Applying Machine Learning-Based Regression Models in the Prediction of Health Insurance Premium

Njoh Nji Mukwa

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
February 2024
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Prof. Dr. Nazım Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Asst. Prof. Dr. Mehmet Ali Tut
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Arif Akkeleş

2. Asst. Prof. Dr. Hüseyin Lort

3. Asst. Prof. Dr. Mehmet Ali Tut

ABSTRACT

It is no doubt that the insurance industry is no stranger to data driven decision making. The field of health insurance has seen profound transformation in recent times driven by technological advancement, data proliferation and evolved healthcare dynamics. Traditional methods for predicting health insurances premiums face several different challenges which can result in inaccurate pricing, adverse selection and suboptimal risk assessment. Some of these limitations including but not restricted to limited data utilization, static models and inefficiency in underwriting.

This thesis project seeks to investigate comprehensively how machine learning based regression models and techniques, including linear regression, polynomial regression and XGBoost regression can be used in insurance to make predictions on health insurance premiums. Using a diverse historic US health insurance dataset gotten from Kaggle containing client insurance charges, demography information, lifestyle factors, these models meticulously tuned, trained, and evaluated. The study does in-depth examination of the methodologies, including exploratory data analysis, feature selection and engineering, hyperparameter optimization, and model evaluation, to determine the predictive accuracy of each model.

Keywords: Health Insurance, Machine Learning, Statistics, Linear Regression, Polynomial, XGBroost, ML Models, Python.

ÖZ

Sigorta endüstrisinin veri odaklı kararlar almaya yabancı olmadığı şüphesizdir. Sağlık sigortası alanında, teknolojik ilerleme, veri çoğalması ve gelişmiş sağlık dinamikleri tarafından yönlendirilen köklü bir dönüşüm yaşanmıştır. Geleneksel yöntemlerle sağlık sigortası primlerini tahmin etme, doğru fiyatlandırmayla, olumsuz seçimle ve altoptimal risk değerlendirmesiyle sonuçlanabilen çeşitli zorluklarla karşılaşmaktadır. Bu sınırlamalar arasında, ancak bunlarla sınırlı olmamak kaydıyla, sınırlı veri kullanımı, statik modeller ve underwritingdeki verimsizlik bulunmaktadır.

Bu tez projesi, sağlık sigortası primlerine ilişkin tahminlerde bulunmak için makine öğrenmesi tabanlı regresyon modelleri ve tekniklerin, lineer regresyon, polinom regresyon ve XGBoost regresyonunun kapsamlı bir şekilde nasıl kullanılabileceğini araştırmayı amaçlamaktadır. Kaggle'dan alınan çeşitli tarihli bir ABD sağlık sigortası veri setini kullanarak, müşteri sigorta ücretleri, demografik bilgiler, yaşam tarzı faktörleri içeren bu modeller özenle ayarlanmış, eğitilmiş ve değerlendirilmiştir. Çalışma, keşifsel veri analizi, özellik seçimi ve mühendisliği, hiperparametre optimizasyonu ve model değerlendirmeyi içeren yöntemleri derinlemesine incelemekte ve her modelin tahmin doğruluğunu belirlemektedir.

Anahtar Kelimeler: Sağlık Sigortası, Makine Öğrenimi, İstatistik, Lineer Regresyon, Polinom, XGBoost, ML Modeller, Python.

DEDICATION

To my beloved Blood Family and Church Family.

ACKNOWLEDGEMENT

Above all, I give thanks to God Almighty and Jesus Christ, my Lord and Saviour, for the blessings, grace, and strength that have sustained me throughout these years.

I extend special thanks to Prof. Mehmet Ali Tut, my dedicated supervisor, for his invaluable efforts and unwavering guidance throughout the course of this thesis.

Heartfelt appreciation to all my esteemed professors and instructors in the Department of Mathematics, Eastern Mediterranean University (EMU) for their expertise and continuous support.

I express special gratitude to my beloved mother, Akwi Salome Njoh, and my fathers in the family for their unwavering love and financial support.

I would like to also convey special thanks to all my siblings, with a special mention to my twin sister Tengu Njoh and also to Chief Dr. Tah Tangwan, for their constant support and motivation.

I am graciously thankful to my spiritual family, the Jesus People Committee, for all your prayers and love.

