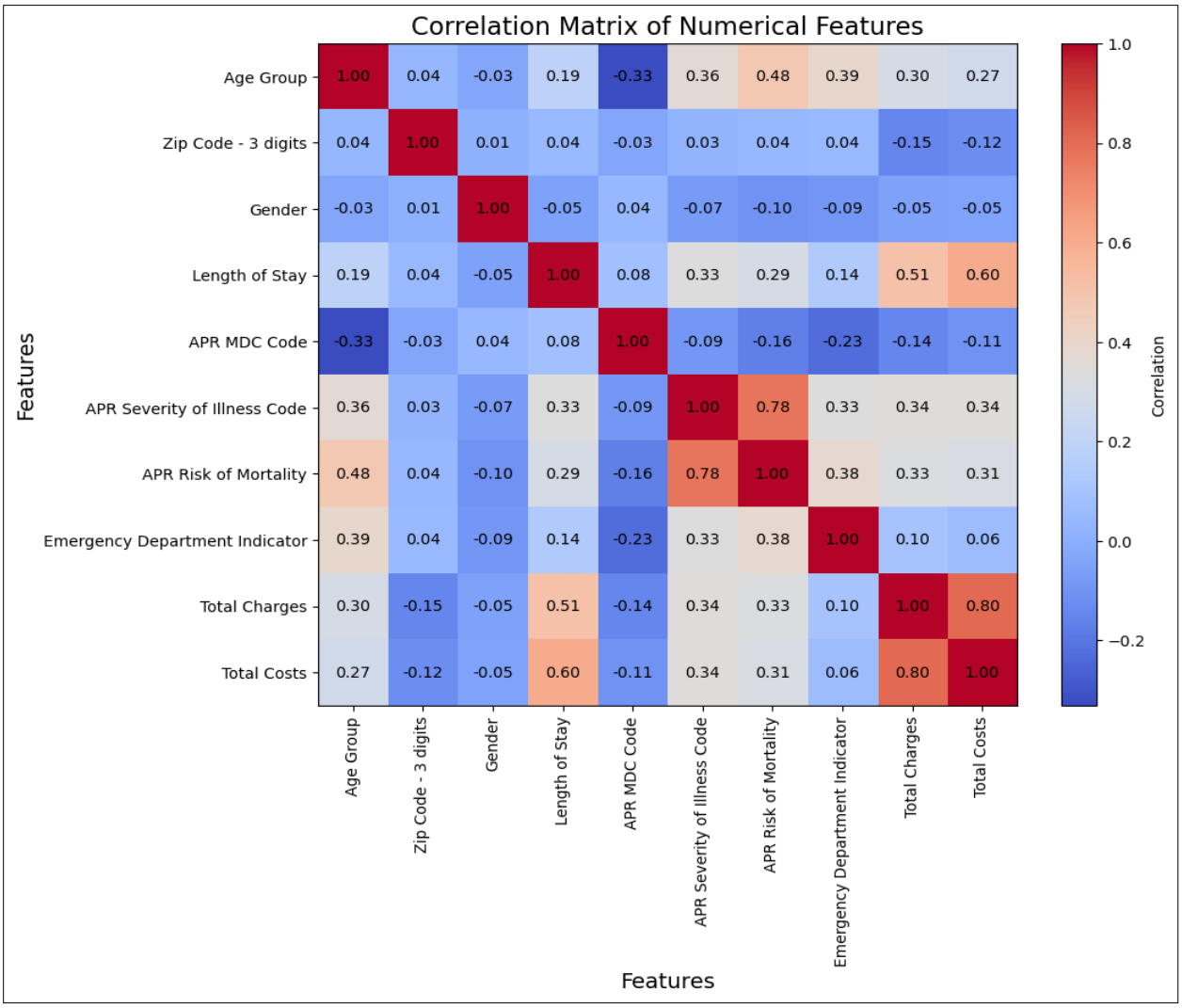
Comparison of Total Charges by Age Group



It shows that Group 5, that is the age group from 70 or older, is paying more charges than the other age group people.

We have plotted a correlation graph between all the numeric features. It is showed in the below Figure 53.

Figure 53



Red shades are used to represent positive correlations and the stronger correlations are indicated by darker hues. Blue represents negative correlations; deeper hues indicate stronger connections. For instance, there is a string positive correlation which is close to 0.8 between APR Severity of Illness Code and APR Risk of Mortality, which makes sense as more severe illness can lead to higher risk of mortality. Likewise Total costs and Total charges also highly positive correlated. Similarly stronger negative correlation is between Age Group and APR MDC Code with value -0.33.

3.5. Data Preparation

After preprocessing and transforming the data we must prepare the data for modelling. To examine the performance of a machine learning model, split data for training, testing and validation. For splitting our transformed data into training, testing and validation sets, we have employed train_test_split function from sklearn.model_selection module. The split was performed so that data distribution remains consistent across the training, testing, and validation sets. This approach ensures that the models are trained, evaluated, and validated on unbiased and representative subsets of the original data. We have split the data in a 60:20:20 ratio, allocating 60% for training, 20% for testing and remaining 20% for validation purposes. Figure 54 shows the code for data preparation.

Figure 54

Code for the Data Preparation

```
from sklearn.model_selection import train_test_split

# Define features (X) and target variable (y)
X = data_encoded.drop(columns=['Total Charges', 'Transformed_Charges', 'log_transform_charges', 'Inverse_Charges'])
y = data_encoded['Transformed_Charges']  # Adjust 'target_column' with your actual target column name

# Split data into train and test sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Split train data into train and validation sets (75% train, 25% validation)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25, random_state=42)

# Print the shapes of the resulting datasets
print("Train set shape:", X_train.shape, y_train.shape)
print("Validation set shape:", X_val.shape, y_val.shape)
print("Test set shape:", X_test.shape, y_test.shape)
```

```
Train set shape: (1090495, 169) (1090495,)
Validation set shape: (363499, 169) (363499,)
Test set shape: (363499, 169) (363499,)
```

We are developing a regression model to accurately predict hospital cost prices, which is crucial information for patients. To train this model effectively, we require a comprehensive dataset. Since our data may contain various types of information, including unstructured and varied elements, an extensive training set is necessary. The diverse and complex nature of the

data demands a substantial amount of information to train and fit the model properly. By leveraging a large training set, the model can better capture the nuances and patterns present in the data, ultimately leading to more reliable and accurate predictions of hospital costs.

The validation dataset will be utilized to tune hyperparameters for hospital price prediction model and estimate the metrics and performance in the regression task. Once the model is trained and best-fit model is implemented, test dataset will be employed to evaluate overall performance of the price prediction model. This approach ensures that the model's hyperparameters are optimized on a separate validation set, preventing overfitting, and the final assessment of the model's predictive capabilities is conducted on an unseen test set, providing an unbiased evaluation of its real-world performance.

The sample training data is shown in Figure 55. It has 1090495 rows and 170 columns after splitting the dataset into 60% for training the model and sample of the testing dataset is shown in Figure 56. It has 363499 rows and 170 columns after splitting the dataset into 20% for training the model.

Figure 55

Sample From the Training Dataset

```
X_train.head()
```

	Age Group	Zip Code - 3 digits	Gender	Length of Stay	APR MDC Code	APR Severity of Illness Code	APR Risk of Mortality	Emergency Department Indicator	Total Costs	Hospital County_Albany	...	CCSR New DiagCode Category_07 Signs/symptoms and factors influencing health status	CCSR New DiagCode Category_08 Suicidal ideation/attempt/intentional self-harm
1789359	5	105	1	13	1	4	4	1	60717.00	0	...	0	0
1737379	5	140	2	13	1	3	3	0	18006.62	0	...	0	0
1098702	4	100	1	4	20	1	1	0	3160.53	0	...	0	0
1310766	5	0	1	6	17	4	4	0	25698.94	0	...	0	0
1938066	4	105	1	6	18	2	1	1	16210.02	0	...	0	0

5 rows × 169 columns

Figure 56

Sample From the Test Dataset

```
X_test.head()
```

	Age Group	Zip Code - 3 digits	Gender	Length of Stay	APR MDC Code	APR Severity of Illness Code	APR Risk of Mortality	Emergency Department Indicator	Total Costs	Hospital County_Albany	...	CCSR New DiagCode Category_07 Signs/symptoms and factors influencing health status	CCSR New DiagCode Category_08 Suicidal ideation/attempt/intentional self-harm
1820605	5	140	1	8	5	4	3	0	25300.68	0	...	0	0
1646654	5	110	2	15	1	3	4	1	33705.13	0	...	0	0
728491	3	104	2	1	10	2	1	0	17793.31	0	...	0	0
1437438	5	112	1	1	5	3	2	1	3711.03	0	...	0	0
857584	2	133	2	1	14	1	1	0	2260.67	0	...	0	0

5 rows × 169 columns

The sample of the dataset for validation model is shown in Figure 57. It has 363499 rows and 170 columns after splitting the data further into 20% for validation of our prediction model.

Figure 57

Sample From the Validation Dataset

```
X_val.head()
```

	Age Group	Zip Code - 3 digits	Gender	Length of Stay	APR MDC Code	APR Severity of Illness Code	APR Risk of Mortality	Emergency Department Indicator	Total Costs	Hospital County_Albany	...	CCSR New DiagCode Category_07 Signs/symptoms and factors influencing health status	CCSR New DiagCode Category_08 Suicidal ideation/attempt/intentional self-harm
1381305	3	112	1	1	1	2	1	1	5059.05	0	...	0	0
469058	1	140	2	2	15	1	1	0	1620.95	0	...	0	0
982725	5	141	1	9	10	3	3	1	14988.11	0	...	0	0
155379	4	130	1	8	18	4	4	1	17292.45	0	...	0	0
1335038	3	142	2	3	14	2	2	0	13625.27	0	...	0	0

5 rows × 169 columns

3.6 Data Statistics

During the Data Engineering process, we collected raw data from various government sources. Subsequently, we then carried out data cleaning procedures to remove inconsistencies, errors, and redundancies. We then conducted exploratory data analysis to acquire an

understanding of the data's characteristics, correlations and patterns. After that, we applied transformations to restructure the data into a more suitable format. Finally, we performed data preparation steps to ensure the data was prepared and ready for modeling purposes. A concise overview of these steps is presented in the table below.

Table 11

An overview of the methods used in data processing

Process	**Method**	**Count**
Data Collection	Raw Data (Hospital Inpatient Discharges)	2061634*33
Data Collection	Raw Data (CCS pcode)	329*7
Data Collection	Raw Data (CCS dcode)	1295*2
Data Preprocessing	Merging Data	5826425*35
Data Preprocessing	Removing redundant data	2061634*34
Data Preprocessing	Removing Columns with more missing values	2061634*32
Data Preprocessing	Replacing all NA values in columns Payment Typology 2 to Not Applicable	2061634*32

Data Transformation	Replacing all 'Length of Stay" values that are 120+ to "130" signifying large number	2061634*32
Data Preprocessing	Removing Missing Data: Checking missing percentage and dropping rows with NA for columns having missing percentage < 1%	2050956*32
Data Transformation	Encoding/Mapping Age Group into numerical categories	2050956*32
Data Preprocessing	Removing Redundant Columns: Dropping Repetitive information columns or columns already categorized to broader categories	2050956*24
Data Preprocessing	Removing Redundant Columns: Dropping Repetitive information columns	2050956*20
Data Transformation	Encoding/Mapping Emergency Department Indicator	2050956*20
Data Transformation	Encoding/Mapping APR Risk of Mortality	2050956*20
Data Transformation	Encoding/Mapping Gender	2050956*20
Data Transformation	Changing data types of columns which has numeric values from object to int/float	2050956*20

Data Preprocessing	Outlier Handling: Removing Outliers	1817493 *20
Data Transformation	Target Variable Normalization	1817493*23
Data Transformation	Dummy Encoding	1817493*173
Data Preparation	Train-Test Split: Will result in 1090495 data rows for training; 363499 data rows for validation and 363499 data rows for testing	1817493 *170

The raw dataset collected from the New York state health department website consists of 2061634 rows and 33 columns. CCS for procedure codes (Agency for Healthcare Research and Quality) has 329 rows and 7 columns, CCS for diagnostic codes (Agency for Healthcare Research and Quality) dataset has 1295 rows and 2 columns.

