AN EFFICIENT STACKING APPROACH FOR HEALTHCARE INSURANCE COST PREDICTION

A MINI PROJECT REPORT

Submitted by

LAVANYA S (221801028)
MONISHA M (221801034)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY, CHENNAI

NOV 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**AN EFFICIENT STACKING APPROACH FOR HEALTHCARE INSURANCE COST PREDICTION**" is the bonafide work of **LAVANYA S (221801028), MONISHA M (221801034)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. J.M. Gnanasekar
Professor and Head
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Mr. Thiyagarajan.G
Assistant Professor
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN., B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN., B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN., Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN., M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J.M. GNANASEKAR., M.E., Ph.D, Professor and Head of the Department of Artificial Intelligence and Data Science for her guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Mr. THIYAGARAJAN G., B.E., M.E., Ph.D., Assistant Professor, Department of Artificial Intelligence and Data Science and coordinator, Dr. P. INDIRA PRIYA., M.E., Ph.D., Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

In the rapidly evolving landscape of the insurance industry, accurately predicting healthcare insurance premiums has become a critical challenge. Premium amounts are determined by a range of factors, including the policyholder's age, medical history, and potential risk factors such as chronic diseases, lifestyle choices, and genetic predispositions. The goal of this project is to develop an advanced machine learning model that accurately predicts health insurance premiums based on multiple medical and demographic characteristics. Using an overlay approach, the model combines the strengths of CatBoost, XGBoost, and RandomForest algorithms to improve the reliability and accuracy of predictions.

The dataset used in this project includes features such as age, history of diabetes, blood pressure problems, transplants, chronic diseases, height, weight, Body Mass Index (BMI), known allergies, cancer history in the family, and the number of major surgeries. These variables are carefully preprocessed, numeric variables expanded through polynomial feature engineering to capture complex relationships within the data. A key preprocessing step was to remove outliers using the interquartile range (IQR) method to ensure the generalizability of the model.

Hyperparameter tuning was performed using GridSearchCV to optimize each model in the stack, resulting in a final model that effectively balanced bias and variance. The underlying learning algorithms (CatBoost, XGBoost, RandomForest) work in parallel to capture different aspects of the dataset, and the meta-model, Ridge Regression, combines their predictions to provide more accurate results. The performance of the models is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R-Squared (R^2), demonstrating that the stacking approach outperforms individual models by reducing prediction error while also improving overall accuracy. The system is designed to be integrated into a web-based Django framework, allowing users to input their medical details and receive an estimated premium instantly.

ANNEXURE I

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Health costs are a serious problem in the world, as medical expenses are constantly increasing based on the fields of treatment, drugs, and technology. As a result, medical insurance has turned into an important mechanism for managing these expenses. However, the calculation of health insurance premiums is often complex and lacks transparency. Traditionally, insurance companies have relied on generalized models that use broad demographic factors like age, gender, and income to determine premiums. While these models are useful, they often overlook more individualized and specific health factors that can vary greatly from person to person, such as chronic diseases, previous surgeries, and lifestyle choices.

1.2 NEED FOR THE STUDY

The need for this study is driven by several pressing challenges in the current healthcare insurance landscape. First, there is an increasing demand for more personalized insurance pricing models. Policyholders are often unaware of the exact reasons behind their premium amounts, leading to frustration and mistrust in insurance companies. While some individuals might feel that they are overpaying for insurance despite being in good health, others may be underinsured because their specific health risk are not adequately accounted for.

Moreover, the rise in the availability of healthcare data presents an opportunity to shift towards more accurate, individualized premium predictions. With access to vast amounts of health-related data, such as medical history, biometric measurements, and lifestyle indicators, machine learning models can be trained to predict insurance premiums with greater precision. This enables the insurance industry to move towards risk-based pricing, where premiums are directly tied to an individual's unique health profile. The integration of SHAP values ensures

that the model not only makes accurate predictions but also provides a clear explanation of the major factors contributing to a person's premium, offering transparency and fostering trust between insurers and policyholders

1.3 OBJECTIVES OF THE STUDY

The main objectives of this study are as follows:

- To create a machine learning model capable of accurately predicting healthcare insurance premiums based on a combination of health-related and demographic factors.
- To employ a stacking approach that combines the predictive power of multiple models (CatBoost, XGBoost, RandomForest) to enhance accuracy and robustness.
- To utilize Polynomial Feature Expansion to capture non-linear relationships in the numeric data, improving the model's ability to make accurate predictions.
- To integrate SHAP (SHapley Additive exPlanations) into the model to provide clear, interpretable insights into how each factor affects the premium prediction.
- To design a system that allows users to input their personal health data and receive not only a predicted premium but also a detailed explanation of the major factors driving the prediction, enabling more personalized and fair pricing models for policyholders.

By meeting these objectives, this study provides a valuable tool for both insurance companies and policyholders. Insurance companies benefit from a more accurate and transparent method of pricing premiums, while policyholders gain insight into how their health and lifestyle choices influence the cost of their insurance. The integration of machine learning models and SHAP values ensures that the system is both cutting-edge in terms of predictive performance and user-friendly in terms of interpretability.

1.4 OVERVIEW OF THE PROJECT

The core objective of this project is to develop a system that predicts healthcare insurance premiums using a stacking ensemble of machine learning models. The models used include CatBoost, XGBoost, and RandomForest, which are combined to leverage their strengths. By stacking these models, the system achieves higher accuracy than any individual model could provide. The dataset used in this project consists of several health-related and demographic features such as Age, Diabetes, Blood Pressure Problems, Any Transplants, Any Chronic Diseases, Height, Weight, BMI, Known Allergies, History of Cancer in the Family, and Number of Major Surgeries. These features are transformed using Polynomial Feature Expansion for numerical variables, which helps capture complex relationships between the factors.

The output of the model is a predicted insurance premium based on the input data. In addition to predicting the premium, the model incorporates SHAP values to explain how each feature influences the prediction. This makes the system not only predictive but also interpretable. SHAP values break down the prediction into the contributions of each feature, allowing the user to understand which factors have the most significant impact on their premium. The system presents both the predicted premium and an explanation of the top five features that contributed to the prediction, making it a transparent and user-friendly solution for insurance premium calculation.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

The healthcare insurance industry has undergone significant changes over the years, particularly with the increasing availability of vast amounts of health-related data and the emergence of advanced data analytics techniques. Machine learning models have demonstrated great potential in improving insurance premium prediction accuracy by leveraging this data. Traditional methods often relied on static actuarial approaches, which, while effective for broad population groups, fail to account for individual-specific health conditions and risk factors. As more personalized healthcare data becomes available, machine learning offers an opportunity to design more sophisticated, dynamic models for premium calculation.

