Chapter 1: Introduction

1.1 Description

In the pursuit of unraveling the complexities inherent in healthcare economics, understanding the dynamics of medical costs and expenses stands pivotal. The Medical Cost Dataset, a meticulous compilation of diverse variables, offers an unprecedented opportunity to delve into the multifaceted realm of healthcare expenses. This dataset serves as a foundational resource for researchers, analysts, and healthcare stakeholders, enabling an in-depth analysis of the factors shaping medical charges.

The Medical Cost Dataset offers a fertile ground for various research avenues such as uncovering billing patterns in the medical domain. Predicting future healthcare costs using data-driven models. Exploring intricate relationships between variables and incurred charges.

The dataset's insights are invaluable across sectors such as researchers can develop predictive models enhancing healthcare resource allocation. The Insurers can refine pricing strategies for improved market efficiency and the policymakers gain informed insights to enhance the healthcare system.

1.2 Problem Statement

Despite advancements in healthcare, the intricacies of medical costs remain an enigmatic challenge, perpetuating inefficiencies and disparities within the healthcare landscape. Understanding the multifaceted factors influencing medical charges is crucial for optimizing resource allocation, refining pricing strategies, and informing policy decisions. However, the lack of comprehensive insights into these intricate dynamics hampers the efficiency and equitable distribution of healthcare services. To address this, we aim to analyze the medical costs analysis of health expenses dataset to answer the following existing gap queries:

1. **Complexity in Cost Determinants**: The multitude of variables influencing medical charges, ranging from demographic factors like age and gender to lifestyle choices such as smoking habits, presents a complex interplay that requires meticulous analysis.
2. **Regional Disparities**: Discrepancies in healthcare expenditure across different geographic regions underscore the need to comprehend regional influences on medical costs.
3. **Limited Predictive Models**: The absence of robust predictive models hinders the accurate estimation and prediction of future healthcare expenses. Regional Disparities: Discrepancies in healthcare expenditure across different geographic regions underscore the need to comprehend regional influences on medical costs.

1.3 Objectives

The scope of this project encompasses several key components:

1. **Identify Cost Determinants:** Investigate the relationships between demographic factors (age, gender), lifestyle choices (smoking habits), and BMI on medical charges to pinpoint key cost determinants.
2. **Predictive Modeling:** Develop robust predictive models leveraging machine learning algorithms to forecast future healthcare costs based on the dataset's variables, enabling proactive resource allocation and financial planning.
3. **Regional Disparities Analysis:** Explore and quantify regional disparities in healthcare expenditure by examining how geographic locations influence medical charges, allowing for targeted interventions and policy recommendations.
4. **Patterns in Medical Billing:** Uncover nuanced patterns in medical billing by analyzing interactions between variables, elucidating billing trends and factors contributing to cost variations.
5. **Impact on Family Healthcare Expenses:** Assess the impact of family size (represented by the 'Children' variable) on medical charges, providing insights into family-related healthcare costs and their implications.
6. **Policy and Industry Recommendations:** Derive actionable insights to inform policy decisions, refine pricing strategies for insurers, and recommend efficient resource allocation practices for healthcare providers based on the dataset's findings.

7. **Longitudinal Analysis:** Conduct longitudinal analysis, if applicable, to understand how medical charges evolve over time concerning demographic changes, lifestyle modifications, or policy interventions.
8. **Validity and Generalizability:** Validate the models and findings against external datasets or similar studies to ensure the validity and generalizability of the research outcomes.
9. **Stakeholder Engagement and Dissemination:** Engage stakeholders—such as healthcare practitioners, policymakers, insurers, and researchers—to disseminate findings, foster collaboration, and translate research insights into actionable strategies.
10. **Contribution to Healthcare Economics:** Contribute to the advancement of healthcare economics and policy by providing a comprehensive understanding of the factors influencing medical costs and suggesting avenues for improving the efficiency and equity of healthcare systems.

1.4 Scope of the Project

The scope of this project encompasses several key components:

1. **Variable Analysis:**
    - **Demographic Correlations:** Investigate correlations between age, gender, and medical charges to understand how these demographic factors impact healthcare expenses.
    - **BMI Impact:** Assess the relationship between BMI and medical charges to discern the influence of weight status on healthcare costs.
    - **Smoking Habits:** Explore the substantial impact of smoking habits on healthcare expenses and potential variations in charges.
    - **Family Dynamics:** Analyze the influence of the number of children or dependents on medical charges to understand family-related healthcare expenses.
2. **Predictive Modeling:**
    - Develop predictive models to forecast future medical charges based on the dataset's variables, enabling proactive financial planning and resource allocation.
3. **Regional Disparities:**
    - Investigate and quantify regional disparities in healthcare expenditure, discerning how geographic regions influence medical charges.
4. **Pattern Recognition:**

- Uncover intricate patterns in medical billing by analyzing interactions between variables to identify trends and factors contributing to cost variations.

5. **Policy and Industry Applications:**
    - Derive actionable insights to inform policy decisions, refine pricing strategies for insurers, and recommend efficient resource allocation practices for healthcare providers based on the dataset's findings.
6. **Longitudinal Analysis (if applicable):**
    - Conduct longitudinal analysis to understand the evolution of medical charges over time concerning demographic changes, lifestyle modifications, or policy interventions.
7. **Validation and Generalization:**
    - Validate the models and findings against external datasets or similar studies to ensure their validity and generalizability.
8. **Stakeholder Engagement:**
    - Engage stakeholders—such as healthcare practitioners, policymakers, insurers, and researchers—to disseminate findings, foster collaboration, and translate research insights into actionable strategies.
9. **Contribution to Healthcare Economics:**
    - Contribute to advancing healthcare economics and policy by providing a comprehensive understanding of the factors influencing medical costs and suggesting avenues for improving the efficiency and equity of healthcare systems.

1.5 Project Planning Activities

To ensure efficient project execution, several planning activities will be undertaken:

1.5.1 Team-Member Wise Work Distribution Table

Team member name	Work
Bhavika Sharma	• Data pre processing • Model Development (train and test) • Fine tuning • Comparative Analysis

	• Reporting
Khyati Raghav	• Data collection • Feature selection • Data splitting • Model Development (train and test) • Model evaluation

Table 1.1 Work Distribution Table

1.5.2 PERT Chart

A PERT (Program Evaluation and Review Technique) chart will be created to outline project milestones, dependencies, and timelines. This chart will provide a visual representation of the project's schedule, ensuring that tasks are completed in a logical sequence.

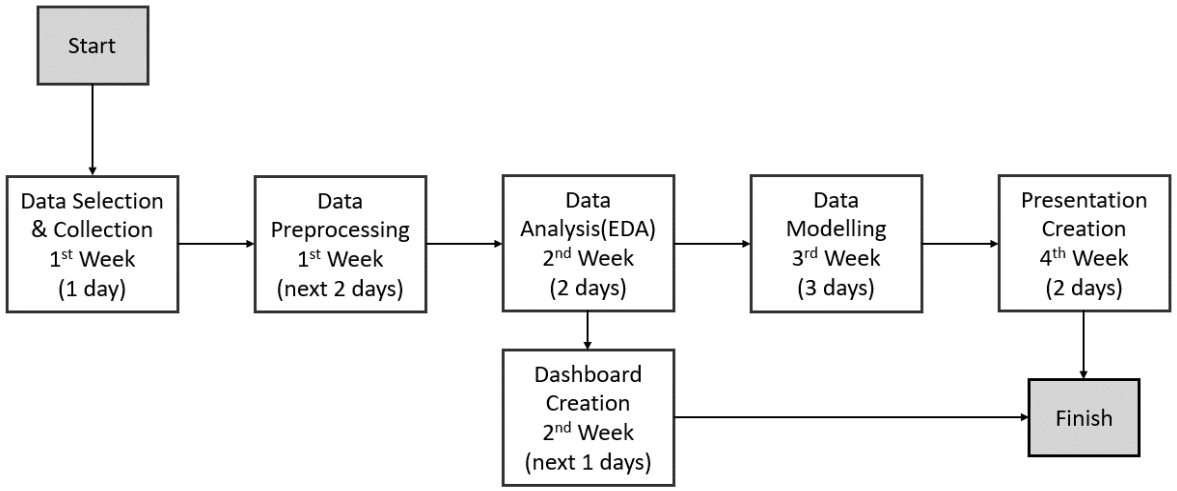

Figure 1.1 PERT Chart

A **PERT (Program Evaluation Review Technique)** chart is primarily a project management tool used to plan and schedule tasks involved in a project. It breaks down tasks into a visual representation, showing dependencies and timelines. However, its direct application to medical cost analysis using linear regression might be indirect. Linear regression is a statistical method

used to model the relationship between variables, often used in predicting costs based on various factors in healthcare.

To integrate a PERT chart with linear regression for medical cost analysis:

Identify Variables: Determine the factors influencing medical costs (e.g., age, type of treatment, comorbidities, geographic location, etc.). Each of these becomes a variable in your linear regression model.

Data Collection: Gather data on these variables and their corresponding medical costs. Ensure the data is comprehensive and representative.

Data Preprocessing: Clean and preprocess the data. This involves handling missing values, normalizing data, and encoding categorical variables if any.

Model Building: Use linear regression to create a model that predicts medical costs based on the identified variables. The model equation would resemble something like:
Cost = b0 + b1 * Variable1 + b2 * Variable2 + ... + bn * VariableN
Here, b0, b1, b2... bn are the coefficients that the regression algorithm calculates based on the provided data.

Evaluate Model Performance: Use techniques like R-squared, Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to assess how well your model fits the data.

PERT Chart Integration: Once the model is built, a PERT chart can be used to outline the steps or tasks involved in the process:

Task 1: Data Collection
Task 2: Data Preprocessing
Task 3: Data Analysis
Task 4: Model Building
Task 4: Presentation Creation
Task 5: Dashboard Creation

Schedule and Dependencies: Determine the dependencies between these tasks and estimate the time required for each. For instance, Model Building can only begin after Data Collection and Preprocessing are complete. This creates a sequence of tasks represented in the PERT chart.

The PERT chart provides a visual representation of the sequential steps involved in the analysis. However, it's essential to note that while linear regression helps predict medical costs, the PERT chart mainly aids in managing the workflow and dependencies within the project of medical cost

Creating a PERT (Program Evaluation Review Technique) chart for a medical cost analysis project involving linear regression would involve breaking down the entire process into specific tasks and depicting their dependencies. Here's a detailed breakdown of the tasks involved:

Task 1: Project Initiation

Description: Define the scope and objectives of the medical cost analysis project.
Dependencies: None
Duration: 1 week
Deliverables: Project scope document, objectives defined.

Task 2: Data Collection

Description: Gather comprehensive data on medical expenses and associated factors (age, treatment type, location, etc.).
Dependencies: None (can start concurrently with Task 1)
Duration: 2-4 weeks
Deliverables: Raw dataset

Task 3: Data Preprocessing

Description: Cleanse data, handle missing values, normalize variables, encode categorical data, and split into training and testing sets.
Dependencies: Data Collection
Duration: 1-2 weeks
Deliverables: Pre-processed dataset

Task 4: Variable Identification

Description: Identify key variables affecting medical costs (age, treatment type, comorbidities, etc.).

Dependencies: Data Collection

Duration: 1 week

Deliverables: List of key variables

Task 5: Model Building (Linear Regression)

Description: Build a linear regression model using the preprocessed data to predict medical costs based on identified variables.

Dependencies: Data Preprocessing, Variable Identification

Duration: 3-6 weeks

Deliverables: Trained regression model

Task 6: Model Evaluation

Description: Assess the performance of the regression model using metrics like R-squared, MSE, RMSE, etc.

Dependencies: Model Building

Duration: 1-2 weeks

Deliverables: Evaluation report

Task 7: Interpretation of Results

Description: Analyze the results, understand the impact of variables on medical costs, draw conclusions.

Dependencies: Model Evaluation

Duration: 1 week

Deliverables: Report on insights and conclusions

Task 8: Reporting and Documentation

Description: Compile all findings, results, and insights into a comprehensive report.

Dependencies: Interpretation of Results

Duration: 1-2 weeks

Deliverables: Final report and documentation

Task 9: Presentation Preparation

Description: Prepare a presentation summarizing the project, methodology, results, and conclusions.

