INSURANCE PREMIUM PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

MAHIMA.A (963319104035)
POOJA.J (963319104044)

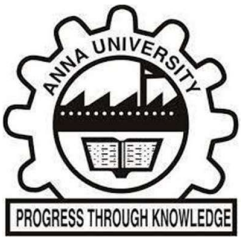
in partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY, PALKULAM

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"INSURANCE PREMIUM PREDICTION USING MACHINE LEARNING"** is the bonafide work of **"MAHIMA.A (963319104035), POOJA.J (963319104044)"** who carried out the project work under my supervision.

SIGNATURE
Mrs.R.Sahila Devi,M.Tech,
HEAD OF THE DEPARTMENT
Assistant Professor,
Department of CSE,
Rohini College of Engineering& Technology, Palkulam-629401

SIGNATURE
Mr.I.Sivaprasad Manivannan,M.E,MBA
SUPERVISOR
Assistant Professor,
Department of CSE,
Rohini College of Engineering& Technology, Palkulam-629401

Submitted for the project Viva Voce held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We acknowledge with great gratitude to all those who helped us to make this project a great success. At the very outset, we express our thanks to the almighty **GOD** who blessed with a healthy situation and has best owed upon us the required skill to pursue this technical course.

We would like to express sincere gratitude and thanks to our Honorable Chairman **Shri.K.NEELA MARTHANDAN,**for providing adequate facilities.

We express our deepest gratitude to our Honorable Pro chairman **Dr.N. NEELA VISHNU, MBA.,** for his invaluable guidance and motivation.

We would like to express sincere gratitude and thanks to our Beloved Managing Director **Dr.V. M. BLESSY GEO, Msc., Ph.D.,** Her kind words ofencouragement and support made our project reality.

We wish to express our profound gratitude to our principal **Dr.R. RAJESH, Ph.D.,** for the kind advice, valuable guidance and whose suggestions encouragement.

We express our sincere thanks to our Head of the department **Mrs. R. SAHILA DEVI, M.Tech.**, Assistant Professor, Department of Computer Science for her motivation, inspiration and encouragement to undertake this project.

We express our sincere gratitude and thanks to our Project Coordinator **Mrs. R. SAHILA DEVI, M.Tech.**, Assistant Professor, Department of Computer Science who gave constant inspiration throughout the project work.

We are thankful to our guide **Mr.I.SIVAPRASAD MANIVANNAN, M.E.,** Assistant Professor, Department of Computer Science for his suggestions and also for extending good guidance to complete the project successfully.

We are extremely thankful to our parents and friends who were the backbone of our success in all aspects of the project work.

ABSTRACT

Health insurance is a necessity nowadays, and almost every individual is linked with a government or private health insurance company. Factors determining the amount of insurance vary from company to company. The premium amount is determined by a variety of factors, including the type of insurance, the level of coverage, and the individual's risk profile. Health Insurance is a type of insurance that covers medical expenses. A person who has taken a health insurance policy gets health insurance cover by paying a particular premium amount. There are a lot of factors that determine the premium of health insurance. The purpose of this project is to investigate different features to observe their relationship, and plot a multiple linear regression based on several features of individual such as age, physical/family condition and location against their existing medical expense to be used for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium. Ultimately, the goal of insurance premiums is to enable insurers to accurately assess and manage risk, while providing policyholders with the coverage they need to protect themselves and their assets. By understanding the factors that go into premium calculations, individuals and businesses can make informed decisions about the type and level of coverage they need, while also ensuring they are getting the best value for their money.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BMI	Body Mass index
SVM	Support Vector Machine
ML	Machine Learning
CART	Classification and Regression Tree algorithm
XAI	Explainable Artificial Intelligence
GDP	Gross Domestic Product
MSE	Mean Square Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

1.1.1 INSURANCE PREMIUM

Insurance is a crucial aspect of modern life, providing individuals and businesses with protection against financial risks. One of the most significant components of an insurance policy is the premium, which is the amount of money paid by the policyholder to the insurer in exchange for coverage. The premium amount is determined by a variety of factors, including the type of insurance, the level of coverage, and the individual's risk profile. Health Insurance is a type of insurance that covers medical expenses. A person who has taken a health insurance policy gets health insurance cover by paying a particular premium amount. There are a lot of factors that determine the premium of health insurance. The amount of the premium for a health insurance policy depends on person to person, as many factors affect the amount of the premium for a health insurance policy. Let’s say age, a young person is very less likely to have major health problems compared to an older person. Thus, treating an older person will be expensive compared to a young one. That is why an older person is required to pay a high premium compared to a young person. Just like age, many other factors affect the premium for a health insurance policy.

Health insurance is one of the most critical purchases a person makes every year. One-third of GDP is spent on health insurance, and everyone requires some degree of health care. The healthcare rates constantly fluctuate every year due to different variables such as medical changes, pharmaceutical trends, and political considerations. Different types of insurance have different premium structures. For

example, health insurance premiums are typically paid monthly or annually, while car insurance premiums are usually paid on a six-month or annual basis. The premium amount for each type of insurance is based on a variety of factors, including the policyholder's age, gender, region, BMI, children, smoker and some of their past medical and surgical history.

In addition to these factors, insurers also consider the level of coverage requested by the policyholder when calculating the premium. Higher levels of coverage typically result in higher premiums, as the insurer is taking on more risk in the event of a claim. The suggested work's goal is to anticipate a person's insurance costs and to identify patients with health insurance policies and medical information, regardless of whether they have any health problems. Several sorts of health insurance must be anticipated for a patient. It is possible to estimate an individual's health insurance costs based on the level of emergency department treatment they receive depending on the type of health insurance they possess. To determine health insurance premium quotes, there are several factors that have to be taken into consideration when defining a premium, such as pre-existing diseases, age, gender, family medical history, lifestyle. Based on this evaluation, the health insurance premium quote via the provided factors to predict the range of the health insurance premium quote for each group of people.

Ultimately, the goal of insurance premiums is to enable insurers to accurately assess and manage risk, while providing policyholders with the coverage they need to protect themselves and their assets. By understanding the factors that go into premium calculations, individuals and businesses can make informed decisions about the type and level of coverage they need, while also ensuring they are getting the best value for their money.

1.1.2 MACHINE LEARNING

Machine learning is a subset of artificial intelligence that involves building algorithms and models that allow computer systems to learn and make predictions based on data without being explicitly programmed. Machine learning algorithms can identify patterns and relationships in data and use this information to make decisions and predictions. Machine learning algorithms can be broadly categorized into three types: supervised learning, unsupervised learning, and reinforcement learning. Machine learning algorithms can be applied to a wide range of applications, including image and speech recognition, natural language processing, predictive maintenance, fraud detection, and personalized recommendations. One of the key benefits of machine learning is its ability to automate decision-making processes. By using machine learning algorithms to analyze data and make predictions, organizations can save time and reduce the risk of human error. Machine learning can also improve the accuracy and efficiency of decision-making processes. Machine learning algorithms require large amounts of data to be effective. The quality and quantity of the data used to train machine learning models have a significant impact on the accuracy and effectiveness of the resulting predictions. It is therefore important for organizations to ensure that they have high-quality, relevant data that is representative of the problem they are trying to solve. Machine learning is a rapidly growing field, and there are many different tools and frameworks available for building and deploying machine learning models. As machine learning continues to evolve, there are many exciting developments and applications emerging. For example, researchers are exploring the use of machine learning in healthcare to improve diagnosis and treatment plans. In addition, there are concerns around the ethical implications of using machine learning to automate decision-making processes. It is therefore important for organizations to consider

these risks and challenges when developing and deploying machine learning models. In conclusion, machine learning is a powerful technology that has the potential to transform many industries and applications. By using machine learning algorithms to analyze data and make predictions, organizations can improve decision-making processes and achieve better outcomes. However, it is important for organizations to carefully consider the potential risks and challenges associated with machine learning and take steps to mitigate these risks.

1.2 PROJECT OBJECTIVES

In this paper, we will use the Python programming language for the implementation and trained the machine learning-based model for the prediction of health insurance premiums. Initially, the dataset and the necessary python libraries and packages were imported. The dataset consisted of over 1300 entries and seven columns, namely charges, smoking, region, children, BMI, sex, age and some of their past medical and surgical history.

This dataset was used to predict the health insurance premium. Thereafter, an exploratory data analysis was performed. the dataset was checked for null values. Since there were no null values in the dataset, the statistical summary of the dataset was analysed. The statistical summary included the count, mean, standard deviation, and various other statistics related to the columns available in the dataset—age, BMI, number of children, and health insurance charges and some of their past medical and surgical history.By predicting the approximate cost, we can avoid overspending in taking policies and since the AI model is generating the output, we can avoid having bias towards customers. This will raise customers trust and increase their satisfaction towards the insurance provider.

CHAPTER 2
LITERATURE SURVEY

2.1 Health Insurance Premium Prediction using XGboost Regressor.(2022)

C. Jyothsna, K.Srinivas, B. Bhargavi, A. E. Sravanth, A. T. Kumar and J. N. V. R. S. Kumar

The suggested work's goal is to anticipate a person's insurance costs and to identify patients with health insurance policies and medical information, regardless of whether they have any health problems. Several sorts of health insurance must be anticipated for a patient. It is possible to estimate an individual's health insurance costs based on the level of emergency department treatment they receive depending on the type of health insurance they possess. Multi-Linear, Decision Tree, Random Forest, and Gradient Boosting Regression were some of the regression models employed in this study. After comparing the accuracies, it was determined that Gradient Boosting was the most accurate of all the methods, with an accuracy of 87 percent. Finally, using the best model, the Telegram- integrated chatbot is trained with instructions to communicate with the user and estimates the insurance premium.

2.2 Applying a Genetic Algorithm to Determine Premium Rate of Occupational Accident Insurance.(2021)

J. J. -C. Ying, C. -K. Chang and Y. -T. Chang

At present, the Occupational Accident Labor Insurance premium rate is calculated based on the business categories in Taiwan. The premium rate is calculated as a combination of the experience rate and the manual rate for each business

category. The traditional actuarial methods are based on many hypotheses to calculate future actual claims and adjust the rate for each business category. Unfortunately, with such adjustments, the risk level of the insured in the business category will be affected. To accurately estimate the size of actual losses for specific industries, we propose a genetic algorithm applied grouping to determine the premium rate for occupational accidents. The proposed approach has been evaluated using the real-world dataset from the Bureau of Labor Insurance in Taiwan that includes occupational accident insurance data from 2009 to 2015. The results demonstrate that the method is practicable at predicting the applicable premium. The proposed method differs from Taiwan's prevailing occupational accident premium rate calculation method. Moreover, it is efficient at selecting the best group of the Standard Industrial Classification from the genetic algorithm. Lastly, the accuracy of the estimates of the total claim amounts are analyzed.

2.3 Health Insurance Cost Prediction using Machine Learning Algorithms (2022)

R. D, M. S. K and D. J

Insurance is the policy people buy to preserve themselves from losing money if something awful happens to them or their belongings. Insurance can be provided to the individuals for their health, car, education, business, home etc. Among them, health insurance plays the vital role as humans are prone to many deadly diseases and accidents nowadays. The cost of medical treatment is unbelievable and unbearable. In this study, health insurance cost is predicted by Machine Learning algorithms such as Multinomial Logistic Regression and Random Forest. Comparing the performance measures of the algorithms, gives the better way to understand and gives the determination about the best algorithm to be used for cost prediction.

2.4 Hybrid Regression Model for Medical Insurance Cost Prediction and Recommendation.(2022)

N.V.Sailaja, M.Karakavalasa, M. Katkam, D. M, S. M and D. N. Vasundhara

As the global value of gross insurance premiums continues to rise beyond $5 trillion, we know that most of these costs are avoidable. The cost prediction of people's medical insurance is a useful method for improving the transparency of health care. We use different regression models to analyse personal health data to predict insurance amounts for individuals in this study. The cost of insurance premiums is influenced by a few factors. Health insurers would benefit from using a Stacking Regression model to predict insurance costs for individuals. According to this perception using recommendation graphs, a person can lower his insurance costs by lowering his BMI, moving to a different area, or switching to a non-smoker.

2.5 Health Insurance Amount Prediction Using Supervised Learning (2022)

S. Aggarwal and Anmol

Medical Insurance cost prediction is prime distress. A Medical Insurance company can only make money if it collects quite it spends on the medical aid of its beneficiaries. Medical Insurance companies are a troublesome task as determining premiums for his or her customers. Mechanism Knowledge stands a part of Reproduction Intellect and computing which spotlight the consumption of data and controls to imitate the method that persons absorb, increasingly employed on the situation exactness. Prediction means affecting the product of estimation afterwards the situation can be situated on a documented dataset, in addition, original data though computing the likelihood of a particular outcome like whether a customer will

mix in thirty days. Comparative analysis of Machine Learning Algorithms. Compare new techniques with existing techniques using various outputs. We will use the dataset for training the model. Which regression gives the best accuracy and who will take less time.

2.6 Health Insurance Cost Prediction Using Regression Models.(2022)
S. Panda, B. Purkayastha, D. Das, M. Chakraborty and S. K. Biswas

India's government spends 1.5 percent of its annual GDP on public healthcare, which is significantly less than that of other countries. Global public health spending, on the other hand, has almost doubled in line with inflation in the last two decades, reaching US $\{\$\}8.5$ trillion in 2019, or 9.8% of global GDP. Multinational multi-private sectors provide around 60% of comprehensive medical treatments and 70% of out-patient care, which charge patients astronomically high fees. Because of the rising expense of quality healthcare, increased life expectancy, and the epidemiological shift toward non-communicable diseases, health insurance is becoming an essential commodity for everyone. Insurance data has increased dramatically in the last decade, and carriers now have access to it. The health insurance system explores predictive modelling to boost its business operations and services. Computer algorithms and Machine Learning (ML) is used to study and analyse the past insurance data and predict new output values based on trends in customer behaviour, insurance policies, and data-driven business decisions, and support in formulating new schemes. Additionally, ML has found enormous and potential applications in the insurance industry. Thus, this paper develops a real-time insurance cost price prediction system named ML Health Insurance Prediction System (MLHIPS) using ML algorithms which will aid the insurance companies in the market for easy and rapid determination of values of premiums and thereby curb

down health expenditure. The proposed model incorporates and demonstrates different models of regression such as Ridge Regression, Lasso Regression, Simple Linear Regression, Multiple Linear Regression and Polynomial Regression to anticipate insurance costs and assess model outcomes. In the proposed model, the Polynomial Regression model has achieved better results with an RMSE value of 5100.53 and R-squared value of 0.80 compared to all the other models.