A growing body of literature focuses on the application of machine learning in the healthcare insurance domain, particularly in the areas of premium prediction and risk assessment. Several studies have explored the use of ensemble methods, which combine multiple models to enhance prediction accuracy. Additionally, the interpretability of machine learning models has become a critical focus area, with techniques like SHAP (SHapley Additive exPlanations) gaining attention for providing insights into model predictions. This review of literature will discuss the background and framework related to the integration of machine learning models, ensemble approaches, and interpretability tools for healthcare insurance premium prediction.

2.2 FRAMEWORK OF LCA (LITERATURE CRITICAL ANALYSIS)

The critical analysis of existing literature on healthcare insurance premium prediction highlights the limitations of traditional actuarial models and the potential benefits of advanced machine learning approaches. A well-structured framework of Literature Critical Analysis (LCA) helps to examine the existing research and identify gaps that the current project aims to address. The LCA framework for this

review is based on three key components: predictive modeling, feature engineering, and interpretability.

Predictive Modeling:

Research indicates that ensemble methods, particularly stacking models such as CatBoost, XGBoost, and RandomForest, have outperformed single-model approaches in terms of predictive accuracy. Each of these models brings unique strengths: CatBoost efficiently handles categorical features, XGBoost is known for its scalability and performance, and RandomForest is effective in managing noise and variance. Multiple studies have reported improved
performance when using stacked models compared to individual models. This suggests that leveraging multiple models can capture more complex patterns in the data.

In the context of healthcare insurance, studies have shown that traditional linear models fail to capture non-linear relationships between health-related features and insurance premiums. Polynomial Feature Expansion, as highlighted in several studies, provides a way to address this limitation by transforming features into higher-dimensional space, thus allowing models to capture more intricate relationships between variables.

Feature Engineering:

The literature emphasizes the importance of feature engineering in improving the performance of machine learning models. Features such as age, body mass index (BMI), pre-existing medical conditions (e.g., diabetes, hypertension), and family history of diseases have been consistently shown to be strong predictors of healthcare costs and, consequently, insurance premiums. Several studies have noted that careful preprocessing of numeric and categorical features, along with the creation of interaction terms, can significantly boost model performance.

Recent research also suggests that incorporating polynomial features, particularly in models like CatBoost and XGBoost, allows for capturing higher-order interactions among variables. This is critical in healthcare insurance, where factors like BMI,

number of surgeries, and family medical history may have combined effects on premium amounts.

Model Interpretability:

One of the main challenges associated with machine learning models in insurance applications is their complexity and lack of interpretability. Insurers and policyholders alike require explanations for how predictions are made. This has led to the growing adoption of interpretability tools, particularly SHAP, in recent research. SHAP provides a unified measure of feature importance, offering insights into how each feature contributes to the final prediction.

Literature highlights that integrating SHAP values into machine learning models offers transparency and fosters trust in premium predictions. Multiple studies have demonstrated the value of SHAP in healthcare-related applications, allowing both insurers and customers to understand which factors (e.g., age, medical history) have the most significant impact on their premiums. This is particularly relevant as personalized healthcare insurance becomes more prevalent.

In summary, the literature highlights the importance of combining advanced machine learning models with feature engineering and interpretability techniques to address the limitations of traditional actuarial models. The current project builds on these insights by implementing a stacking ensemble of CatBoost, XGBoost, and RandomForest, using polynomial feature expansion and integrating SHAP values to enhance both accuracy and interpretability in healthcare insurance premium prediction.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

In the healthcare insurance sector, the current systems predominantly rely on traditional actuarial methods and machine learning models to predict premium amounts. The most widely used approaches involve linear regression, which assumes a linear relationship between the input features and the target variable (premium). While linear regression models are simple and interpretable, they often fall short in capturing the complex and non-linear relationships that exist between an individual's health conditions and the insurance premium.

Additionally, single-model approaches, such as XGBoost, have gained popularity due to their strong performance in various prediction tasks. XGBoost, known for its gradient boosting framework, provides accurate predictions but still relies heavily on its single-model approach. Although XGBoost offers some improvements over linear regression by capturing more complex relationships, it is limited in its ability to fully explore interactions between multiple features and risk factors, particularly in healthcare scenarios where the interdependence of features such as age, BMI, chronic diseases, and medical history significantly affects premium calculations.

The limitation of relying on single models is that they may not generalize well across diverse datasets and may underperform when faced with noisy or incomplete data. Furthermore, many of these models lack transparency in terms of explainability, making it difficult for insurers to justify the premium amounts to customers based on their individual health profiles. Thus, there is a need for more robust and interpretable models that can better account for the non-linearities and interactions present in healthcare data.

3.2 PROPOSED SYSTEM

The proposed system addresses the limitations of existing models by introducing a stacked ensemble approach, which combines multiple machine learning models—namely CatBoost, XGBoost, and RandomForest—along with advanced feature engineering techniques such as Polynomial Feature Expansion. By stacking these models, the system leverages the strengths of each model: CatBoost's ability to handle categorical features efficiently, XGBoost's robust performance, and RandomForest's resistance to overfitting. This ensemble approach not only enhances predictive accuracy but also improves generalization across different datasets.

Additionally, the proposed system integrates SHAP (SHapley Additive exPlanations) to provide interpretability. By generating SHAP values for each prediction, the system explains the contribution of each feature to the final premium amount, offering transparency and building trust in the prediction process. Policyholders will now be able to understand which health factors (e.g., age, BMI, medical history) have the greatest impact on their premium, empowering them with actionable insights into their healthcare and insurance decisions.

3.3 FEASIBILITY STUDY

The feasibility of the proposed system has been evaluated from three key perspectives: technical, operational, and economic.

Technical Feasibility: The proposed solution is technically feasible as it builds upon well-established machine learning techniques and frameworks. The use of CatBoost, XGBoost, and RandomForest ensures that the system can efficiently handle large-scale healthcare datasets with diverse feature sets, including both categorical and numeric data. The SHAP framework for interpretability has been successfully applied in various domains, making it a reliable tool for this project.