Dependencies: Reporting and Documentation

Duration: 1 week

Deliverables: Presentation material

1.6 Organization of the Report

Chapter 1, In this healthcare economics research project, the meticulously curated Medical Cost Dataset is explored to unravel the complexities of medical charges. Addressing gaps in understanding cost determinants and regional disparities, the project aims to develop predictive models, analyze billing patterns, and assess family dynamics' impact on healthcare expenses. The comprehensive objectives encompass variable analysis, predictive modeling, regional disparities investigation, pattern recognition, policy applications, longitudinal analysis, and stakeholder engagement. Through careful planning activities like team-wise work distribution and a PERT chart, the project aims to contribute valuable insights to healthcare economics, informing policy decisions and enhancing the efficiency of healthcare systems.

Chapter 2, In chapter 2 the literature review encompasses diverse studies addressing healthcare economics challenges. "Best Care at Lower Cost" advocates a learning healthcare system for quality improvement. "Big Data in Health Care" explores big data's potential to reduce costs, emphasizing specific data needs and policy implications. "Estimation and Prediction of Hospitalization" emphasizes BMI's impact, highlighting the efficacy of linear regression in forecasting healthcare costs. "Comparison of Statistical and Machine Learning Models" focuses on cost prediction in the Oncology Care Model, emphasizing the importance of flexible modeling. "Machine Learning Approaches for Predicting High-Cost" explores global healthcare expenditures, predicting patient-level expenses for efficient healthcare management. "Medical Cost Prediction Model" in Japan adopts machine learning for accurate cost forecasts. "Machine Learning versus Regression Modelling" favors Random Forest for precise healthcare cost prediction, especially for high-cost subjects.

Chapter 3, In chapter 3 The methodology for analyzing the Medical Cost Dataset involves systematic data understanding, cleaning, and exploration. Descriptive statistics, data cleaning, and visualization techniques are employed, followed by feature engineering and selection of regression models such as Linear Regression and Random Forest. The dataset is split for training and validation, and model performance is evaluated using metrics like MSE and R-squared. Ethical considerations and documentation are prioritized. The approach aims to derive meaningful insights into factors influencing medical charges while ensuring transparency and reproducibility in the analysis.

Chapter 4, In chapter 4 the implementation details showcase a thorough process from data collection to model development and evaluation. The dataset, extracted from Kaggle, focuses on medical cost analysis for healthcare expenses. Data preparation involves feature selection, handling missing values, and encoding categorical features. Linear Regression and Random Forest Regression models are developed using Python, and their performance is assessed on metrics like RMSE, MAE, and R2 Score. The Random Forest Regressor and AdaBoost Regressor demonstrate superior accuracy, outperforming other models. Decision Tree and XGBRegressor show lower errors, highlighting better prediction accuracy. The comparative analysis emphasizes the trade-offs between model complexity, accuracy, and potential overfitting, guiding the choice of suitable models for medical cost analysis.

Chapter 5, In chapter 5 the conclusion highlights the project's focus on efficient system design for the Medical Cost Dataset, emphasizing regression algorithms like AdaBoost and Random Forest. Future work suggests real-time integration into healthcare systems, longitudinal analysis, global health cost examination, and continuous model improvement. Collaboration, interpretability, and adherence to ethical standards remain paramount for positive healthcare impacts.

Chapter 2: Literature Review

2.1 **Summary of the research papers studied**

A. *Elvan DUMAN (2022)* ***" Implementation of XGBoost method for healthcare fraud detection"***

The paper titled "Implementation of XGBoost Method for Healthcare Fraud Detection" explores the application of machine learning algorithms, specifically XGBoost, for detecting fraud in healthcare systems. The healthcare sector is facing challenges related to rising costs and an exponential increase in medical data. The paper focuses on addressing these issues by employing XGBoost, a gradient-boosted decision tree algorithm, and comparing its performance with other supervised algorithms like Random Forest, Logistic Regression, and decision trees.

The study uses the List of Excluded Individuals/Entities (LEIE) database, containing information about excluded providers, to label fraud in the Medicare Part B dataset. The results indicate that XGBoost outperforms traditional machine learning algorithms in terms of performance metrics such as Area Under the Curve (AUC), precision, recall, and F1 score. The paper discusses the challenges of imbalanced data in healthcare fraud detection and proposes a method based on XGBoost to address this issue.

Overall, the research contributes to the ongoing efforts in leveraging machine learning for healthcare fraud detection and emphasizes the importance of efficient methods to handle the increasing volume of electronic medical data. The findings suggest that XGBoost is a promising algorithm for enhancing the accuracy of fraud detection in healthcare systems.

B. *David W. Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar (2014)* ***"Big Data in Health Care: Using Analytics to Identify and Manage High-Risk and High-Cost Patients".***

The rapid integration of electronic health records in the US health care system augments the availability of clinical data, coinciding with significant advancements in clinical analytics, forming what's termed as big data. This convergence presents unprecedented prospects to curtail health care expenses. The research identifies six pivotal areas where big data utilization can notably diminish costs: high-cost patients, readmissions, triage, decompensation, adverse events, and optimizing treatments for multi-organ diseases. It delves into the potential insights from clinical analytics, requisite data types, and essential infrastructure (analytics, algorithms, registries, etc.) for effective analysis and implementation. The findings underscore policy implications for regulatory oversight, privacy concerns, and the necessity of supporting analytics-focused research, providing a roadmap for leveraging big data to enhance care quality while containing expenses in the US health care system.

C. *Ahmed I. Taloba, Rasha M. Abd El-Aziz, Huda M. Alshanbari, and Abdal-Aziz H. El-Bagoury (2022)* ***"Estimation and Prediction of Hospitalization and Medical Care Costs Using Regression in Machine Learning"***

This research underscores the pervasive impact of medical costs in people's lives, linking factors like BMI, aging, and smoking to heightened healthcare expenses. Highlighting the urgency of cost-effective obesity prevention strategies, it emphasizes the significance of addressing obesity early on in global health agendas. Employing genetic variants as instrumental variables, the study uses extensive public datasets to predict the influence of BMI on healthcare expenses. A Multiview learning approach integrates BMI information from various records, employing a hierarchical perception structure to discern crucial elements for informative data representations. Comparing statistical and machine-learning methods like linear regression, naive Bayes classifier, and random forest algorithms, the study finds that linear regression demonstrates the highest accuracy (97.89%) in forecasting overall healthcare costs. This approach offers a predictive method with significant financial implications, offering insights into managing and projecting healthcare expenses.

D. *Madhu Mazumdar, Jung-Yi Joyce Lin, Wei Zhang, Lihua Li, Mark Liu, Kavita Dharmarajan, Mark Sanderson, Luis Isola & Liangyuan Hu Liangyuan Hu (2020)* ***"Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by Oncology Care Model (OCM) data"***

The Oncology Care Model (OCM) aims to enhance cancer care quality while reducing costs. It employs a Gamma generalized linear model (Gamma GLM) with a log-link for risk adjustment. However, when applying the national model to an academic medical center (AMC), the predicted expense for patient episodes did not align with observed expenses. To address this discrepancy, the study fitted the Gamma GLM to the AMC data and compared it with Random Forest (RF) and Partially Linear Additive Quantile Regression (PLAQR) models. Additionally, a simulation study explored these methods' comparative performance, specifically assessing non-linearity and interaction effects, often overlooked in cost prediction studies. This investigation sheds light on enhancing cost prediction accuracy in the OCM framework, offering insights into improving cancer care cost estimation through flexible modeling approaches.

E. *Chengliang Yang, Chris Delcher, Elizabeth Shenkman & Sanjay Ranka (2018)* ***"Machine learning approaches for predicting high-cost high need patient expenditures in health care"***

This research delves into the expansive realm of global healthcare economics, spotlighting the significant rise in healthcare expenditures worldwide. Specifically, it focuses on the disproportionate allocation of resources to a minority of high-cost, high-need (HCHN) patients, prompting the need for more efficient healthcare delivery. The study harnesses the potential of information technology and big data in healthcare, aiming to predict patient-level expenditures, particularly for HCHN individuals. By employing machine learning models on extensive longitudinal administrative claims data, the research unveils promising insights. It identifies temporal patterns in healthcare spending, enhances predictive accuracy for HCHN patients, and quantifies the impact of various factors on expenditure prediction. The findings showcase the feasibility of forecasting healthcare costs and suggest avenues for targeted interventions. The paper outlines methodologies,

results demonstrating predictive capabilities, and discusses strategies to further enhance expenditure prediction for improved healthcare management.

F. F. *Takeshima T, Keino S, Aoki R, Matsui T, Iwasaki K (2018)* ***"Development of Medical Cost Prediction Model Based on Statistical Machine Learning Using Health Insurance Claims Data"***

The Japanese government's mandate for insurers to develop population health management strategies necessitates robust cost prediction models for evaluation. However, the challenge arises from individuals being covered for multiple ailments, rendering traditional linear models inadequate. To address this, a novel approach leveraging statistical machine learning methods was developed for medical cost prediction. This innovative model aims to accommodate the complexities of multiple diseases within one insured individual. By harnessing the power of machine learning, this research presents a refined and adaptable framework capable of more accurately forecasting medical costs, aligning with the government's initiative for efficient population health management in Japan.

G. *Alexandre Vimont, Henri Leleu & Isabelle Durand-Zaleski (2022)* ***"Machine learning versus regression modelling in predicting individual healthcare costs from a representative sample of the nationwide claims database in France"***

This research assesses predictive models for individual healthcare cost estimation, crucial for innovative provider payment methods. Comparing a neural network (NN), random forest (RF), and a generalized linear model (GLM), the study uses a French National Health Data Information System subset. Factors include demographics, pre-existing conditions, and healthcare service usage. Results show similar performance among models after demographic adjustment, but RF excels with additional variables like pre-existing conditions and prior costs. In the full model, RF outperforms, achieving 47.5% adjusted R-squared, 1338€ mean absolute error, and 67% hit ratio, surpassing GLM (34.7% R-squared, 1635€ error, 58% hit ratio) and NN (31.6% R-squared, 1660€ error, 55% hit ratio). RF proves superior, especially for high-cost subjects and diverse conditions, while

GLM is suitable for understanding predictor contributions. This study advocates for RF in precise medical cost prediction scenarios.

H. *San Wang, Jieun Han, SeYoung Jung, Tae Jung Oh, SenYa , Sanghee Lim , Hee Hwang , Ho-Young Lee & Haeun Lee (2022)* ***"Development and implementation of patient-level prediction models of end-stage renal disease for type 2 diabetes patients using fast healthcare interoperability resources"***

This study aimed to predict the 5-year risk of end-stage renal disease (ESRD) in type 2 diabetes mellitus (T2DM) patients using machine learning (ML). They developed an algorithm using various ML classifiers and implemented it into electronic medical records using Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR). The model achieved an AUROC of 0.95 and AUPRC of 0.79 with XGBoost, outperforming other models. Features like serum creatinine, serum albumin, and medication days of insulin were vital for prediction. The model, integrated into EMR, provides a risk score within 5 years for ESRD in T2DM patients. The SHAP analysis helped interpret the model's features. A dashboard was designed for clinical use, aiding in identifying modifiable risk factors. The model's precision and recall rates showed clinical relevance, presenting a significant advancement in ESRD risk prediction for T2DM patients using ML.

2.2 Comprehensive Overview of the studies

Title of Research Paper	Author(s) detail with year	Findings
Implementation of XGBoost method for healthcare fraud detection	Elvan DUMAN (2022)	The paper explores XGBoost for healthcare fraud detection, outperforming traditional methods significantly.