2.7 Clustering Application for Data-Driven Prediction of Health Insurance Premiums for People of Different Ages.(2021)

T. Omar, M. Zohdy and J. Rrushi.

A health insurance premium is a monthly fee that is paid in a health plan to typically pay for medical, surgical, prescription drug and sometimes dental expenses incurred by the insured. Since 2010, the affordable care act has prohibited insurance companies from denying coverage to patients with pre-existing conditions and has allowed children to remain on their parents; insurance plans until they reached the age of 26. Creating the policy is an important and challenging task. To determine health insurance premium quotes, there are several factors that must be taken into consideration when defining a premium, such as pre-existing diseases, age, gender, family medical history, lifestyle, etc. In this paper, a combination of the K-means algorithm and the Elbow method is developed to accurately group people in an optimal number of clusters based on similarity. Based on this evaluation, the health insurance premium quote via the provided factors to predict the range of the health insurance premium quote for each group of people.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Health insurance premium prediction is a vital task in the insurance industry, which involves estimating the expected cost of providing health insurance to individuals or groups. The existing system for health insurance premium prediction typically involves using actuarial methods and statistical models. Actuarial methods analyse the historical data of healthcare utilization, medical claims, and demographic characteristics of the insured individuals. The actuarial techniques are based on statistical models that estimate the likelihood of future healthcare costs based on these historical data. The models may include factors such as age, gender, medical history, family history, and lifestyle habits to predict the likelihood of future health risks and associated costs. In recent years, machine learning algorithms have also been used to predict health insurance premiums. These algorithms use more complex models that can analyse vast amounts of data and identify patterns that traditional actuarial methods may miss. Machine learning models may consider additional variables such as socioeconomic status, geospatial data, and environmental factors to improve the accuracy of premium predictions. Overall, the existing system for health insurance premium prediction involves a combination of actuarial methods, statistical models, and machine learning algorithms. The goal is to estimate the expected costs of providing healthcare to individuals or groups accurately. This allows insurance companies to price their policies accurately and minimize their financial risk while providing affordable health coverage to their customers.

3.1.1 DISADVANTAGES

- Limited data availability: The accuracy of health insurance premium predictions depends on the availability and quality of historical data. In some cases, there may be limited data available, which can affect the accuracy of premium predictions.
- Bias in data: Historical data used for health insurance prediction may contain biases, such as racial or gender bias. These biases can affect the accuracy of premium predictions and lead to discriminatory pricing.
- Privacy concerns: The use of personal health data for health insurance prediction raises concerns about privacy and data protection. Insurers must ensure that they comply with all relevant data protection regulations and guidelines to protect their customers' privacy.

3.2 PROPOSED SYSTEM

In the proposed system we are predicting the insurance cost by using a machine learning algorithm. Insurance companies are increasingly using artificial intelligence and machine learning algorithms to predict insurance costs. These algorithms can analyse large amounts of data and identify patterns that may not be immediately apparent to humans. These models use statistical analysis to estimate the probability of a particular event occurring and the cost associated with that event. Actuaries use data from past events to create these models, which can then be used to predict future events.

We get the set of data from the customers that include factors such as their age, gender, BMI, smoker, region of living. Once you have the data, you need to pre-process it. This involves cleaning the data. Once you have pre-processed the data and

engineered the features, you need to select a machine learning model that is appropriate for the problem at hand.

In our project we are using Adaboost regression algorithm. After selecting the algorithm, the next step is to train the selected machine learning model using the pre-processed data and engineered features. Once the model is trained, you need to evaluate its performance. This involves testing the model on a validation dataset and comparing its predicted outputs with the actual outputs. After the model is tuned and its performance is satisfactory, it can be deployed in a production environment. In our project we are deploying the model in the cloud environment. Ensure that the machine learning model is optimized for deployment in the cloud. This may involve converting the model to a format that is supported by the deployment method, optimizing the model's performance, and reducing its size. Once the cloud environment is set up, we will deploy the machine learning model.

Insurance companies use statistical models and actuarial science to predict insurance costs. Actuarial science is a branch of mathematics that applies statistical methods to assess risk in the insurance industry. Insurance companies use historical data and other relevant information to estimate the probability of a particular event occurring and the potential cost associated with that event. To predict insurance costs, insurance companies analyse various factors, such as: Demographic information, Personal factors, Type of coverage, Risk factors and Claims history. Demographic information: Age, gender, location, occupation, and marital status are some of the factors that affect insurance rates.

Personal factors: Health history, driving record, credit score, and other personal factors may also influence the insurance rate. Type of coverage: The type of insurance coverage, such as liability, comprehensive, and collision coverage, can also affect the

insurance cost. Claims history: An individual's claims history can influence the insurance cost. If someone has a history of filing claims, they may be considered a higher risk, and their insurance premiums may be higher. Risk factors: Some insurance policies are based on the level of risk involved. For example, homeowners' insurance rates are affected by the likelihood of natural disasters occurring in a particular area.

Using these factors and other relevant information, insurance companies create mathematical models that predict the likelihood of certain events occurring and the potential cost associated with those events. These models help insurers set their rates and premiums, ensuring that they are pricing their policies correctly based on the level of risk involved.

3.2.1 ADVANTAGES

- Enhanced Risk Assessment:

  Predictive models can leverage historical data on health insurance claims to identify patterns and trends that can help to improve risk assessment. By analysing data on past claims, a predictive model can identify individuals or groups that are more likely to experience certain health issues or require certain treatments. This can help insurance providers to better tailor their policies to individual needs and reduce the risk of unexpected claims.
- Improved Cost Control:

  Health insurance providers face significant challenges in controlling costs, particularly as healthcare costs continue to rise. Predictive models can help to identify individuals who are at high risk of developing chronic conditions or requiring expensive treatments, allowing insurance providers to

take proactive steps to manage these costs. By identifying high-risk individuals early on, insurance providers can provide preventative care, manage chronic conditions more effectively, and reduce the overall cost of healthcare.

- Increased Customer Satisfaction:

  Predictive modelling can help insurance providers to offer more personalized policies and services to their customers. By using data to better understand customer needs and preferences, insurance providers can tailor their offerings to meet individual needs and provide a better customer experience. This can lead to higher customer satisfaction and loyalty.

3.3 HARDWARE SPECIFICATION

Processor	: Intel i5
Operating System	: MacOS/Windows/Linux
RAM	: 8GB
Hard Disk	: 200 GB

3.4 SOFTWARE SPECIFICATION

Language	: Python
IDE	: Jupyter Notebook
Analysing Software	: Excel
Visualization	: Tableau

3.5 SOFTWARE DESCRIPTION

3.5.1 PYTHON

Python is a high-level language. It is an open source and interpreted language. It is the popular language used by most of the data scientists for various projects. Python language contains various functions to deal with statistics and mathematics. It provides many libraries to deal with different data science projects. Because of its ease of use and easiest syntax make this language suitable for scientific and research related tasks. It is also easily adopted by people who do not have any technical background.

The python language containing a greater number of packages related to the machine learning, deep learning, computer vision and natural language processing, these libraries are rapidly upgrading its features makes this language to produce the outcome in efficient manner. When it comes to the problems related to NLP, and analysis developers opted for the python language because it provides a large collection of frameworks which helps to solve complex problems easily and can build the data science application.

Following are some useful features of Python language:

1. It uses simple syntax; hence the programs are easier to read and write.
2. It provides large library support.
3. Python's interactive mode makes it easier to test the codes.
4. Python can be embedded into any application to provide a programmable interface.
5. It provides large communication support.
6. Python is a platform independent language, which allows the developers to run the code anywhere such as Windows, MacOS, Linux and UNIX.

3.5.2 Jupyter Notebook

Jupyter Notebook is an interactive computational environment which is web based for making notebook documents. It contains a list of input/output cells which can be used for coding, marking, heading and text. Jupyter kernel is a program which is used for handling various kinds of requests and providing responses to the user. It contains many features and provides a way to download the working document into many formats such as pdf, html file and ipybn file.

3.5.3 Excel

Excel is a most popular tool for data analysis due to its features. They are Data manipulation such as sorting, filtering and pivot tables. Data visualization such as making a variety of charts and graphs to represent the data. Statistical analysis such as regression and hypothesis testing. Data cleaning includes the features such as removing duplicates and merging data from different sources.

Excel is suitable for only a small amount of dataset, when but for large amount of dataset the analysts can understand the structure of the dataset and make some small operations such as attribute's name or value changing based on some condition.

3.5.4 Tableau

Tableau is a popular data visualization software that allows users to make Interactive dashboards and visualization graphs such as bar chart, Gantt chart, line chart, pie chart and many more plots from a variety of data resources. It is mainly used in finance, business to show the complex data in an effective visualization manner. The tableau is widely used when the dataset is large and complex. It provides a user-friendly environment, and the wide range of features makes the environment easier to both the beginners and experienced analysts.

CHAPTER 4

SYSTEM DESIGN

Fig 4.1: System Architecture

4.1 ARCHITECTURE DIAGRAM

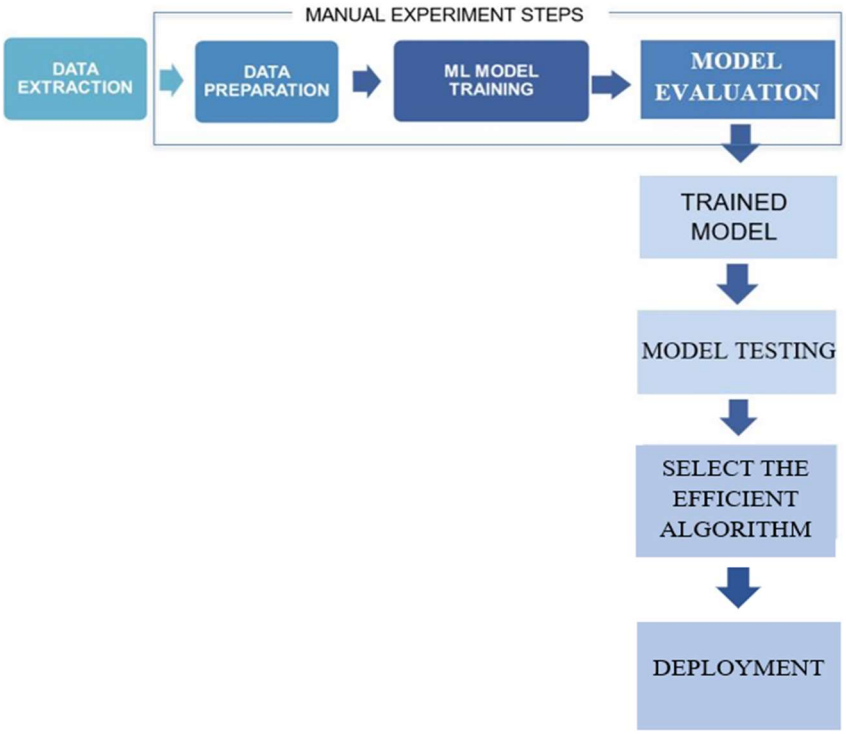


Fig 4.2 Machine Learning Architecture Diagram

An insurance premium cost prediction ML architecture would involve several key components, including data collection and pre-processing, feature engineering, model selection and training, and model evaluation and deployment. The data collection and pre-processing component would include gathering relevant data from various sources, cleaning and transforming the data to prepare it for use in training the machine learning model. This may involve data cleaning techniques like handling missing values, data normalization, and data scaling. The feature engineering component would involve selecting and extracting the most relevant features from the cleaned and transformed data, based on domain knowledge and statistical analysis. The model selection and training component would involve selecting an appropriate machine learning algorithm (e.g., regression, decision trees, neural networks) and training it on the pre-processed and engineered data. The model would be evaluated and fine-tuned using techniques like cross-validation to optimize its performance. Finally, the model evaluation and deployment component would involve evaluating the model's performance on a holdout dataset, deploying the model to a production environment, and monitoring its performance over time to ensure that it continues to provide accurate and reliable predictions.

4.2 WORKFLOW DIAGRAM

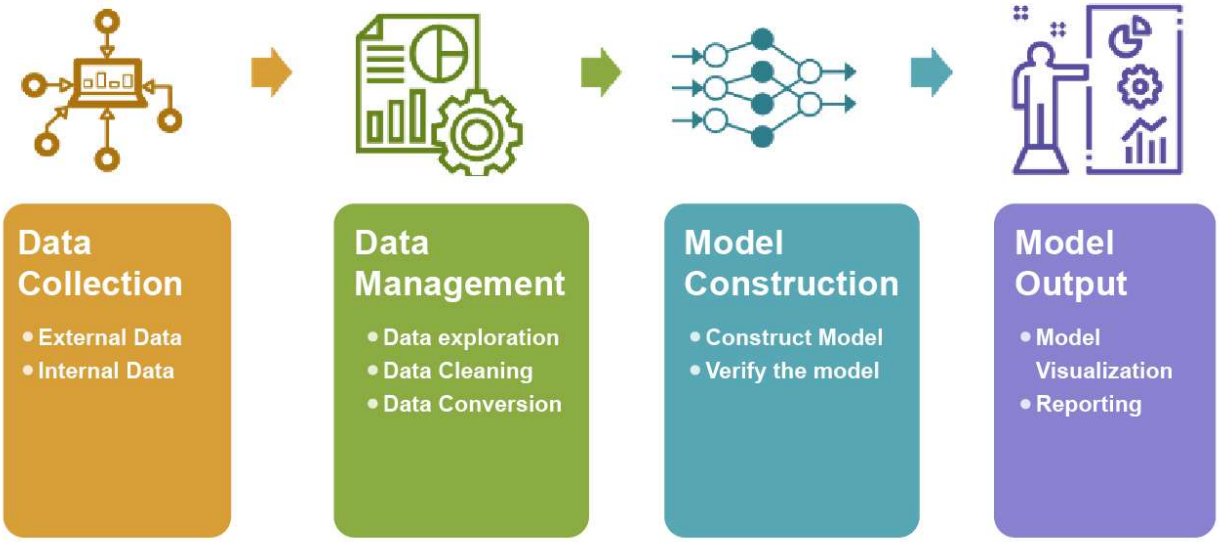


Fig 4.3: Workflow Diagram

Step 1: Performing the Data Analysis and Feature Engineering, the dataset was analysed to check the relationship between the various columns.

Step 2: Data Visualisation In the previous step, the dataset was cleaned so that the model could be trained and visualised. In this step, the data was visualised to obtain useful information.

Step 3: Training and Evaluating a model

Step 4: The accuracy of the model is reported.

- To get the accuracy we train the model using machine learning algorithms. After evaluation we select an algorithm which gives us good accuracy to test the model.
- In our project we have an interface page where the user needs to enter their personal details such as age, BMI value, whether they are a smoker or non-smoker and some other medical conditions they have now.
- They are asked to mention whether they have undergone any surgery before. The price of the policy differs from one customer to other customer based on the attributes they have entered.
- The final stage is to predict the amount by selecting the predict button. This will take you to the new page where the amount id displayed in rupees.

CHAPTER 5
SYSTEM IMPLEMENTATION

5.1 MODULES

- ➢ Data pre-processing
    1. Collecting Dataset
    2. Importing Libraries and Datasets
    3. Data Preparation
    4. Encoding Categorical Data
    5. Feature Scaling
    6. Normalization
- ➢ Model Evaluation
    1. Splitting dataset into training set and testing set
    2. Applying different Machine learning models
    3. Evaluating the classification model.