Operational Feasibility: Operationally, the system is user-friendly and can be easily integrated into existing insurance platforms. The inclusion of SHAP

explanations makes the system transparent and understandable, improving both customer experience and insurer decision-making processes. The user-friendly interface allows insurance agents and policyholders to interact with the system without requiring extensive technical knowledge.

Economic Feasibility: From an economic perspective, the system is cost-effective due to the open-source nature of the tools and frameworks used, such as Python, CatBoost, and SHAP. The ensemble approach also reduces the need for manual intervention in premium calculation, saving time and resources for insurance companies. By improving prediction accuracy and reducing potential premium miscalculations, the system offers long-term cost benefits through improved customer satisfaction and reduced claim risks.

In conclusion, the proposed system is both technically and economically feasible and offers significant operational advantages over the existing system, making it a valuable addition to the healthcare insurance industry.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- To effectively run the healthcare insurance premium prediction system, the following hardware components are recommended to ensure smooth processing and high performance:
- Processor: Intel Core i5 or higher (or equivalent AMD Ryzen processor):
  A multi-core processor is necessary for handling complex machine learning computations and model training efficiently.
- RAM: Minimum 8 GB (16 GB recommended):
  Adequate RAM ensures smooth operation, especially when processing large datasets and performing model training that requires significant memory usage.
- Storage: Minimum 256 GB SSD (Solid-State Drive)
  Fast storage (SSD) helps in quicker data retrieval, model loading, and efficient handling of read/write operations for datasets.
- Graphics Processing Unit (GPU): Optional, but recommended for faster model training (e.g., NVIDIA GeForce GTX 1050 or higher).
  While not mandatory, a dedicated GPU can significantly accelerate training times for machine learning models, especially when working with large datasets or deep learning models.
- Other peripherals: Keyboard, mouse, monitor, and internet connectivity for smooth operation and integration with cloud services (if necessary).

4.2 SOFTWARE REQUIREMENTS

The following software components are necessary for implementing the proposed system:

- Operating System: Windows 10/11, macOS, or any Linux distribution (e.g., Ubuntu 20.04 or higher)

- Programming Languages: Python 3.8 or higher: Python will be used for developing the machine learning models, data processing, and system integration.

Libraries/Packages/Frameworks:

- Django Framework: Version 4.0 or above is used to create the web interface. Django provides a robust and scalable platform for integrating the machine learning model and user interaction.
- CatBoost: For handling categorical features and building an efficient gradient boosting model.
- XGBoost: For constructing gradient-boosted trees to enhance predictive performance.
- RandomForest: For building an ensemble model that can handle non-linear relationships in the data.
- SHAP (SHapley Additive exPlanations): For interpreting the output of the model by generating explanations for each prediction.
- scikit-learn: For preprocessing, model evaluation, and machine learning utilities such as polynomial feature expansion.
- Pandas: For data manipulation and handling large datasets efficiently.
- NumPy: For numerical computations and matrix operations.
- Matplotlib/Seaborn: For generating data visualizations and SHAP plots.
- Joblib: For saving and loading machine learning models.

Integrated Development Environment (IDE):

- Jupyter Notebook or VS Code (with Python extensions): These environments allow for interactive coding and visualization, making it easier to work with data science projects.
- Browser: A modern web browser (e.g., Chrome, Firefox) is needed for users to interact with the Django web interface.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The system architecture is designed to predict healthcare insurance premium amounts based on the personal and medical information provided by the user. The architecture integrates machine learning models and a web-based user interface built using Django. The core functionality involves taking user inputs, preprocessing the data, applying the stacking machine learning model, and then interpreting the model's output using SHAP (SHapley Additive exPlanations) for explainability.

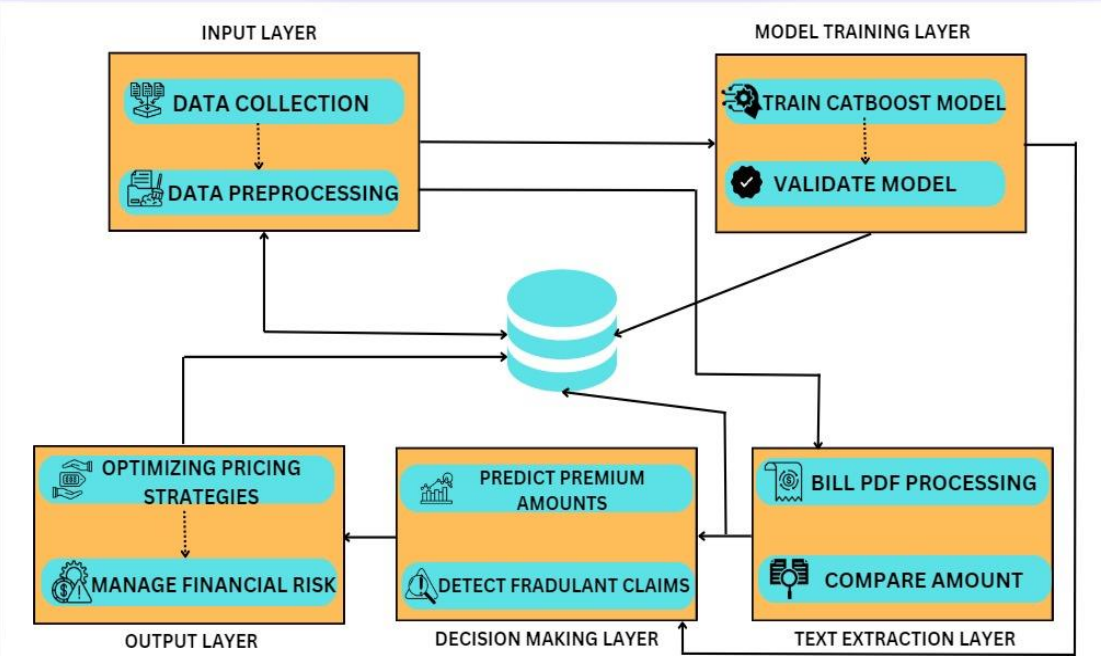

Figure 1: System Architecture

The system has multiple components:

- User Interface: A Django-based web application allows users to input their data and view predicted premium amounts along with feature interpretations.
- Pre-trained Model: The stacking model, which combines CatBoost, XGBoost, and RandomForest, is used for premium prediction.