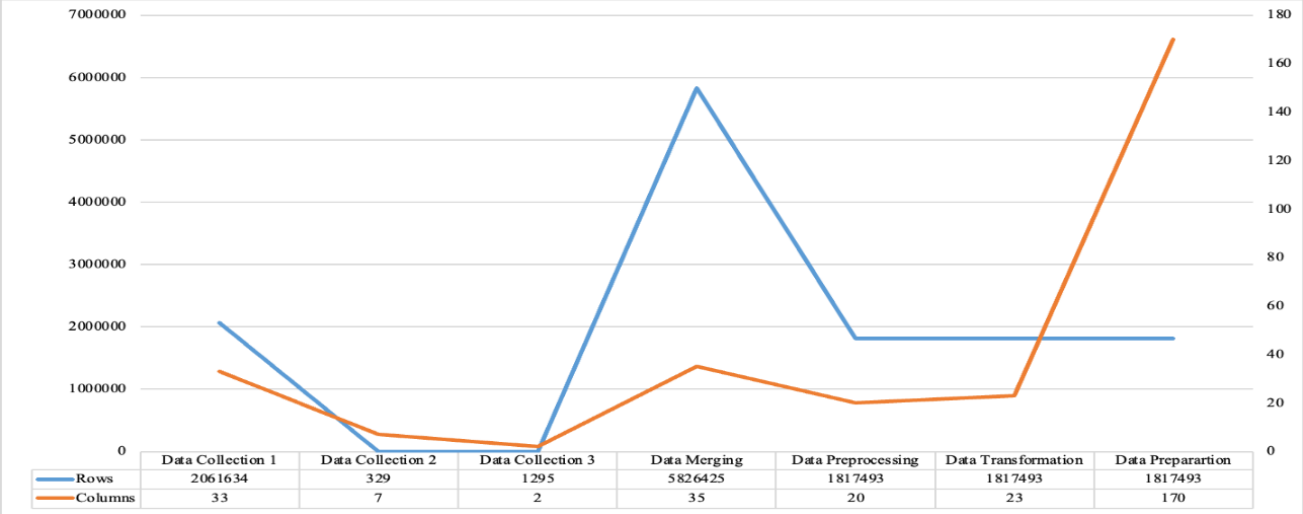
During Data Preprocessing several preprocessing steps were undertaken Merging the different data sources into a single dataset (resulting in 5,826,425 rows), removing unnecessary columns removing columns with a high percentage of missing values, replacing NA (missing) values in the "Payment Typology 2" column with "Not Applicable", Checking for missing value percentages and dropping rows with NAs for columns exceeding a certain missing percentage threshold (resulting in 2,050,956 rows), Removing redundant or repetitive information columns, including those already categorized into broader categories (down to 2,061,634 rows, then 2,061,620 rows) Outlier handling by removing extreme outlier values (down to 1817493 rows). After deleting unnecessary data, merging, and pre-processing datasets, we have a dataset with 1817493 rows and 20 columns.

During Data Transformation the preprocessed data was used to Encoding/mapping categorical variables like Age Group, Emergency Department Indicator, APR Risk of Mortality, and Gender into numerical categories Replacing "Length of Stay" values which are 120+ with 130 to signify a large number, Changing data types of columns with numeric values from object to int/float Dummy encoding (creating binary columns for categorical variables) and we are left with a dataset containing 1817493 rows and 170 columns.

Finally, in the data preparation step, we split the dataset by the ratio of 60:20:20, resulting in 1090495 rows and 170 features for training dataset, 363499 rows and 170 features for validation dataset, 363499 rows and 170 features for testing dataset. An overview of how the amount of data varies at each stage of the data engineering process is shown in Figure 58.

Figure 58

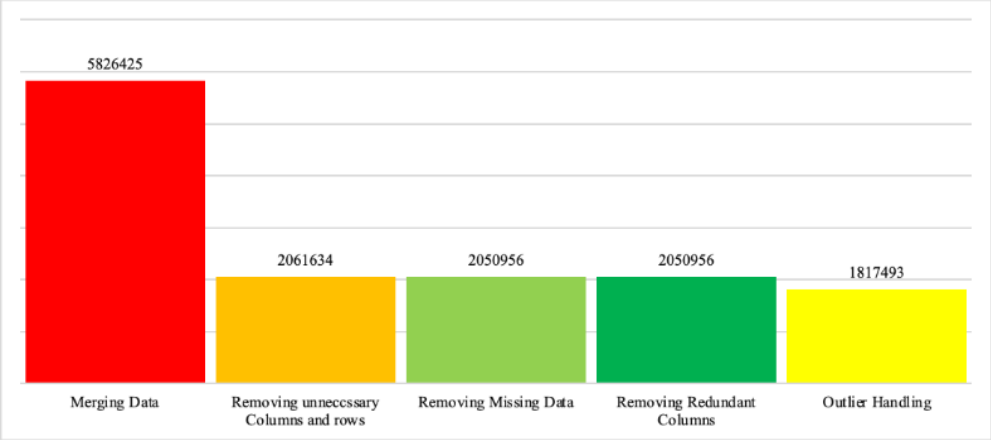
Number of data at every phase



A step-by-step variation in the amount of data in the process of data pre-processing can be found in Figure 59.

Figure 59

Amount of data from Data Pre-Processing at each level

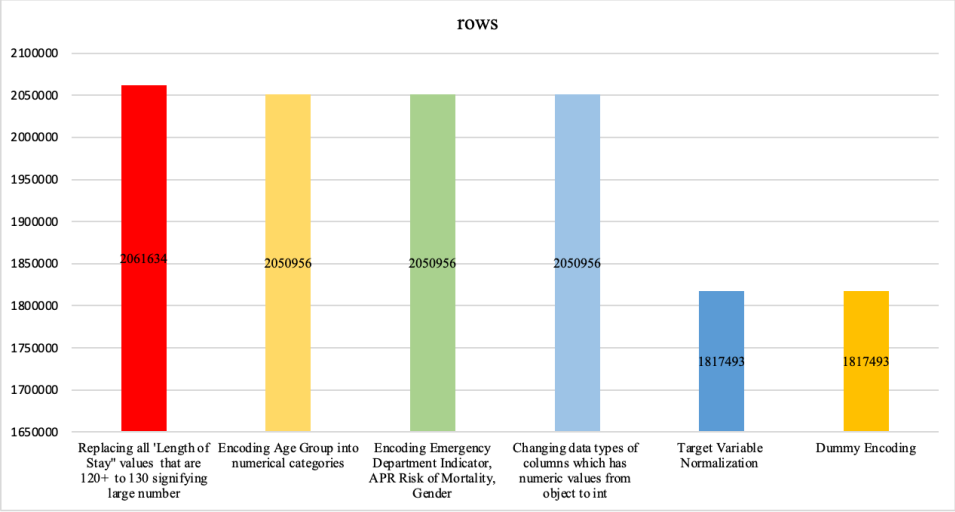


Note. Amount of data from Data Pre-Processing at each level

The data volume change for each stage of the data transformation process is shown in figure 60.

Figure 60

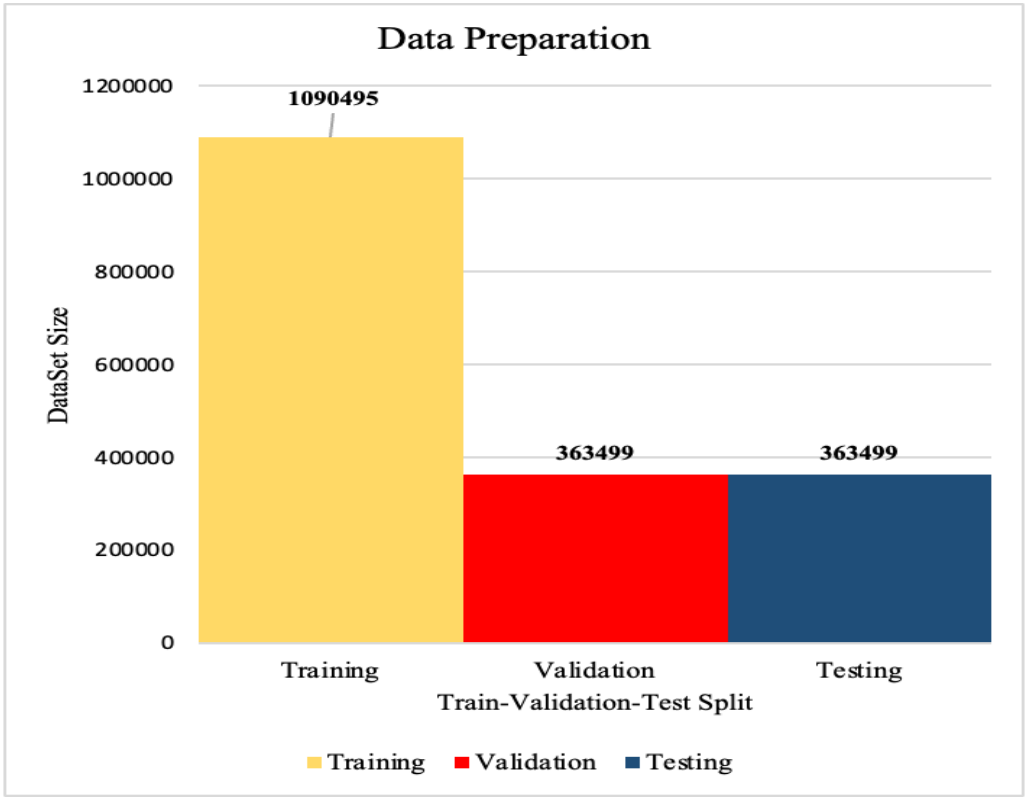
Quantity of data in all steps carried for Data Transformation



The count for data in train, test and validation samples after process of data transformation is shown in Figure 61.

Figure 61

Quantity of data in Train, Validation and Test Datasets



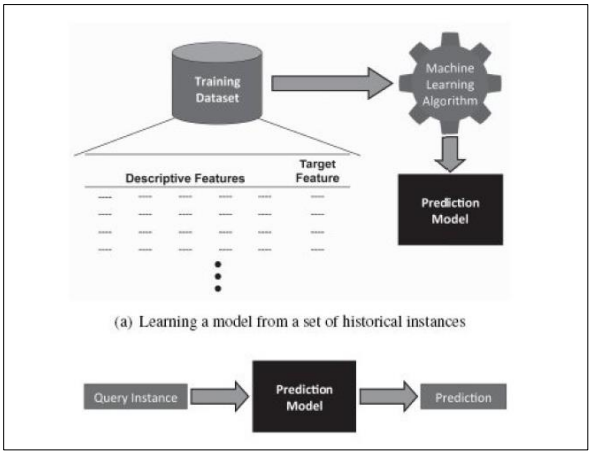
4.Modeling

4.1Model Proposals

Our goal in this research is to efficiently determine healthcare expenses by utilizing the capability of supervised machine learning algorithms. Machine learning techniques are used to find correlations between many independent variables like patient characteristics, treatment protocols, and healthcare expenditures and other technical procedural and diagnostic codes used in the healthcare domain. The ultimate objective is to create predictive models that is able to forecast the cost for medical care accurately that can support healthcare decision-making that is both economical and efficient, as well as financial planning and resource allocation. Figure 62 shows the basic overview of a supervised regression problem. All the independent variables in our dataset like age group, length of stay, procedural codes of treatment received etc. on the target feature which is the Total Charges for each patient.

Figure 62

Summary of Supervised Regression problem



Several essential tasks are included in our workflow to create and improve the prediction models. First, we will preprocess the data, which entails standardizing numerical features, encoding categorical variables, and addressing missing values. Another critical stage is featuring engineering, in which we generate new features to enhance model performance by extracting existing pertinent data. After that, we will choose suitable machine learning algorithms and use methods like grid search and cross-validation to adjust their hyperparameters. Lastly, in order to make sure that our models fulfill the criteria for precise healthcare cost estimation, we will assess their performance using metrics like mean squared error, root mean square error, and $R2$ score. With these initiatives, we hope to create solid and trustworthy prediction models that can offer insightful information.

We intend to model the relationship between independent factors that might impact the cost of healthcare using a range of regression approaches, such as polynomial regression and linear regression. Our goal is to precisely estimate healthcare expenses by leveraging variables including patient demographics, medical history, length of hospital stay, and severity of sickness by the use of regression models. We also plan to investigate ensemble approaches, like random forest, which integrate several models to increase prediction accuracy. Next, our goal is to investigate how well the XGBoost and gradient boosting algorithms can predict healthcare expenditures. Gradient boosting, which takes advantage of variables like patient demographics and medical history, gradually increases prediction accuracy by successively training weak learners to rectify errors produced by prior models. In a similar vein, XGBoost, an enhanced gradient boosting implementation, improves performance by regularization and parallel computing strategies, offering reliable forecasts by using a variety of characteristics like the length of hospital stay and the intensity of illness.