5.2 MODULES DESCRIPTION

- ➢ **DATA PREPROCESSING**

Pre-processing is an important step in machine learning, which can transform the raw data collection into a suitable format which can be easily understood and handled by machine algorithms. Pre-processing is a required task for cleaning the data and making it suitable knowledge for a machine learning model that also increases the efficiency and accuracy of a model. It involves following steps

1. Collecting Dataset

To create a machine learning model, the first thing we require is a dataset, where the machine or deep learning model completely works on it. To collect the information related to a particular problem in proper format is known as a dataset. The data may be in structured or unstructured format while we collect it. The dataset varies, because the medical dataset is entirely different from the business dataset. Each dataset is different from another dataset. To use this dataset in our code, we usually download the dataset in CSV file format. However, sometimes we may also need to use an HTML or xlsx file. Datasets can be used in various fields, such as business, science and medical, to analyse and identify the pattern from the data. The dataset can come in different forms, they are text related data, image data, audio data or numerical data. These datasets are typically analysed by using machine learning algorithms, statistical models, or other data analysis techniques to identify the patterns, which can be useful in the decision-making process. In machine learning, a dataset is a collection of records that can be used to provide training, testing and evaluating the model. Mainly the dataset contains dependent and independent attributes, where the independent attributes are the input features and dependent attributes are the target column. The quantity and quality of the dataset is very important for the success of a machine learning model. A well dataset must cover the records in all the variations of the related domain.

2. Importing Libraries and Datasets

The Insurance.csv datasets are downloaded from authorised sources and imported. In machine learning, libraries are collections of prewritten code or functions that can provide features which makes the developers and data scientist to work easier without having to write code from scratch for building machine learning models. Libraries contain algorithms and techniques for creating, training and

evaluating models. To perform data pre-processing using Python, we need to import predefined libraries. The five main libraries for pre-processing are Pandas, NumPy, Matplotlib, Seaborn and Sci-Kit Learn.

- Pandas

Pandas is a library for data handling and analysis, which is consistently used in pre-processing to process the dataset before they are given as the input to the machine learning models during the training time.

- NumPy

NumPy is a Python library for processing arrays. It can execute all the mathematical and statistical operations over the dataset.

- Matplotlib

Matplotlib is a popular data visualization library that provides a variety of tools for creating high quality plots, graphs and charts.

- Seaborn

Seaborn is another popular data visualization library. It provides a high-level interface for creating graphs, charts and plots.

- Sci-kit Learn

Sci-kit Learn is a popular library mainly used for machine learning problems. It contains several algorithms and techniques for handling, processing, training, testing and evaluating machine learning models.

3. Data Preparation

In general, data preparation is considered as the most difficult stage in machine learning, and includes data cleaning, data pre-processing, data wrangling, and feature engineering. Data preparation involves transforming raw data into a format where the machine learning algorithms can deal with, in order to uncover insights or make predictions. The Insurance.csv dataset was downloaded from their corresponding authorised website. The dataset contains 1339 records and 7 columns. The data preparation process may consist of several steps, however, the most significant one involves processing the missing or incomplete data in the Insurance dataset. Data cleaning includes identifying and correcting errors or mistakes in the Insurance dataset. Dropping columns that include missing or incomplete data, since missing and incomplete data affect the efficiency of the machine learning model. Therefore, it is important to process the missing and incomplete data in the dataset. For the Insurance dataset, we check which columns have null values, positive and negative infinity values. The infinity values are replaced with the null value, after all these null values are replaced by their own corresponding column's mean value. Then once again we checked if the dataset contains any null values and if it is, then we apply the drop null value function to drop the null value containing rows from the dataset. Then we write on a user-defined function named compress. The compress function takes the entire dataset as input, and it tries to reduce the size of the dataset. It works by converting int 64-bit values into int 32-bit values, int 32 bit into int 16 bit, int 16 bit into int 8 bit, int 8 bit into int 4 bit and int 4 bit into int 2 bit. After applying this function, a new dataset is generated.

4. Encoding Categorical Data

In machine learning, encoding is the process of converting the text data into numerical format which can be easily processed by machine learning algorithms. The input data to a machine learning algorithm must be in numerical format but in general the data we have collected is not fully numerical data so to convert those data into appropriate format we have to encode the data before giving it as an input to the model. The input data to a machine learning algorithm can come in various formats such as text, images, audio, or other complex data structures. However, machine learning algorithms typically require numerical inputs. Therefore, encoding techniques are used to convert this data into a numerical format. The popular encoding methods are Label encoding, one-hot encoding and word embeddings.

5. Feature scaling

Featuring scaling is the process of scaling numerical features on the dataset to ensure that they are on the same scale. This is one of the important features because having the attributes on the same scale required less timing for training when compared with the different scales.

6. Normalization

This involves transforming data into standard distribution form for processing the dataset easily and the normalization also improves the performance of the model.

MODEL EVALUATION

Now we need to apply the different Machine and Deep learning algorithms on the pre-processed data.

- **Splitting dataset into training set and testing set**

Splitting the main dataset into a training set and testing set is an important step in creating and evaluating machine and deep learning models. The primary reason for this is to assess how well the model can perform on the unseen data. The training dataset is extremely important, and it is used to train the machine and deep learning model, which involves training and minimizing the error rate between actual and predicted values in the training set. If we evaluate the model performance by using the same data that we have used for training, it may give highest accuracy because the model already studied the training data well. The main purpose of the testing dataset is to evaluate the developed model's performance on unseen data. By doing so, we can determine and confidently say the performance of the model. Therefore, splitting the main dataset into training and testing is very important to ensure that there is no overfitting. It is also used in hyperparameter tuning to evaluate the model with different parameters.

Training Set: It is a subset of dataset to train the machine and deep learning model, where output of each record is already known.

Testing set: It is a subset of dataset to test the machine and deep learning model, by using the test set, the model can predict the output.

- **Loss Function**

A loss function is a mathematical function that measures the difference between the actual and predicted values, here the actual values are obtained from Training Set Testing Set the dataset and the predicted values are the values which have been predicted by the model. Generally, the actual values are denoted by y and

the predicted values are denoted by ybar. The goal of the loss function is to adjust the model's parameters to minimize the loss function

Accuracy Score: Accuracy score is used to evaluate the performance of a classification model. It measures the percentage of correct predictions on the testing dataset by the machine learning model.

Accuracy score = (Number of correct predictions) /
(Total number of predictions) *100%

- **Mean Square Error**

Mean Square Error (MSE) is a popularly used loss function and evaluation metric for machine learning problems. It evaluates by taking the average squared difference between the actual values and the predicted values.

MSE = 1/n * ∑(y_pred - y_true)^2

Where,

n is the total number of instances in the dataset

y_pred is the predicted value of the target variable for an instance

y_true is the actual value of the target variable for the same instance

- **Root Mean Square Error**

Root Mean Square Error (RMSE) is a commonly used evaluation metric for machine learning problems. This is like the Mean Square Error (MSE). The RMSE is nothing, but this is the square root of the MSE, and it measures the average distance between the actual and predicted values.

RMSE = sqrt(1/n * ∑(y_pred - y_true)^2)

Where,

n is the total number of instances in the dataset

y_pred is the predicted value of the target variable for an instance

y_true is the actual value of the target variable for the same instance

- **R2 Score**

R2 score, also known as the coefficient of determination, is a statistical metric used to evaluate the performance of a regression model. It measures the proportion of variance in the dependent variable that is explained by the independent variables (i.e., the features used to make predictions). The R2 score ranges from 0 to 1, with higher values indicating a better fit between the model's predictions and the actual data. An R2 score of 1 indicates that the model is able to perfectly predict the dependent variable based on the independent variables, while a score of 0 indicates that the model is no better than simply predicting the mean value of the dependent variable.

R2 = SSR/TSS

Where,

TSS is the total sum of squares

SSR is the regression sum of squares

- **Mean Absolute Error (MAE)**

Mean Absolute Error (MAE) is a commonly used metric to measure the accuracy of a regression model. It measures the average absolute difference between the predicted values and the actual values. To calculate the MAE, you first take the absolute difference between each predicted value and its corresponding actual value, and then calculate the mean of these absolute differences. Mathematically, it can be expressed as:

MAE = (1/n) * Σ|yi - xi|

Where,

n is the number of data points in the dataset

yi is the predicted value for the i-th data point

xi is the actual value for the i-th data point

- **Mean Absolute Percentage Error**

Mean Absolute Percentage Error (MAPE) is a commonly used metric to measure the accuracy of a forecasting model. It measures the percentage difference between the predicted values and the actual values. To calculate the MAPE, you first take the absolute percentage difference between each predicted value and its corresponding actual value, and then calculate the mean of these absolute percentage differences. Mathematically, it can be expressed as:

$$\mathbf{MAPE = (1/n) * \Sigma(|(yi - xi) / xi|) * 100}$$

Where,

n is the number of data points in the dataset

yi is the predicted value for the i-th data point

xi is the actual value for the i-th data point

5.3 ALGORITHMS

5.3.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision

boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence the algorithm is termed as Support Vector Machine.

There are two main types of SVM: linear SVM and nonlinear SVM. Linear SVM is used when the data is linearly separable, while nonlinear SVM is used when the data is not linearly separable. Nonlinear SVM uses kernel functions to map the data to a higher dimensional space where it can be linearly separated. SVM has several advantages over other classification algorithms. It is effective in high-dimensional spaces, it works well with both linearly separable and nonlinearly separable data, and it is less prone to overfitting. SVM is used in a wide range of applications, including image classification, bioinformatics, text classification, and more.

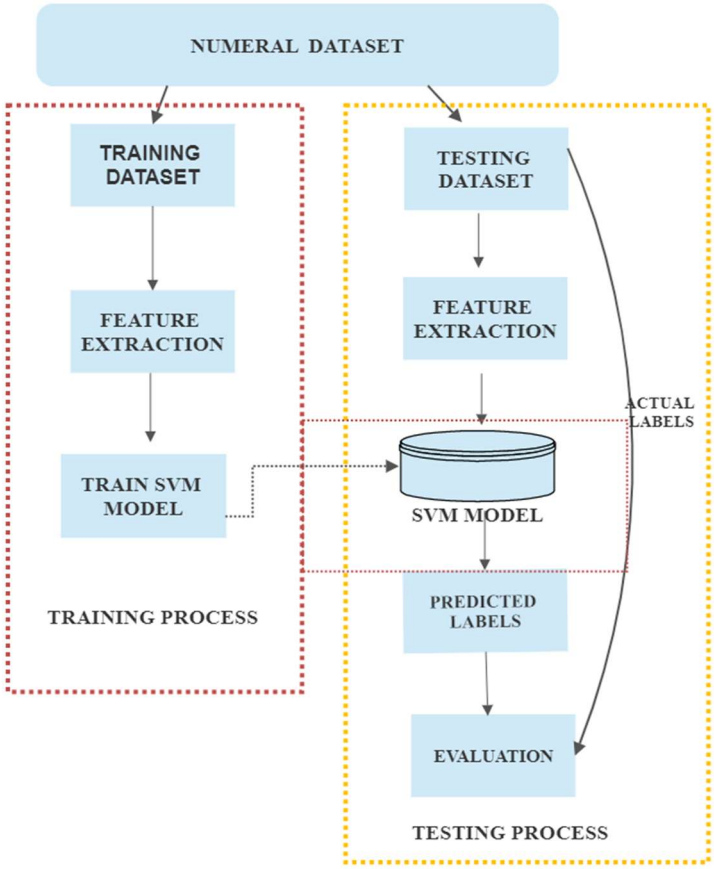


Fig 5.1 Schematic diagram of Support Vector Machine

5.3.2 RANDOM FOREST

Random Forest is a very popular algorithm in machine learning. The Random Forest algorithm is mostly used for classification and regression problems. Random forest algorithm belongs to the ensemble learning algorithms, which produce the output based on the multiple combined models. The ensemble technique improves the accuracy and produced the output in an effective manner. The problem of Decision tree is overfitting, where the decision tree model performs well in training time and not well in testing time. To overcome the problem, we are using random forest algorithm.

Decision trees: The random forest algorithm uses decision trees as its basic building blocks. A decision tree is a flowchart-like structure that makes decisions based on a set of rules or conditions. Randomness: The random forest algorithm introduces randomness in two ways. First, it selects a random subset of the input features at each node of the decision tree to split on. This helps to reduce overfitting and increase the diversity of the trees. Second, it uses bootstrapping to create multiple random subsets of the input data to train each decision tree. Ensemble learning: The random forest algorithm builds multiple decision trees using the randomly selected input features and bootstrapped data subsets. Each decision tree is trained independently on a different subset of the input data, which leads to different trees making different predictions. Combining outputs: Once the decision trees are trained, the random forest algorithm combines their outputs to make a final prediction. For classification, the algorithm takes the majority vote of the class predictions made by the individual trees. For regression, the algorithm takes the average of the output values predicted by the individual trees. Performance: Random Forest is known for its high performance and robustness. It works well with both high-dimensional and low-dimensional data, handles missing values and noisy data well, and is less prone to overfitting than single decision trees. The Random Forest

algorithm is very effective in both the regression and classification problems when compared to the decision tree algorithm. Random forest has been used in a wide range of applications, including bioinformatics, image classification, text classification, fraud detection, and more. Random forest is a powerful and flexible machine learning algorithm that is widely used for both classification and regression tasks. Its ability to handle high-dimensional data and noisy data makes it a popular choice in many real-world applications.

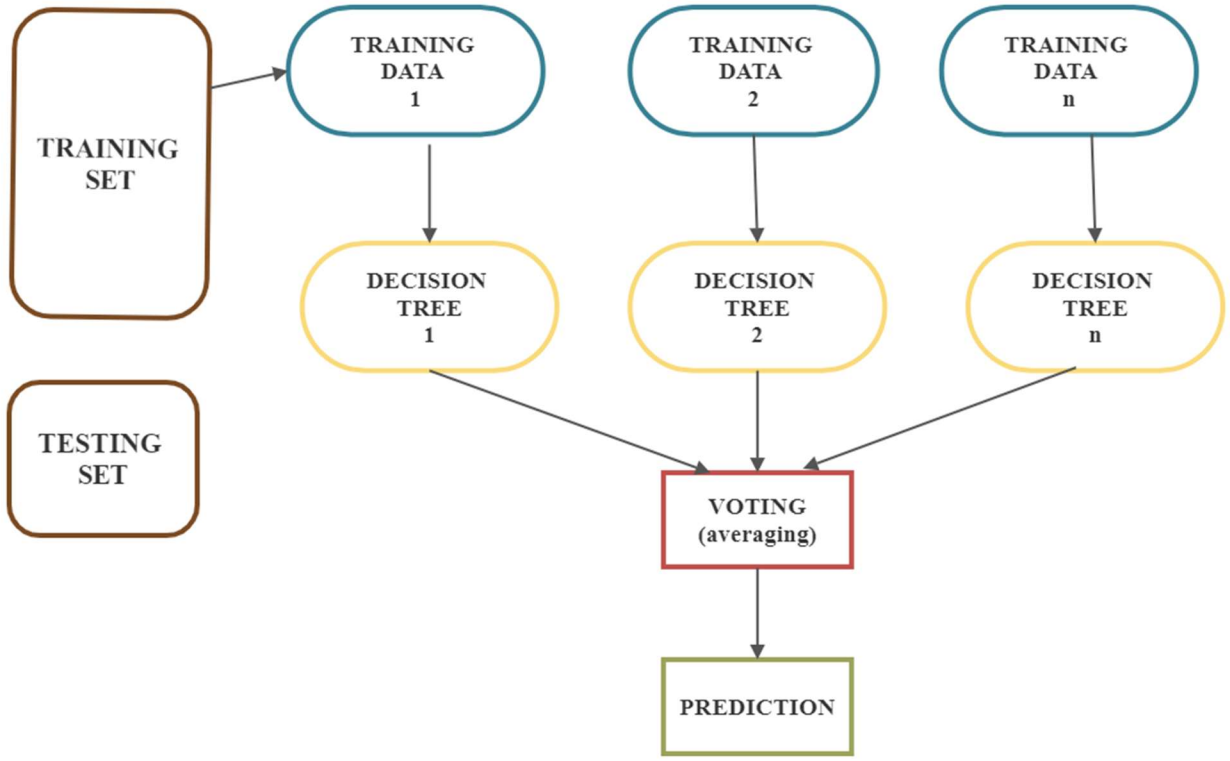


Fig 5.2: Random Forest Architecture

5.3.3 DECISION TREE

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes

represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed based on features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. To build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into subtrees. Decision Trees usually mimic human thinking ability while deciding, so it is easy to understand.