Big Data in Health Care: Using Analytics to Identify and Manage High-Risk and High-Cost Patients	A. David W. Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar (2014)	Electronic health records' integration in the US health system amplifies clinical data availability, aligning with advanced clinical analytics, forming big data. Six key areas for cost reduction emerge, leveraging analytics insights, necessitating specific data and infrastructure, with policy implications for efficient care delivery.
Estimation and Prediction of Hospitalization and Medical Care Costs Using Regression in Machine Learning	Ahmed I. Taloba, Rasha M. Abd El-Aziz, Huda M. Alshanbari, and Abdal-Aziz H. El-Bagoury (2022)	Research highlights medical cost impacts, BMI's influence, and urgency for obesity prevention. Using diverse datasets and models, linear regression excels, forecasting healthcare costs with 97.89% accuracy, aiding expense projection.
Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by Oncology Care Model (OCM) data	Madhu Mazumdar, Jung-Yi Joyce Lin, Wei Zhang, Lihua Li, Mark Liu, Kavita Dharmarajan, Mark Sanderson, Luis Isola & Liangyuan Hu Liangyuan Hu (2020)	. The Oncology Care Model (OCM) seeks better cancer care at lower costs using a Gamma GLM. Adapting it for an academic medical center, the study compared Gamma GLM with RF and PLAQR models. A simulation study emphasized enhancing cost predictions, highlighting the

		importance of flexible modeling in OCM.
Machine learning approaches for predicting high-cost high need patient expenditures in health care	Chengliang Yang, Chris Delcher, Elizabeth Shenkman & Sanjay Ranka (2018)	This study explores global healthcare economics, emphasizing rising expenditures and the skewed resource allocation to high-cost patients. Using machine learning on extensive data, it predicts patient-level expenses, focusing on high-need individuals, revealing patterns and enhancing predictions for efficient healthcare management.
Development of Medical Cost Prediction Model Based on Statistical Machine Learning Using Health Insurance Claims Data	Takeshima T, Keino S, Aoki R, Matsui T, Iwasaki K (2018)	. Japan's insurer mandate for population health management demands precise cost prediction models. Overcoming challenges of multiple ailments, a novel machine learning approach was crafted, enhancing accurate medical cost

		forecasts for effective healthcare management.
Machine learning versus regression modelling in predicting individual healthcare costs from a representative sample of the nationwide claims database in France	Alexandre Vimont, Henri Leleu & Isabelle Durand-Zaleski (2022)	The study compares predictive models for healthcare cost estimation. Random Forest (RF) outperforms Neural Network (NN) and Generalized Linear Model (GLM), especially for high-cost subjects and varied conditions. RF excels in precise cost prediction.
"Development and implementation of patient level prediction models of end stage renal disease for type 2 diabetes patients using fast healthcare interoperability resources"	San Wang, Jieun Han, SeYoung Jung, Tae Jung Oh, SenYa , Sanghee Lim , Hee Hwang , Ho Young Lee & Haeun Lee (2022)	ML predicts 5-year ESRD risk in T2DM with XGBoost (AUROC 0.95, AUPRC 0.79), aiding clinical intervention via EMR.

Table 2.1 Literature Review Comparison Table

Chapter 3: System Design and Methodology

3.1 System Design

3.1.1 Introduction to System Design

Designing a system around the Medical Cost Dataset involves structuring it for efficient storage, analysis, and accessibility. Here's a high-level overview of how you might design a system to work with this dataset:

Data Storage: Use a relational database management system (RDBMS) like MySQL, PostgreSQL, or SQLite to store structured data efficiently. Create a table with columns corresponding to the dataset attributes (ID, Age, Sex, BMI, Children, Smoker, Region, Charges). Define appropriate data types for each column (integers, floats, strings, etc.).

Data Processing: Implement routines to clean and preprocess incoming data, handling missing values, inconsistencies, and formatting issues. Derive additional features , such as categorizing BMI into weight categories or encoding categorical variables like 'Region' and 'Sex' for analysis.

Analysis and Modeling: Develop tools or scripts to perform EDA, generating visualizations and summary statistics to understand data distributions, correlations, and outliers. Create predictive models using machine learning or statistical approaches. For instance, regression models to predict 'Charges' based on other variables or clustering techniques to identify patterns in medical billing.

System Components: Develop an interface (web-based or otherwise) for users to access and query the dataset. This could involve creating an API that allows querying specific data or retrieving statistical summaries. Implement security protocols to protect sensitive patient information, ensuring compliance with healthcare data regulations (such as HIPAA in the US). Ensure the system can handle large volumes of data and perform computations efficiently, especially if dealing with extensive analyses or predictive modeling.

Accessibility and Collaboration: Create comprehensive documentation detailing the dataset's schema, meaning of each attribute, and guidelines for its usage. Implement version control systems (e.g., Git) and collaboration platforms to enable multiple researchers or teams to work simultaneously on analyses and models.

Ethical Considerations: Ensure compliance with privacy laws and regulations regarding patient data. Anonymize or de-identify data when necessary to protect patient confidentiality. If conducting research involving human subjects, ensure proper ethical review and consent procedures are followed.

Designing a system around this dataset involves considerations for data integrity, security, performance, and ethical handling of sensitive medical information to enable researchers, analysts, and policymakers to derive meaningful insights while respecting patient privacy and confidentiality.

3.1.1 System Architecture

Designing a system architecture for handling the Medical Cost Dataset involves considering data storage, processing, analysis, and accessibility. Here's a high-level system architecture:
Components:

1. **Database:**
    - **Type:** Use a relational database (e.g., MySQL, PostgreSQL) for structured data.
    - **Tables:** Create a table to store the dataset with columns for ID, Age, Sex, BMI, Children, Smoker, Region, and Charges.
    - **Indexes:** Implement indexes on columns used frequently in queries for faster retrieval.

2. **Data Processing:**
    - **ETL (Extract, Transform, Load):** Develop ETL processes to clean and preprocess incoming data, handling missing values, outliers, and transforming data types.
    - **Feature Engineering:** Implement scripts to derive additional features if necessary.

3. **API/Backend:**
    - **RESTful API:** Develop an API for interacting with the dataset. This API should allow querying, updating, and inserting data.
    - **Backend Server:** Implement a backend server to handle requests, perform data retrieval, and execute computations.

4. **Security:**
    - **Data Encryption:** Implement encryption for sensitive data, especially if the system is accessed over the internet.
    - **Authentication and Authorization:** Implement user authentication and authorization mechanisms to control access to the dataset.

5. **Analysis and Modeling:**
    - **Machine Learning Models:** Develop models for predictive analysis based on the dataset, integrating them into the system.
    - **Analytics Engine:** Implement an analytics engine to perform complex analyses on the dataset, generating insights.

6. **Frontend/Application:**
    - **User Interface:** Develop a user-friendly interface for users to interact with the dataset, query data, and visualize insights.
    - **Dashboard:** Create dashboards for easy visualization of key metrics and trends.

7. **Collaboration and Version Control:**
    - **Version Control System:** Use a version control system (e.g., Git) for managing changes to the system and dataset.
    - **Collaboration Tools:** Implement tools for multiple users or teams to collaborate on analyses and share findings.

8. **Scalability and Performance:**
    - **Load Balancing:** If the system experiences high traffic, implement load balancing to distribute requests efficiently.
    - **Caching:** Use caching mechanisms to store frequently accessed data for faster retrieval.

9. **Monitoring and Logging:**
    - **Logging:** Implement logging mechanisms to record system activities and errors for troubleshooting.
    - **Monitoring Tools:** Use monitoring tools to track system performance, resource usage, and user activities.

10. **Ethical Considerations:**
    - **Privacy Measures:** Implement strict privacy measures to ensure compliance with healthcare data regulations.
    - **Audit Trails:** Maintain audit trails to trace data access and modifications for accountability.

11. **Backup and Recovery:**
    - **Regular Backups:** Implement regular backups of the dataset to prevent data loss.
    - **Recovery Procedures:** Develop procedures for recovering data in case of system failures.

This architecture is designed to ensure the efficient storage, processing, and analysis of the Medical Cost Dataset while considering security, scalability, and ethical considerations in handling healthcare data.

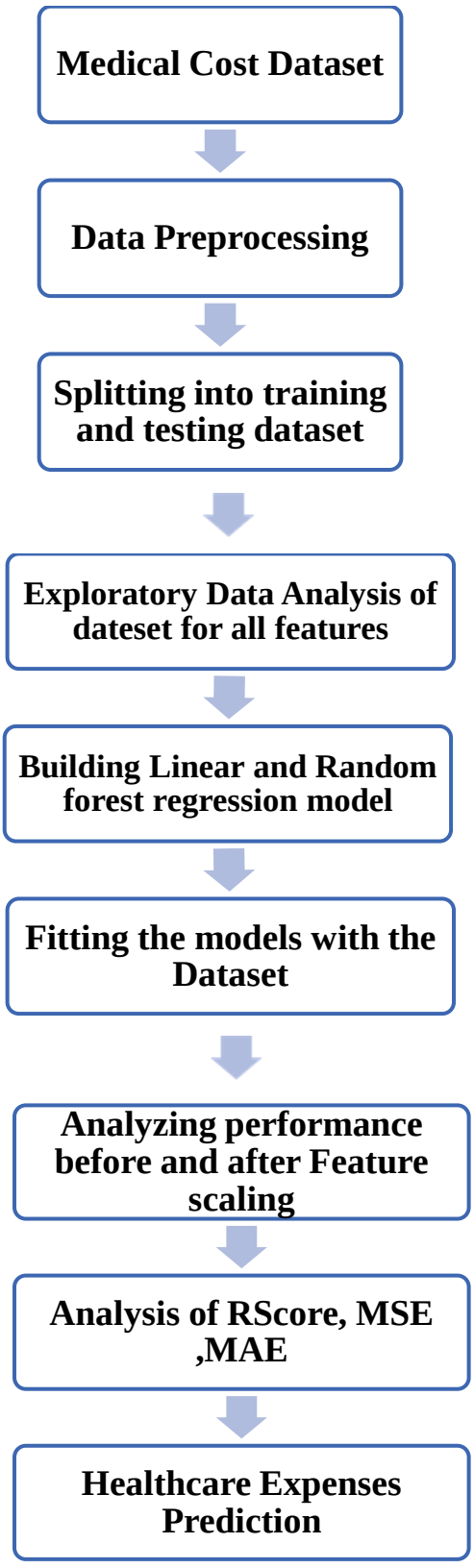

Fig 3.0 – Block Diagram

Medical Cost Dataset :

The database used here is taken from Kaggle and it is collection of medical expense personal data, which contain anonymous information about people. these data will act as a method learning

object to generate functional information. In Table 1, the attributes such as BMI and age are continuous variables, and the attributes such as smoker and sex are categorical variables

Data Preprocessing:

After importing the data we import the necessary libraries for data preprocessing like numpy , pandas, and seaborn. Then data cleaning , removal of null values if any and duplicate rows are deleted . A target value is then extracted i.e. y including charges and the independent set x including age , BMI , children , smoker and encoding of the categorical values such as smoker is done in this phase.

Splitting into training and test dataset :

After data preprocessing using the train_test_split function of the sklearn model_selection library we split the dataset into four different parts namely X_train , X_test , y_ train , y_test two of theses are the dependent datasets and other two are the independent datatsets.

Exploratory Data analysis of dataset for all features :

Descriptive statistics, data visualisations, data cleaning is done to scrutinize each feature, including the unique ID, patient age, gender, BMI, children count, smoking status, region, and medical charges. Identify correlations between age and expenses, assess potential gender-based cost variations, and examine how BMI and smoking habits impact healthcare costs. Explore regional disparities in healthcare expenditure and analyze the influence of family size on medical charges .

Building Linear Regression and Random Forest regressor models:

We decided to choose these models as the target value of our data was the numerical values .To build the linear regression model we used the LinearRegression() function after importing it from the sklearn library under linear_model .Linear regression is apt for the Medical Cost Dataset as it efficiently models the linear relationships between variables (age, BMI, etc.) and medical charges. Random Forest is employed in the Medical Cost Dataset due to its ability to handle complex, nonlinear relationships among variables like age, BMI, and smoking status. By aggregating predictions from multiple decision trees, it enhances predictive accuracy . For this the function RandomForestRegressor() is imported from the sklearn library under the ensemble category .

Fitting the models with dataset :

For fitting the models we use the fit function and fit the x_train and the y_train datasets i.e. training datasets .

Analysis of RScore, MSE ,r2score,MAE:

Mean squared error is calculated using cross_val_score and then the mean of the MSE is calculated and then the r2 score higher the cv lower is the MSE and r2score obtained.

In case of random forest we calculate the rf_score using the r2_score obtained earlier and the varying cross val score affects the rf_score .

Healthcare Expenses Prediction:

Linear regression algorithm and random forest are used to estimate the healthcare costs of the patients such as obesity (BMI) using certain devices such as smartphones and smart devices.

The model performance evaluation reveals distinct characteristics among various machine learning algorithms applied to the Medical Cost Analysis for Healthcare Expenses dataset. Among the models assessed, the Random Forest Regressor and AdaBoost Regressor stand out with superior accuracy metrics.

The Random Forest Regressor achieves an accuracy of 83.85%, indicating its efficacy in predicting medical expenses. AdaBoost Regressor performs even better, boasting an accuracy of 84.1%.

In comparison, the Linear Regression model, while demonstrating decent performance with a 77% accuracy, is surpassed by ensemble methods.

A comprehensive comparative analysis indicates that Decision Tree and XGBRegressor exhibit the lowest Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), reflecting superior prediction accuracy. These models also excel in explaining variance, as evidenced by their higher R2 scores.

However, they come with higher model complexity, while Linear Regression and AdaBoost Regressor offer simpler alternatives.