Finally, I extend immense gratitude to the wonderful people of Cyprus and Eastern Mediterranean University (EMU) for providing me with a memorable experience to cherish for a lifetime.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 General

The integration of Artificial Intelligence (AI) in the field of medical science within the last few decades has drastically involved with application of data engineering and analytics methods along with advanced machine learning algorithms to identify patterns within vast medical datasets and produce insightful predictions and outcomes. As medicine increasingly depends on different types of data variables like imaging, histopathological, and biochemical data, the amount of available information is rapidly growing, creating opportunities for machine learning advancements. AI and ML systems are poised to play a more significant role in healthcare, handling tasks ranging from research and data organization to pattern recognition and predictive analytics. These systems have the potential to revolutionize medical diagnostics and even contribute to treatment decisions.

Although the health sector has experienced rapid technological advancement and innovation, nonetheless it still remains vulnerable serious uncertainties, dangers and risks. Individuals, are all exposed to different types of threats, which can vary in their severity and nature. These potential hazards encompass accidents, illnesses, and even mortality. Overall, people turn to experience of happiness and productivity when these risks are could be avoided or even eliminated. However, because risk cannot totally be

avoided or eliminated the financial industry has developed a number of financial products to allow individuals and corporations mitigate these risks. Amongst which is insurance; a policy, in which policyholders receive financial protection or reimbursement against losses from an insurer. Policies which cover for medical expenses are known as health insurance. After purchasing a health insurance, the policy holder is required to pay a periodic amount to the insurer called premium.

How much a health insurance cost can be influences by several factors and it is determined by the process called underwriting. Insurance underwriting is carried out by actuaries or trained financial expects called underwriters who assess and evaluate the risks of insuring an individual or asset using several underwriting tools and processes to analyze factors involved to set price and premium for insurance policies. Underwriters rely on computerized probabilistic methods and actuarial data to assess the potential risk involved with insuring an individual or asset and based on this assessment they set the premium price for the insurance policy. The aim is to ensure that persons or assets with high-risk pay higher cost and premiums to maintain an equivalent level of protection compared to those with considerably lower-risk.

AI and ML play important role in the health insurance sector. Here are some:

- By analyzing large datasets, AI can identify patterns and generate insights, aiding in the early detection of fraud, abuse, waste management, and claims utilization, resulting in potential cost savings.
- Using machine learning tools and techniques, AI enables dynamic data analysis of health insurer data and electronic health records, supporting better decision-making in networks, claims, pricing, and risk management.

- AI facilitates more efficient claims adjudication and prior authorization workflows, delivering faster and more accurate predictive analytic reports.
- Robotic process automation powered by AI streamlines administrative processes, reducing operational expenses and optimizing resources for technical functions.
- AI tools can predict diseases and develop personalized treatments, leading to improved health outcomes and cost reductions.
- Chatbots enhance member engagement, customer service, triage services, and appointment scheduling, saving administrative costs.
- AI promotes remote and telehealth services utilization for triage, primary care, disease management, and diagnostic services, increasing accessibility and cost-effectiveness.

A data-driven and predictive modelling approach may shift healthcare from illness management to wellness management, improving diagnosis and treatment planning while reducing waste and overutilization, ultimately leading to better health outcomes and claims cost reduction. This approach can enhance insurers' claims ratios and competitiveness.

1.2 Challenges of traditional methods

For several decades, traditional methods of predicting health insurance premiums faced several challenges, which has impacted the accuracy of pricing and financial sustainability. Some of these challenges include:

1. Limited Data Utilization: Traditional methods rely on limited historical claims data, which may not capture the full spectrum of variables affecting healthcare costs. This can result in underestimation or overestimation of premiums. [20]

2. Static Models: Many traditional models use static, actuarial approaches that do not adapt well to changing dynamics of healthcare products. They fail to account for emerging treatments, shifts in demographics, or advancements in medical technology. [22]
3. Risk Pooling Issues: Traditional models often struggle with adverse selection, where healthier individuals may opt for lower coverage, leaving a riskier, costlier pool. This can lead to higher premiums for the sicker population. [20]
4. Inefficiencies in Underwriting: Manual underwriting processes can be time-consuming and prone to errors. Traditional underwriting doesn't harness advanced data analytics, leading to suboptimal risk assessment. [21]
5. Lack of Personalization: Traditional models often provide one-size-fits-all pricing, neglecting individual health behaviors and risk profiles. Personalization is limited, and this can result in unfair pricing. [21]
6. Regulatory Challenges: Regulatory constraints can limit innovation in premium prediction models, making it difficult for insurers to adapt to changing market conditions efficiently. [21]
7. Data Privacy and Security: Handling sensitive health data in traditional models raises concerns about data privacy and security, which can impact insurers' ability to leverage big data for better predictions. [22]
8. Rate Regulation: Government-imposed rate regulations can limit insurers' ability to set premiums based on actuarial principles, potentially leading to market distortions. [22]

Addressing these challenges involves integrating advanced data analytics, artificial intelligence, and improved regulatory frameworks. This is crucial for developing more accurate and equitable health insurance premium prediction models.