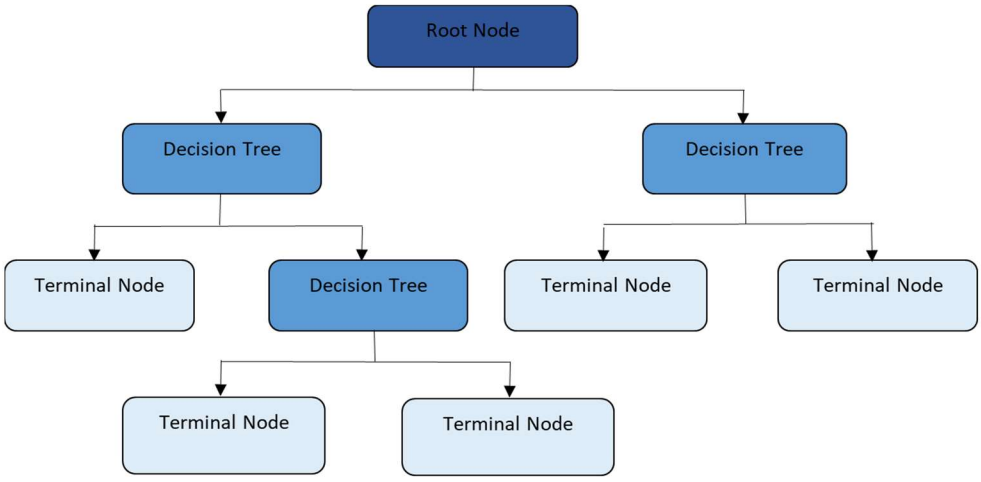


Fig 5.3: Decision Tree Architecture

5.3.4 LINEAR REGRESSION

The linear regression algorithm is a statistical method that aims to model the relationship between a dependent variable and one or more independent variables. It involves fitting a linear equation to the observed data and using this equation to make predictions or to explain the relationship between the variables. The basic steps of the linear regression algorithm are:

- Collect data: The first step in linear regression is to collect the data for the dependent and independent variables. The data should be a representative sample of the population of interest.

- Analyze the data: The next step is to analyse the data to identify any patterns or relationships between the variables. This may involve calculating descriptive statistics, plotting scatter plots, and calculating correlation coefficients.
- Choose the model: The next step is to choose the appropriate linear regression model for the data. This may involve selecting between simple linear regression, multiple linear regression, polynomial regression, or other types of regression models.
- Fit the model: The next step is to fit the chosen model to the data using a method such as ordinary least squares regression, maximum likelihood estimation, or gradient descent.
- Evaluate the model: The final step is to evaluate the model's performance and its ability to make accurate predictions. This may involve calculating measures such as the coefficient of determination (R-squared), root mean squared error (RMSE), or mean absolute error (MAE). Once the linear regression model has been fit to the data, it can be used to make predictions

about the value of the dependent variable based on the values of the independent variables.

Linear regression is widely used in various fields such as finance, economics, social sciences, engineering, and machine learning. Linear regression can be used to build models that predict the value of a dependent variable based on the values of one or more independent variables, to identify trends in data and to analyse the relationship between two or more variables. Linear regression can be used to test hypotheses about the relationship between variables, in risk analysis to identify the factors that contribute to risk and to quantify the impact of those factors.

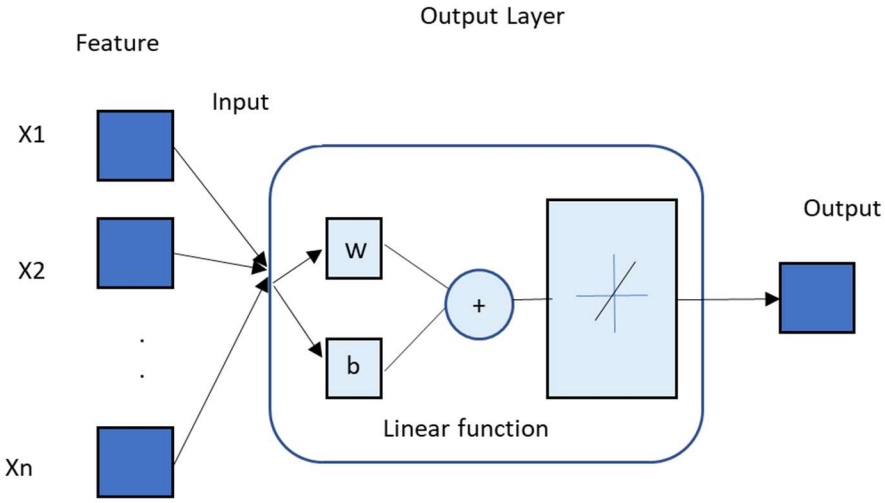


Fig 5.4: Linear regression Architecture

5.3.5 ADABOOST REGRESSION

Adaboost (Adaptive Boosting) regression is a machine learning algorithm that is used for regression problems. It is an ensemble method that combines multiple weak learners (base models) to create a strong learner. Adaboost was initially designed for classification problems, but it can also be adapted for regression problems. In Adaboost regression, a set of weak learners (e.g., decision trees) are trained sequentially. Each weak learner is trained on a modified version of the training data. The modifications are based on the errors made by the previous weak learner. The idea is to give more weight to the examples that were misclassified by the previous learner, and less weight to the correctly classified examples.

The training process of Adaboost regression can be summarized in the following steps:

- Initialize the weights: Each training example is assigned an initial weight. Train a weak learner: A weak learner is trained on the modified training data, where the weights of the examples are adjusted based on the errors made by the previous learner.
- Calculate the error: The error of the weak learner is calculated on the training data.
- Calculate the weight of the weak learner: The weight of the weak learner is calculated based on its error.
- Update the weights: The weights of the training examples are updated based on the weight of the weak learner.
- Repeat: Steps 2-5 are repeated for a predefined number of iterations, or until the error reaches a certain threshold.
- Combine the weak learners: The final model is created by combining the weak learners with their respective weights. Adaboost regression is a powerful algorithm that can handle nonlinear and complex relationships

between the input variables and the output variable. It is particularly useful in situations where the data is noisy or there are many irrelevant features.

However, it is sensitive to outliers in the data, which can lead to overfitting. To mitigate this, it is recommended to pre-process the data and remove any outliers before applying the algorithm.

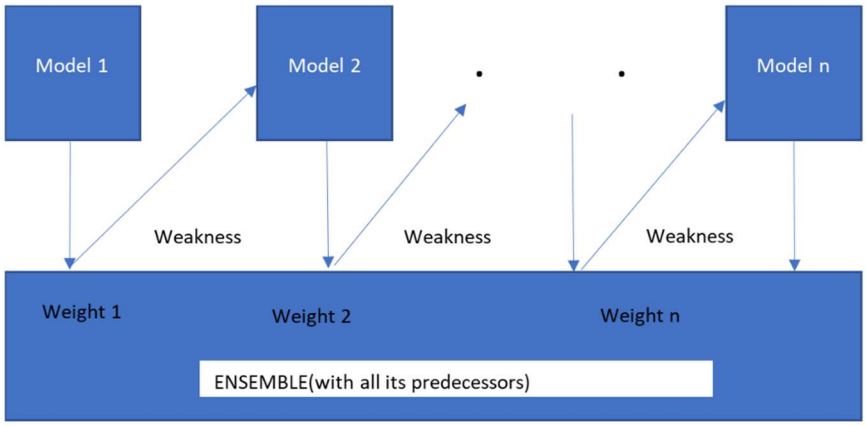


Fig 5.5: Ada boost Algorithm Architecture

CHAPTER 6
SYSTEM TESTING

6.1 INTRODUCTION

The purpose of testing is to identify the errors. Testing is defined as the process of discovering every fault or weakness in a product. The main aim of testing is to ensure that the software system meets user expectation and does not fail in any manner. Testing is necessary for quality assurance, Error detection, Cost effective, Improved performance, Security and Compliance. There are various types of tests, each of the types addresses testing requirements.

6.2 TYPES OF TESTS

6.2.1 UNIT TESTING

Unit testing is the process which involves the designing of the test cases for the program that can validate internal program's logic is working properly and the program produces outputs for all the valid inputs. The complete unit testing done after the testing of each individual units of the software. This is a kind of structural testing, that depends on the knowledge of its construction. Unit tests ensure that each business process' path performs accurately and contains the well-defined possible inputs and their corresponding results. By using this testing any individual components of a software application are tested in isolation manner.

This is done by provides the possible inputs to the isolated unit and check the outputs against the expected outcome. Any antagonism between the actual and expected results are classified as errors which is needed to be fixed before moving to the testing of next phases. Unit testing identified the defects in the earlier development stage, which avoids the unwanted costs for the defect reducing in the

later development process. It improves the maintainability of the code by ensuring changes on one unit does not affects the other separated units of the software.

6.2.2 INTEGRATION TESTING

Integration tests are used to testing the integrated components of software to determine if they can run as one combined program or not. Integration testing especially used at the problems which may arise from the components of software combination. This software testing technique where the software modules are combined and tested to ensure whether the combined components function together correctly. The main aim of integration testing is to discover the errors that arise from interaction of the modules. The objective of this testing is to ensure that the data is being passed between the modules correctly. It contains two approaches top-down and bottom-up. In top-down integration testing higher level modules are tested first then the lower-level modules whereas in bottom-up approach lower level tested first next, higher-level modules are tested.

6.2.3 FUNCTIONAL TESTING

Functional testing is a types of software testing that verifies the system can function against the business requirements and their specifications. The main aim of the functional testing is to ensure that the application is working, and all these functions of the application are working correctly. Tests are based on the requirements and their specifications which is conducted to identify the errors between the actual and expected result.

Functional testing is based on the following items:

- Valid Input: Identified the classes which accepts the valid inputs.
- Invalid Input: Identified the classes which rejects the invalid inputs.

• Functions: Identified the functions which must be exercised.

• Output: Identified the classes of the software where outputs must exercise.

6.2.4 SYSTEM TESTING

System testing is a type of software testing that can verifies the functionality and behaviour of a complete software system, rather than testing the software as an individual component. The main purpose of the system testing is to evaluate the system with the functional and non-functional requirements. Those requirements including security, reliability and performance. System testing is conducted after the completion of integration testing and before the user acceptance testing. It involves testing all the interconnected components to ensure that, they can work together as expected. The test involves positive and negative testing, boundary testing, load testing and security testing. The system testing helps to ensure that the software application is of high quality and meets the user's requirements.

6.2.5 WHITE BOX TESTING

White box testing is a type of software testing that evaluates the internal structure of the software application to identify the defects in the application. It is popularly known as open-box testing. In this testing technique, the tester has the access to the review the source code, doing statistical and dynamic analysis to verify the behaviour of the software. The main aim of the testing is to ensure the application functioning as expected and free from errors, security vulnerabilities. It is mainly used for testing the complex software applications.

6.2.6 BLACK BOX TESTING

Black box testing evaluates the functionality of an application without examining its implementation details. In this testing technique, the tester is not concerned with the code, only focusing on the inputs and outputs. It can be performed manually or automated tool. It is suitable for the application developed by different teams because it does not require the knowledge of the code. The aim is identifying the defect and ensure whether the actual and expected results are matched. Black box testing ensures the quality and reliability of software application by verifying their behaviour without examining their implementation details.

6.2.7 ACCEPTANCE TESTING

Acceptance testing examines the system meets its business requirements and specifications and it is ready for the deployment or not. The aim of this testing is to ensure whether the application meets user requirements, it is free of errors and suitable for the production environment. It is performed by the stakeholders, representatives and end-users, who evaluates the application's functionalities. Acceptance testing including the user, operations and contracts. It includes different scenarios such as boundary testing, security testing, positive and negative testing. It is necessary to ensures the reliability and quality of the software application.

CHAPTER 7
RESULT

Medical insurance cost predictors can help insurance companies control the cost of healthcare by identifying individuals who are more likely to require expensive medical treatments or procedures. This can help insurance companies develop cost-effective healthcare strategies and pricing models, which can ultimately benefit the consumers. Therefore, in this thesis we are developing a machine learning model to predict the cost of insurance policy. In this thesis we have taken a dataset from GitHub which includes age, gender, BMI, smoker, region, number of children as parameters some of their past medical and surgical history.

We have tested the machine learning model with 5 algorithms. According to the obtained evaluation metrics results, the AdaBoost algorithm offers the best precision, accuracy in predicting the insurance policy cost.