Considering interpretability and accuracy, Decision Tree or XGBRegressor might be preferable, whereas Random Forest strikes a balance between accuracy and model complexity.

The results underscore the importance of fine-tuning hyperparameters, validating models on separate test sets, and addressing potential overfitting concerns to ensure robust generalization in the domain of medical cost analysis for healthcare expenses.

3.1.3 Data Sources

The data source consists of a downloaded CSV file that is available on Kaggle website. The reference is given as below.

Link: https://www.kaggle.com/datasets/nanditapore/medical-cost-dataset

3.1.4 Data Preprocessing:

Data preprocessing steps that were followed to prepare the data for modeling are as follows.

- Data cleaning
- Feature Engineering
- Feature Transformation
- Handling Missing, Null, Duplicate values.

3.1.5 Modelling:

The machine learning models we employed for estimating delivery time was Random Forest Regressor. This algo was used because of numerical target value as well as low effect of outliers on its result, though the data source we used contained almost not outliers considering larger no number of data we had.

Since we were using python for machine learning algorithm we used sci-kit learn library for model selection.

3.1.6 Training and Validation:

The training dataset taken was 85% of the data and 15% was used for testing. We also used cross validation technique and performance metrics used was R2_score and for error detection we used MSE, RMSE, MAE.

R2_Score is the measure that provides information about the goodness of fit of a model it is given by:

$$R^2 = 1 - \frac{\Sigma_i^N (y_i - \hat{y}_i)^2}{\Sigma_i^N (y_i - \bar{y})^2}$$

Fig 3.1 Formula for R2 coefficient (R2

MSE is Mean Squared Error it measures the amount of error in the statistical model it is given by:

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

Fig 3.2 Formula for Mean Squared Error

RMSE is Root Mean Squared Error is the average difference between values predicted by a model and the actual values it is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Fig 3.3 Formula for Root Mean Squared Error (RMSE)

MAE is Mean Absolute Error it is the magnitude of difference between the prediction of an observation and the true value of that observation it is given by:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - y_i^p|}{n}$$

Fig 3.4 Formula for Root Mean Squared Error (RMAE)

3.1.7 Challenges and Trade-offs:

The only challenge we encountered was while removing the outliers as the data was already clean enough and thus removing the outliers leaning the model toward biasness thus we had to make a trade off with the outliers and didn't remove them.

3.2 Algorithm Used

There are a lot of algorithms available in machine learning models but choosing one of them to address a particular problem is totally dependent on the type of problem or the target values of your data. Choosing an accurate model is crucial part of a data science project. So, we did in our project.

After carefully analyzing the all the aspects of our dataset we chose Regression algorithm to be used in our project.

First, we used Linear Regression model and the results we get were considerably good. With different combinations of Test size and Random state we could only achieve a highest score of 66.12 %.

Although the result was not that satisfying thus, we used another Regression model Random Forest Regressor that gave us an R2_score of 82.45% with different combination of Test size and Random states.

3.2.1 Linear Regression Algorithm

Linear regression is a foundational machine learning algorithm used for predictive analysis and understanding relationships between variables. It works by fitting a linear equation to observed data to predict outcomes based on input features.

The algorithm seeks to find the best-fitting line (in the case of simple linear regression) or hyperplane (in multiple linear regression) that minimizes the difference between predicted and actual values. This line represents the relationship between the independent variable(s) and the dependent variable, which is continuous.

The process begins by defining the hypothesis function, typically represented as:

The goal is to learn the coefficients that best fit the data. This is done by using an optimization technique like Ordinary Least Squares (OLS) or gradient descent to minimize the difference between predicted and actual values, often quantified by a cost function like Mean Squared Error (MSE).

During training, the algorithm adjusts the coefficients iteratively to minimize the cost function, updating the model's parameters until convergence.

Once trained, the model can predict outcomes for new data by applying the learned coefficients to the input features.

Linear regression's simplicity and interpretability make it a valuable tool for tasks like price prediction, trend analysis, and understanding the relationship between variables. However, its effectiveness relies on the assumption of a linear relationship between the input features and the target variable, which might not hold in all cases. Regularization techniques like Ridge or Lasso

regression can mitigate issues like overfitting and enhance model performance in complex scenarios.

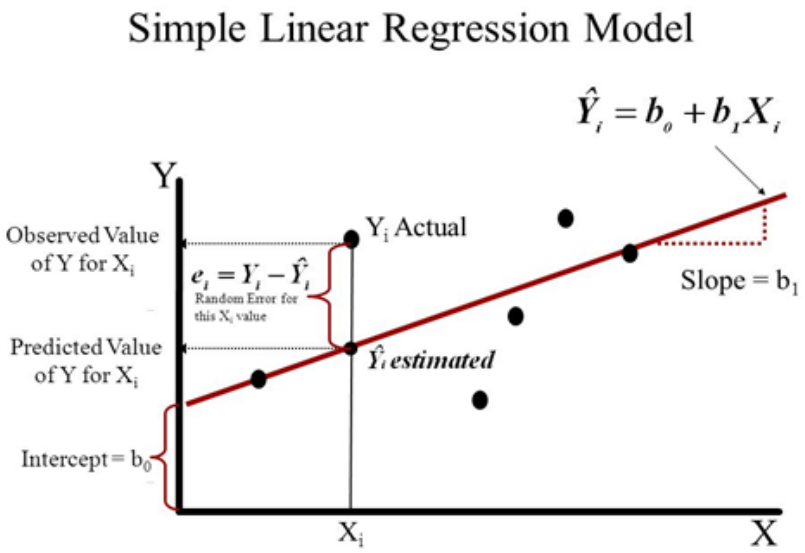

Figure 3.5 Linear Regression

3.2.2 Random Forest Regressor Algorithm

The Random Forest Regressor is a powerful ensemble learning algorithm used in machine learning for regression tasks. It's constructed from multiple decision trees, creating a forest of trees that work together to make predictions.

Training Process:

1. **Ensemble of Trees:** It builds a multitude of decision trees during training, each trained on a random subset of the dataset (bootstrap aggregating or "bagging"). This randomness helps introduce diversity among the trees.
2. **Random Feature Selection:** At each node of every tree, instead of considering all features for splitting, a random subset of features is chosen. This enhances independence among the trees and reduces the risk of overfitting.
3. **Decision Tree Training:** Each tree is trained on a subset of the data, and for each split in the tree, the algorithm selects the best feature among the randomly chosen subset of features.

Prediction Process:

When making predictions:

- Each tree in the forest predicts an outcome.

- For regression tasks, the predicted values from individual trees are averaged (for example, taking the mean) to get the final prediction.

Benefits:

- **Accuracy:** Random Forests tend to deliver high accuracy and robustness due to their ensemble nature, reducing overfitting compared to individual decision trees.
- **Versatility:** Suitable for a wide range of problems, handling both categorical and numerical data.
- **Feature Importance:** It provides insight into feature importance, indicating which features contribute most to predictions.

Random Forest Regressors are effective for various applications like predicting stock prices, real estate valuation, medical diagnosis, and more. They excel in scenarios where high accuracy and robustness are required, and understanding feature importance is valuable. Adjusting parameters like the number of trees, depth, and feature selection criteria can further optimize their performance for specific tasks.

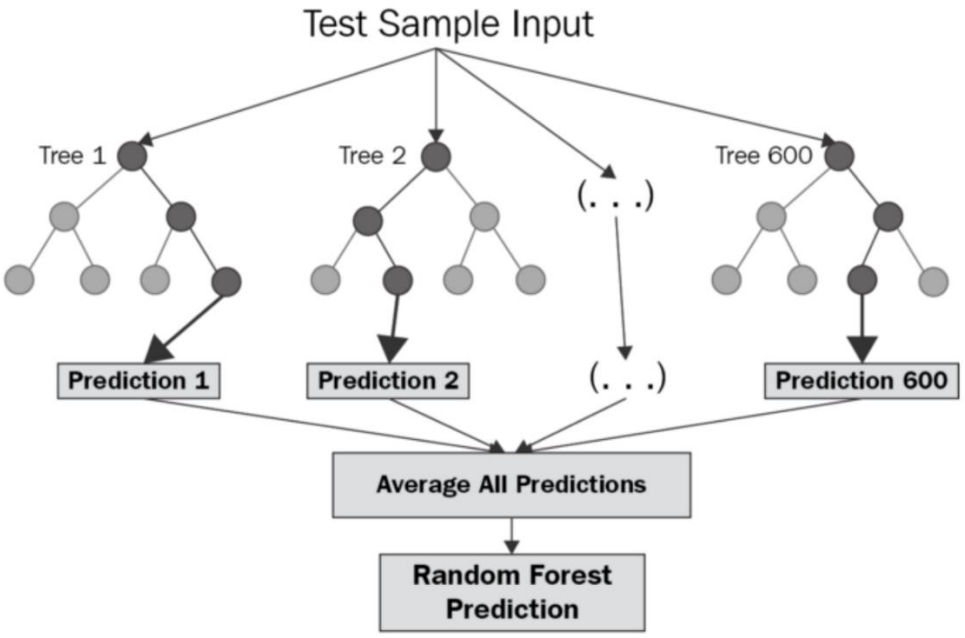

Figure 3.6 Random Forest Regressor

3.2.3 Decision Tree Algorithm

The decision tree algorithm in linear regression is a method that employs a tree-like structure to make predictions in regression tasks. Unlike traditional linear regression, which uses a continuous function to predict output, decision trees recursively split the input space into smaller regions and fit a simple model within each of these regions.

Here's how the algorithm works:
Splitting Nodes: The decision tree starts with the entire dataset and selects the best feature and split point to create two child nodes.

Node Splitting Criteria: For linear regression in decision trees, this usually involves minimizing the variance of the target variable within the regions created by the splits.

Leaf Nodes: Eventually, the splitting stops when a predefined stopping criterion is reached, such as a maximum depth of the tree, a minimum number of samples in a node, or no further gain in variance reduction.

Prediction: In the case of linear regression, the prediction at each leaf node is the mean (or any linear function) of the target variable values within that node.

Model Interpretation: While decision trees themselves might not inherently provide coefficients or coefficients in the way linear regression does, understanding how the tree makes decisions can still offer insights into feature importance and relationships between predictors and the target variable.

In linear regression within decision trees, the idea is to approximate the relationships between features and the target variable within local regions rather than using a global linear function. This approach can capture non-linear relationships effectively and handle interactions between features, making it a powerful tool for regression tasks.

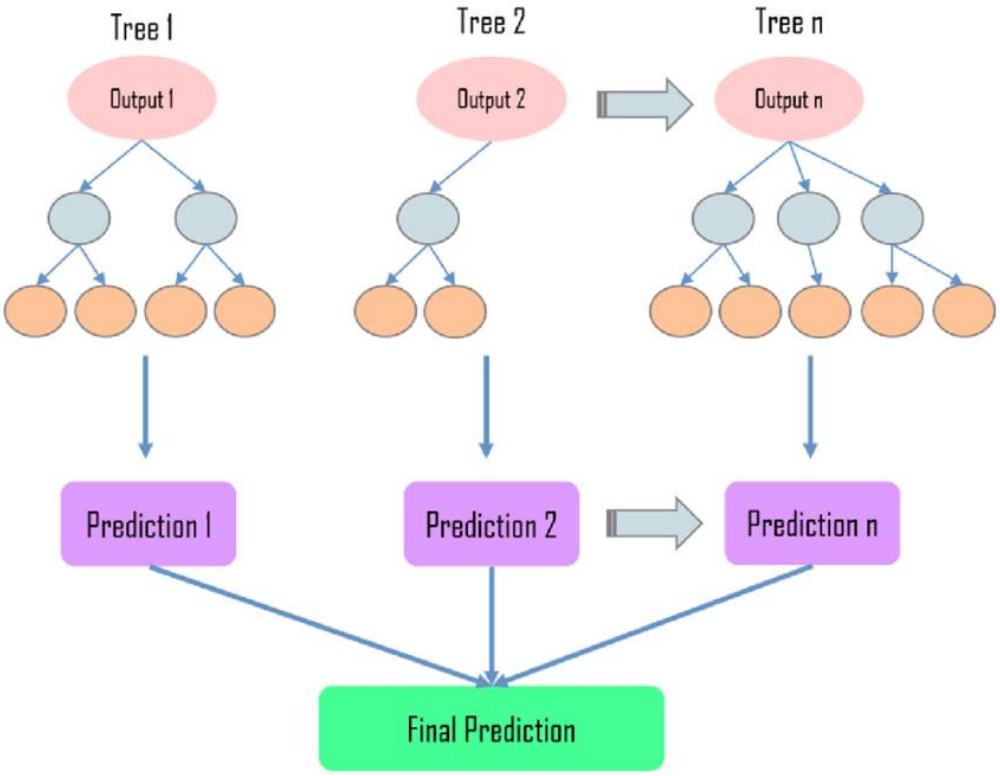

Figure 3.7 Decision Tree Regressor

3.2.4 AdaBoost Regressor

The AdaBoost Regressor algorithm in linear regression is an ensemble learning method that combines multiple weak learners (typically simple linear regression models) to create a strong regression model. It iteratively adjusts the weights of the training instances based on their performance, focusing more on instances that were previously poorly predicted, thus emphasizing the difficult-to-predict cases.