- Model Interpretation: SHAP is used to interpret the results and provide insights into the most influential features.
- Database: Stores user information, predicted premium amounts, and feature importance explanations for further analysis.

5.2 MODULE DESCRIPTION

5.2.1 PREPROCESSING MODULE

The first module handles the preprocessing of the data entered by the user. The steps include:

- Outlier Removal: The system cleans the dataset to remove any outliers using the IQR method, ensuring that extreme values do not distort the predictions.
- Feature Engineering: Numeric features such as Age, Height, Weight, and BMI are used to create polynomial features. This enhances the model's ability to capture non-linear relationships between features.

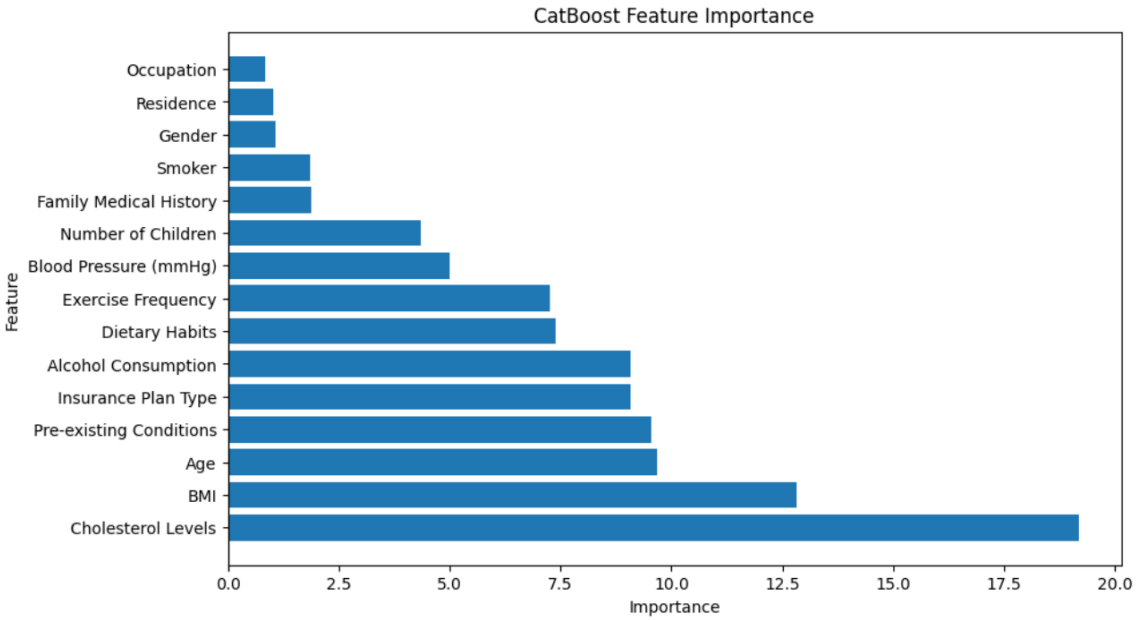

Figure 2: Feature Importance Plot

5.2.2 PREDICTIVE MODEL TRAINING MODULE

The second module involves the training of a stacking model to predict premium prices. This includes:

1. **Stacking Model Workflow**
2. **Base Models**: Three base machine learning models—CatBoost, XGBoost, and RandomForest—are chosen for their robust predictive capabilities. Each model is trained separately on the input features to predict premium prices.
3. **Outputs of Base Models**: After training, each base model produces an initial prediction for the premium amount. These predictions are then used as inputs for the next layer in the stacking process.
4. **Meta-Learner (Ridge Regression)**: The predictions generated by the base models are fed into a Ridge regression model, which acts as a meta-learner. Ridge regression is chosen because it reduces overfitting while minimizing prediction error by adding an L2 regularization term.
5. **Model Tuning with GridSearchCV**: To achieve optimal performance, each base model undergoes hyperparameter tuning using GridSearchCV. This process iterates over a range of hyperparameter combinations to identify those that maximize the accuracy of each model.

Key Formulas

In the process of calculating the premium amount using stacking, we use the following formulas:

1. Ridge Regression:

Ridge regression incorporates an L2 regularization term, penalizing large coefficients to reduce overfitting. The premium prediction, y^\hat{y}y^, is calculated as:

$$\hat{y} = \beta_0 + \sum_{j=1}^{k} \beta_j x_j$$

where:

- β0 is the intercept,
- βj are the coefficients for each base model's prediction,
- xj is the prediction from the j-th base model.

The Ridge regression objective function includes an additional penalty term, given by:

$$J(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{k} \beta_j^2$$

Here, λ is a regularization parameter tuned through GridSearchCV to control the balance between model complexity and predictive accuracy.

2. Model Predictions (Base Models):

Each base model (CatBoost, XGBoost, RandomForest) generates predictions that serve as inputs to the meta-learner. Their calculations depend on internal algorithms:

- **CatBoost**: Predicts by gradient boosting, reducing prediction errors over iterations.
- **XGBoost**: Uses optimized gradient boosting, adjusting weights to improve the prediction accuracy.
- **RandomForest**: Averages predictions from multiple decision trees to produce a final estimate.