7.1 RESULTING GRAPHS

Support Vector Machine Algorithm Metrics

R2 Score	MSE	RMSE	MAE	MAPE
-0.04	16.723946	12.932	8.419	1.11

TABLE 7.1: Support Vector Graph

Random Forest Algorithm Metrics

R2 Score	MSE	RMSE	MAE	MAPE
0.91	13.295379	3.646	2.519	0.33

TABLE 7.2: Random Forest Graph

Decision Tree Algorithm Metrics

R2 Score	MSE	RMSE	MAE	MAPE
0.92	11.104021	3.332	1.317	0.07

TABLE 7.3 Decision Tree Graph

Linear Regression Algorithm Metrics

R2 Score	MSE	RMSE	MAE	MAPE
0.92	11.343595	3.368	1.943	0.16

TABLE 7.4 Linear Regression Graph

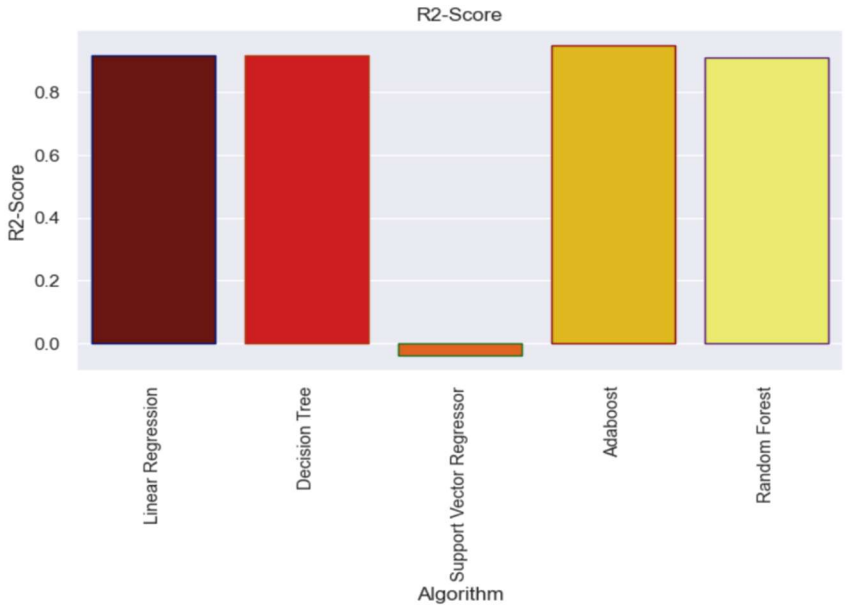
Adaptive Boosting Algorithm Metrics

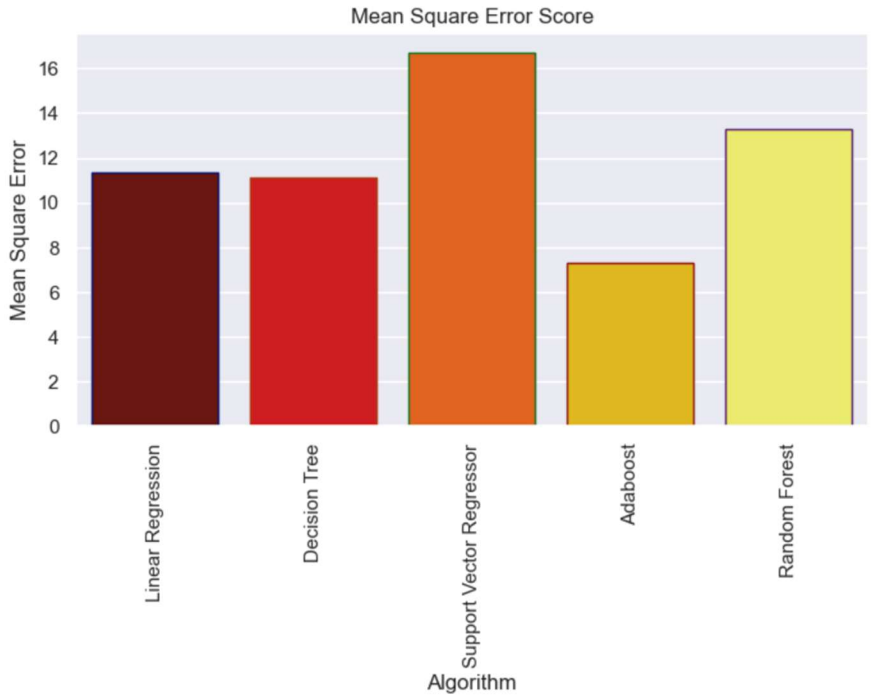
R2 Score	MSE	RMSE	MAE	MAPE
0.95	7.316735	2.704	1.004	0.06

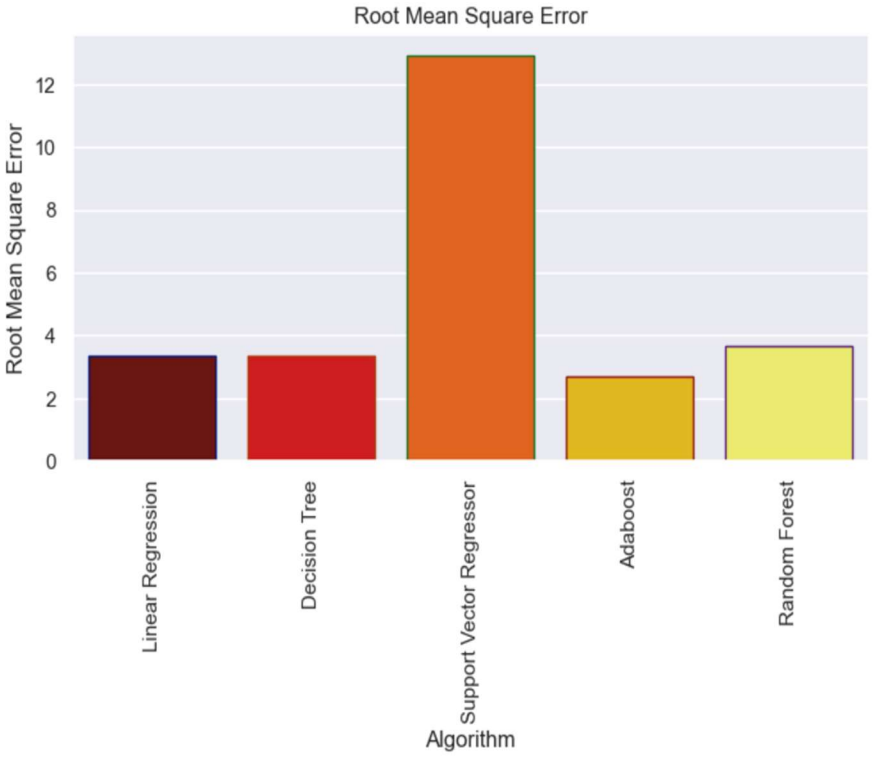
TABLE 7.5 AdaBoost Graph

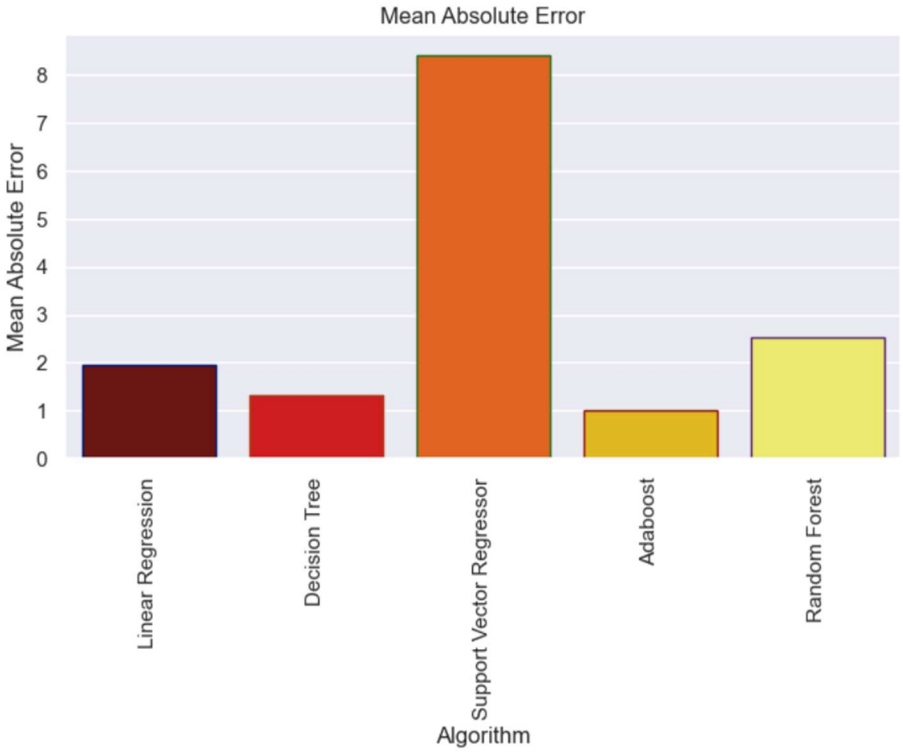
7.1.1 METRICS

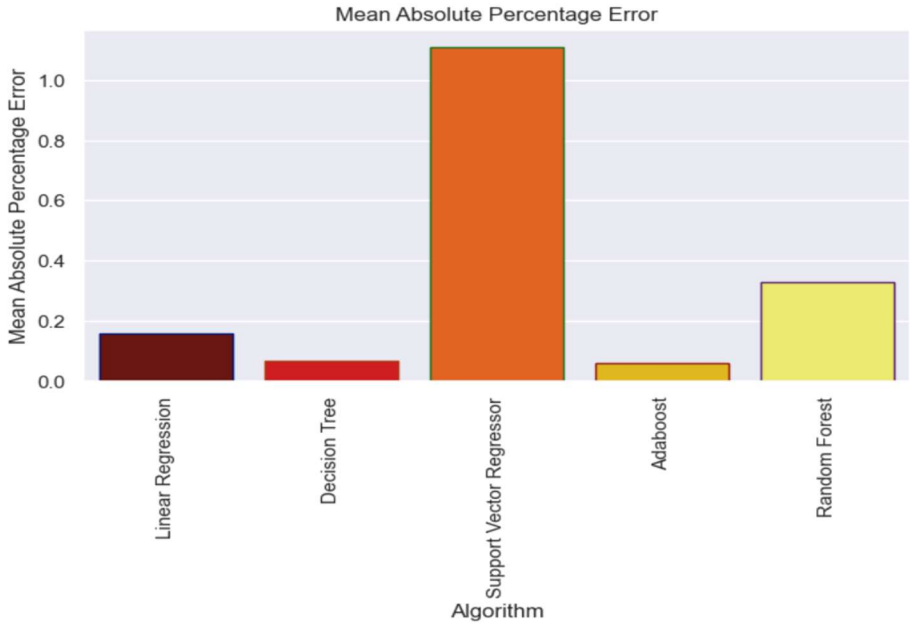
1. R2 Score
2. Mean Square Error Score
3. Root Mean Square Error Score
4. Mean Absolute Error Score
5. Mean Absolute Percentage Error Score




Mean Square Error Score
Mean Square Error
0
2
4
6
8
10
12
14
16
Linear Regression
Decision Tree
Support Vector Regressor
Adaboost
Random Forest
Algorithm


Root Mean Square Error
Root Mean Square Error
0
2
4
6
8
10
12
Linear Regression
Decision Tree
Support Vector Regressor
Adaboost
Random Forest
Algorithm


Mean Absolute Error
Mean Absolute Error
0
1
2
3
4
5
6
7
8
Linear Regression
Decision Tree
Support Vector Regressor
Adaboost
Random Forest
Algorithm


Mean Absolute Percentage Error
Mean Absolute Percentage Error
0.0
0.2
0.4
0.6
0.8
1.0
Linear Regression
Decision Tree
Support Vector Regressor
Adaboost
Random Forest
Algorithm

7.1.2 ANALYSIS

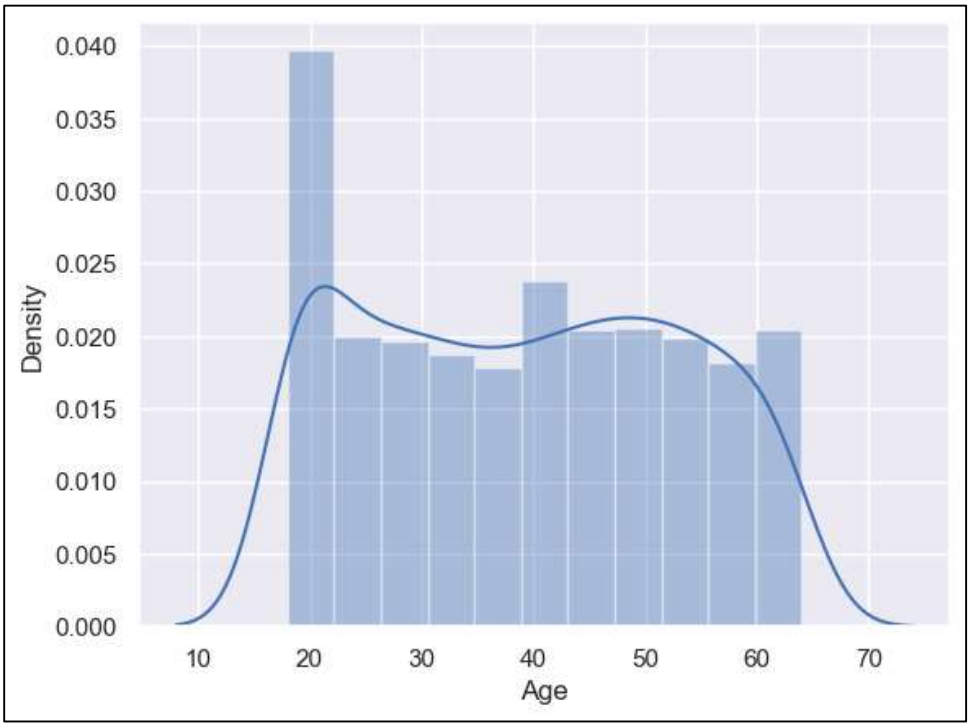


Fig 7.1: Age and Density analysis – Bar Chart

The Dataset contains attributes such as Age, Sex, BMI, Children, Smoker, Region, Charges.

- ❖ The Age attribute has the **Minimum** value for Age as 18
- ❖ The Age attribute has the **Maximum** age as 64
- ❖ The Age attribute has the **Mean** age as 39.20
- ❖ The Age attribute has the **Median** age as 39.0

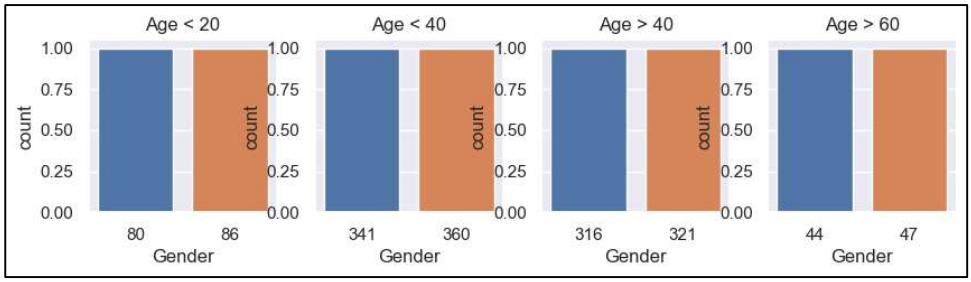


Fig 7.2: Age and Gender Analysis – Bar Chart

Based on the information we find that we must give priority to the customers who has the age between 20 to 60

From this bar chart both the male as well as female have taken the insurance, we understand that the gender attribute has not performed any impact on the output.