Here's a breakdown of the AdaBoost Regressor algorithm in linear regression:

Base Estimator: Initially, each training instance is given an equal weight, and a weak learner (e.g., a simple linear regression model) is trained on the data. The weak learner aims to predict the target variable.

Instance Weight Adjustment: After the first iteration, the algorithm increases the weights of instances that were inaccurately predicted and decreases the weights of correctly predicted

instances. This adjustment highlights the importance of misclassified or poorly predicted instances in subsequent iterations.

Sequential Model Fitting: The subsequent weak learners are trained by focusing more on the instances with increased weights, aiming to correctly predict the previously misclassified data points.

Model Combination: The final model is a weighted combination of these weak learners. Each learner contributes to the final prediction based on its performance in the iterations. In linear regression, this combination may involve assigning weights to the predictions made by individual linear regression models.

The AdaBoost Regressor in linear regression aims to sequentially improve the overall regression performance by emphasizing the mispredicted instances and learning from them. By combining multiple weak models, it constructs a strong regression model capable of handling complex relationships between predictors and the target variable while mitigating overfitting.

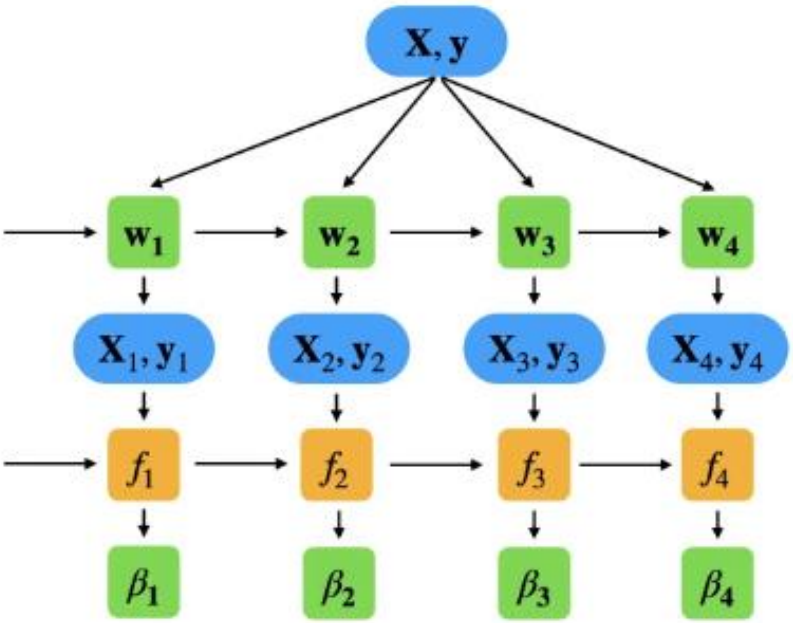

Figure 3.8 AdaBoost Regressor

3.2.5 XGB Regressor

XGBRegressor stands for eXtreme Gradient Boosting Regressor, a powerful machine learning algorithm belonging to the gradient boosting family. It's known for its efficiency, speed, and accuracy in handling regression tasks.

Here's an overview of XGBRegressor:

Gradient Boosting: XGBRegressor works on the principle of gradient boosting, which sequentially combines multiple weak learners (decision trees in this case) to create a robust predictive model.

Tree Ensemble: It constructs an ensemble of decision trees iteratively, with each subsequent tree aiming to correct the errors made by the previous ones. These decision trees are generally shallow (have limited depth) to prevent overfitting.

Objective Function: XGBRegressor uses a specific objective function (often squared loss for regression tasks) and optimizes it using gradient descent. It minimizes the residuals (differences between predicted and actual values) at each iteration to improve the model's accuracy.

Regularization: It implements regularization techniques like shrinkage (learning rate), subsampling, and column sampling to prevent overfitting and enhance the model's generalization capability.

Parallel Computing and Optimization: XGBRegressor is optimized for speed and efficiency. It supports parallel processing and offers options for hardware optimization, making it significantly faster than many other boosting algorithms.

Hyperparameter Tuning: Fine-tuning hyperparameters is crucial for optimal performance. Parameters such as tree depth, learning rate, and regularization settings can be adjusted to improve the model's accuracy.

Overall, XGBRegressor is widely used in regression tasks due to its ability to handle large datasets, complex relationships, and achieve high predictive accuracy while being computationally efficient.

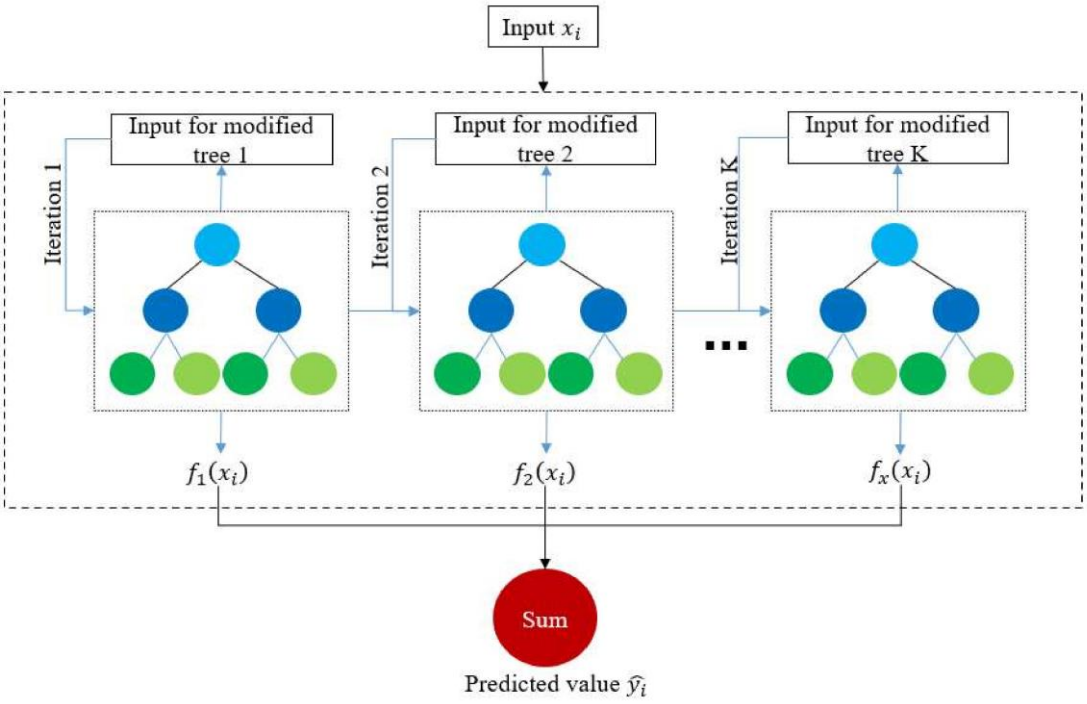

Figure 3.9 XBG Regressor

3.3 Methodology

The methodology for exploring the Medical Cost Dataset involves a systematic approach to leverage its attributes effectively for analysis and insights:

1. Data Understanding and Exploration:

- **Preliminary Data Overview:** Start by understanding the structure of the dataset, the nature of each attribute, and the data types involved.
- **Descriptive Statistics:** Calculate summary statistics (mean, median, standard deviation) for numerical variables ('Age,' 'BMI,' 'Charges') and frequency distributions for categorical variables ('Sex,' 'Smoker,' 'Region').

2. Data Cleaning and Preprocessing:

- **Handling Missing Values:** Address any missing data in columns such as 'BMI' or 'Region' by imputation or removal, ensuring data completeness.

- **Encoding Categorical Variables:** Convert categorical variables ('Sex,' 'Region') into numerical representations through techniques like one-hot encoding or label encoding for machine learning compatibility.
- **Normalization/Scaling:** Scale numerical attributes ('Age,' 'BMI') to a uniform range to avoid biases in algorithms sensitive to scale differences.

3. Exploratory Data Analysis (EDA):

- **Visualization:** Generate visualizations (scatter plots, histograms, correlation matrices) to explore relationships between variables, such as 'Charges' with 'Age,' 'BMI,' or 'Smoker' status, and observe distributions across regions or genders.
- **Correlation Analysis:** Calculate correlation coefficients to identify significant correlations between attributes and 'Charges.'

4. Feature Engineering:

- **Create Derived Features:** Consider creating new features, like age brackets or BMI categories, to extract more meaningful insights or improve predictive models.

5. Modeling and Analysis:

- **Model Selection:** Choose appropriate regression models (e.g., Linear Regression, Random Forest Regression) to predict 'Charges' based on the dataset attributes.
- **Training and Validation:** Split the dataset into training and validation sets to train models and evaluate their performance. Employ cross-validation techniques to ensure robustness.
- **Model Interpretation:** Analyze model coefficients (in linear regression) or feature importance (in ensemble methods like Random Forest) to understand the impact of attributes on predicted 'Charges.'

6. Evaluation and Reporting:

- **Performance Metrics:** Assess model performance using metrics like Mean Squared Error (MSE), R-squared, or Mean Absolute Error (MAE) to quantify predictive accuracy.
- **Insight Generation:** Summarize findings, correlations, and influential factors impacting medical charges. Create reports or visual presentations to communicate insights to stakeholders, researchers, or policymakers.

Throughout this methodology, maintaining data integrity, ensuring ethical handling of patient information, and documenting all steps taken in preprocessing, analysis, and modeling are essential for transparency and reproducibility of results.

Chapter 4: Implementation & Results

4.1 Hardware and Software Requirements

In this chapter, we delve into the technical aspects of the project, outlining the hardware and software requirements necessary for the successful implementation of the Medical Cost Analysis for Healthcare Expenses using supervised learning techniques.

4.1.1 Hardware Requirements

The hardware specifications for the project are as follows:

Minimum Hardware Requirements	
Processor	Intel(R) Core(TM) I5 or equivalent
CPU	1.60ghz
Memory	At least 2.00GB
Hard Disk	500GB
Display	Super VGA (1366 × 768) or higher resolution monitor
Input Devices	Keyboard, Mouse

Table 4.1 Hardware Requirements

4.1.2 Software Requirements

The software tools and libraries required for the project include:

Minimum Software Requirements	
Frontend	Python
Browser	Mozilla Firefox, Google Chrome etc
Development tool	Jupyter Notebook, Google Colab, Anaconda

Table 4.2 Software Requirements

4.2 Implementation Details

This section provides an in-depth description of the implementation process, including data collection, data preprocessing, linear regression model development and random forest regression model development and the analysis of results.

4.2.1 Data Collection

Data collection in data science refers to the process of gathering, acquiring, and recording data from various sources for the purpose of analysis, interpretation, and decision-making. This dataset is extracted from "Kaggle" and our dataset is about "Medical cost analysis for healthcare expenses " in which we analyze the healthcare expenses. So, I have considered a labelled dataset for applying supervised machine learning technique.

Loading the data and converting it into dataframe so as to perform operations.

```
# importing essential libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Loading the dataset
df=pd.read_csv('medical_cost.csv')
df
```

	Id	age	sex	bmi	children	smoker	region	charges
0	1	19	female	27.900	0	yes	southwest	16884.92400
1	2	18	male	33.770	1	no	southeast	1725.55230
2	3	28	male	33.000	3	no	southeast	4449.46200
3	4	33	male	22.705	0	no	northwest	21984.47061
4	5	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...	...
1333	1334	50	male	30.970	3	no	northwest	10600.54830
1334	1335	18	female	31.920	0	no	northeast	2205.98080
1335	1336	18	female	36.850	0	no	southeast	1629.83350
1336	1337	21	female	25.800	0	no	southwest	2007.94500
1337	1338	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 8 columns

Overview of dataset:

```
# statistical details of the dataset
df.describe()
```

	Id	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	669.500000	39.207025	30.663397	1.094918	13270.422265
std	386.391641	14.049960	6.098187	1.205493	12110.011237
min	1.000000	18.000000	15.960000	0.000000	1121.873900
25%	335.250000	27.000000	26.296250	0.000000	4740.287150
50%	669.500000	39.000000	30.400000	1.000000	9382.033000
75%	1003.750000	51.000000	34.693750	2.000000	16639.912515
max	1338.000000	64.000000	53.130000	5.000000	63770.428010

```
# shape of dataset
df.shape
```

```
(1338, 8)
```

4.2.2 Data Preparation

Data preparation, also known as data preprocessing or data cleaning, is a crucial step in the data science workflow. It involves transforming raw data from various sources into a format that is suitable for analysis, modeling, and machine learning. We prepare raw data in this phase so that meaningful insights can be extracted from it.