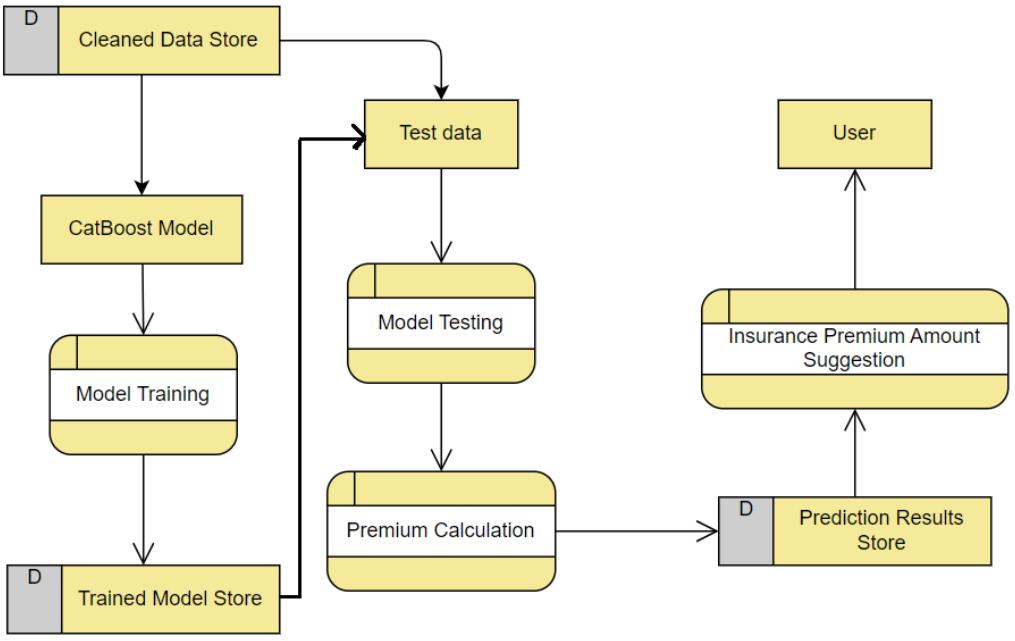

Figure 3: Prediction Module DFD

5.2.3 PREDICTION & EVALUATION MODULE

This module predicts the premium based on the pre-trained stacking model. The user input is processed through the trained model, and the premium amount is predicted.

- Evaluation Metrics: The system uses metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) to evaluate the model's performance and compare it with individual models, achieving an R^2 score of 0.904.

```
Stacking MSE: 2311550.8872222975, RMSE: 1520.3785341888702
Mean Absolute Error (MAE): 747.3858755699139
R-squared (R²): 0.9043719355393718
```

Figure 4: Evaluation Module

5.2.4 SHAP INTERPRETATION MODULE

Once the premium amount is predicted, SHAP is used to provide an explanation of which features had the most influence on the prediction.

- SHAP Value Calculation: SHAP values are calculated for each feature in the dataset, showing how much each feature increased or decreased the premium prediction.
- User Interpretation: The user interface displays the most influential factors that led to the final premium prediction, providing transparency in how the decision was made.

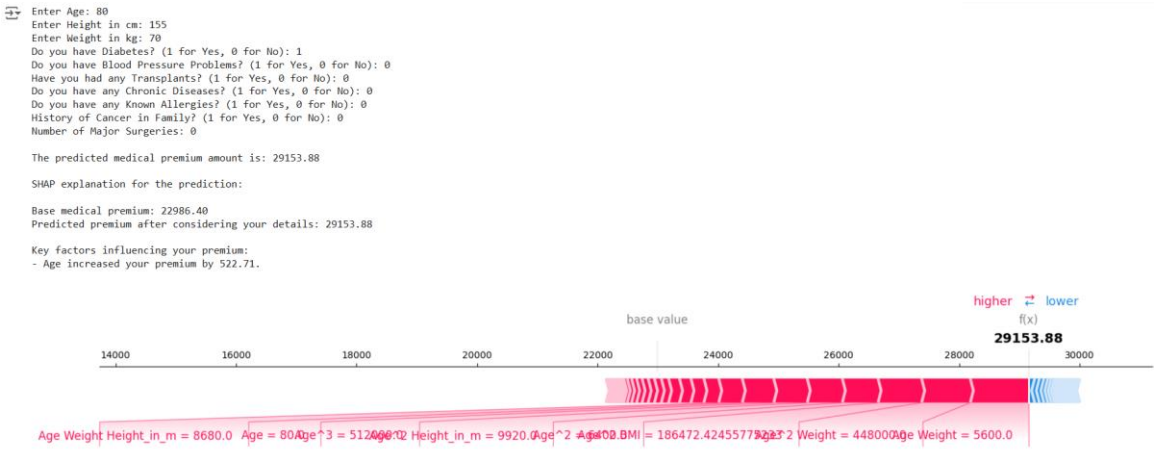

Figure 5: SHAP Plot

5.2.5 WEB INTERFACE MODULE

This module involves creating the Django-based web application, which allows users to input their personal and medical information and receive real-time premium predictions.

- User Input: Users enter details such as Age, Weight, Height, Diabetes, etc.
- Prediction Output: The system displays the predicted premium along with SHAP interpretations, highlighting the key factors influencing the prediction.

CHAPTER 6

RESULT AND DISCUSSION

6.1 RESULTS

The results section summarizes the outcomes of the healthcare insurance premium prediction system and discusses its performance metrics.

- Model Performance: The stacking model (CatBoost, XGBoost, and RandomForest with Ridge as a meta-learner) achieved the following performance metrics on the test dataset:
- Mean Absolute Error (MAE): 747.39
- Root Mean Squared Error (RMSE): 1520.38
- R-squared (R^2): 0.9044

These values indicate a high level of accuracy in predicting healthcare insurance premium amounts, with a low average error in the predictions and a strong fit between the predicted and actual values.

SHAP Interpretation: The SHAP values provided insight into which features influenced the premium predictions the most. Key influential features across most predictions were:

- Age: Older age groups were more likely to have higher premiums.
- BMI (Body Mass Index): Higher BMI values increased the premium amount.
- History of Chronic Diseases: Individuals with chronic conditions were assigned higher premiums.
- Number of Major Surgeries: A higher number of surgeries significantly increased the premium prediction.
- The interpretation provided transparency to users, allowing them to understand why their premium was set at a specific level.

6.2 DISCUSSION

The discussion reflects on the model's performance, strengths, and areas for future improvement.

- Model Strengths:

The stacking approach improved the model’s accuracy by combining the strengths of multiple models. CatBoost handled categorical features well, XGBoost optimized generalization, and RandomForest added robustness to the predictions.

SHAP provided model interpretability, which is critical for user trust in predictions. Users could see exactly which features contributed the most to their premium calculation, making the system more transparent and explainable.

- Challenges:

Data Quality: The accuracy of the predictions is highly dependent on the quality of the data provided. Inconsistent or incomplete data can negatively impact the model’s performance.

Feature Complexity: Handling a mix of categorical and numerical features was challenging, especially when deciding the number of polynomial features to add for numeric columns.

- Future Improvements:

Model Generalization: Although the model performed well on the test set, its ability to generalize to unseen data could be further improved by increasing the diversity of the training dataset.

User Experience: Enhancing the web interface to provide more detailed explanations or offer suggestions on how users can lower their premiums based on their inputs could increase the system’s utility.