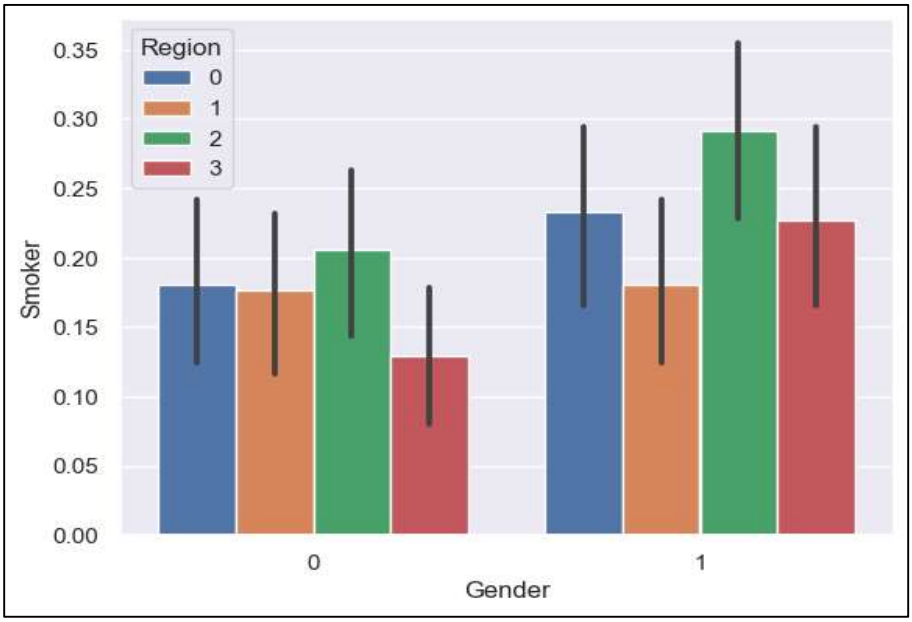


Fig 7.3: Smoker and Gender Analysis – Bar Chart

The graph is the correlation of Gender, Smoker and Region

- The Region 0 is North-East
- The Region 1 is North-West
- The Region 2 is South-East
- The Region 3 is South-West

According to the above bar chart, female smokers are comparatively high in numbers in region 2 which is South-East. Male smokers are comparatively high in numbers in region 2 which is South-East. From this information we get to know that customers in the region 2 have high possibility of getting the insurance policy.

Similarly, we have a high number of male and female smokers in the region 0 which is North-East, so there is a possibility of getting many numbers of insurance policy takers.

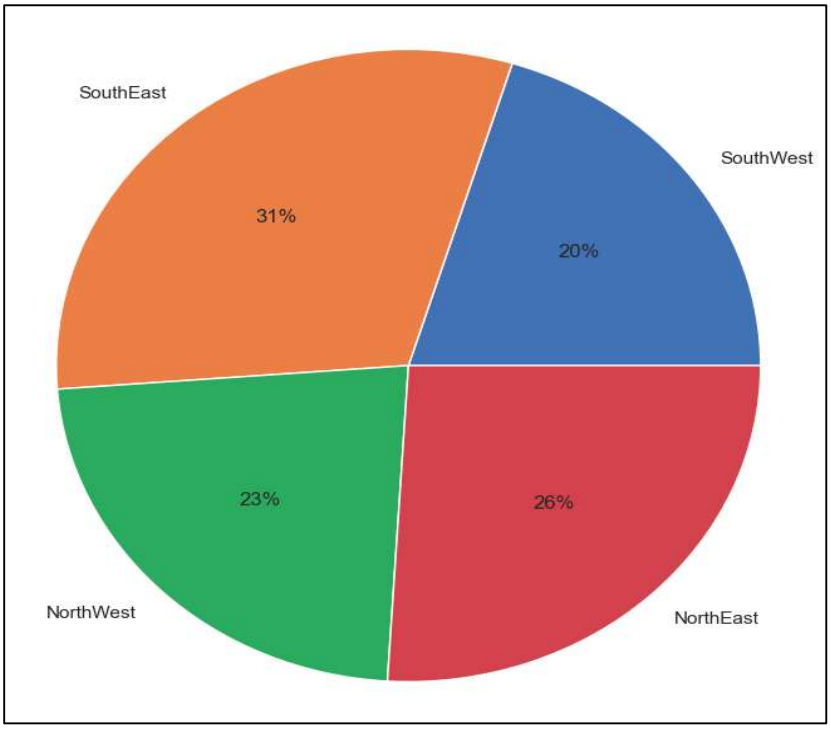


Fig 7.4: Regional analysis – Pie Chart

Next, we have created a column split target median, in that we got 9000 as median value for charges. Here we have replaced the values greater than median as 1 and values lower than median as 0.

Overall regions count,

For South-East region: 364

For South-West region: 325

For North-West region: 325

For North-East region: 324

After analyzing we find that,

South-East region:

Among 364 records only 85 holds the median value as 1

South-West region:

Among 325 records only 130 holds the median value as 1

North-West region:

Among 325 records only 96 holds the median value as 1

North-East region:

Among 324 records only 109 holds the median value as 1

While comparing all four regions we should focus on regions with a high percentage of target median the regions are South-east which holds 31% and North-East which holds 26%. So that profit amount yield by the organization will be high.

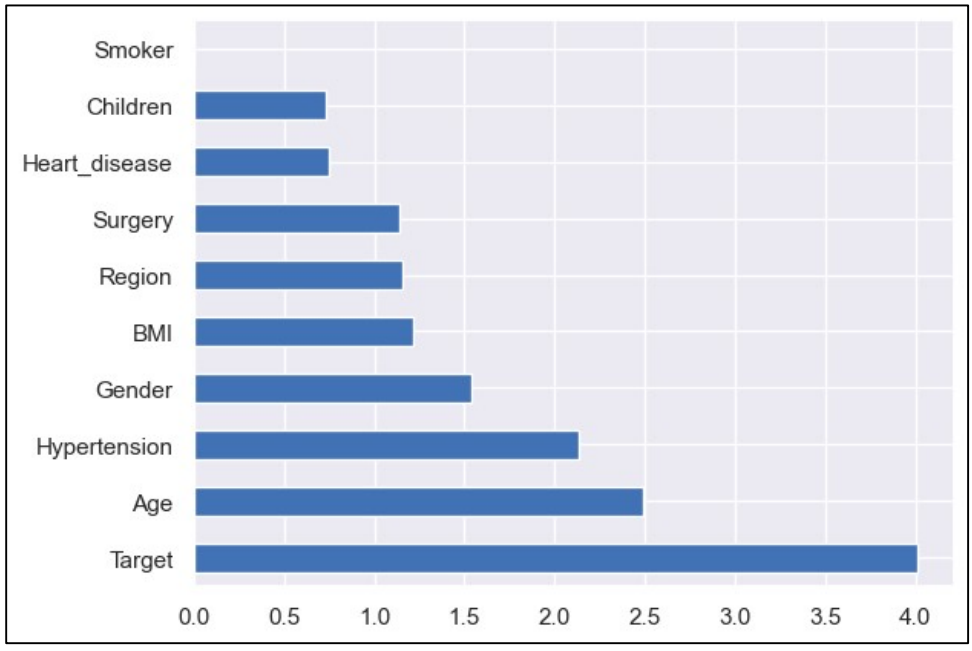


Fig 7.5: Mutual Information Score – Bar Chart

By using the mutual_info_classif function from sklearn we identified the important feature of attribute.

To visualize the mutual information scores for each feature, here x-axis represents the mutual information scores, and y-axis represents the feature names. The age attribute bar in the chart indicate that the corresponding feature is highly informative and has a strong information with the target variable, while smoker attribute being the low bar indicates that the feature is less informative and has a weak relationship with the target variable. Thus, the chart is used to identify the most informative features for classification.

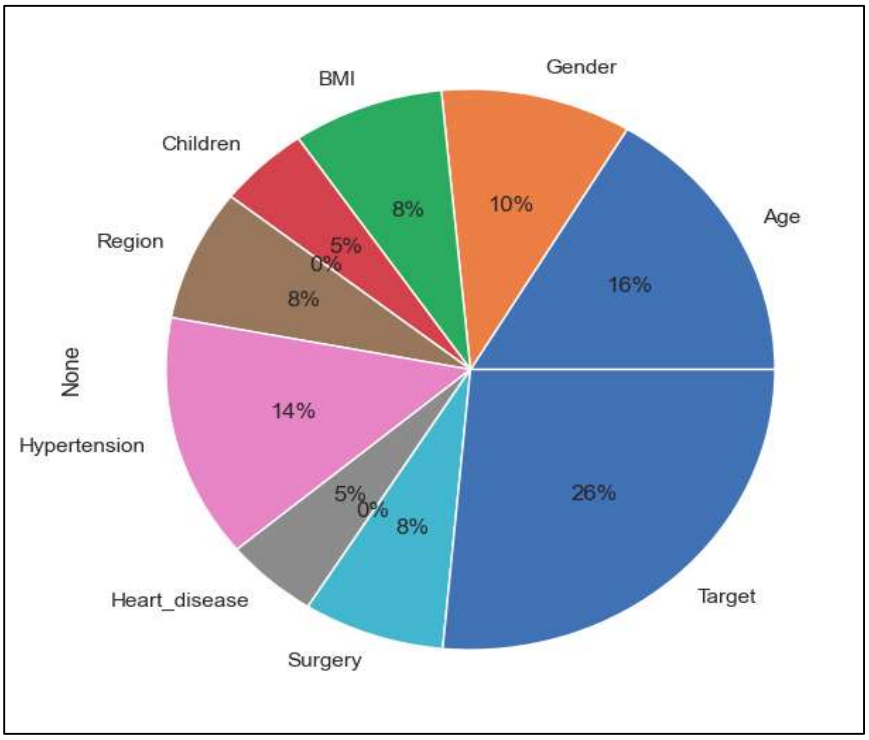


Fig 7.6: Importance of the Feature – Pie Chart

By using the ExtraTreesClassifier function from sklearn we identified the important feature of attribute. It works by constructing many decision trees and combining their predictions to produce a result.

The size of each slice is proportional to the feature's importance relative to the other features. Gender, Age feature has the highest importance while, Smoker feature has the lowest importance. By focusing on the most important features, we can potentially reduce the dimensionality of the problem and improve the accuracy of the model.

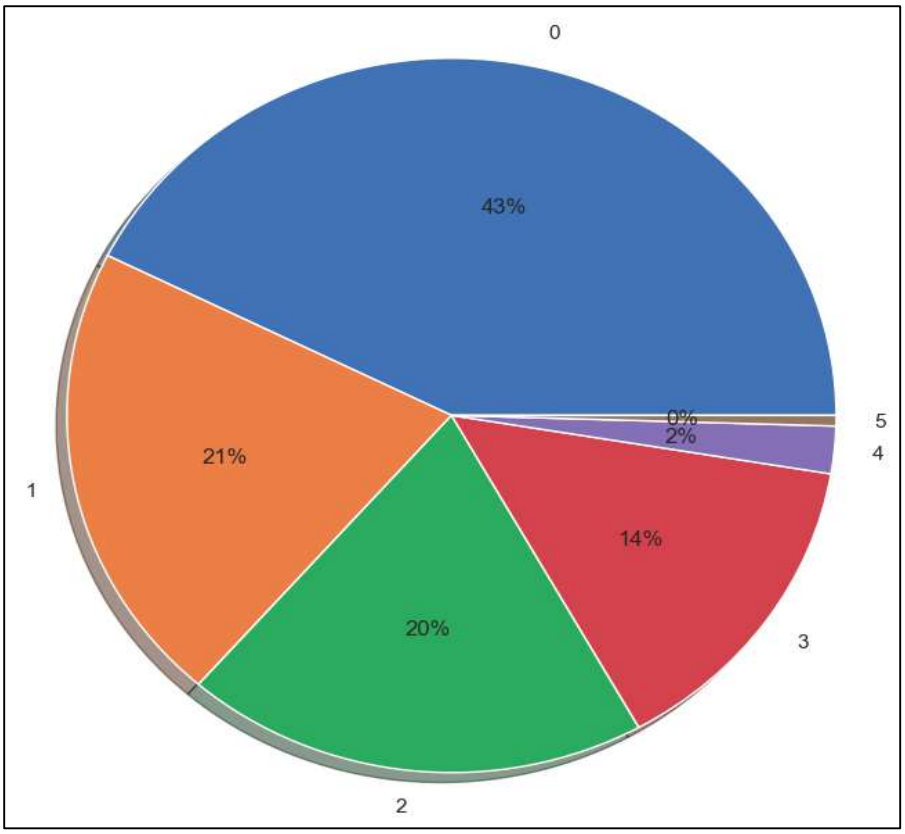


Fig 7.7: Children attribute pattern -Pie Chart

The above pie chart is useful in identifying patterns or trend in the data, such as whether health policy takers with more children are less or more likely to take the insurance.

In our data set,

- 43% of policy takers have no child
- 21% of policy takers have 1 child

- 20% of policy takers have 2 children
- 14% of policy takers have 3 children
- 2% of policy takers have 4 children
- 0% of policy takers have 5 children

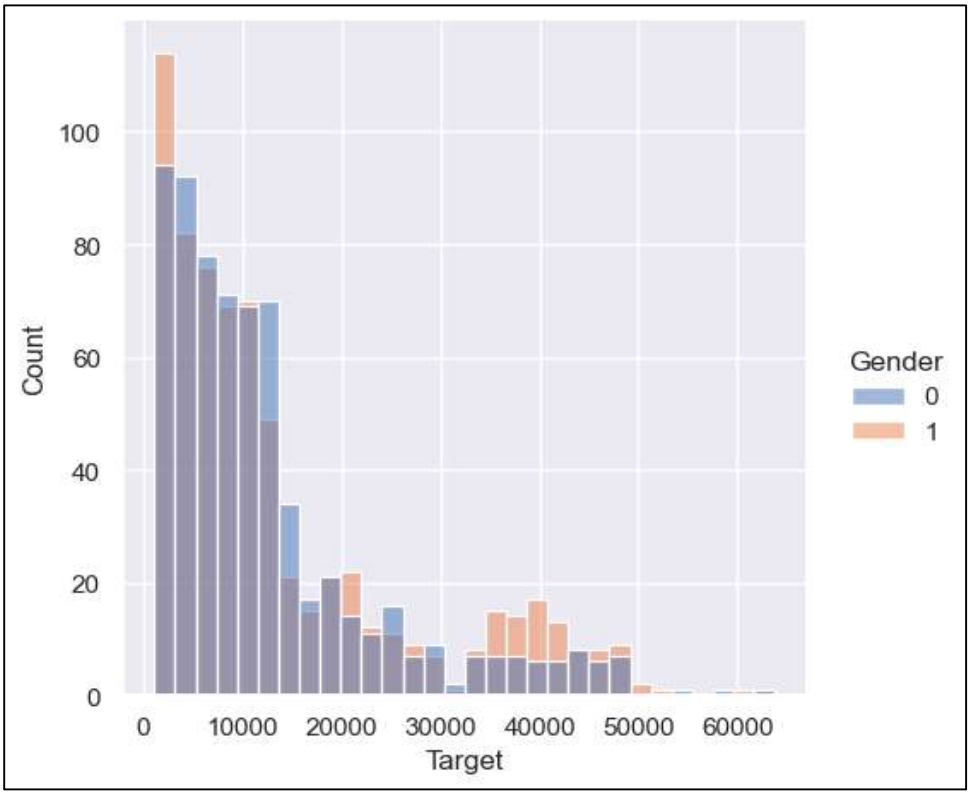


Fig 7.8: Policy Takers and Target Analysis – Bar Chart

The x-axis represents the amount of money invested in insurance policy and y-axis represents the number of policy takers. Blue indicates female policy takers and red indicates male policy takers. We can state that female policy takers tend to take low amount premium policy plans than male policy takers. So, the target customer is male policy takers, as we can see from the graph, they tend to invest increased wealth.