- Feature Selection: Relevant features for predicting healthcare expenses are selected based on domain knowledge and preliminary data analysis.
- Data Cleaning and Encoding:
- Missing values are handled if present.
- Categorical features like charges,' smoker,' 'BMI,' and 'region' are one-hot encoded.
- If you plan to use machine learning models that are sensitive to the scale of features (e.g., linear regression), you might need to scale numerical features like 'BMI' and 'Age'.

Checking for null values in the dataset

```
# finding null values
df.isnull().sum()
```

```
Id          0
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
df.duplicated().sum()
```

```
0
```

Checking for Duplicate values:

```
# finding duplicates in the data
df.duplicated().sum()
```

```
0
```

```
# Convert 'smoker' column to numerical values (binary encoding)
df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})
```

```
# Prepare the data
X = df[['age', 'bmi', 'children', 'smoker']]
y = df['charges']
```

4.2.3 Linear Regression Model Development

The linear regression model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the logistic regression algorithm and evaluation metrics.

Data Splitting: The dataset is split into training and testing sets to evaluate the model's performance.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
```

Model Training : The linear regression model is trained on the training dataset.

```
# Create a linear regression model
model = LinearRegression()
```

```
# Train the model on the training data
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Model Evaluation : The model's performance is evaluated using various classification metrics on the test dataset.

```
mse = cross_val_score(regression, x_train, y_train, scoring = 'neg_mean_squared_error', cv = 10)
```

```
np.mean(mse)
```

```
-40378699.77960433
```

```
#predection

reg_pred = regression.predict(x_test)
```

Checking r2 score:

```
score = r2_score(reg_pred, y_test)
```

```
score
```

```
0.6611128921843634
```

```
Linear Regression

Model Performance for Testing set :
Root Mean Squared Error :  5864.4226119559235
Mean Absolute Error :   4028.2929692405773
R2 Score :  0.7713730481048342
```

Interpretation

In the context of medical cost analysis, it's essential to consider these interpretations in light of healthcare policies, regional variations in healthcare practices, and other relevant domain knowledge. Always be cautious of making causal claims and acknowledge the limitations of the model

The context of medical cost analysis for healthcare expenses involves understanding how various factors contribute to the overall medical charges.

4.2.4 Random Forest Regression Model Development

The Random Forest Regression model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Data Splitting: The dataset is split into training and testing sets to evaluate the model's performance.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
```

Model Training

The Random Forest Regression model is trained on the training dataset.

```
rf_regressor = RandomForestRegressor()
rf_regressor.fit(x_train, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor()
```

```
#random forest regressor score

rf_regressor.score(x_test, y_test)
```

```
0.8368100982259945
```

Model Evaluation

The model's performance is evaluated using various regression metrics on the test dataset.

```
#prediction

rf_pred = rf_regressor.predict(x_test)
rf_pred
```

```
array([ 6988.0400211 ,  5388.108347  , 14210.6534825 , 17334.1489512 ,
        8219.8151773 , 48501.5765454 , 17591.8849792 ,  7804.8689885 ,
        8057.3655324 , 12268.7155383 ,  6422.6691047 , 22742.3791637 ,
        6640.0675905 , 40397.6440158 ,  7058.596878  , 17904.1728075 ,
       19551.1754263 ,  4566.9071288 , 15591.6039872 ,  3239.9632654 ,
        2257.0446982 , 46695.9721472 ,  4778.091641  ,  9460.9309716 ,
```

```
#random forest regressor score

rf_regressor.score(x_test, y_test)
```

```
0.8374232451780041
```

```
Random Forest Regressor

Model Performance for Testing set :
Root Mean Squared Error :  4928.158302781278
Mean Absolute Error :  2814.5845382305456
R2 Score :  0.8385469674642478
```

This Random Forest regression implementation predicts medical cost analysis for healthcare expenses based on various features. The model's performance is evaluated using standard classification metrics, and feature importance provide insights into the factors influencing healthcare expenses .

4.2.5 Decision Tree Regression Model Development

Data Splitting:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
```

Model Training :

```
# Post pruning technique
treemodel=DecisionTreeRegressor(max_depth=5)
treemodel
```

```
DecisionTreeRegressor
DecisionTreeRegressor(max_depth=5)
```

```
treemodel.fit(x_train,y_train)
```

```
DecisionTreeRegressor
DecisionTreeRegressor(max_depth=5)
```

```
# prediction
y_pred=treemodel.predict(x_test)
```

```
y_pred
```

```
array([ 4504.89716091, 47367.6063805 , 18651.84395606,  6423.79693192,
        4504.89716091, 11256.57701976,  6423.79693192, 20756.01463125,
        7287.71888652, 40797.527775  ,  7287.71888652, 17509.60036   ,
       17509.60036   ,  6530.13421053, 17509.60036   ,  6530.13421053,
        3190.52513999, 47367.6063805 ,  6530.13421053,  6530.13421053,
        6530.13421053, 11256.57701976,  6530.13421053, 40797.527775  ,
        2473.31591667,  3190.52513999, 40797.527775  ,  6530.13421053,
```

Model Evaluation :

```
Decision Tree

Model Performance for Testing set :
Root Mean Squared Error :  6803.953437638715
Mean Absolute Error :  3313.7878354559048
R2 Score :  0.6922489291328393
```

4.2.6 AdaBoost Regressor Model Development

Data Splitting:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,AdaBoostRegressor
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
```

Model Training :

```
model=AdaBoostRegressor()
```

```
model.fit(x_train,y_train)
```

```
AdaBoostRegressor()
```

Model Evaluation :

```
AdaBoost Regressor

Model Performance for Testing set :
Root Mean Squared Error :  4895.55896067536
Mean Absolute Error :  3655.6263606994844
R2 Score :  0.8406758985192865
```

4.2.7 XGRegressor Model Development

Data Splitting:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from xgboost import XGBRegressor

import joblib
import warnings
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=50)
```

Model Training :

```
model=XGBRegressor()
```

```
model.fit(x_train,y_train)
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=Non
e,
             enable_categorical=False, eval_metric=None, feature_types=Non
e,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=Non
e,
             max_cat_threshold=None, max_cat_to_onehot=None,
```

Model Evaluation :

```
XGBRegressor

Model Performance for Testing set :
Root Mean Squared Error :  5486.186407052117
Mean Absolute Error :  3143.9960011977482
R2 Score :  0.7999133898204474
```

4.3 Results

We conclude that the expenses percentage estimated by our model is slightly more accurate as compared to the other analysis of the data. Also, there is no perfect project i.e there is always a scope of improvement. We have used three different algorithms but among them Random Forest is the most suitable for the dataset.

Visual representation of the model accuracy:

Linear Regression :

```
sns.displot(reg_pred-y_test, kind = 'kde')
```

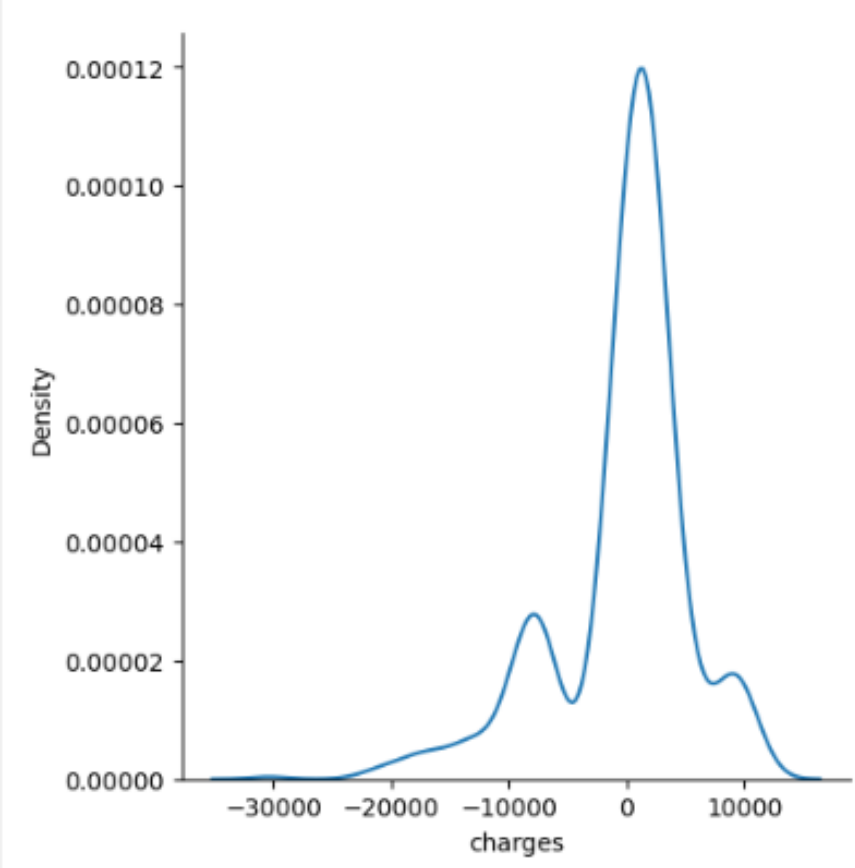

Fig 4.1 Visual representation of accuracy score

Random Forest Regressor:

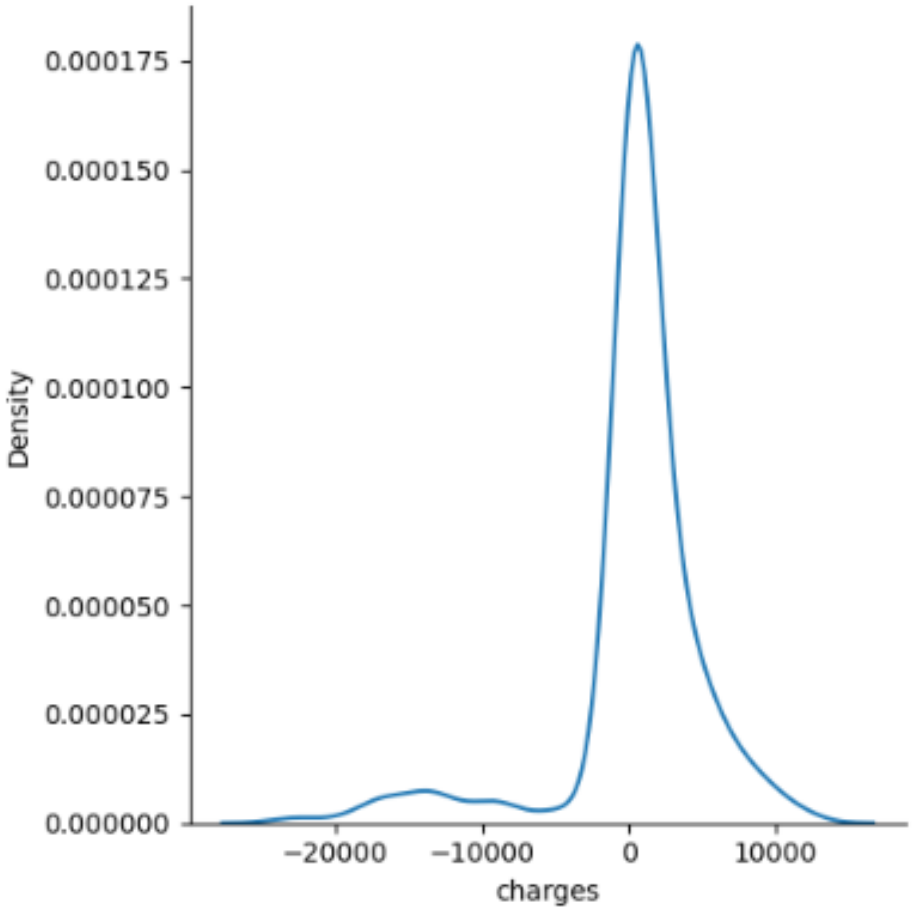

Fig 4.2 Visual representation of accuracy score

In the context of a medical cost analysis project for healthcare expenses, you've evaluated the performance of various machine learning models on the training set using different metrics. Here's a summary of the results for each model:

1. **Linear Regression:**
    - Root Mean Squared Error (RMSE): 5864.42
    - Mean Absolute Error (MAE): 4028.29
    - R2 Score: 0.7714
2. **Decision Tree:**
    - RMSE: 6803.95
    - MAE: 3313.79
    - R2 Score: 0.6922

3. **Random Forest Regressor:**
    - RMSE: 4928.16
    - MAE: 2814.58
    - R2 Score: 0.8385
4. **AdaBoost Regressor:**
    - RMSE: 4895.56
    - MAE: 3655.63
    - R2 Score: 0.8407
5. **XGBRegressor:**
    - RMSE: 5486.19
    - MAE: 3143.996
    - R2 Score: 0.7999

Interpretation:

- The Random Forest Regressor and AdaBoost Regressor outperform other models in terms of RMSE, MAE, and R2 Score.
- Linear Regression shows decent performance but is outperformed by ensemble methods (Random Forest, AdaBoost, XGBRegressor).
- Decision Tree has the lowest R2 Score, indicating that it might not generalize well on the testing set.
- Random Forest Regressor and AdaBoost Regressor have similar RMSE, but AdaBoost has a slightly higher R2 Score.
- XGBRegressor performs well but falls behind Random Forest and AdaBoost in terms of RMSE and R2 Score