Overall, the system demonstrated a high level of accuracy and transparency in predicting healthcare insurance premiums, with SHAP providing valuable insights into the key factors influencing the predictions.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In this project, we developed a healthcare insurance premium prediction system using a stacking ensemble model that combines the strengths of CatBoost, XGBoost, and RandomForest, with Ridge regression as the meta-learner. The system demonstrated high predictive accuracy, with metrics such as a Mean Absolute Error (MAE) of 747.39 and an R-squared value of 0.9044, indicating a strong fit between the predicted and actual premium values.

A key feature of the system is the integration of SHAP (SHapley Additive exPlanations), which allows for model interpretability by identifying the features that most significantly impacted each prediction. This transparency is essential for user trust, as it provides a clear understanding of how factors like age, BMI, chronic diseases, and the number of surgeries contribute to premium calculations.

The system has practical applications for insurance companies and customers alike. Insurers can leverage the model to predict premiums with greater accuracy, while customers gain insight into the factors influencing their premiums, leading to greater transparency in the insurance process.

7.2 FUTURE ENHANCEMENT

While the current system is effective, there are several avenues for future improvement and enhancement:

- Expanding the Dataset: The system could benefit from training on a larger, more diverse dataset that covers various regions and healthcare systems. This would improve the model's ability to generalize and ensure it works effectively across different populations.
- Real-time Data Integration: Integrating real-time data sources, such as wearables or health monitoring apps, could allow for dynamic premium adjustments based on the user’s lifestyle changes and current health status.

- Improved User Interface: The current user interface could be enhanced by offering users suggestions on how to lower their premiums based on their health data. This could include recommendations on weight management, regular check-ups, or managing chronic conditions.
- Feature Engineering: Future versions of the system could explore additional feature engineering techniques, including advanced transformations of numerical features or more sophisticated encoding of categorical variables, to further enhance model performance.
- AI-driven Fraud Detection: Enhancing the fraud detection module with advanced AI techniques such as deep learning could improve the system's ability to identify fraudulent claims submitted for premium predictions.
- Mobile Application: Developing a mobile application version of the system could increase accessibility and convenience for users who want to estimate their insurance premiums on the go.

In summary, while the system already provides accurate and interpretable premium predictions, future enhancements could further improve its effectiveness, usability, and scalability in the healthcare insurance domain.

APPENDIX

A1.1 SAMPLE CODE

1. PREPROCESSING MODULE

- views.py:

```
import os
import joblib
import pandas as pd
import logging
from django.conf import settings
from django.shortcuts import render
from .forms import PremiumPredictionForm
import traceback
logging.basicConfig(level=logging.INFO)
model_path = os.path.join(settings.BASE_DIR, 'predictor', 'saved_models',
'premium_prediction_model.pkl')
model = joblib.load(model_path)
def predict(request):
    if request.method == 'POST':
        form = PremiumPredictionForm(request.POST)
        if form.is_valid():
            input_data = form.cleaned_data
            height_cm = float(input_data['Height'])
            height_in_m = height_cm / 100
            bmi = float(input_data['Weight']) / (height_in_m ** 2)
            features = [
                int(input_data['Age']),
                int(input_data['Diabetes']),
                int(input_data['BloodPressureProblems']),
                int(input_data['AnyTransplants']),
                int(input_data['AnyChronicDiseases']),
                height_cm,
```

```
                float(input_data['Weight']),
                height_in_m,
                bmi,
                int(input_data['KnownAllergies']),
                int(input_data['HistoryOfCancerInFamily']),
                int(input_data['NumberOfMajorSurgeries'])
            ]
            try:
                prediction = model.predict([features])
                prediction_value = prediction[0]
                return render(request, 'predictor/result.html', {'prediction':
prediction_value})
            except Exception as e:
                logging.error(f"Error during prediction: {e}")
                logging.error(traceback.format_exc())
                return render(request, 'predictor/predict.html', {'form': form,
'error': str(e)})
        else:
            form = PremiumPredictionForm()
        return render(request, 'predictor/predict.html', {'form': form})
```

2. PREDICTIVE MODEL TRAINING MODULE

- models.py:

```
from django.db import models
class UserData(models.Model):
    age = models.IntegerField()
    diabetes = models.IntegerField(choices=[(0, "No"), (1, "Yes")],
default=0)  # No = 0, Yes = 1
    blood_pressure_problems = models.IntegerField(choices=[(0, "No"), (1,
"Yes")], default=0)
```

```
    any_transplants = models.IntegerField(choices=[(0, "No"), (1, "Yes")],
default=0)
    any_chronic_diseases = models.IntegerField(choices=[(0, "No"), (1,
"Yes")], default=0)
    height = models.FloatField()
    weight = models.FloatField()
    height_in_m = models.FloatField()
    bmi = models.FloatField()
    known_allergies = models.IntegerField(choices=[(0, "No"), (1, "Yes")],
default=0)
    history_of_cancer_in_family = models.IntegerField(choices=[(0, "No"),
(1, "Yes")], default=0)
    number_of_major_surgeries = models.IntegerField()
    def _str_(self):
        return f"UserData(age={self.age}, diabetes={self.diabetes},
blood_pressure_problems={self.blood_pressure_problems},
weight={self.weight})"
```

3. PREDICTION AND EVALUATION MODULE

- Stacking approach:

```
from sklearn.ensemble import StackingRegressor
from sklearn.linear_model import Ridge
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
base_models = [ ('catboost', CatBoostRegressor(loss_function='RMSE',
cat_features=categorical_features, verbose=0)),
('xgboost', XGBRegressor(learning_rate=0.1, n_estimators=100)),
 ('random_forest', RandomForestRegressor(n_estimators=100))]
meta_model = Ridge()
```

```
stacking_model = StackingRegressor(estimators=base_models,
final_estimator=meta_model, cv=5)
stacking_model.fit(X_train_final, y_train)
y_pred_stack = stacking_model.predict(X_test_final)
mse_stack = mean_squared_error(y_test, y_pred_stack)
rmse_stack = np.sqrt(mse_stack)
print(f"Stacking MSE: {mse_stack}, RMSE: {rmse_stack}")
mae = mean_absolute_error(y_test, y_pred_stack)
r2 = r2_score(y_test, y_pred_stack)
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R²): {r2}")
```