7.2 SCREENSHOTS

7.2.1 DATA PRE-PROCESSING

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split

In [2]: import warnings
        warnings.filterwarnings('ignore')
        sns.set_theme(style="darkgrid")

In [3]: df = pd.read_csv('/Users/subashkrishnan/Desktop/Project/datanew.csv')
        df.head()
```

Out[3]:

	Age	Sex	BMI	Children	smoker	region	Hypertension	Heart_disease	Cancer	Surgery	Charges
0	54	female	47.410	0	yes	southeast	0	0	1	0	63770
1	45	male	30.360	0	yes	southeast	0	0	1	0	62592
2	52	male	34.485	3	yes	northwest	0	0	1	0	60021
3	31	female	38.095	1	yes	northeast	0	0	1	0	58571
4	33	female	35.530	0	yes	northwest	0	0	1	0	55135

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             1338 non-null   int64
 1   Sex             1338 non-null   object
 2   BMI             1338 non-null   float64
 3   Children        1338 non-null   int64
 4   smoker          1338 non-null   object
 5   region          1338 non-null   object
 6   Hypertension    1338 non-null   int64
 7   Heart_disease   1338 non-null   int64
 8   Cancer          1338 non-null   int64
 9   Surgery         1338 non-null   int64
 10  Charges         1338 non-null   int64
dtypes: float64(1), int64(7), object(3)
memory usage: 115.1+ KB
```

```
In [8]: df.describe()
```

Out[8]:

	Age	BMI	Children	Hypertension	Heart_disease	Cancer	Surgery	Charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	0.367713	0.242152	0.121076	0.227205	13269.928999
std	14.049960	6.098187	1.205493	0.482363	0.428546	0.326338	0.419183	12110.012755
min	18.000000	15.960000	0.000000	0.000000	0.000000	0.000000	0.000000	1121.000000
25%	27.000000	26.296250	0.000000	0.000000	0.000000	0.000000	0.000000	4740.000000
50%	39.000000	30.400000	1.000000	0.000000	0.000000	0.000000	0.000000	9381.500000
75%	51.000000	34.693750	2.000000	1.000000	0.000000	0.000000	0.000000	16639.250000
max	64.000000	53.130000	5.000000	1.000000	1.000000	1.000000	1.000000	63770.000000

This data set contains 1338 data points with 7 independent features and 1 target feature.

```
In [9]: df.isnull().sum()

Out[9]: Age              0
        Sex              0
        BMI              0
        Children         0
        smoker           0
        region           0
        Hypertension     0
        Heart_disease    0
        Cancer           0
        Surgery          0
        Charges          0
        dtype: int64

In [10]: df.head(2)
```

Out[10]:

	Age	Sex	BMI	Children	smoker	region	Hypertension	Heart_disease	Cancer	Surgery	Charges
0	54	female	47.41	0	yes	southeast	0	0	1	0	63770
1	45	male	30.36	0	yes	southeast	0	0	1	0	62592

```
In [11]: df.shape

Out[11]: (1338, 11)
```

Data.isnull().sum() function is used to whether this dataset contains any null values or not.

```
In [12]: le_sex = LabelEncoder()
         le_smoker = LabelEncoder()
         le_region=LabelEncoder()

In [13]: df['Gender']=le_sex.fit_transform(df.Sex)
         df['Smoker']=le_smoker.fit_transform(df.smoker)
         df['Region']=le_region.fit_transform(df.region)

In [14]: df
```

Out[14]:

	Age	Sex	BMI	Children	smoker	region	Hypertension	Heart_disease	Cancer	Surgery	Charges	Gender	Smoker	Region
0	54	female	47.410	0	yes	southeast	0	0	1	0	63770	0	1	2
1	45	male	30.360	0	yes	southeast	0	0	1	0	62592	1	1	2
2	52	male	34.485	3	yes	northwest	0	0	1	0	60021	1	1	1
3	31	female	38.095	1	yes	northeast	0	0	1	0	58571	0	1	0
4	33	female	35.530	0	yes	northwest	0	0	1	0	55135	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1333	18	male	34.430	0	no	southeast	1	0	0	0	1137	1	0	2
1334	18	male	33.660	0	no	southeast	1	0	0	0	1136	1	0	2
1335	18	male	33.330	0	no	southeast	1	0	0	0	1135	1	0	2
1336	18	male	30.140	0	no	southeast	1	0	0	0	1131	1	0	2
1337	18	male	23.210	0	no	southeast	1	0	0	0	1121	1	0	2

1338 rows × 14 columns

```
In [15]: df.drop(["Sex",'smoker','region'],axis=1,inplace=True)
         df.rename(columns={"age":'Age',
                            'bmi':"BMI",
                            'children' : "Children",
                            'charges' : "Charges"
                            },inplace=True)

In [16]: df = df[['Age','Gender','BMI','Children','Smoker','Region','Hypertension','Heart_disease','Cancer','Surgery','Charges']]
         df.head()
```

Out[16]:

	Age	Gender	BMI	Children	Smoker	Region	Hypertension	Heart_disease	Cancer	Surgery	Charges
0	54	0	47.410	0	1	2	0	0	1	0	63770
1	45	1	30.360	0	1	2	0	0	1	0	62592
2	52	1	34.485	3	1	1	0	0	1	0	60021
3	31	0	38.095	1	1	0	0	0	1	0	58571
4	33	0	35.530	0	1	1	0	0	1	0	55135

```
In [17]: print(df.Charges.mean())
         print(df.Charges.max())
         print(df.Charges.min())

         13269.928998505231
         63770
         1121
```

After dividing the data into training and validation data it is trained.

7.2.2 EVALUATION METRICS

```
In [84]: from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
         from sklearn.metrics import mean_absolute_percentage_error

In [85]: from sklearn.linear_model import LinearRegression
         from sklearn.tree import DecisionTreeRegressor
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.svm import SVR
         from sklearn.ensemble import AdaBoostRegressor

In [86]: lin_reg=LinearRegression()
         lin_reg.fit(x_train,y_train)
         y_pred=lin_reg.predict(x_test)
         r2 = r2_score(y_test,y_pred)
         mse = mean_squared_error(y_test,y_pred)
         mae = mean_absolute_error(y_test,y_pred)
         rmse = np.sqrt(mean_squared_error(y_test,y_pred))
         mape = mean_absolute_percentage_error(y_test,y_pred)
         N = len(y_test)
         adj_r2 = (1-r2)*(N-1)/(N-6-1)
         print('              Linear Regression')
         print('                  ----------')
         print('')
         print("R2 Score : ", r2)
         print("Mean Score Error Score : ", mse)
         print("Root Mean Score Error Score : ", rmse)
         print("Mean Absolute Error Score :", mae)
         print("Mean Absolute Percentage Error Score :", mape)
         print("Adjusted R2 Score : ", adj_r2)

                       Linear Regression
                           ----------

         R2 Score :  0.9268945502185703
         Mean Score Error Score :  11343595.378431058
         Root Mean Score Error Score :  3368.0254420700353
         Mean Absolute Error Score : 1943.435585196211
         Mean Absolute Percentage Error Score : 0.16839562960740737
```

```
In [87]: from sklearn.tree import DecisionTreeRegressor
         dec_tree=DecisionTreeRegressor()
         dec_tree.fit(x_train,y_train)
         y_pred=dec_tree.predict(x_test)
         r2 = r2_score(y_test,y_pred)
         mse = mean_squared_error(y_test,y_pred)
         mae = mean_absolute_error(y_test,y_pred)
         rmse = np.sqrt(mean_squared_error(y_test,y_pred))
         mape = mean_absolute_percentage_error(y_test,y_pred)
         N = len(y_test)
         adj_r2 = (1-r2)*(N-1)/(N-6-1)
         print('              Decision Tree Regressor')
         print('                  -------------')
         print('')
         print("R2 Score : ", r2)
         print("Mean Score Error Score : ", mse)
         print("Root Mean Score Error Score : ", rmse)
         print("Mean Absolute Error Score :", mae)
         print("Mean Absolute Percentage Error Score :", mape)
         print("Adjusted R2 Score : ", adj_r2)

                       Decision Tree Regressor
                           -------------

         R2 Score :  0.9284385203791737
         Mean Score Error Score :  11104021.27238806
         Root Mean Score Error Score :  3332.26968782361
         Mean Absolute Error Score : 1317.205223880597
         Mean Absolute Percentage Error Score : 0.07517886755587956
```

In [88]:
```
from sklearn.svm import SVR
svr=SVR()
svr.fit(x_train,y_train)
y_pred=svr.predict(x_test)
r2 = r2_score(y_test,y_pred)
mse = mean_squared_error(y_test,y_pred)
mae = mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(mean_squared_error(y_test,y_pred))
mape = mean_absolute_percentage_error(y_test,y_pred)
N = len(y_test)
adj_r2 = (1-r2)*(N-1)/(N-6-1)
print('              Support Vector Regressor')
print('                  -------------')
print('')
print("R2 Score : ", r2)
print("Mean Score Error Score : ", mse)
print("Root Mean Score Error Score : ", rmse)
print("Mean Absolute Error Score :", mae)
print("Mean Absolute Percentage Error Score :", mape)
print("Adjusted R2 Score : ", adj_r2)
```

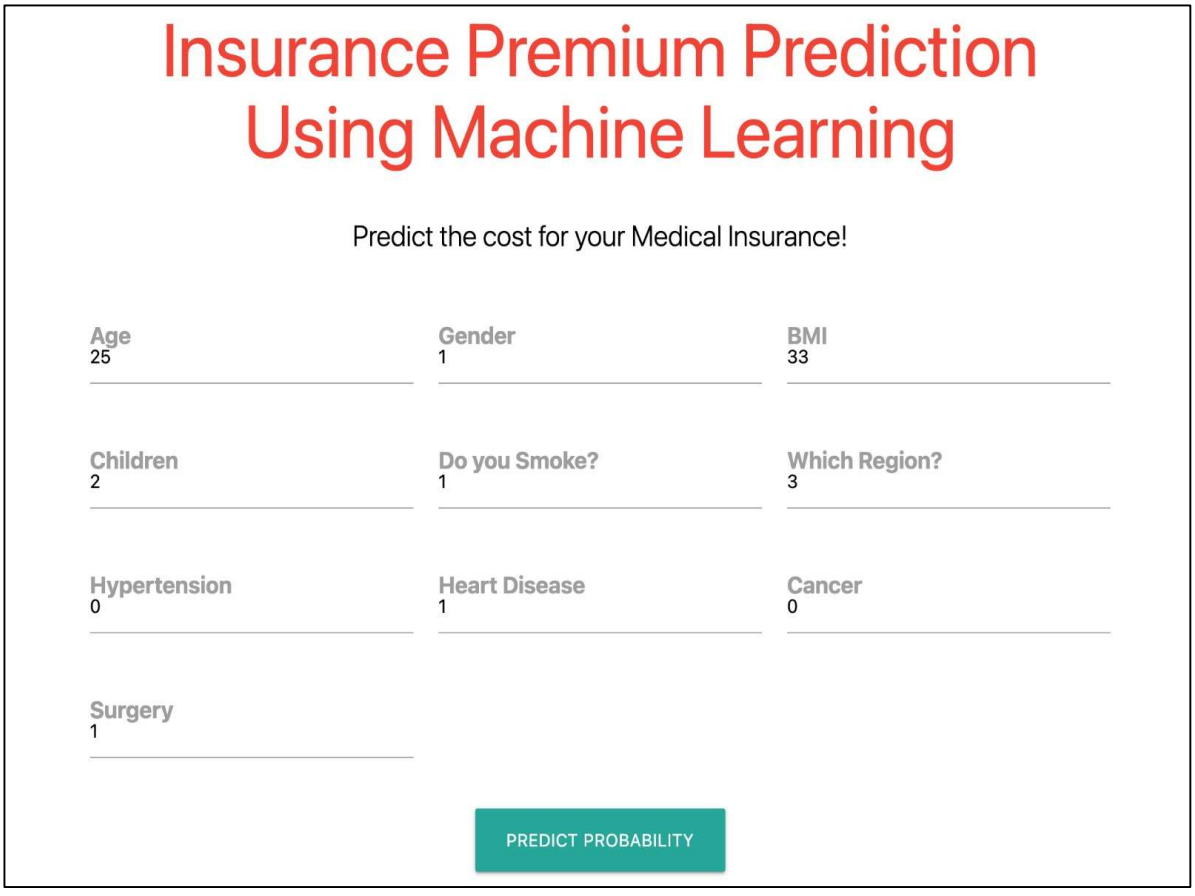
```
              Support Vector Regressor
                  -------------

R2 Score :  -0.07779902157785146
Mean Score Error Score :  167239460.75978735
Root Mean Score Error Score :  12932.109679390573
Mean Absolute Error Score : 8419.180272842306
Mean Absolute Percentage Error Score : 1.1108835942534017
```