Linear Regression

The idea behind linear regression is to fit a line to the target feature, such the cost of healthcare, based on the underlying relationships and correlations between the independent features. By simulating the link between the independent variables, such as hospital location, type of admission, and degree of illness, this model can be used as a baseline to compute the cost. The equation below helps to express the relationship:

$$Y = \beta_0 + \beta_1 X \qquad (1)$$

where Y represents the healthcare cost and X represents the dependent variables,

β_0 *is the slope and* β_1 *is the intercept*

This predictive model assumes a linear association between the independent variables and the target variable, enabling healthcare providers and policymakers to anticipate costs based on patient characteristics. Additionally, for the Medicare cost prediction we have many dependent variables that might impact and must be considered together for prediction. Therefore, multiple linear regression is performed. The relationship is as below.

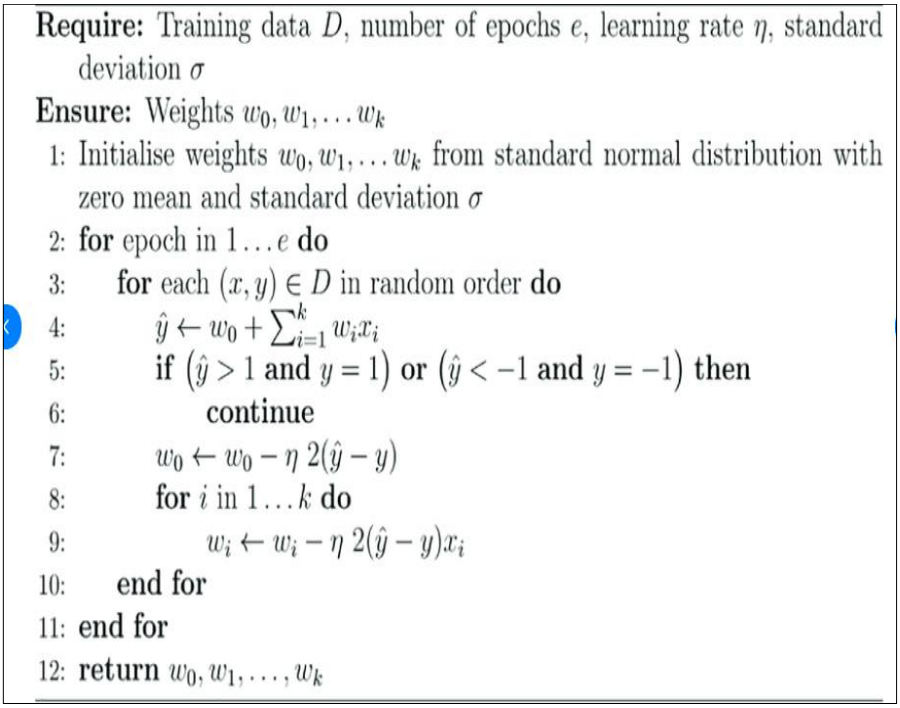
$$\mathrm{Y} = \beta_0 + \beta_1 X + \beta_2 \mathrm{X} + \beta_3 X + \ldots\ldots\ldots \beta_0 + \beta_{\mathrm{n}} X \qquad (2)$$

The basic goal of cost prediction through linear regression is to fit a line to the current data so that the errors between the projected and actual values are as little as possible. By minimizing these errors, the model aims to accurately depict the underlying link between the independent variable(s) and the dependent variable. In order to minimize the sum of squared errors, sometimes referred to as the residual sum of squares (RSS), the line's parameters (slope and intercept) must be adjusted iteratively. The interpretability and simplicity of linear regression make it very helpful for predicting healthcare costs. The coefficients assigned to each

independent variable are immediately understandable and usable to all the users, facilitating a straightforward interpretation of the model's predictions. Figure 63 shows the overview of steps in Linear Regression.

Figure 63

Overview of Linear Regression problem



Require: Training data D, number of epochs e, learning rate η, standard deviation σ
Ensure: Weights $w_0, w_1, \ldots w_k$
Initialise weights $w_0, w_1, \ldots w_k$ from standard normal distribution with zero mean and standard deviation σ
for epoch in $1 \ldots e$ **do**
  for each $(x, y) \in D$ in random order **do**
    $\hat{y} \leftarrow w_0 + \sum_{i=1}^{k} w_i x_i$
    if $(\hat{y} > 1$ **and** $y = 1)$ **or** $(\hat{y} < -1$ **and** $y = -1)$ **then**
      continue
    $w_0 \leftarrow w_0 - \eta\, 2(\hat{y} - y)$
    for i in $1 \ldots k$ **do**
      $w_i \leftarrow w_i - \eta\, 2(\hat{y} - y) x_i$
  end for
end for
return $w_0, w_1, \ldots, w_k$

Polynomial Regression

Polynomial regression can capture more intricate relationships between patient attributes and expenses for estimating healthcare expenditures. Polynomial regression expands on the capabilities of linear regression where it does not assume that the relationship between the dependent and independent variable is linear. The equation of the line which would be fit will have the formula below:

$$Y=\beta_0+\beta_1 X+\beta_2 X^2+\beta_3 X^3+\ldots+\beta_n X^n \quad (3)$$

Y represents the healthcare cost, X represents the independent variables, and β0,β1,…,βn are the coefficients of the polynomial terms.

Polynomial regression aims to minimize errors between predicted and actual values by adjusting coefficients to fit observed data, enhancing prediction accuracy in non-linear relationships. Despite less straightforward interpretability due to higher-order terms, it offers flexibility in capturing complex relationships. This method is valuable for healthcare cost prediction when linear models fail to fully capture the intricate connections between all the unique patient attributes and the expenses they incurred.

Random Forest

Random Forest provides a potent solution for healthcare expense prediction as it utilizes an ensemble of decision trees. The Random Forest model trains each tree using a distinct subset of the information, which allows it to capture various facets of the intricate correlations between patient attributes and healthcare costs. This model can efficiently handle non-linear correlations, interactions between several components, and outliers, this technique is advantageous in the prediction of healthcare costs. It is an ensemble learning method that combines different decision trees to make accurate predictions of healthcare costs. This is supervised learning.

The prediction of the Medicare cost is made using majority votes from all the individual trees in the model.

$$Y = \frac{1}{k}\sum_{i=1}^{k} y_i(X) \tag{4}$$

Y represents the target output whereas in our project it is the hospital cost prediction, k is the number of trees we have taken to build the random forest model. y-i(X) represents the prediction of the i-th tree and X here are input features like length of stay, Gender, Procedure code etc. In predicting the continuous target like hospital cost we refer the decision tree as regression random forest tree. The pseudo code for random forest regression algorithm is in Figure 64.

Figure 64

Algorithm for random forest regression

Algorithm 1: Pseudo code for the random forest algorithm

To generate c classifiers:
for i = 1 to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

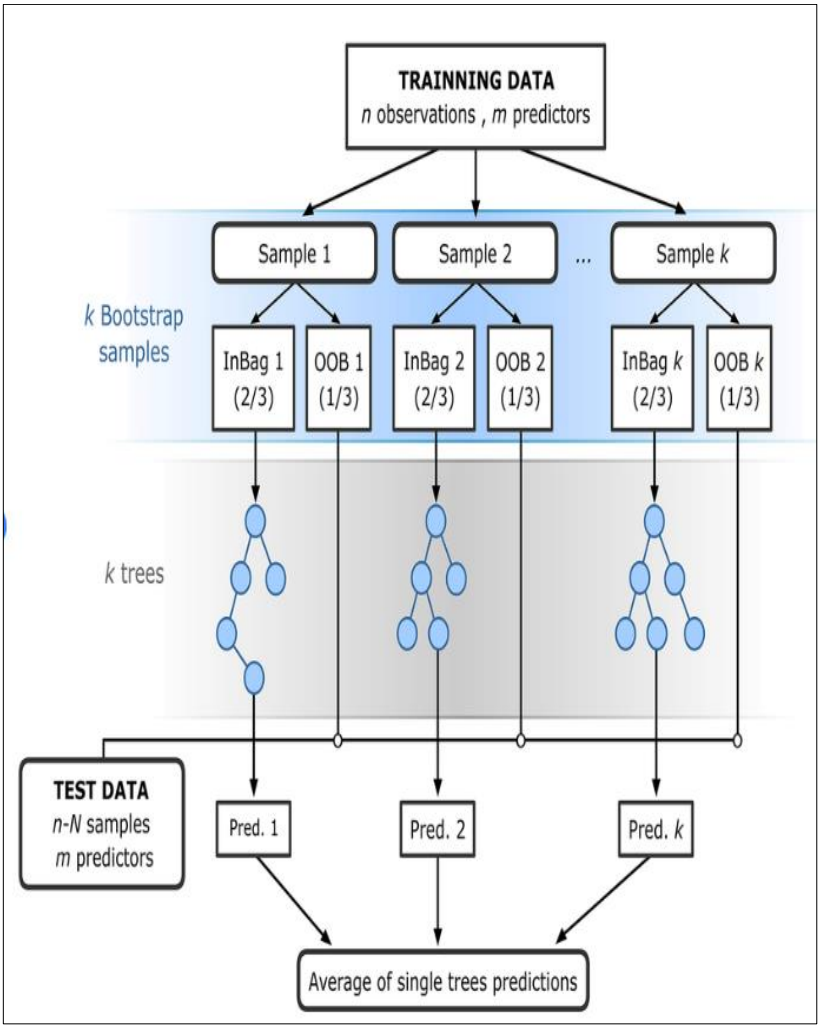
BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N, N_1,..., N_f, where F has f possible values (F_1, ... , F_f)
  for i = 1 to f **do**
    Set the contents of N_i to D_i, where D_i is all instances in N that match
    F_i
    Call BuildTree(N_i)
  end for
end if

In random forest regression randomly select data and forms a subset with replacement and trains on it and this method is called bootstrapping. We can get the prediction for each tree and aggregates and takes the average of all the predictions. This method is called aggregation.

Bootstrapping and aggregation make the model not to be overfit and the flowchart mentioned in Figure 65. The aggregation formula is mentioned in formula (4).

Figure 65

Bootstrapping and aggregation for Random Forest Regression Model



A decision tree in a random forest ensures that has given low bias but gives high variance. But by aggregating all the decision trees in the forest gives the low variance. At each node of the tree a feature of a subset is chosen randomly to calculate mean square error (MSE) for regression tasks.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

In the above formula (5) n is the number of data points, y-i is the actual value of the data points and hat(y-i) is the predicted value and this is the formula helps to understand the error.

XG Boost

XGBoost short for eXtreme Gradient Boosting, is an advanced ensemble machine learning algorithm known for its efficiency and effectiveness across various applications especially in complex predictive modeling tasks like healthcare cost prediction. This algorithm enhances the gradient boosting method by sequentially building an ensemble of decision trees, with each tree learning from the errors of its predecessors. This capability allows XGBoost to effectively model intricate relationships within healthcare data that include variables such as patient demographics, medical histories, treatment types and outcomes.

In XGBoost model the initial predictions are set to a constant value often mean of the target values for regression problems such as cost prediction, which minimizes loss over the training data. The model iteratively improves through a series of steps where gradient of the loss function is calculated for each instance in the training dataset. A new regression tree is then built targeting negative gradients or pseudo-residuals from the previous step. Each tree is constructed using standard decision tree algorithms, focusing on splits that best separate high gradients from low gradients to effectively reduce overall loss. The model is updated by adding the output of the new tree scaled by a learning rate which helps in controlling complexity of the model and prevents overfitting.

The mathematical foundation of XGBoost involves an additive strategy where each subsequent model is trained to predict the gradient of the loss function with respect to the predictions of earlier models. The prediction at each iteration of the model is given by the formula:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i) \tag{6}$$

In the above formula (6), $\hat{y}_i^{(t)}$ represents the prediction at iteration t, $\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration, η denotes the learning rate, and f_t(x_i) is the output of the new tree at iteration t. Trees f_t is developed to minimize the objective function at each step:

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{7}$$

In the above equation (7), ℓ is the loss function that quantifies the difference between actual and predicted values, and Ω is the regularization term that penalizes the complexity to enhance model generalization. The XGBoost algorithm's capability to efficiently handle high-dimensional data, its methodical execution of gradient boosting and advanced regularization options makes it an ideal model for predicting healthcare costs.