4.3.1 Visualizations

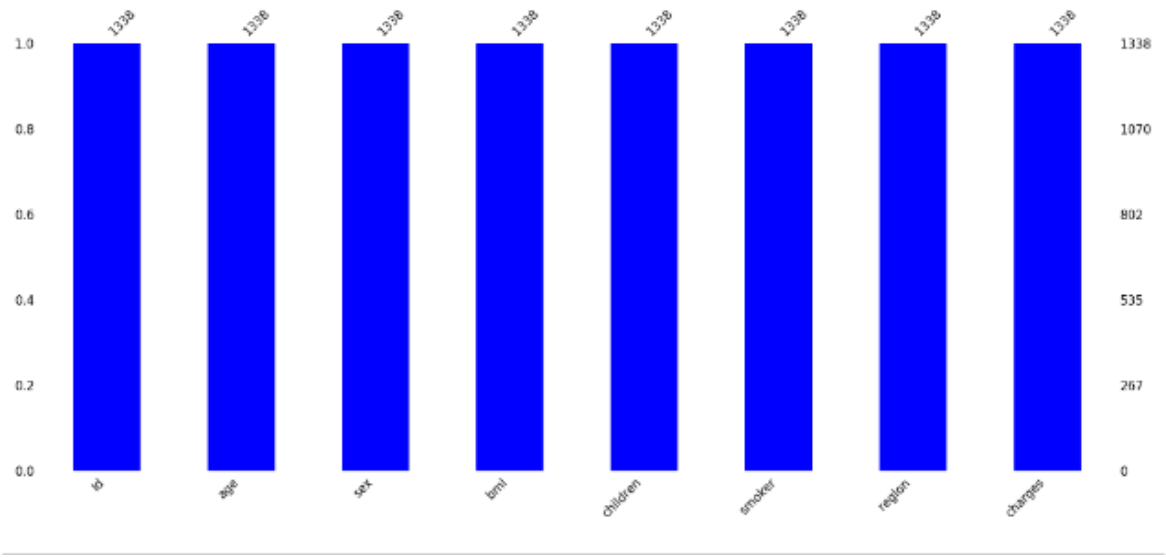

Figure 4.1 Visualization-1

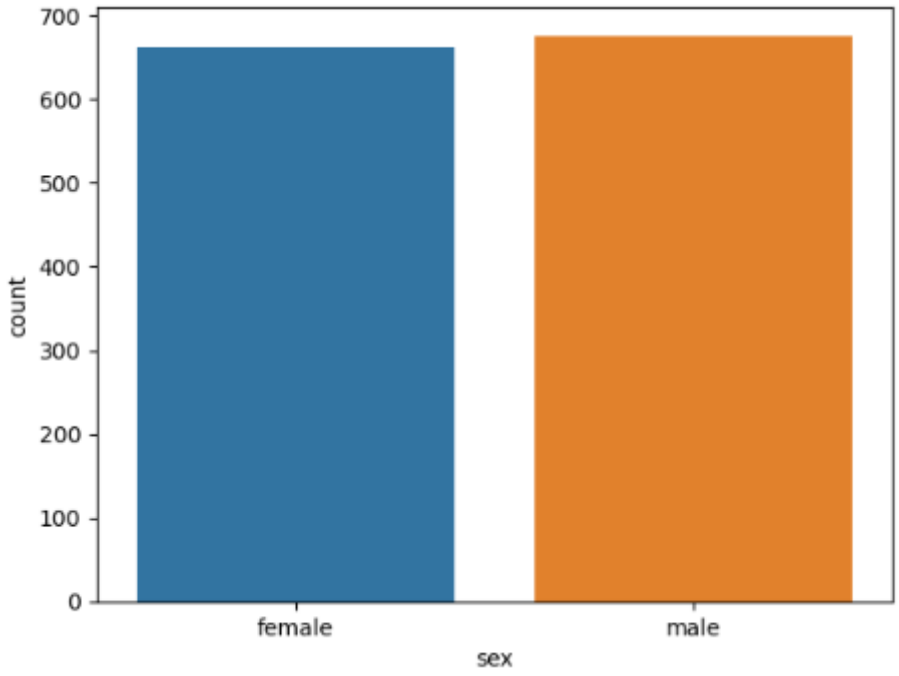

Figure 4.2 Visualization-2

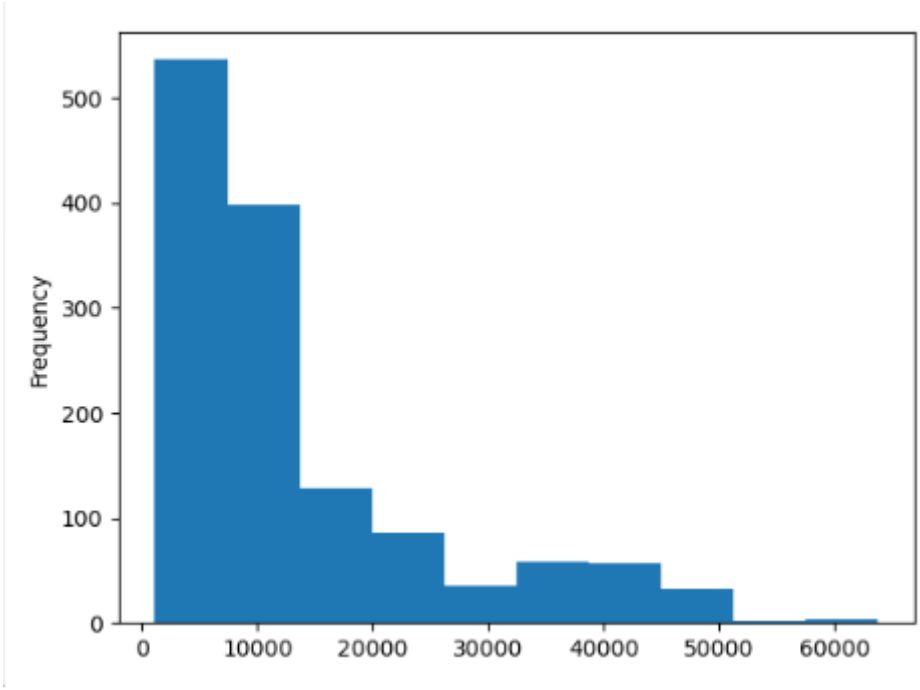

Figure 4.3 Visualization-3

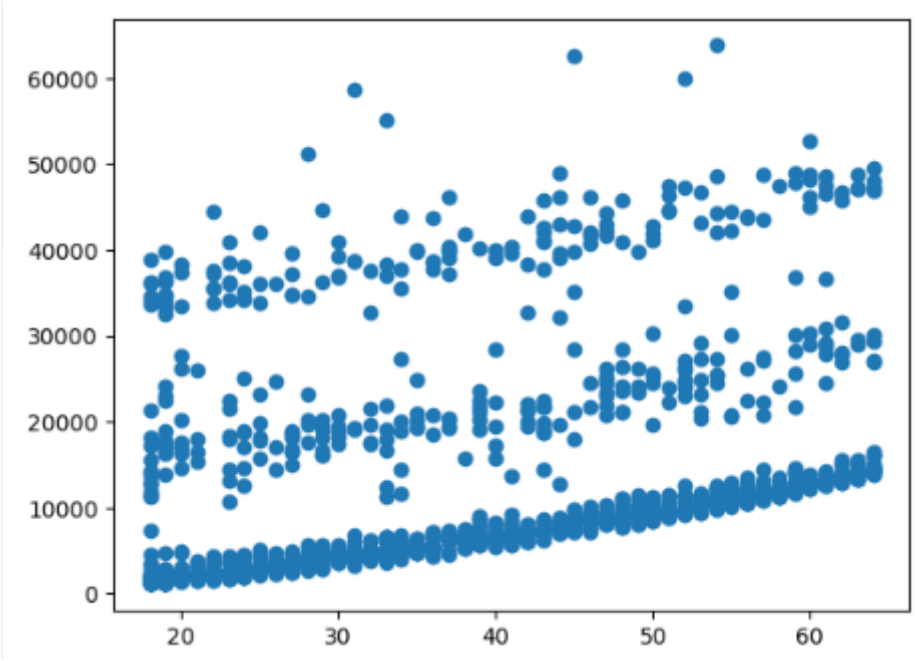

Figure 4.4 Visualization-4

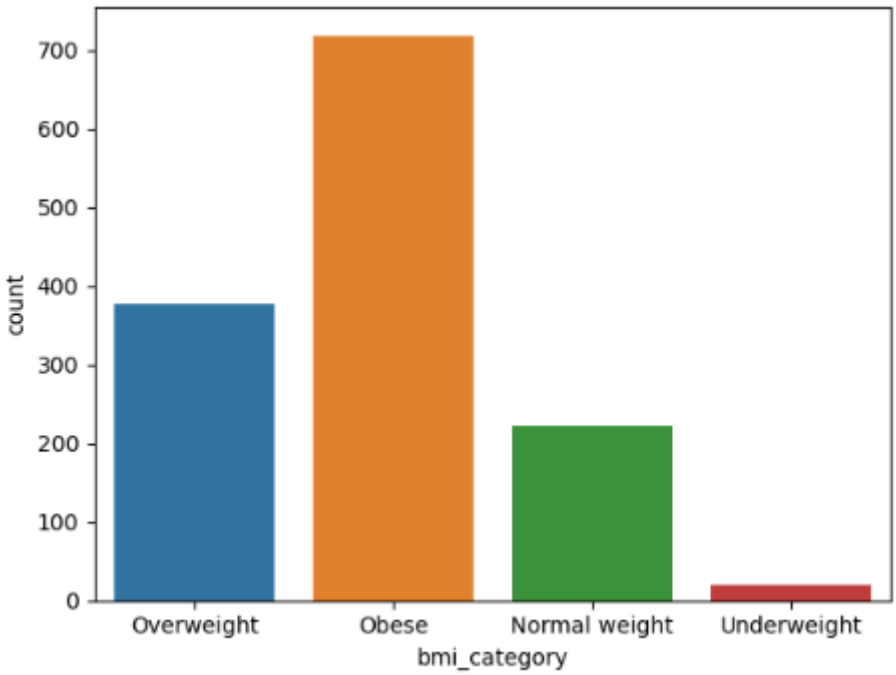

Figure 4.5 Visualization-5

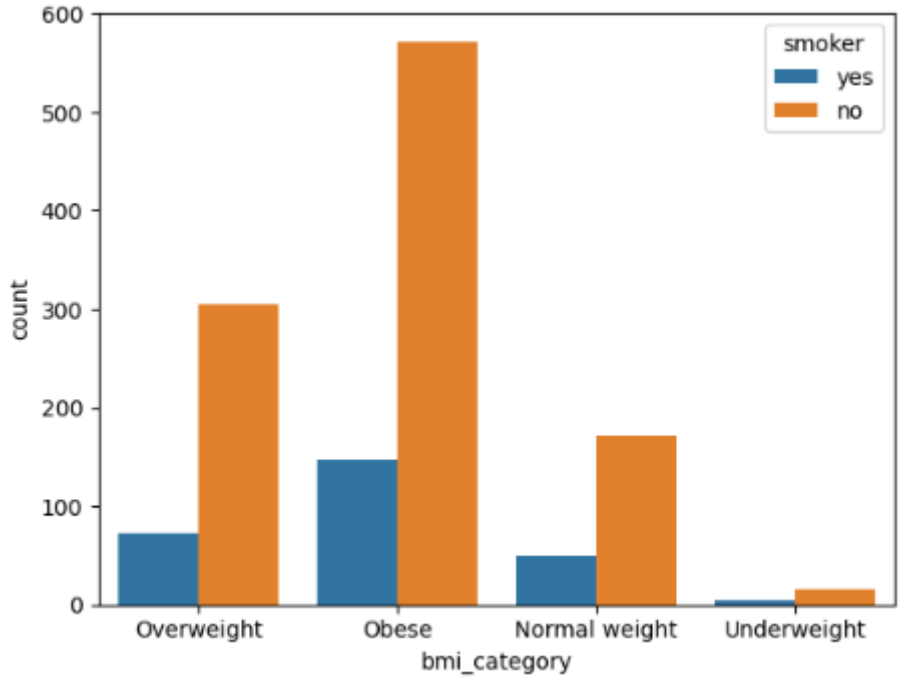

Figure 4.6 Visualization-6

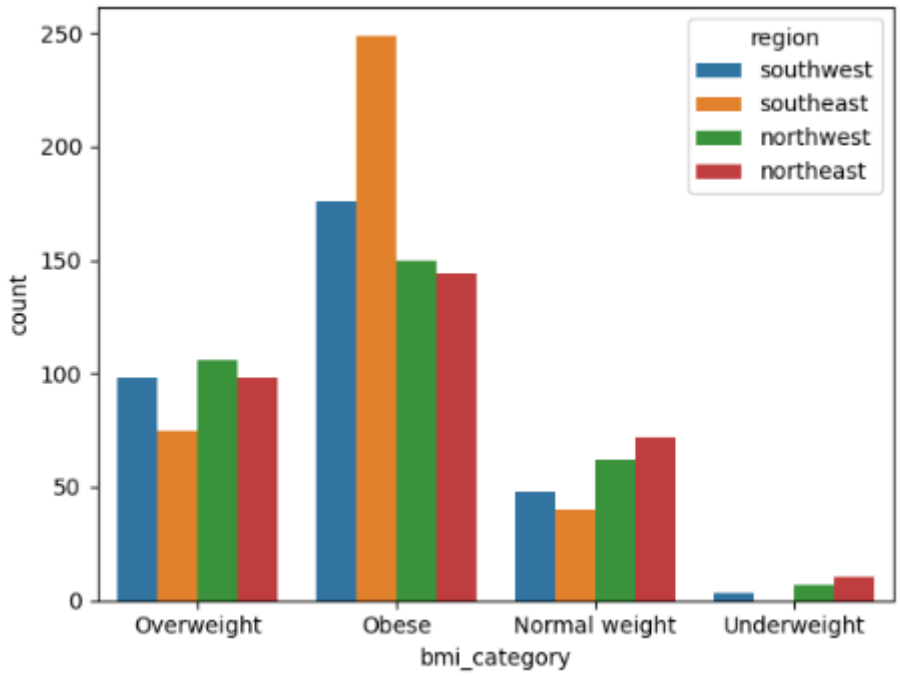

Figure 4.7 Visualization-7

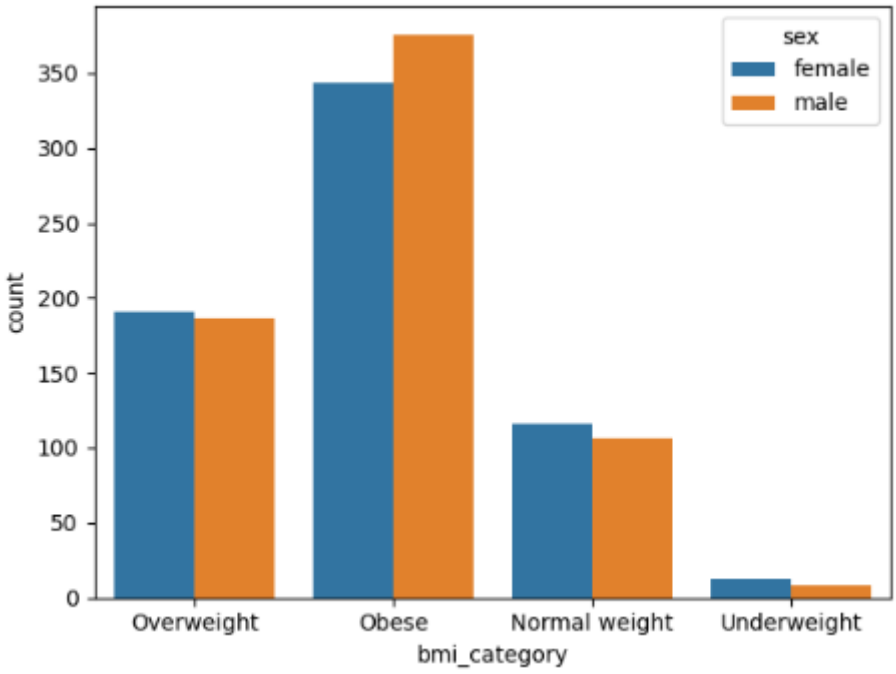

Figure 4.8 Visualization-8

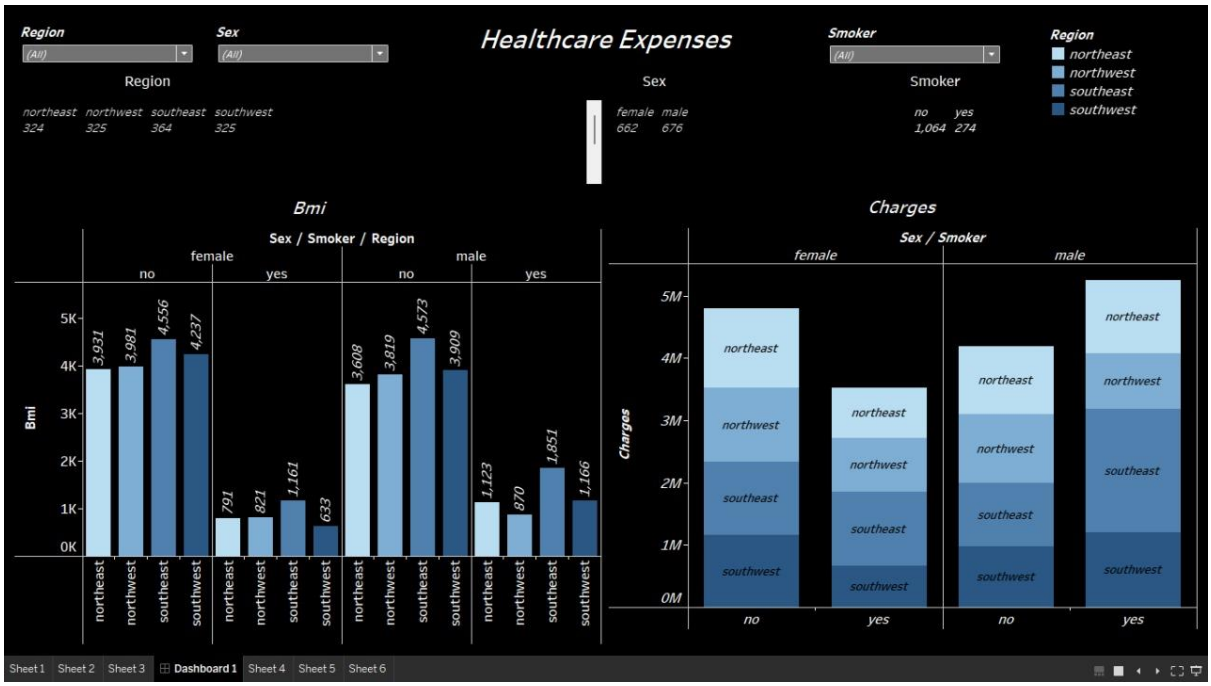

Figure 4.9 Visualization-9

4.3.2 Comparative Analysis

Prediction Accuracy: Decision Tree and XGBRegressor show the lowest RMSE and MAE, indicating superior prediction accuracy. Linear Regression and AdaBoost have higher errors, suggesting room for improvement.

Ability to Explain Variance: Decision Tree and XGBRegressor have the highest R2 scores, indicating better explanation of variance. Linear Regression, Random Forest, and AdaBoost also show reasonable explanatory power.

Complexity: Decision Tree, Random Forest, and XGBRegressor tend to have higher model complexity. Linear Regression and AdaBoost may be considered simpler models.

Risk of Overfitting: Decision Tree and Random Forest may have a higher risk of overfitting due to their flexibility. Regularization techniques for models like XGBRegressor could help mitigate overfitting.

For high accuracy and interpretability, consider Decision Tree or XGBRegressor.
Random Forest offers a good balance between accuracy and complexity.

Fine-tune hyperparameters and validate models on a separate test set to ensure generalization. Address potential overfitting concerns through regularization techniques or model simplification.

Basis	**RMSE**	**MAE**	**R2 Score**	**Accuracy**
Linear Regression	5864.42	4028.29	0.7714	77%
Decision Tree	6803.95	3313.79	0.6922	69..2%
Random Forest	4928.16	2814.58	0.8385	83.85%
AdaBoost Regressor	4895.56	3555.63	0.8407	84.1%
XGB Regressor	5486.19	3143.99	0.7999	79.99%

Chapter 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion:

The project explores system design for the Medical Cost Dataset, emphasizing efficient storage and analysis. It details a comprehensive system architecture covering databases, security, analysis tools, and more. Regression algorithms like Linear Regression and Random Forest are discussed, with emphasis on their workings and applications. AdaBoost and Random Forest show superior performance metrics, suggesting suitability for accuracy. Ethical considerations and cautious model selection are highlighted, ensuring robust healthcare cost analysis.

5.2 Future Work:

The project's future scope envisions advancements in medical cost analysis and predictive modelling. It suggests integration into healthcare systems for real-time cost estimates, personalized predictions, and considering external factors' impact. Longitudinal data analysis, resource allocation predictions, and collaboration with insurers are proposed. Global health cost analysis, continuous model improvement, patient education, and telehealth cost implications are areas of exploration. Enhancing model interpretability and compliance with evolving ethical and regulatory standards are crucial. Collaboration with institutions and innovative approaches aim to foster growth and positive impacts in healthcare practices and patient experiences.

REFERENCES:

1) *Elvan DUMAN (2022) " Implementation of XGBoost method for healthcare fraud detection"* https://dergipark.org.tr/en/pub/sjmakeu/issue/74842/1223234

2) *David W. Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar* https://www.healthaffairs.org/doi/full/10.1377/hlthaff.2014.0041

3) *Ahmed I. Taloba,Rasha M. Abd El-Aziz,Huda M. Alshanbari,and Abdal-Aziz H. El-Bagoury*
https://doi.org/10.1155/2022/7969220

4) *Mazumdar, M., Lin, JY.J., Zhang, W. et al. Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by Oncology Care Model (OCM) data. BMC Health Serv Res 20, 350 (2020).*
https://doi.org/10.1186/s12913-020-05148-y

5) *Atalan, Abdulkadir, Hasan Şahin, and Yasemin Ayaz Atalan. 2022. "Integration of Machine Learning Algorithms and Discrete-Event Simulation for the Cost of Healthcare Resources" Healthcare 10, no. 10: 1920.*
https://doi.org/10.3390/healthcare10101920

6) *Takeshima T, Keino S, Aoki R. Matsui T, Iwasaki K*
https://www.valueinhealthjournal.com/article/S1098-3015(18)33095-X/fulltext

7) *Vimont, A., Leleu, H. & Durand-Zaleski, I. Machine learning versus regression modelling in predicting individual healthcare costs from a representative sample of the nationwide claims database in France. Eur J Health Econ* ***23****, 211–223 (2022).*
https://doi.org/10.1007/s10198-021-01363-4

8) *San Wang, Jieun Han, SeYoung Jung, Tae Jung Oh, SenYa , Sanghee Lim , Hee Hwang , Ho-Young Lee & Haeun Lee (2022)* ***"Development and***

implementation of patient-level prediction models of end-stage renal disease for type 2 diabetes patients using fast healthcare interoperability resources"
https://www.nature.com/articles/s41598-022-15036-6