1.3 Design of The Study

This design of this study involves the analysis of historical health insurance dataset from of an undisclosed region. The scope of this analysis will basically be applying machine learning techniques to process the data and running different regression models on the data. The accuracy of the models is then compared based on performance matrix.

Firstly, we will explore the mathematical background of the different regression models implemented. Next, fundamental machine learning concepts as it pertains to this study, followed by initiating the machine learning process beginning with exploratory data analysis and data preprocessing to make sense of the dataset.

With very large dataset with several attributes, it is difficult to use the traditional manual or computational analyses and avoid errors in the results. This is due to the fact that processed are many and not automated or streamlined to avoid leakages, errors and inefficiencies. Therefore, it is better to use machine learning for analysis using the best popular programing tool know as Python distributed by Anaconda. The line of codes is carried out in Jupiter notebook in Anaconda.

Chapter 2

MATHEMATICAL BACKGROUNDS

This study employs regression analysis, implemented in Python on the Jupiter notebook interface, to develop and train machine learning models to predict health insurance premiums. Regression analysis, a statistical method for investigating relationships between variables, finding broad application in fields like economics, engineering, biology, and insurance. Its predictive power and ability to explore variable relationships make it suitable for this research.

In regression analysis, variables are characteristics taking on different values. In a simple model, the independent variable (denoted as 'x') predicts the dependent variable ('y'). Multiple regression involves more than two variables, with independent variables (x_1, x_2, x_3, etc.) predicting a single dependent variable (y). A scatter diagram, plotting values of x against y, visually depicts the relationship, whether increasing, decreasing, or none. The correlation coefficient, ranging from -1 to 1, quantifies this relationship. A coefficient of 1 indicates a perfect positive relationship, -1 represents a perfect negative relationship, and 0 signifies no relationship. Recognizing patterns like clusters adds valuable insights [2].

There are different regression analysis techniques, and the choice of a particular method to employ will depend on a number of factors. Some of these factors include the type of variables involved, the observed pattern of the scatter plot, the number and

properties of the independent variables involved. The different regression techniques involve with is study are examined in the subsequent sections:

2.1 Linear Regression

If y and x have some degree of linear relationship, then the corresponding straight-line relationship is expressed as

$$Y = \beta_0 + \beta_1 x + \varepsilon \quad (1)$$

Equation above is called a simple linear regression model, where β_0 is *y* intercept and $\beta_1$is the slope. The slope β_1, represents how the average of *y* changes for one unit increase in *x* [1]. In practical scenarios, the variables rarely display a perfect linear relationship, and the error term ε accommodates the deference between the observed *y* value and the linear prediction ($\beta_0 + \beta_1 x$). Essentially, the error term (residual error or unexplained variation), ε is a random variable representing the model's inability to precisely fit the data. This value is determined by subtracting the predicted values (obtained from the regression equation) from the actual dependent variable values:

$$\varepsilon_i = y_i - (\beta_0 + \beta_1 x_i) \quad (2)$$

The sum of squared errors (SSE) is the common measure for assessing the overall error in the regression model. It is computed by summing the squared error terms for each individual data point. The goal of the regression model is to minimize the SSE, which indicates a better fit between values predicted and actual values. R-squared (R^2), or the coefficient of determination, is a statistical measure that quantifies the proportion of variance in the dependent variable explained by the independent variable(s). It indicates the goodness of fit in the regression model, with a higher R^2 indicating a better fit. R^2 can take values between 0 and 1where 0 signifies no explanatory power, and 1 denotes perfect explanation of variance. R^2 is calculated by [1].