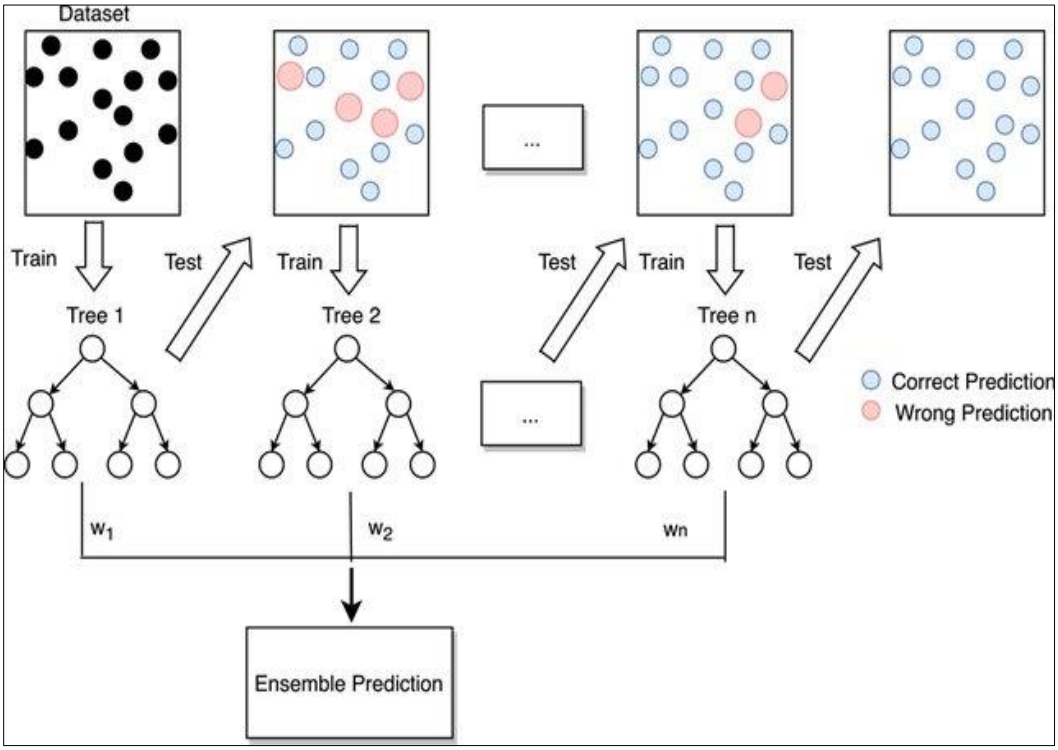
Gradient Boosting Regressor

A widely used boosting technique in machine learning for regression and classification problems is gradient boosting. One type of ensemble learning technique is called "boosting," in which the model is trained successively, with each new model attempting to improve upon the one before it. And we will see how we can employ this technique in our use case which is predict the healthcare price for patients. The concept of boosting involves training weak learners one after the other to improve their predecessors. This implies that the algorithm will always pick up

some knowledge—a tiny step in the right direction, but not correct. The algorithm increases prediction strength as it proceeds by progressively fixing the earlier mistakes.

Figure 66

Gradient boosting machine learning method's flow diagram



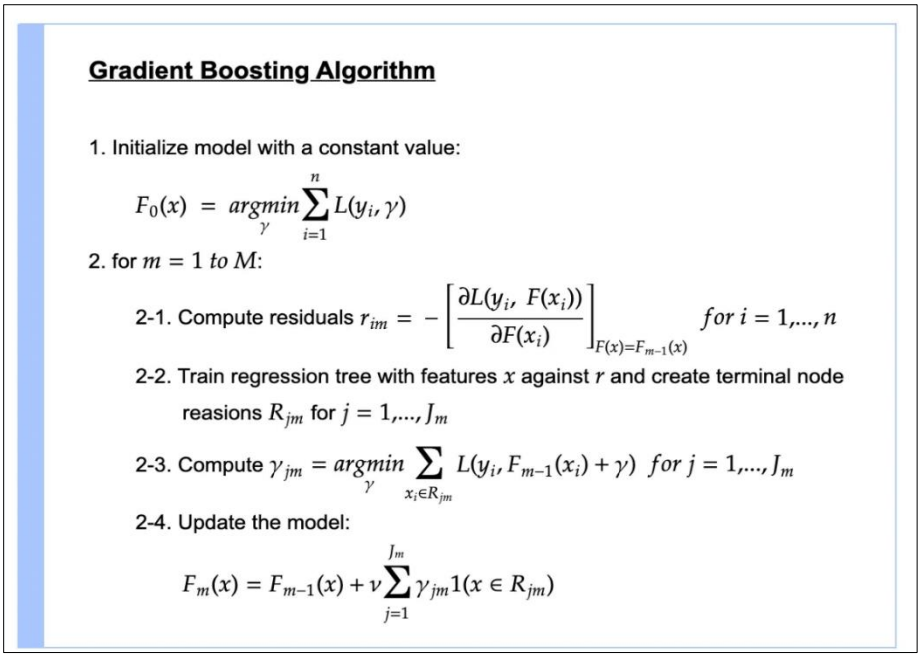
This ensemble method excels in capturing the intricate correlations between patient attributes and healthcare costs, adeptly handling non-linear relationships. Gradient Boosting Regression is a supervised learning technique that uses many decision trees to get precise cost estimates for healthcare. The expected hospital cost is represented by the desired output, Y, and the input characteristics, which include length of stay, gender, operation code, and others, are indicated by X. The decision trees are called regression trees in this regression situation. A new decision tree is trained using the residuals, or mistakes, of the predictions made by the prior model at each iteration. The predictions from the new tree are then added to the predictions from

the previous iteration to update the predictions. New trees are trained on the residuals in an iterative manner until a stopping criterion—such as a minimal error threshold or a limit number of iterations—is satisfied. Gradient Boosting Regression's capacity to handle overfitting is one of its advantages. Without overfitting to individual trees, the ensemble model may capture complex patterns and connections in the data by integrating numerous weak models, each trained on the residuals of the prior model.

In addition, Gradient Boosting Regression has a variety of hyper-parameters that may be adjusted to maximize the model's performance and avoid overfitting, including learning rate, number of estimators, and maximum depth of trees. The number of trees in the ensemble, the complexity of individual trees, and the contribution of each new tree to the overall model are all controlled by these hyper-parameters.

Figure 67

Gradient Boosting Algorithm



The optimal split is found by computing the mean squared error (MSE) of a randomly chosen subset of features at each regression tree node. The most important patterns and interactions between characteristics are captured through this feature selection process, which increases the predictive potential of the model.

4.2Model Support

Analytic/ML Environment, platform, and tools

Healthcare expenses are rapidly increasing each year and predicting healthcare costs have become significant in the US. H-predict leverages data from the government database and hospital records and has used Machine learning models to predict the cost of healthcare expenses. We have used a local system equipped with 8GB RAM and a 64-bit processor with an 8 GB graphic card to run the models. The data we are using for predicting the cost of healthcare is of size less than 2 GB and stored the CSV file in MS Office. We have used Python in Jupyter Notebook to divide the data into training, validation, and testing datasets to evaluate the best prediction model. A wealth of libraries offered by Python gives us an obvious choice to select this platform to run Machine Learning models like Linear Regression, Polynomial Regression, Random Forest, Gradient Boost and XG Boost.

Libraries used for model development.

As Python has plethora of libraries we can use for Machine Learning model development, the libraries we have used for our project for listed in Table 12 below.

Table 12

List of libraries used for model development

	Library	Method	Usage
Scikit-learn	sklearn.linear_model	Linear Regression	To import and evaluate the linear regression model.
	sklearn.metrics	Mean_squared_error, r2_score	Imported metrics to evaluate the model
	sklearn.model_selection	train_test_split	To split the data into training, validation, and testing datasets
	sklearn.decomposition	PCA	For dimensionality reduction
	sklearn.preprocessing	StandardScalar	To standardize the data
	sklearn.ensemble	RandomForestRegressor	Evaluate the random forest regression model.
	sklearn.model_selection	GridSearchCV, RandomizedSearchCV	Tuning the parameters
	sklearn.preprocessing	PolynomialFeatures	To evaluate polynomial regression.
xgboost	xgboost	xgb	To evaluate the XG Boost model.
Pandas	DataFrame	info, drop, head, shape, tail	Read data frames, manipulating, them to see the statistics of data before building models.
Numpy	np.log, np.sqrt	Transformation	To transform the right-skewed data.
Matplotlib and seaborn	Pyplot, sns	Plot graphs	Plotting various graphs to understand data and to plot metrics to compare models.

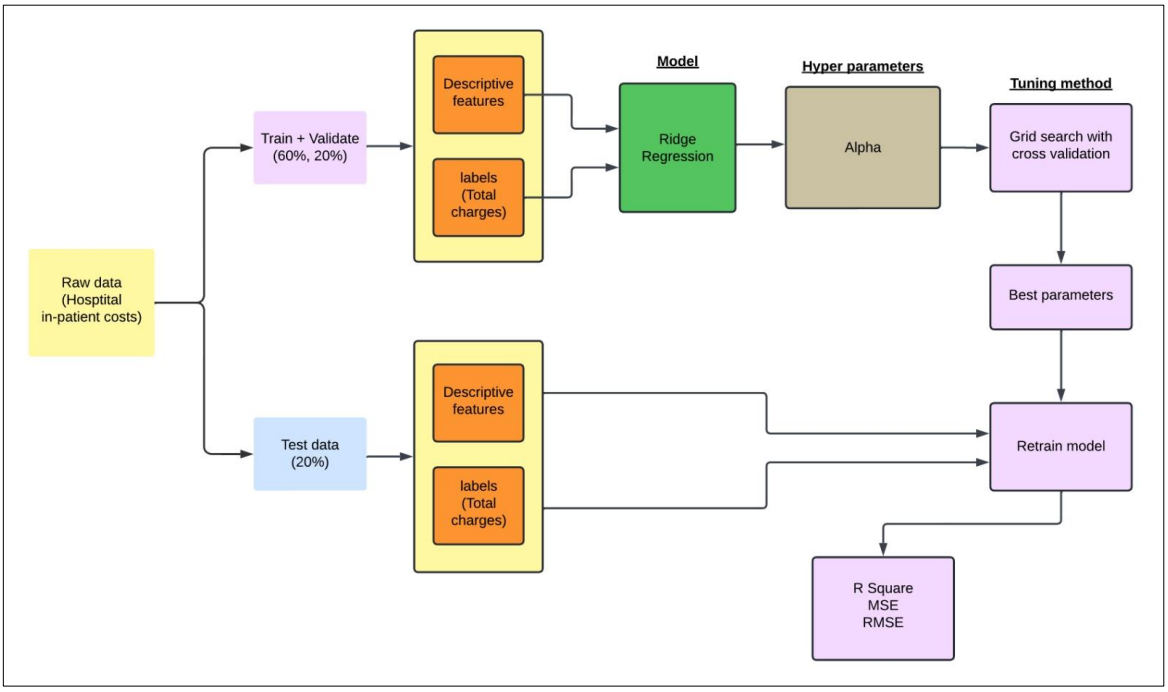
Model architecture and data flow

The raw dataset is divided into training, validation, and testing datasets into 60:20:20 by using train_test_split importing from sklearn.model_selection library. We have built Linear regression, Polynomial regression, Random Forest regression, Gradient Boosting Regressor and Extreme Gradient Boost regression models to predict the target variable healthcare cost. Additionally, to capture the complex relationships between a variety of independent variables such as patient demographics, medical history, length of hospital stay, and degree of illness and the target variable, healthcare cost, each model was meticulously built and adjusted to maximize predictive performance.

Linear regression. The prepared dataset has been used to build a linear regression model to maintain an appropriate distribution for an objective assessment of the model. By importing LinearRegressor from sklearn.linear_model we build the individual model. When predicting the target variable from input features this model acts as a baseline. To get better performance we have used the regularization method by implementing the Ridge regression model importing Ridge () from sklearn.linear_model. The main hyperparameter is alpha varied with multiple values like 0.01,0.1,1,10,100,1000 used and GridsearchCV is implemented with 5-fold Cross-validation to minimize negative mean squared error. This aims to enhance the model's prediction performance and also helps to generalize the relationships. After grid search, the best parameters were identified, and the optimal value fits the model for future predictions. Then the model is evaluated using metrics like r2 value, root mean squared error(RMSE) and mean absolute error(MSE). System architecture diagram for linear regression model is shown in Figure 68 below.