4. SHAP INTERPRETATION MODULE

- Shap:

```
import shap
import numpy as np
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
import joblib
model = joblib.load('premium_prediction_model_with_shap.pkl')
def generate_shap_interpretation(shap_values, feature_names, prediction,
base_value):
    explanation = ""
    shap_values_array = shap_values.values  # Extract actual values from the
Explanation object
    shap_impact = np.abs(shap_values_array[0]).mean(axis=0)
    sorted_indices = np.argsort(shap_impact)[::-1
    explanation += f"\nBase medical premium: {base_value:.2f}\n"
    explanation += f"Predicted premium after considering your details:
{prediction:.2f}\n\n"
    explanation += "Key factors influencing your premium:\n"
```

```
    for idx in sorted_indices[:5]:  # Top 5 features
        feature_name = feature_names[idx]
        impact_value = shap_values_array[0, idx]
        if impact_value > 0:
            explanation += f"- {feature_name} increased your premium by
{impact_value:.2f}.\n"
        else:
            explanation += f"- {feature_name} decreased your premium by {-
impact_value:.2f}.\n"
    return explanation
```

5. WEB INTERFACE MODULE

- predict.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Predict Premium</title>

    {% load static %}
    <link rel="stylesheet" href="{% static 'css/style.css' %}">
</head>
<body>
    <div class="container">
        <h1>Enter Your Health Information</h1>
        <form method="POST" action="{% url 'predict' %}">
            {% csrf_token %}
            {{ form.as_p }}

            {% if error %}
```

```
            <div class="error">{{ error }}</div>
        {% endif %}

            <button type="submit" class="btn">Predict</button>
        </form>
        <a href="{% url 'home' %}" class="btn">Back to Home</a>
    </div>
</body>
</html>
```

A1.2 SCREENSHOTS

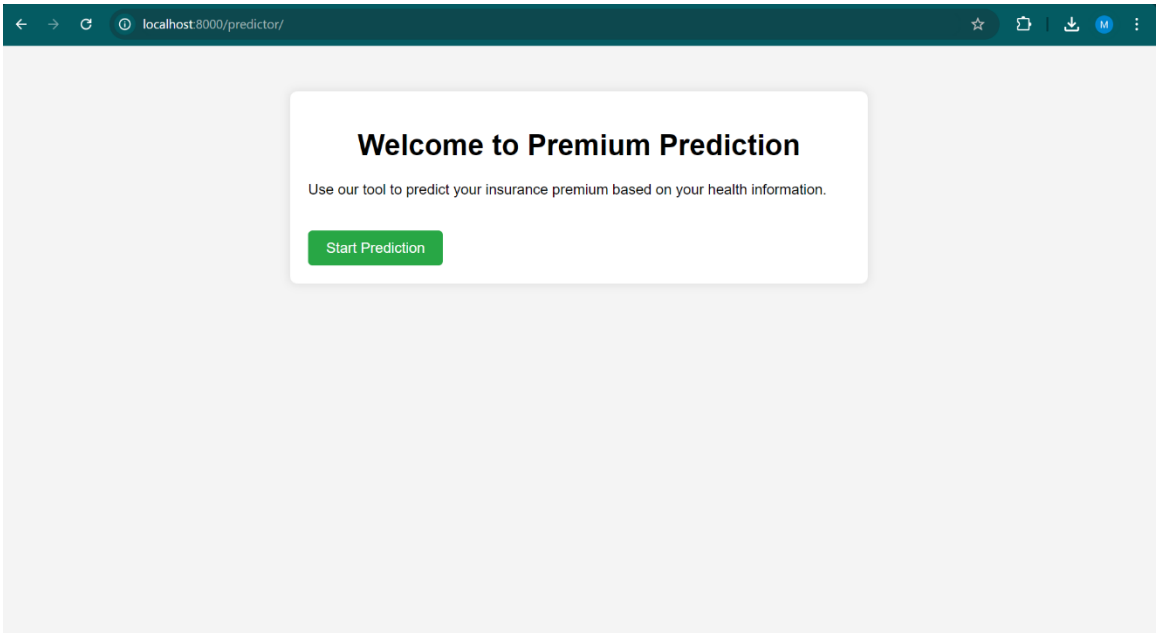

Figure 6: Welcome Page

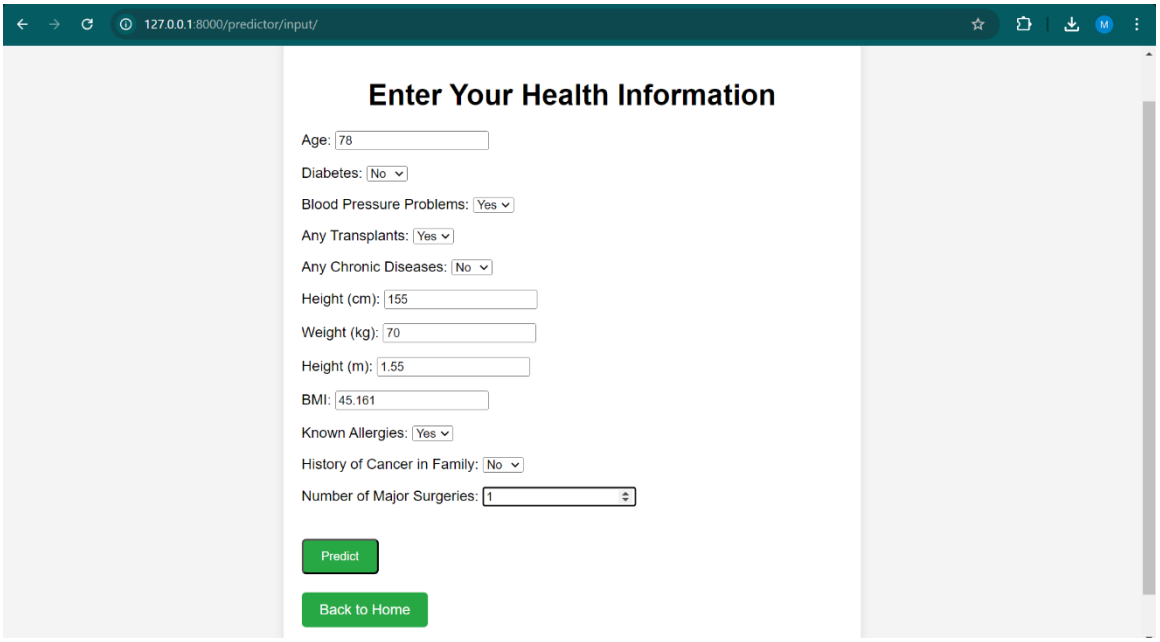

Figure 7: Health Information Form

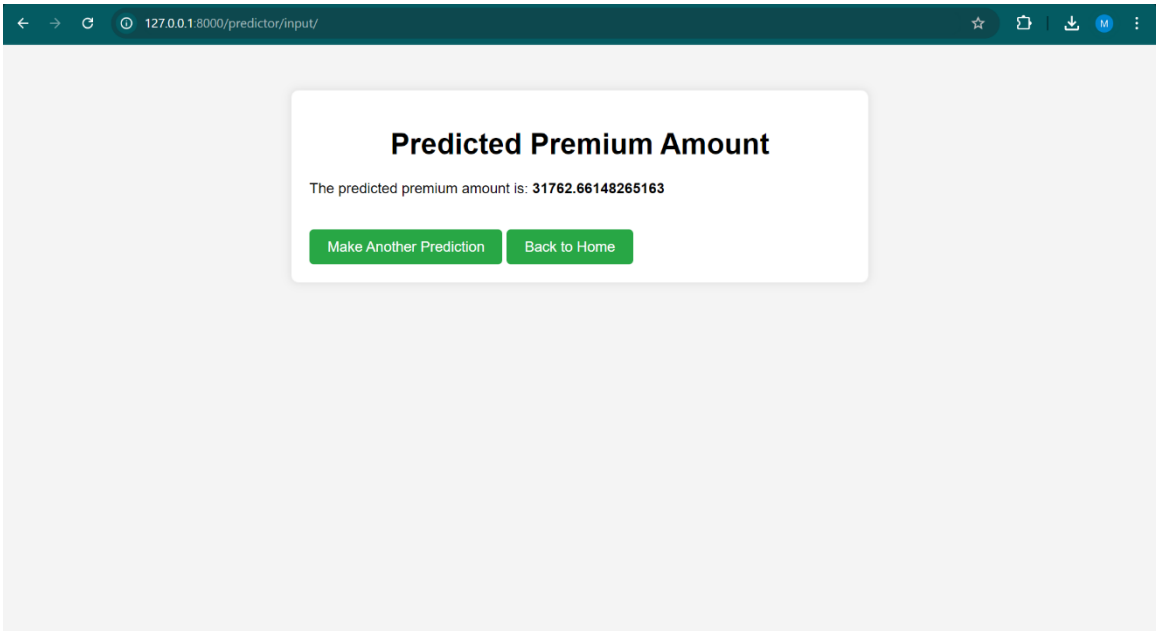

Figure 8: Premium Amount Prediction Page

	Feature	Importance
8	Cholesterol Levels	19.179511
2	BMI	12.815353
0	Age	9.666807
6	Pre-existing Conditions	9.540930
5	Insurance Plan Type	9.088518
12	Alcohol Consumption	9.073223
10	Dietary Habits	7.381803
9	Exercise Frequency	7.251366
7	Blood Pressure (mmHg)	5.001564
4	Number of Children	4.353818
11	Family Medical History	1.883353
3	Smoker	1.845562
1	Gender	1.075927
14	Residence	1.014165
13	Occupation	0.828102

Figure 9: Feature Importance