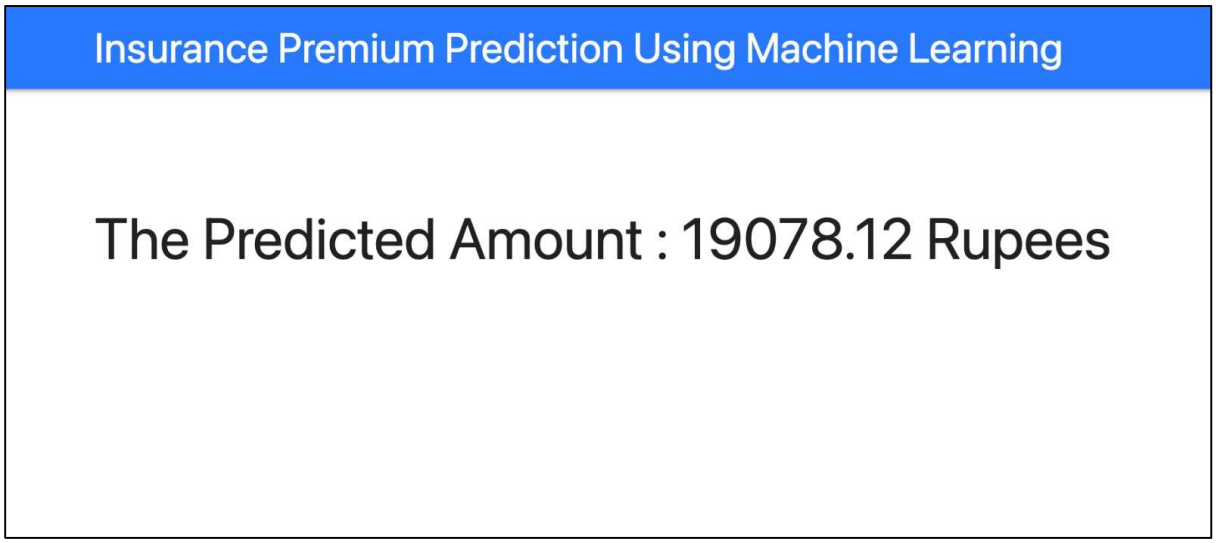
In [89]:
```
from sklearn.ensemble import AdaBoostRegressor
abr=AdaBoostRegressor(n_estimators=100,learning_rate=2)
abr.fit(x_train,y_train)
y_pred=abr.predict(x_test)
r2 = r2_score(y_test,y_pred)
mse = mean_squared_error(y_test,y_pred)
mae = mean_absolute_error(y_test,y_pred)
rmse = np.sqrt(mean_squared_error(y_test,y_pred))
mape = mean_absolute_percentage_error(y_test,y_pred)
N = len(y_test)
adj_r2 = (1-r2)*(N-1)/(N-6-1)
print('              Adaptive Boosting Regressor')
print('                  -------------')
print('')
print("R2 Score : ", r2)
print("Mean Score Error Score : ", mse)
print("Root Mean Score Error Score : ", rmse)
print("Mean Absolute Error Score :", mae)
print("Mean Absolute Percentage Error Score :", mape)
print("Adjusted R2 Score : ", adj_r2)
```

```
              Adaptive Boosting Regressor
                  -------------

R2 Score :  0.9528462342945718
Mean Score Error Score :  7316735.487312313
Root Mean Score Error Score :  2704.9464851106227
Mean Absolute Error Score : 1004.3536940298509
Mean Absolute Percentage Error Score : 0.062478326458459604
```

```
In [90]: from sklearn.ensemble import RandomForestRegressor
         rfr=RandomForestRegressor()
         rfr.fit(x_train,y_train)
         y_pred=rfr.predict(x_test)
         r2 = r2_score(y_test,y_pred)
         mse = mean_squared_error(y_test,y_pred)
         mae = mean_absolute_error(y_test,y_pred)
         rmse = np.sqrt(mean_squared_error(y_test,y_pred))
         mape = mean_absolute_percentage_error(y_test,y_pred)
         N = len(y_test)
         adj_r2 = (1-r2)*(N-1)/(N-6-1)
         print('               Random Forest Regressor')
         print('                   -------------')
         print('')
         print("R2 Score : ", r2)
         print("Mean Score Error Score : ", mse)
         print("Root Mean Score Error Score : ", rmse)
         print("Mean Absolute Error Score :", mae)
         print("Mean Absolute Percentage Error Score :", mape)
         print("Adjusted R2 Score : ", adj_r2)

                       Random Forest Regressor
                           -------------

         R2 Score :  0.9143159941605659
         Mean Score Error Score :  13295379.421802869
         Root Mean Score Error Score :  3646.2829596457364
         Mean Absolute Error Score : 2519.94448056764
         Mean Absolute Percentage Error Score : 0.3300051313167054
```

USER INTERFACE PAGE





The interface page of the project is designed to predict insurance premium cost based on input variables such as age, BMI, gender, number of children, and smoking status and some of their past medical and surgical history.

- Header: This section contains the title of the project, such as "Insurance Premium Prediction Using Machine Learning".
- Input Form: This section contains a form that allows the user to input the various variables that are used to predict the insurance premium cost. The form would typically have fields for Age, BMI, Gender, Number of Children, Smoker status and some of their past medical and surgical history.
- Predict Button: This button is used to submit the data entered by the user. After the user has entered all the required information, they can click on this button to submit the form and start the prediction.

  Output Section: This section would display the predicted insurance premium cost based on the inputs provided by the user. The predicted cost could be displayed as a Rupees amount.

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

This paper provided a simple and efficient prediction model for insurance companies to help them predict and estimate health insurance premium charges. Regressor algorithms are very useful in many applications. But for insurance claims, when compared to SVM, Random Forest, Decision tree, linear regressor algorithms Adaboost regressor gives slightly higher performance with good accuracy. The Machine Learning algorithms other than these five, can also be used to predict the insurance claim. In this paper, only five of the ML algorithms are used and evaluated. ML algorithms reduce the human work and makes easier to compute the insurance claims automatically. Moreover, these models can also manage large amount of data.

8.2 FUTURE ENHANCEMENT

As we move towards an increasingly data-driven world, the importance of utilizing real-time data cannot be exaggerated. In the context of insurance premium prediction, the use of live data sets has the potential to significantly enhance the accuracy and relevance of our models. While the collected dataset used in our current ML project has provided valuable insights, live data can offer even more up-to-date information that reflects current market trends and customer behaviour.

One of the main advantages of live data sets is their ability to provide immediate feedback. This can be particularly useful in the insurance industry, where changes in risk factors can occur rapidly and unpredictably. By incorporating live data, our ML project can adjust to these changes in real-time, resulting in more accurate premium predictions and ultimately, better risk management. Additionally, the use of live data can enable us to identify emerging trends and patterns that may not have been evident in our collected dataset. Working with live data sets also poses its own unique challenges, such as ensuring data quality and managing the sheer volume of information. However, by data management tools and techniques, we can overcome these obstacles and tap into the full potential of live data. In sum, the future enhancement of our insurance premium prediction ML project lies in the use of live data sets, which offer a more dynamic and responsive approach to risk assessment and management.

Another area for future enhancement is the integration of external data sources, such as weather data or demographic data, which could help to identify correlations and patterns that may not be immediately apparent from our existing data sets. Additionally, the use of predictive analytics could allow us to forecast future trends and anticipate changes in risk factors, enabling us to proactively adjust our premium pricing models and risk management strategies.

The implementation of explainable AI (XAI) techniques has the potential to improve our understanding and interpretation of the factors that drive premium predictions. This, in turn, can help us build trust with our customers and regulators. With XAI, we can better explain how our predictive models arrive at their conclusions, providing more transparency and clarity in our decision-making process. These techniques enable data scientists to identify the most important features in the data, visualize how the model makes decisions, and explain individual predictions to stakeholders.

APPENDIX

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

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

import warnings
warnings.filterwarnings('ignore')
sns.set_theme(style="darkgrid")

df = pd.read_csv('/Users/subashkrishnan/Desktop/Insurance.csv')
df.head()

df.info()

df.describe()

df.isnull().sum()

df.head(2)

df.shape
```

```
le_sex = LabelEncoder()
le_smoker = LabelEncoder()
le_region=LabelEncoder()

df['Gender']=le_sex.fit_transform(df.sex)
df['Smoker']=le_smoker.fit_transform(df.smoker)
df['Region']=le_region.fit_transform(df.region)

df.head()

df.drop(["sex",'smoker','region'],axis=1,inplace=True)

df.rename(columns={"age":'Age',
                   'bmi':"BMI",
                   'children' : "Children",
                   'charges' : "Charges"
                   },inplace=True)

df = df[['Age','Gender','BMI','Children','Smoker','Region','Charges']]

df.head()

df.Charges.mean()
df.Charges.max()

df.Charges.min()
```

```
sns.distplot(df.Charges)

df.Charges

df['Target'] = df['Charges'].apply(int)

df.head(10)

df.corr()

sns.heatmap(df.corr(),annot=True)

df.Target.median()

df.Target.mean()

df.Target.max()

df.Target.min()

df['SplitTargetMean']=df['Target'].apply(lambda x:1 if x>=9382 else 0)

df['SplitTargetMedian']=df['Target'].apply(lambda x:1 if x>=13269 else 0)

df.head()
```

```
df.SplitTargetMean.value_counts()

df.SplitTargetMedian.value_counts()

df.columns

sns.boxenplot(df.Target)

sns.boxenplot(df.BMI)
df.drop("SplitTargetMean",axis=1,inplace=True)

df.head()

#Age
sns.distplot(df.Age)

df.Age.min(), df.Age.mean(),df.Age.median(), df.Age.max()

age_less_20 = df[(df['Age']<=20)]
age_less_20.shape[0]

age_less_20.head()

age_less_20.SplitTargetMedian.value_counts()
```

```
age_less_20.Smoker.value_counts()

var1 =
age_less_20[(age_less_20['Smoker']==1)&(age_less_20['SplitTargetMedian']==1)]

len(var1)

age_less_40 = df[(df['Age']<=40)]
age_less_40.shape[0]

age_less_40.SplitTargetMedian.value_counts()

age_less_40.Smoker.value_counts()

var2 =
age_less_40[(age_less_40['Smoker']==1)&(age_less_40['SplitTargetMedian']==1)]
len(var2)

age_greater_40 = df[(df['Age']>40)]
age_greater_40.shape[0]

age_greater_40.SplitTargetMedian.value_counts()

age_greater_40.Smoker.value_counts()
```

```
var3 =
age_greater_40[(age_greater_40['Smoker']==1)&(age_greater_40['SplitTargetMed
ian']==1)]

len(var3)

age_greater_60 = df[(df['Age']>60)]
age_greater_60.shape[0]

age_greater_60.SplitTargetMedian.value_counts()

age_greater_60.Smoker.value_counts()

var4 =
age_greater_60[(age_greater_60['Smoker']==1)&(age_greater_60['SplitTargetMed
ian']==1)]

len(var4)

a=age_less_20.Gender.value_counts()
print(a)
print("----------------------------")
b=age_less_40.Gender.value_counts()
print(b)
print("----------------------------")
c=age_greater_40.Gender.value_counts()
```

```
print(c)
print("----------------------------")
d=age_greater_60.Gender.value_counts()
print(d)
print("----------------------------")

plt.figure(figsize=(10,2))
plt.subplot(1,4,1)
plt.title("Age < 20")
sns.countplot(a)
plt.subplot(1,4,2)
plt.title("Age < 40")
sns.countplot(b)
plt.subplot(1,4,3)
plt.title("Age > 40")
sns.countplot(c)
plt.subplot(1,4,4)
plt.title("Age > 60")
sns.countplot(d)

#Gender - 0 Female
#Gender - 1 Male
sns.barplot(x='Gender',y='Smoker',hue='Region',data=df)
df.Region.unique()

df1 = pd.read_csv('/Users/subashkrishnan/Desktop/Insurance.csv')
df1.head(2)
```

```
lst1=[]
for i in df1.region.unique():
    lst1.append(i)
    lst1.append(df1[(df1['region']==i)&df['SplitTargetMedian']==1].shape[0])
print(lst1)
print(df1['region'].value_counts())

dict1={
    'SouthWest':85, 'SouthEast':130, 'NorthWest':96, 'NorthEast':109
}

labels=[]
sizes=[]
for x,y in dict1.items():
    labels.append(x)
    sizes.append(y)
plt.figure(figsize=(8,8))
plt.pie(sizes,labels=labels,autopct='%1.0f%%')
plt.axis('equal')

sns.kdeplot(df.BMI)

df.head(10)
df.BMI.mean(), df.BMI.median()

bmi_greater_stm1 = df[(df['BMI']>30.4)&(df['SplitTargetMedian']==1)]
```

```
bmi_greater_stm1

bmi_less_stm1 = df[(df['BMI']<=30.4)&(df['SplitTargetMedian']==1)]

bmi_less_stm1

sns.barplot(y='Age',x='SplitTargetMedian',hue='Gender',data=df)


sns.barplot(y='BMI',x='SplitTargetMedian',hue='Gender',data=df)

df.head(2)

x=df.drop(['Charges','SplitTargetMedian'],axis=1)
y=df.Charges
y=y.astype('int')
```

REFERENCES

[1] Pesantez-Narvaez, J., Guillen, M., & Alcañiz, M. (2019). Predicting motor insurance claims using telematics data—XGBoost versus logistic regression. Risks, 7(2), 70

[2] Gupta, S., & Tripathi, P. (2016, February). An emerging trend of big data analytics with health insurance in India. In 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH) (pp. 64-69). IEEE.

[3] C. C. a. A. Semanskee, "Analysis of UnitedHealth Group's Premiums and Participation in ACA Marketplaces," 2016.

[4] Prasad, K.S., Reddy, N.C.S. & Puneeth, B.N. A Framework for Diagnosing Kidney Disease in Diabetes Patients Using Classification Algorithms. SN COMPUT. SCI. 1, 101 (2020)

[5] Varun K L Srivastava, N. Chandra Sekhar Reddy, Dr. Anubha Shrivastava, "An Effective Code Metrics for Evaluation of Protected Parameters in Database Applications", International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.1.3, 2019.

[6] Singh, Anshy, Shashi Shekhar, and Anand Singh Jalal. "Semantic based image retrieval using multi-agent model by searching and filtering replicated web images." 2012 World Congress on Information and Communication Technologies. IEEE, 2012.

[7] R. Nafeena Abdul Munaf, K. Karthikeyan, S. Joe Patrick Gnanaraj, T.A. Sivakumar, N. Muthukumaran, "An ML Approach for Household Power Consumption," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022, pp. 784-791.

[8] N. Muthukumaran, N. R. G. Prasath and R. Kabilan, "Driver Sleepiness Detection Using Deep Learning Convolution Neural Network Classifier," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, India, 2019, pp. 386-390.

[9] Mccord, Michael, and M Chuah. 2011. ―Spam Detection on Twitter Using Traditional Classifiers.‖ In International Conference on Autonomic and Trusted Computing, 175–86. Springer.

[10] Sturm, Roland, Ruopeng An, Josiase Maroba, and Deepak Patel. The S.P.Panimalar, R.T. Subhalakshmi (2021). "Privacy and Security aspects of COVID 19 Image in Big Data Era", in International Journal of Innovative Research in Scicence , Engineering and Technology, vol. 10, Issue 1pp 529-537

[11] Singh, R., Ayyar, M. P., Pavan, T. S., Gosain, S., & Shah, R. R. (2019, September). Automating Car Insurance Claims Using Deep Learning Techniques. In 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM) (pp. 199-207). IEEE.

[12] Yerpude, P., Gudur, V.: Predictive modelling of crime dataset using data mining. Int. J. Data Min. Knowl. Manag. Process (IJDKP) 7(4) (2017)

[13] D; Manigandan S. K; Deepa J. Health Insurance Cost Prediction using Machine Learning Algorithms. Ramya
2022 (ICECAA)

[14] Chaparala Jyothsna; K. Srinivas; Bandi Bhargavi; Akuri Eswar Sravanth; Atmuri Trinadh Kumar; J.N.V.R. Swarup Kumar
Health Insurance Premium Prediction using XGboost Regressor
2022 (ICAAIC)

[15] Josh Jia-Ching Ying; Chi-Kai Chang; Yen-Ting Chang . Applying a Genetic Algorithm to Determine Premium Rate of Occupational Accident Insurance
2020 (ICMLC)

[16] N Venkata Sailaja; Mounika Karakavalasa; Meera Katkam; Devipriya M; Sreeja M; D N Vasundhara. Hybrid Regression Model for Medical Insurance Cost Prediction and Recommendation .2021 (ICISSGT)

[17].Tallal Omar; Mohamed Zohdy; Julian Rrushi. . Clustering Application for Data-Driven Prediction of Health Insurance Premiums for People of Different Ages
2021(ICCE)

[18]. Shruti Aggarwal; Anmol. Health Insurance Amount Prediction Using Supervised Learning 2022 (ICTACS)