Figure 68

System Architecture Diagram for Linear Regression

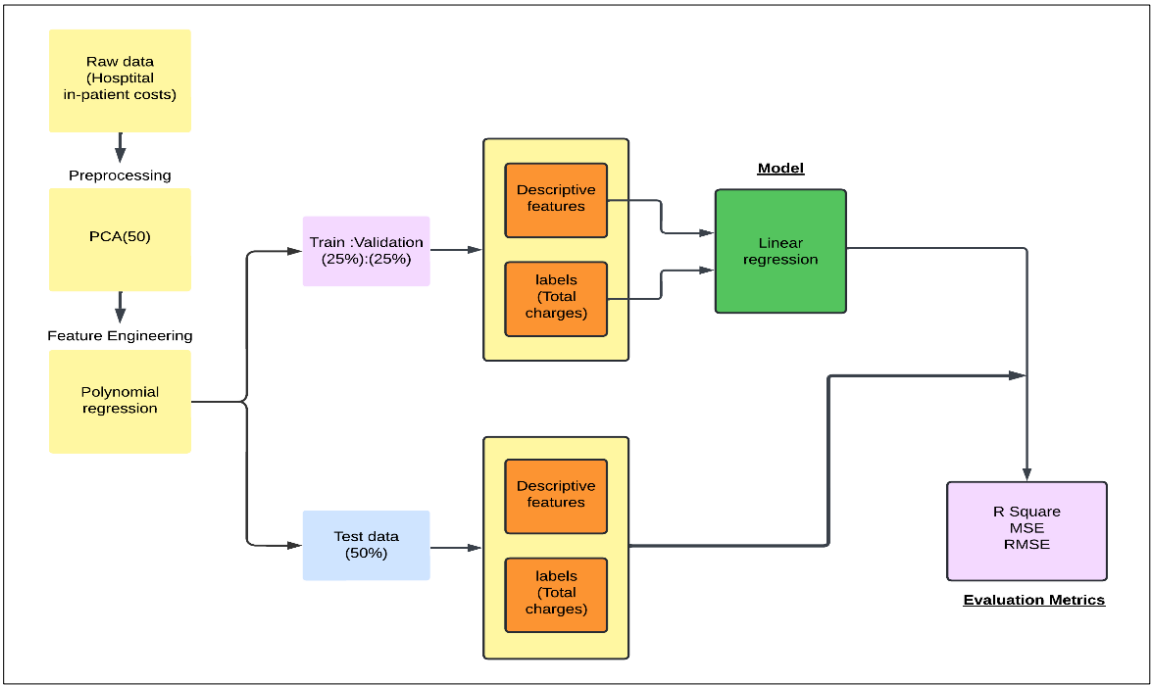


Polynomial Regression. Polynomial regression emerged as a compelling approach to model the non-linear relationships between patient attributes and healthcare costs. However, due to computational constraints and memory limitations, we had to strategically allocate our data for training and testing. Specifically, we utilized 25% of the available data for training and 50% for testing purposes. This decision was driven by the kernel's inability to handle larger datasets, as it would trigger memory errors and halt the computation process. We used the PolynomialFeatures and LinearRegression modules of the robust scikit-learn framework to construct polynomial regression. The original input data were converted into polynomial features using the PolynomialFeatures converter, which allowed the linear regression model to capture non-linear correlations. We chose a polynomial degree of two to prevent too complicated models while still being able to capture non-linear trends. Furthermore, we used principal Component Analysis

(PCA) to minimize computational load and overfitting risk by reducing the dimensionality of the feature space and keeping just the top 50 principal components.

Figure 69

System Architecture Diagram for Polynomial Regression

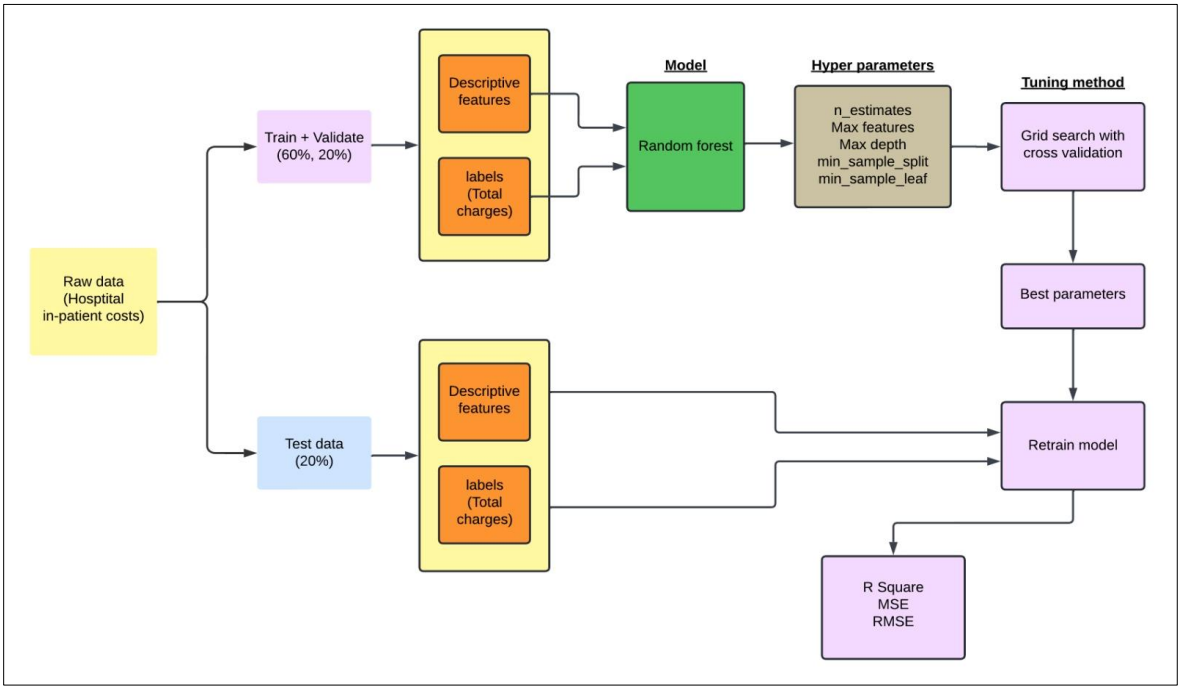


Random Forest. After dividing the preprocessed data into training, validation and testing datasets as mentioned above, initializing RandomForestRegressor importing from sklearn.ensemble library. This is suitable for non-linear relationships. This model can also be used to handle automatic interactions between features with no need for feature engineering. Key parameters are n_estimators and random_state. Then the model is trained by fitting the model and making predictions for the test data. Evaluate by using metrics like MSE, r2 value and RMSE. A hyperparameter tuned random forest model was built then to get the optimized model and increase the prediction accuracy through GridsearchCV with the value of 5 for crossvalidation(CV). Hyperparameters are taken in the model like n_estimators(number of trees),

max_features(maximum features taken), max_depth(tree maximum depth), min_samples_split(minimum number of samples to split the node internally), min_samples_leaf(minimum number of samples required to be at leaf). After this the model trained and evaluated based on the metrics like Mean squared error and R-squared score. The trained models are saved in the pickle file for future tasks to deploy easily. The system architecture and data flow are shown in Figure 70 below.

Figure 70

System architecture and data flow for Random Forest regression model

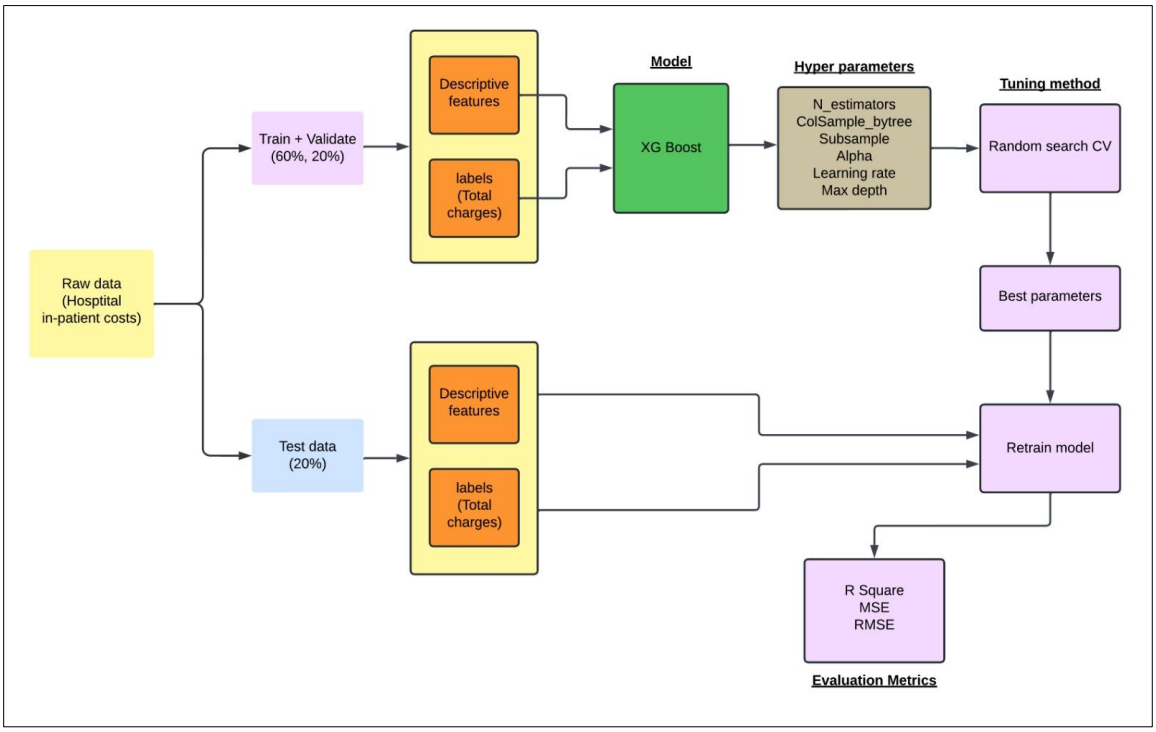


XG Boost. The XG Boost is the short name for extreme gradient boost here has been used to predict the costs based on different features of hospital inpatient discharge cost dataset. This is an extended version of gradient boosting designed for better performance. It is a standalone library which we have imported xgb from xgboost to build the baseline model after dividing data into training, validation and testing sets into 60:20:20 ratio. To do hyperparameter

tuning by importing RandamizedSearchCV from sklearn.model_selection with hyperparameters like max depth, learning rate, N_estimators, Colsample_bytree, Subsample, Alpha taking 3-fold Cross validation (CV) to reduce negative mean squared error. RandomizedsearchCV fixed the number of parameters setting instead of trying every parameter combination like in GridSearchCV. RandomizedSearchCV explores larger areas faster in cases of large data. This is a parallel process which enhances commutating efficacy. The performance of the model is evaluated by metrics like Mean squared error and R-squared score importing from sklearn.metrics. The architecture and dataflow are shown in Figure 71 below.

Figure 71

System architecture and data flow for XG Boost regression model

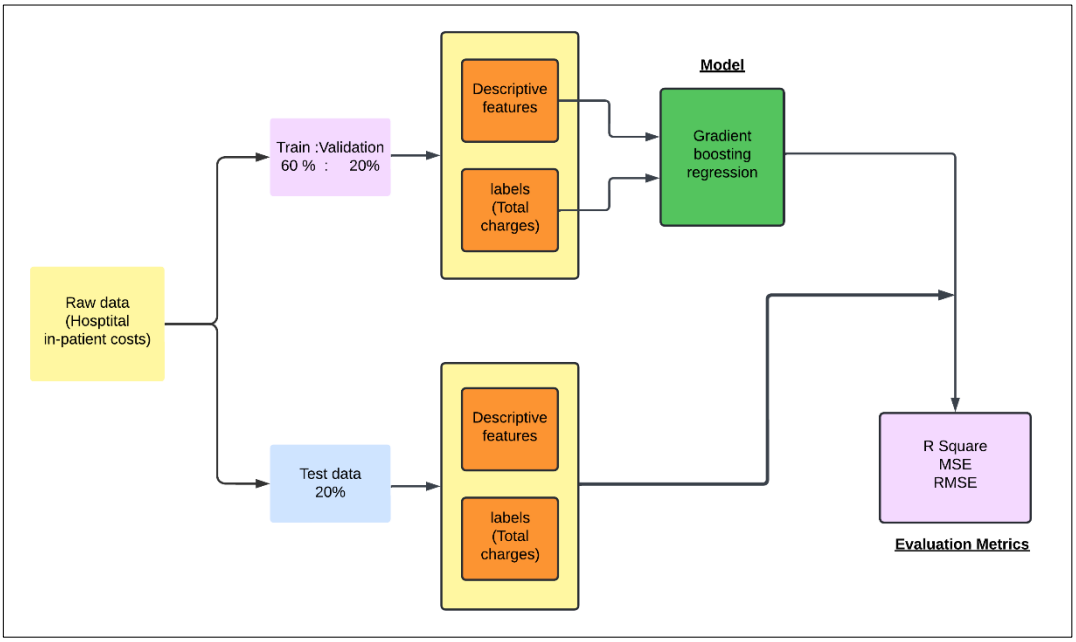


Gradient Boosting Regression. Gradient Boosting is a one of the powerful machine learning methods in regression problems which constructs an additive model in forward steps

enabling the optimization of arbitrary loss function. In regression, decision trees are fitted on negative gradient in every stage of loss function. GradientBoostingRegressor is imported from sklearn.ensemble library to build the individual model. After dividing the data into training, validation and testing data into 60:20:20 ration using fit method the model is trained. This is iterative process where it builds decision trees to reduce loss function. To get more accurate predictions hyperparameter tuning can used by GridSearchCV for optimization with hyperparameters n_estimators, learning rate, max_depth, min_samples_split, min_samples_leaf to reduce negative mean squared error. Then the model is trained and fit before evaluating the model. By using evaluation metrics like MSE, RMSE and r-squared importing metrics from sklearn.metrics. The architecture is shown in Figure 72. But due to computational limitations we could not run the hyperparameter tuned for gradient boosting regression model.