```
[4] data.duplicated().sum()
0

[5] data = data.drop(columns=['Blood Pressure'])

[7] data = data.drop(columns=['Residence'])

[6] data = data.drop(columns=['Occupation'])

data.head()
```

ıer of en	Insurance Plan Type	Predicted Premium	Pre-existing Conditions	Blood Pressure (mmHg)	Cholesterol Levels	Exercise Frequency	Dietary Habits	Family Medical History	Alcohol Consumption
1	Basic	482.279824	Heart Disease	143/87	5.9 LDL/HDL	Weekly	High-fat	Yes	Moderate
0	Premium	264.971796	Asthma	120/65	4.5 LDL/HDL	Daily	Balanced	Yes	Low
0	Comprehensive	428.248902	Hypertension	110/78	5.7 LDL/HDL	Rarely	High-sugar	Yes	Moderate
1	Comprehensive	317.307137	Diabetes	105/99	2.9 LDL/HDL	Rarely	High-fat	No	Unknown

Figure 10: Head of dataset

```
Original column names:
['Age', 'Diabetes', 'BloodPressureProblems', 'AnyTransplants', 'AnyChronicDiseases', 'Height', 'Weight', 'Height_in_m', 'BMI', 'KnownAllergies',
Mean Absolute Error (MAE): 971.3924776729407
Mean Squared Error (MSE): 2829541.3451626897
Root Mean Squared Error (RMSE): 1682.124057601784
R-squared (R²): 0.882942848610882
```

Figure 11: Evaluation Metrics

REFERENCES

[1] J. H. Friedman, Stochastic gradient boosting, Comput. Statist. Data Anal., vol. 38, no. 4, pp. 367378, Feb. 2002.

[2] R.RoyandK.T.George,Detecting insurance claims fraud using machine learning techniques, in Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT), Kollam, India, Apr. 2017, pp. 16.

[3] J. West and M. Bhattacharya, Intelligent nancial fraud detection: A comprehensive review, Comput. Secur., vol. 57, pp. 4766, Mar. 2016.

[4] G. Kowshalya and M. Nandhini, Predicting fraudulent claims in auto mobile insurance, in Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT), Coimbatore, India, Apr. 2018, pp. 13381343.

[5] Linghong Zou and Luling Zhou, "The Discussion about preservation and increasing the value of Social health insurance fund," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Dengleng, 2011,

[6] K. Culver, "Blockchain Technologies: A whitepaper discussing how the claims process can be improved," 2016, unpublished.

[7] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure Attribute-Based Signature Scheme With Multiple Authorities for Blockchain in Electronic Health Records Systems," IEEE Access, vol. 6, 2018, pp. 11676-11686.

[8] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," presented at the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1-5.

[9] A. Azaria, A. Ekblaw, T. Vieira, and T. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," presented at the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, 2016, pp. 25-30.

[10] C. Stagnaro, "White Paper: Innovative Blockchain Uses in Health Care," 2016, unpublished.