Figure 72

System architecture and data flow Gradient boosting regression model



4.3 Model Comparison and Justification

Healthcare cost prediction models such as Linear Regression, Polynomial Regression, Random Forest, Gradient Boosting Regressor and XGBoost each contribute to different data characteristics and problem complexities within healthcare analytics. Linear Regression is preferred for its simplicity and interpretability in linear scenarios making it ideal for smaller datasets, whereas Polynomial Regression extends this to fit non-linear patterns even though it risks overfitting. Random Forest is better at handling large, complex datasets by leveraging multiple decision trees to improve prediction accuracy and robustness suitable for datasets rich in features like demographic and medical histories. Gradient Boosting Regressor combines weak models for accurate predictions, particularly effective in large-scale regression and classification tasks. XGBoost excels in scenarios requiring high performance and speed thus optimizing gradient boosting with advanced features like tree pruning and regularization to manage large datasets effectively.

These models not only differ in their approach but also in their inherent advantages and disadvantages as detailed in the accompanying table. Linear and Polynomial Regression are straightforward and flexible but struggle with non-linear data and outlier sensitivity. Random Forest and XGBoost, while robust and powerful for complex datasets, demand careful parameter tuning and are computationally intensive. Gradient Boosting Regressor, though providing high performance and managing missing data internally can be slow and less efficient on very large datasets. The selection of these models is based on their ability to address specific needs of healthcare cost prediction from managing simple linear relationships to tackling complexity and dimensionality of extensive healthcare data which is detailed in Table 13 below.

Table 13

Model comparative analysis

	Advantages	Disadvantages
Linear Regression	• Easy to implement and understand • Requires minimal computational resources • Results are easily interpretable • Provides clear feature importance	• Limitation if relationship isn't linear • Outliers can heavily impact results • May miss complex patterns in data
Polynomial Regression	• Fits a wide range of curvatures • Suitable for non-linear relationships	• Model complexity grows exponentially • Risk of extreme values outside training data range
XGBoost	• Handles large datasets efficiently • Includes L1 and L2 regularization • Built-in routines improve model robustness	• Requires careful parameter tuning • Challenging to deploy in real-time applications
Random Forest	• Works well with many features • Captures interactions without transformation • Provides probabilities for classification and continuous outputs	• Less interpretable than decision trees • With high features can tend to overfit • Training can be expensive with large datasets
Gradient Boosting Regressor	• Combines weak models for accurate predictions to give high performance • Suitable for regression and classification tasks • Automatically provides feature importance scores • Manages missing data internally	• Slow due to sequential tree building • Parameters need careful tuning • Harder to interpret with increased trees • Less efficient on very large datasets

Justifications for Models

Linear regression is ideal when the relationship between variables is linear offering

simplicity and easy interpretation making it a preferred choice for baseline models. For instance, it can predict healthcare costs based on factors like age or health parameters. Polynomial regression is suitable when relationships are non-linear but can be approximated by polynomials, providing a more nuanced understanding of trends. For example, it can model healthcare costs considering interactions between variables like age and comorbidities. Random Forest excels in handling overfitting with large datasets by averaging multiple decision trees useful in predicting patient treatment costs from different features. XGBoost is valued for its performance and speed particularly in structured data handling various data types efficiently making it useful for real-time prediction tasks, like treatment cost prediction using complex healthcare data. Model choice depends on the dataset characteristics, problem complexity, accuracy needs, and computational efficiency.

4.4 Model Evaluation Methods

In our endeavor to accurately predict healthcare costs, we developed a comprehensive approach involving the implementation of five distinct regression models. To evaluate the performance of these models and ensure precise cost estimations, we employed several evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). These metrics play a crucial role in assessing the models' ability to predict healthcare costs accurately, with lower MSE, RMSE, and MAE values indicating better performance, and a higher R^2 value, ranging from 0 to 1, suggesting a better fit between the predicted costs and the actual costs. We can precisely evaluate the effectiveness of our regression models, make necessary adjustments, and eventually provide accurate and dependable cost estimates by using these evaluation criteria. Accurate cost

predictions are crucial for the healthcare industry's effective resource allocation and well-informed decision-making.

Mean Squared Error (MSE)

The average of the squares of the errors, or the average squared difference between the estimated values and the actual value, is measured by the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a process for estimating an unobserved variable) in statistics. As the expected value of the squared error loss, MSE is a risk function. Due to randomness or the estimator's failure to take into consideration data that may result in a more accurate estimate, the MSE is nearly always strictly positive (rather than zero). The average loss on an observed data set, or the empirical risk, can be referred to as an estimate of the real MSE in machine learning, more especially in empirical risk minimization.

Figure 72

MSE Formula

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2. \tag{8}$$

Root Mean Squared Error (RMSE)

One of the most used metrics for assessing the performance of forecasts is the Root Mean Squared Error (RMSE), often known as the root mean square deviation. It calculates the mean difference between the values that a model predicts and the actual values. It offers an estimate of the accuracy—or how effectively the model can anticipate the desired result. A model is considered better if its Root Mean Squared Error value is less. A Root Mean Squared Error value

of zero would indicate a perfect model, or hypothetical model, which would always predict the exact predicted value. The residual (difference between the truth and the predicted) for each data point, its norm, the mean of the residuals, and the square root of the mean are the steps involved in computing RMSE. Since it requires accurate measurements at each anticipated data point, root mean square error (RMSE) is frequently employed in supervised learning applications.

Root means square error can be expressed as RMSE formula in figure 73, where N is the number of data points, y(i) is the i-th measurement, and ŷ(i) is its corresponding prediction.

Figure 73

Root Mean Squared Error (RMSE) formula

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}}, \tag{9}$$

R-squared (R^2)

The percentage of the variation in the dependent variable that can be predicted from the independent variable(s) is known as the coefficient of determination in statistics. It is represented by the symbols R^2 or r^2 and is pronounced "R squared". R-squared shows how much of the variance in the dependent variable can be accounted for by the model's independent variables. On a scale of 0 to 1, 1 denotes an ideal match. An indicator of a model's quality of fit is its R2 value. The statistical measure of how closely the regression predictions match the actual data points in regression is called the R2 coefficient of determination. The regression predictions fully fit the data when the R2 value is 1. In figure 74, we can see the formula for R squared Where RSS = sum of squares of residuals and TSS = total sum of squares.

Figure 74

R-squared *formula*

$$R^2 = 1 - \frac{RSS}{TSS} \quad (10)$$

We can carefully evaluate the performance of the built models and choose the best method for properly estimating healthcare expenses by using these assessment measures.

4.5 Model Validation and Evaluation

Linear Regression

Baseline Model. To establish a baseline for comparison, we implemented a linear regression model with a 60:20:20 split for training, validation, and testing data. The evaluation metrics for this baseline model revealed a Root Mean Squared Error (RMSE) of approximately 40 across the training, testing, and validation sets, indicating a consistent level of error. Furthermore, the Mean Squared Error (MSE) values were 1616 for the training data, 1617 for the validation set, and 1621 for the test set, as depicted in Figure 75. The model's coefficient of determination (R^2) score, which measures the proportion of variance in the target variable that can be explained by the independent variables, was 0.75 for the training, testing, and validation sets. This R^2 value suggests that the baseline linear regression model captures a substantial portion of the variability in the healthcare cost data, providing a solid foundation for further model development and refinement. The figure also represents the actual versus predicted healthcare costs plot for the baseline linear regression model across the training, validation, and test datasets. While the model captures some underlying patterns, the scattered data points

around the perfect fit line indicate that it still exhibits deviations from the true healthcare costs, suggesting room for improvement in prediction accuracy.

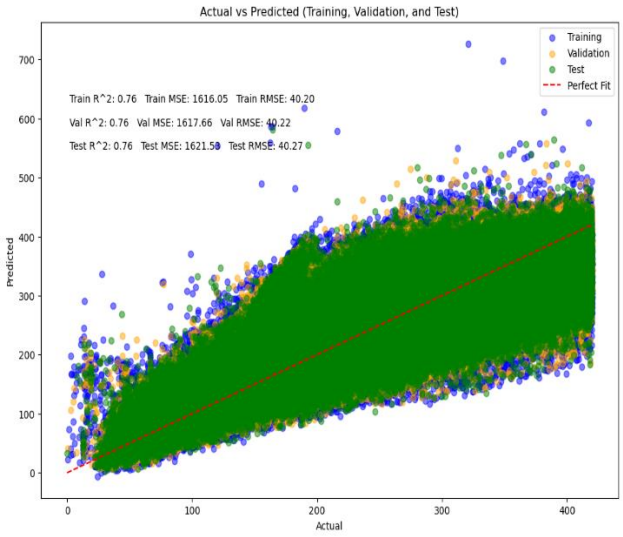
Figure 75

Linear Regression results

```
Evaluation for Linear Regression:
Train Set :
Mean Squared Error (MSE): 1616.0536281829723
Root Mean Squared Error (RMSE): 40.20016950440598
R-squared (R^2) Score: 0.7593232863658076

Validation Set:
Mean Squared Error (MSE): 1617.6616515690644
Root Mean Squared Error (RMSE): 40.220164738213896
R-squared (R^2) Score: 0.7591227680578773

Test Set:
Mean Squared Error (MSE): 1621.5280787170364
Root Mean Squared Error (RMSE): 40.26820183118482
R-squared (R^2) Score: 0.7586097970692673
```



Hyper Parameter Tuned Model or Ridge Regression. To optimize the model's performance and fine-tune its hyperparameters, we conducted an exhaustive grid search using the GridSearchCV technique with 5-fold cross-validation. This approach allowed us to systematically explore the hyperparameter space and identify the optimal configuration. Specifically, we focused on tuning the alpha parameter, which acts as a smoothing factor in the model. After an extensive grid search, the optimal value of alpha was found to be 1, resulting in improved model performance. The tuned model, trained with the optimized smoothing parameter, achieved a Mean Squared Error (MSE) of 1621, a Root Mean Squared Error (RMSE) of 40, and an R-squared (R^2) value of 0.75, as illustrated in figure 76. These evaluation metrics indicate that the hyperparameter tuning process enhanced the model's ability to capture the

underlying patterns in the data, leading to more accurate healthcare cost predictions while maintaining consistency with the baseline model's performance.

Figure 76

Linear Regression Hyperparameter Tuned Model

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
Evaluation for Ridge Regression or hyperparameter tuned linear regression :
Best Parameters: {'alpha': 1}
Mean Squared Error (MSE) of Best Fit: 1621.5254168382644
Root Mean Squared Error (RMSE): 40.26816877930091
R-squared (R^2) Score: 0.7586101933321907
```

Polynomial Regression

Baseline Model. To establish a baseline for comparison and capture potential non-linear relationships in the data, we implemented a polynomial regression model of degree 2 with a 60:20:20 split for training, validation, and testing data. The evaluation metrics for this baseline polynomial regression model revealed a consistent Root Mean Squared Error (RMSE) of approximately 33 across the training, testing, and validation sets, indicating a reasonable level of error. Furthermore, the Mean Squared Error (MSE) values were 1086 for the training data, 1098 for the validation set, and 1100 for the test set, as depicted in Figure 98. The model's coefficient of determination (R^2) score, which measures the proportion of variance in the target variable that can be explained by the independent variables, was 0.83 for the training, testing, and validation sets. This R^2 value suggests that the baseline polynomial regression model captures a significant portion of the variability in the healthcare cost data, including potential non-linear patterns, providing a solid foundation for further model development and refinement.

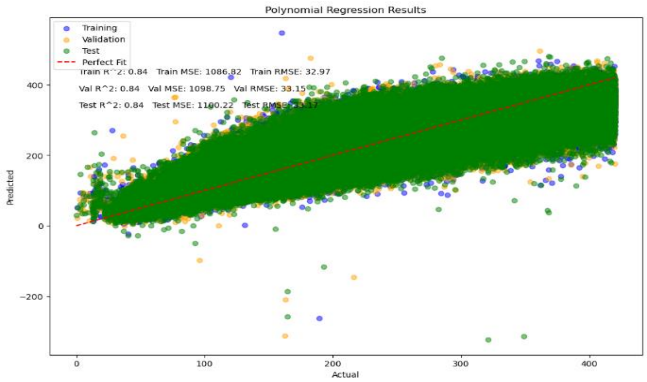
Figure 77

Polynomial Regression

```
Train Set:
Mean Squared Error (MSE): 1087.7724720505926
Root Mean Squared Error (RMSE): 32.98139584751671
R-squared (R^2) Score: 0.837719291915253

Validation Set:
Mean Squared Error (MSE): 1099.4644916088134
Root Mean Squared Error (RMSE): 33.15817382801431
R-squared (R^2) Score: 0.836122337352873

Test Set:
Mean Squared Error (MSE): 1100.8874468029278
Root Mean Squared Error (RMSE): 33.179623970185794
R-squared (R^2) Score: 0.8362949247742417
```



Random Forest

Baseline Model. The RandomForestRegressor function from the Scikit-learn module is used to generate the baseline model for RF. To establish a baseline for comparison and leverage the power of ensemble learning, we implemented a Random Forest model with 100 estimators and a 60:20:20 split for training, validation, and testing data. The evaluation metrics for this baseline Random Forest model revealed a Root Mean Squared Error (RMSE) of approximately 9 for the training data, and 24 for both the test and validation sets. Furthermore, the Mean Squared Error (MSE) values were 84 for the training data, 594 for the validation set, and 597 for the test set, as depicted in Figure 78. The model's coefficient of determination (R^2) score, which measures the proportion of variance in the target variable that can be explained by the independent variables, was 0.98 for the training set, indicating an excellent fit on the training data. However, the R^2 values for the validation and test sets were 0.91, suggesting a slightly lower, but still satisfactory, performance on unseen data. This baseline Random Forest model exhibits the ability to capture complex non-linear patterns and interactions in the healthcare cost data, providing a strong foundation for further model refinement and optimization.

Figure 78

Random Forest for n_estimators = 100

```
Evaluation for Random forest regression with n_estimatoros = 100 :
Random Forest - Train Set:
Mean Squared Error (MSE): 83.94441776289275
Root Mean Squared Error (RMSE): 9.162118628510152
R-squared (R^2) Score: 0.9874982697091403

Random Forest - Validation Set:
Mean Squared Error (MSE): 594.7306317893311
Root Mean Squared Error (RMSE): 24.38709970023765
R-squared (R^2) Score: 0.9114418839083868

Random Forest - Test Set:
Mean Squared Error (MSE): 597.5103670006454
Root Mean Squared Error (RMSE): 24.444025180003504
R-squared (R^2) Score: 0.911051093942437
```

The baseline Random Forest model with n_estimators = 10 demonstrates a high R-squared (R^2) score of 0.98 on the training set, indicating a good fit to the training data. However, its performance on the validation and test sets is relatively poorer, with R^2 scores of 0.90 and 0.90, respectively, accompanied by higher Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values. This discrepancy in performance between the training and validation/test sets suggests potential overfitting or a mismatch between the data distributions, necessitating further model tuning or evaluation based on the specific requirements of the hospital cost prediction use case.

Figure 79

Random Forest for n_estimators = 10

```
Evaluation for Random forest regression with n_estimatoros = 10 :
Random Forest - Train Set:
Mean Squared Error (MSE): 118.6110197485897
Root Mean Squared Error (RMSE): 10.890868640681958
R-squared (R^2) Score: 0.9823354188647886

Random Forest - Validation Set:
Mean Squared Error (MSE): 654.1232587331735
Root Mean Squared Error (RMSE): 25.575833490488115
R-squared (R^2) Score: 0.9025980496231842

Random Forest - Test Set:
Mean Squared Error (MSE): 656.0076304478625
Root Mean Squared Error (RMSE): 25.61264590876668
R-squared (R^2) Score: 0.9023428473941634
```

Hyper Parameter Tuned Model. The Random Forest model initially seemed to overfit training data. To address this issue, hyperparameter tuning was performed to enhance the model's accuracy. The optimal hyperparameter values were determined as n_estimators=10, max_depth=10, min_samples_split=5, cv=3, scoring='neg_mean_squared_error', and n_jobs=-1. With these optimized hyperparameters, the Random Forest model achieved the performance metrics as depicted in Figure 80. The tuned model demonstrated a significantly improved R-squared score of 0.95 on the training set, indicating a strong fit to the training data. However, the model's performance on the validation and test sets, while better than the initial baseline, still exhibits room for improvement, with R-squared scores of 0.88 and 0.88, respectively, and relatively higher MSE and RMSE values. These results suggest that the hyperparameter tuning process has enhanced the model's predictive capabilities on the dimensionally reduced dataset, but further refinement or exploration of alternative modeling approaches may be warranted to achieve optimal performance for the hospital cost prediction use case. The tuned Random Forest model achieved an impressive R-squared (R^2) score of 0.95 on the training set, indicating an excellent fit to the training data. However, this high R^2 score, coupled with relatively lower R^2 scores of 0.88 and 0.88 on the validation and test sets, respectively, may suggest potential overfitting on the training data itself, where the model has learned the training data's noise and patterns too closely, leading to poor generalization on the unseen validation and test sets. This discrepancy between the high training performance and lower validation/test performance could be attributed to the model overfitting to the training set, necessitating further techniques such as regularization, ensemble methods, or adjustments to the model's complexity to enhance generalization capabilities and ensure a more robust and reliable predictive model for the hospital cost prediction use case.

Figure 80

Random Forest Hyperparameter Tuned Model CV = 3

```
Train Set:
Mean Squared Error (MSE): 270.8346524409707
Root Mean Squared Error (RMSE): 16.457054792427794
R-squared (R^2) Score: 0.9596649560689138

Validation Set:
Mean Squared Error (MSE): 751.8578661537214
Root Mean Squared Error (RMSE): 27.420026735102237
R-squared (R^2) Score: 0.888044918764468

Test Set:
Mean Squared Error (MSE): 753.9975206226294
Root Mean Squared Error (RMSE): 27.459015288655735
R-squared (R^2) Score: 0.88775549624995
```

To further refine the Random Forest model's performance for the hospital cost prediction use case, a new iteration was undertaken with an increased cross-validation fold of cv=5, while maintaining the previously optimized hyperparameter values: n_estimators=10, max_depth=10, and min_samples_split=5. The scoring metric remained 'neg_mean_squared_error', and n_jobs=-1 was used for parallel processing. We see that our model seemed to give similar results as in previous hypertuned model as shown in figure 81

Figure 81

Random Forest Hyperparameter Tuned Model CV = 5

```
Train Set:
Mean Squared Error (MSE): 487.88824291704145
Root Mean Squared Error (RMSE): 22.08819238681702
R-squared (R^2) Score: 0.9273394540389972

Validation Set:
Mean Squared Error (MSE): 797.8072700874644
Root Mean Squared Error (RMSE): 28.245482295182434
R-squared (R^2) Score: 0.8812028419814678

Test Set:
Mean Squared Error (MSE): 799.9298890131954
Root Mean Squared Error (RMSE): 28.283031821450745
R-squared (R^2) Score: 0.8809177338501397
```

Compared to the previous iteration, this model iteration exhibits a slightly lower R-squared score on the training set (0.92), suggesting a potential reduction in overfitting. However, the validation and test set performances remained same, with R-squared scores of 0.88 and 0.88, respectively. While the increased cross-validation fold aimed to enhance the model's generalization capabilities, the results indicate that further fine-tuning or exploration of alternative modeling approaches may be necessary to achieve optimal performance on the hospital cost prediction task.

Gradient Boosting Regression

Baseline Model. In an effort to explore alternative modeling approaches, we employed the Gradient Boosting Regression algorithm from the sklearn.ensemble module. The dataset was divided into a 60:20:20 split for training, validation, and testing, respectively. The baseline model was trained using the default parameters of the GradientBoostingRegressor. The performance metrics obtained for the Gradient Boosting Regression model are shown in figure 82. The Gradient Boosting Regression model demonstrates consistent performance across the training, validation, and test sets, with R-squared scores of 0.81. These scores indicate a good fit to the data, with approximately 82% of the variance in hospital costs being explained by the model.

However, it is worth noting that the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values are relatively higher compared to the previously explored Random Forest models, suggesting potential room for improvement in terms of prediction accuracy.

While the Gradient Boosting Regression model provides a reasonable baseline performance, further hyperparameter tuning or ensemble techniques may be necessary to

enhance its predictive capabilities and achieve optimal performance for the hospital cost prediction use case.

Figure 82

Gradient Boosting Regression

```
Train Set :
Mean Squared Error (MSE): 1224.8352307994562
Root Mean Squared Error (RMSE): 34.99764607512134
R-squared (R^2) Score: 0.8175869210332826

Validation Set:
Mean Squared Error (MSE): 1223.9091273629397
Root Mean Squared Error (RMSE): 34.9844126342424
R-squared (R^2) Score: 0.8177543230613591

Test Set:
Mean Squared Error (MSE): 1232.253829996663
Root Mean Squared Error (RMSE): 35.10347318993753
R-squared (R^2) Score: 0.8165594503177431
```

XGBoost

Baseline Model. In pursuit of further enhancing the predictive performance for the hospital cost prediction use case, we explored the Extreme Gradient Boosting (XGBoost) algorithm, a powerful and efficient implementation of gradient boosting decision trees. The XGBoost library from Python was utilized, and the dataset was divided into a 60:20:20 split for training, validation, and testing, respectively. The XGBoost model was trained with the hyperparameters like objective ='reg:squarederror' (Objective function for regression tasks, minimizing the squared error), colsample_bytree = 0.3 (Subsample ratio of columns for each tree), learning_rate = 0.1 (Step size shrinkage to prevent overfitting), max_depth = 5 (Maximum depth of the decision trees), alpha = 10 (L1 regularization term on weights), n_estimators = 100 (Number of boosting iterations). The XGBoost model's performance on the hospital cost prediction task yielded the evaluation metrics outlined in Figure 83

Figure 83

XGBoost Baseline Model

```
XGBoost - Train Set:
Mean Squared Error (MSE): 1072.5170413814017
Root Mean Squared Error (RMSE): 32.749305967934674
R-squared (R^2) Score: 0.8402714660363257

XGBoost - Validation Set:
Mean Squared Error (MSE): 1075.464585516861
Root Mean Squared Error (RMSE): 32.79427671891638
R-squared (R^2) Score: 0.8398583955057528

XGBoost - Test Set:
Mean Squared Error (MSE): 1080.254316913756
Root Mean Squared Error (RMSE): 32.86722253117467
R-squared (R^2) Score: 0.8391869914562757
```

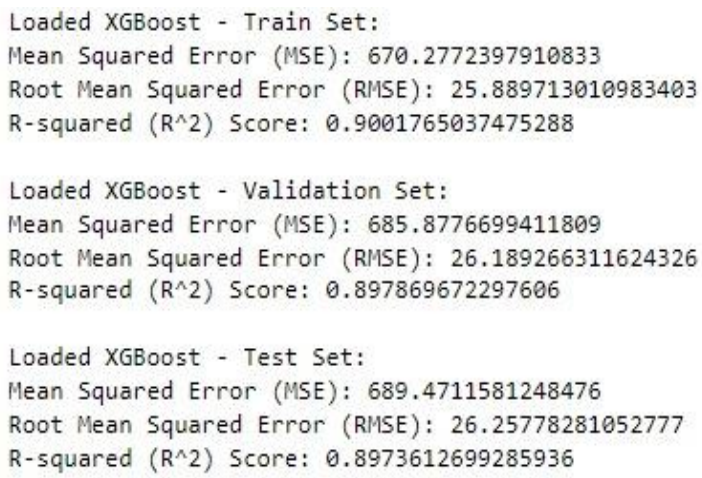
Hyper Parameter Tuned Model. In an effort to further optimize the performance of the Extreme Gradient Boosting (XGBoost) model for the hospital cost prediction use case, we employed hyperparameter tuning to identify the optimal set of parameters that yield accurate and reliable results. The XGBoost model was initialized with a set of base hyperparameters, including the objective function, column and row subsampling rates, learning rate, maximum tree depth, regularization parameter, and the number of boosting iterations like objective ='reg:squarederror' (Objective function for regression tasks, minimizing the squared error), colsample_bytree = 0.3 (Subsample ratio of columns for each tree), learning_rate = 0.1 (Step size shrinkage to prevent overfitting), max_depth = 5 (Maximum depth of the decision trees) and alpha = 10 (L1 regularization term on weights) and n_estimators = 100 (Number of boosting iterations). Subsequently, a hyperparameter tuning process was undertaken, where various combinations of the following hyperparameters were explored of max_depth: np.arange(3, 8), learning_rate of np.logspace(-2, -0.5, num=4), n_estimators: [50, 100, 200], colsample_bytree: np.linspace(0.3, 1, num=3), subsample: [0.7, 1] and alpha: [0, 10, 100]. The dataset was divided

into a 60:20:20 split for training, validation, and testing, respectively. The hyperparameter tuning process involved fitting 3 folds for each of the 50 candidate parameter combinations, totaling 150 fits. This approach aimed to identify the optimal set of hyperparameters that maximized the model's performance.

The hyperparameter-tuned XGBoost model demonstrated a substantial improvement in performance compared to the baseline XGBoost model, achieving lower Mean Squared Error (MSE) values and higher R-squared (R^2) scores across the training, validation, and test sets. The R-squared scores, ranging from 0.903 to 0.907, as shown in figure 84 indicate that the model explains approximately 90% of the variance in hospital costs, which is a significant portion of the variability. This improvement in predictive performance can be attributed to the identification of optimal hyperparameters through the tuning process, allowing the model to better capture the complex patterns and relationships within the data.

Figure 84

XGBoost Hyperparameter Tuned Model



Comparison of Model Evaluation Results

Upon analyzing all the models and analyzing all the results and summarizing them in Table 14, it is evident that the XGBoost Hyperparameter Tuned Model outperforms the other models in terms of generalization performance on both the testing and training data.

Table 14

Comparisons of all the Models

Models	MSE	RMSE	Testing R-Squared	Training R-Squared
Linear Regression	1621	40	0.75	0.75
Linear Regression Hyperparameter Tuned Model	1621	40	0.75	-
Polynomial Regression	1100	33	0.83	0.83
Random Forest for n_estimators = 100	597	24	0.91	0.98
Random Forest for n_estimators = 10	656	25	0.9	0.98
Random Forest Hyperparameter Tuned Model CV = 3	753	27	0.88	0.95
Random Forest Hyperparameter Tuned Model CV = 5	799	28	0.88	0.92
Gradient Boosting Regression	1232	35	0.81	0.81
XGBoost Baseline Model	1080	33	0.84	0.84
XGBoost Hyperparameter Tuned Model	689	26	0.90	0.90

While the Random Forest models appear to have lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values on the testing data, their high training R-squared scores (0.98 and above) indicate potential overfitting. This means that the Random Forest models have learned the training data patterns too well, including noise, leading to poor generalization on unseen test data.

On the other hand, the XGBoost Hyperparameter Tuned Model exhibits a testing R-squared of 0.9, which is comparable to its training R-squared of 0.9, suggesting that the model has effectively captured the underlying patterns in the data without overfitting. Additionally, the XGBoost Hyperparameter Tuned Model achieves the one of the lowest RMSE of 26 while comparing to other models, indicating better prediction accuracy.

The success of the XGBoost Hyperparameter Tuned Model can be attributed to several factors. Firstly, the XGBoost algorithm is a highly efficient and robust implementation of gradient boosting decision trees, which has been proven to be effective in various machine learning tasks, including regression problems. Secondly, the hyperparameter tuning process has played a crucial role in identifying the optimal combination of hyperparameters, such as the learning rate, maximum tree depth, and regularization parameters, which have helped to prevent overfitting and enhance the model's generalization capabilities. Furthermore, the XGBoost algorithm's ability to handle sparse data, automatically handle missing values, and its built-in parallel processing capabilities make it a powerful tool for tackling complex regression problems like hospital cost prediction. Additionally, XGBoost's feature importance analysis can provide valuable insights into the most influential features contributing to the cost predictions, which can aid in further feature engineering and model optimization.

While the Random Forest models demonstrated strong predictive capabilities, with impressive testing R-squared scores of 0.91 for 100 estimators and 0.9 for 10 estimators, they presented significant computational challenges in terms of time and space complexity. Training Random Forest models, especially with many estimators, can be computationally expensive and resource-intensive, often requiring substantial memory and processing power.

The XGBoost model not only exhibited superior prediction accuracy with a testing R-squared score of 0.9, but it also offered significant computational advantages. The XGBoost algorithm is highly optimized for efficient memory usage and parallelization, allowing it to train models faster and with fewer computational resources compared to the Random Forest models.

In summary, the XGBoost Hyperparameter Tuned Model emerges as the best-performing model for the hospital cost prediction task, exhibiting consistent performance on both training and testing data without significant overfitting. Its ability to capture complex patterns, handle sparse data, and leverage hyperparameter tuning makes it a robust and reliable choice for this regression problem.

Limitations and Future Scope

Although the machine learning models have shown encouraging outcomes in hospital cost prediction, many issues still need to be resolved. First, the quality and representativeness of the training data greatly impacts how accurate the predictions are. The predictions may not be accurate or may not generalize effectively to new, unforeseen events if the training data contains mistakes, biases, or is not reflective of real-world circumstances. Furthermore, it's possible that pertinent characteristics that affect hospital expenses were missed or omitted from the dataset, which would have limited the models' ability to predict outcomes. Furthermore, deciphering and comprehending the underlying decision-making process might be difficult due to the intricacy of

certain machine learning algorithms, such as ensemble approaches. The inability to comprehend the results might make it more difficult to discover potential biases or inconsistent patterns in the model's behavior or to communicate the predictions to interested parties. Furthermore, a variety of circumstances, including modifications to healthcare regulations, breakthroughs in medicine, and shifts in the economy, can cause hospital expenses and the factors that influence them to fluctuate over time. When new data becomes available, the trained models might need to be updated or retrained frequently to retain their predicted accuracy.

Many options may be investigated to overcome these restrictions and improve hospital cost prediction performance. The models' prediction ability may be enhanced by regularly adding new pertinent elements to the dataset associated with hospital expenses, such as patient demographics, medical procedures, hospital facilities, and geographic regions. To improve interpretability and transparency, investigating methods such as Shapley additive explanations (SHAP) or Local Interpretable Model-agnostic Explanations (LIME) may help comprehend the models' decision-making procedures and pinpointing the most important characteristics for hospital cost projections.

Examining the potential for transfer learning—in which models that have already been trained on comparable datasets or issues are adjusted for the hospital cost prediction task—could speed up the training process and enhance performance, particularly in situations with sparse data. Creating a strong deployment strategy that incorporates automatic retraining, updating methods, and real-time monitoring might guarantee that the models stay current and correct when new data becomes available or when there are changes in the distribution of the data.

Performing an extensive cost-benefit analysis to evaluate the possible cost reductions or better resource allocation that might arise from precise hospital cost forecasts could support the

deployment and advancement of the predictive models in actual healthcare environments. The accuracy and applicability of hospital cost prediction may be further improved by solving these issues and investigating the field's potential future applications. This will eventually improve healthcare resource management and cost optimization.

References

Morid, M. A., Sheng, O. R., Kawamoto, K., Ault, T., Dorius, J., & Abdelrahman, S. (2019). Healthcare cost prediction: Leveraging fine-grain temporal patterns. *Journal of Biomedical Informatics*, *91*, 103113 from https://doi.org/10.1016/j.jbi.2019.103113

Abdelmoula, B., Torjmen, M., & Abdelmoula, N. B. (2021). Machine learning based prediction tool of hospitalization cost. *2021 22nd International Arab Conference on Information Technology (ACIT)*. https://doi.org/10.1109/acit53391.2021.9677110

Muremyi, R., Dominique, H., Ignace, K., & Niragire, F. (2020). Prediction of out-of-pocket health expenditures in Rwanda using Machine Learning Techniques. *Pan African Medical Journal*, *37*. https://doi.org/10.11604/pamj.2020.37.357.27287

Panay, B., Baloian, N., Pino, J., Peñafiel, S., Sanson, H., & Bersano, N. (2019). Predicting health care costs using evidence regression. *13th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2019*. https://doi.org/10.3390/proceedings2019031074

Taloba, A. I., Abd El-Aziz, R. M., Alshanbari, H. M., & El-Bagoury, A.-A. H. (2022).Estimation and prediction of hospitalization and medical care costs using regression in machine learning. *Journal of Healthcare Engineering*, *2022*, 1–10. https://doi.org/10.1155/2022/7969220

Islam, M., Chandra, P., Mishra, B., Firoz, S., Fahim, A., & Hoque, M. (2024). *Healthcare Cost Patterns and Prediction: Investigating Personal Datasets Using Data Analytics*. https://doi.org/10.22541/au.170602866.68724472/v1

Sushmita, S., Newman, S., Marquardt, J., Ram, P., Prasad, V., Cock, M. D., & Teredesai, A. (2015). Population cost prediction on public healthcare datasets. *Proceedings of the 5th International Conference on Digital Health 2015*. https://doi.org/10.1145/2750511.2750521

Patidar, S., Dudi, S., & Rohit. (2023). Estimating medical insurance cost using linear regression with hyperparameterization, decision tree and Random Forest Models. *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. https://doi.org/10.1109/confluence56041.2023.10048836

Kuo, C.-Y., Yu, L.-C., Chen, H.-C., & Chan, C.-L. (2018). Comparison of models for the prediction of medical costs of spinal fusion in Taiwan diagnosis-related groups by machine learning algorithms. *Healthcare Informatics Research*, *24*(1), 29. https://doi.org/10.4258/hir.2018.24.1.29

Peng, L., & Chen, X. (2019). Pseudocode of asymmetric loss linear regression (ALLR). ResearchGate. Retrieved from https://www.researchgate.net/figure/Pseudocode-of-asymmetric-loss-linear-regression-ALLR_fig2_340472863

Kelleher, J. D., Namee, M. B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Anayltics: Algorithms, Worked Examples, and Case Studies* (1st ed.). The MIT Press.

Rodriguez-Galiano, V., Sánchez Castillo, M., Dash, J., Atkinson, P., & Ojeda-Zujar, J. (2016). Modelling interannual variation in the spring and autumn land surface phenology of the European forest. *Biosciences 13* (9), 3305-3317. https://doi.org/10.5194/bg-13-3305-2016

Guo, H., Nguyen, H., Vu, D.-A., & Bui, X.-N. (2019). Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach. *Resources Policy, 74*. https://doi.org/10.1016/j.resourpol.2019.101474

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

Mani, D., Kesavan, R., & Kumar, S. (2020). Using XGBoost for high-performance predictions of inpatient prolonged stay at multiple hospitals. Journal of Healthcare Engineering, 2020, Article ID 6645198.

Sarwar, S., Dent, A., Faust, K., Richer, M., Djuric, U., Van Ommeren, R., & Diamandis, P. (2018). XGBoost for big data healthcare analytics: Case study of Parkinson's disease progression. In IEEE International Conference on Big Data (Big Data) (pp. 2046-2055).

Kulkarni, A., Ambekar, A., & Hudnkar, S. (2020). Predicting inpatient hospital costs using machine learning approaches. Health Economics Review, 10(1), 34.

Smith, B., & Johnson, J. (2019). Application of deep learning for cost prediction in healthcare. Artificial Intelligence in Medicine, 99, 101702.

Williams, A., et al. (2018). The impact of data preprocessing on the prediction of healthcare costs using machine learning algorithms. Journal of Medical Systems, 42(11), 206.

Brown, C., & Davis, T. (2017). Using Support Vector Machines for healthcare cost prediction: An empirical analysis. Decision Support Systems, 104, 114-122.

Brown and Davis (2017). Study on SVM algorithm for healthcare cost prediction. This specific reference is to be matched with an exact bibliographic entry from published works for accurate citation.

Huang, S., et al. (2022). Predicting healthcare costs using linear regression and machine learning techniques. Journal of Biomedical Informatics, 128, 104019.

Appendix

GitHub is used to manage and preserve the source code created for this project's implementation. Jupyter notebook for Python uploaded. GitHub URL- https://github.com/Kavanaanil/H-Predict

The final merged dataset used for this project is placed in the OneDrive Location- GWAR_SubmissionFiles