R^2 = Explained variation / Total variation

R^2 can also be interpreted as the proportion of the sum of squared errors (SSE) that is explained by the regression model.

$$R^2 = 1 - \left(\frac{SSE}{SST}\right) \tag{3}$$

Where SSE is the sum of squared errors, and SST is the total sum of squares, which represents the total variation in the dependent variable. SST is calculated as:

$$SST = \sum(y_i - \bar{y})^2 \tag{4}$$

Where y_i is the observed value of the dependent variable for each data point, and $\bar{y}$ is the mean of the dependent variable [1].

Regression models hold validity only within the range where the independent variable has observable data. For example, if data (y and x) were collected, and x can take values in the interval $x_1<x<x_2$, the linear regression model serves as an effective approximation within this range. However, for regressor values outside this interval ($x<x_1$ *and* $x>x_2$), the model's performance diminishes, yielding poor or in most cases meaningless results [1][7].

In general, given k independent variables, $x_1, x_2 \ldots x_k$, influencing the response variable y, the relationship is expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \tag{5}$$

This equation known as a multiple linear regression model, involving more than one regressor. The descriptor 'linear' pertains to the linearity in parameters $\beta_0, \beta_1, \ldots, \beta_k$. It emphasizes linearity not just in cases where y has a linear relationship with the x variables, but also in situations where some x exhibits a nonlinear relationship [1].

The β, denotes the population regression coefficient while b is used as the sample regression coefficient. In theory, β is used for definition, but in practices b is used since the regression analysis is conducted on samples. The linear regression model for a sample data is given by:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k + \varepsilon \quad (6)$$

2.1.1 Ordinary Least Square Estimation of Parameters

The OLS estimation uses calculus to derive values for the parameters β_0 and β_1 that minimize the residual sum of squares (RSS or SSE). RSS is given by [1]:

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} ((\beta_0 + \beta_1 x_i) - y_i)^2 \quad (7)$$

Taking partial derivatives of the RSS with respect to β_0 and β_1:

$$\frac{\partial RSS}{\partial \beta_0} = 2 \sum_{i=1}^{n} \left((\beta_0 + \beta_1 x_i) - y_i\right)$$

and

$$\frac{\partial RSS}{\partial \beta_1} = 2 \sum_{i=1}^{n} x_i((\beta_0 + \beta_1 x_i) - y_i)$$

To get the minima, set these derivatives equal to zero, yielding the normal equations:

$$\sum y_i = n\beta_0 + \beta_1 \sum x_i$$

$$\sum y_i x_i = \beta_0 \sum x_i + \beta_1 \sum x_i^2$$

where n is the sample size.

Simultaneously solving these two equations gives the estimates for the regression coefficients β_0 and β_1 that minimize the RSS:

$$\beta_1 = \frac{n(\sum xy) - (\sum x)(\sum y)}{(n\sum x^2) - (\sum x)^2} = \frac{\sum \left(x - \bar{x}\right)\left(y - \bar{y}\right)}{\sum \left(x - \bar{x}\right)^2} = \frac{S_{xy}}{S_{xx}} \quad (8)$$

Where $S_{xx} = \sum(x - \bar{x})^2$

$$\beta_0 = \frac{\sum y - b_1 \sum x}{n} = \bar{y} - b_1 \bar{x} \tag{9}$$

Therefore, the derivatives used in the OLS method are the partial derivatives of the RSS with respect to β_0 and β_1 [1].

In the case of two independent variables x_1 and x_2, the parameters β_0, β_1 and β_2 can be calculated manually by the following formula;

$$\beta_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)} \tag{10}$$

$$\beta_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)} \tag{11}$$

$$\beta_0 = \bar{Y} - b_1 \bar{X_1} - b_2 \bar{X_2} \tag{12}$$

Alternative, a matrix can be applied to the systems of equations to generalize the solution for any number of input variables. In general, the linear regression equation is defined as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \tag{13}$$

Then, partially differentiating the coefficients of the RSS, the following equations are gotten:

$$n\beta_0 + \beta_1 \sum x_1 + \beta_2 \sum x_2 \quad + \cdots + \beta_k \sum x_k = \sum y$$

$$\beta_0 \sum x_1 + \beta_1 \sum x_1^2 + \beta_2 \sum x_1 x_2 + \cdots + \beta_3 \sum x_k = \sum x_1$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\beta_0 \sum x_k + \beta_1 \sum x_1 x_k + \beta_2 \sum x_2 x_k + \cdots + \beta_k \sum x_k^2 = \sum x_k y$$

The generalized matrix form is:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{2i} & \cdots & \sum_{i=1}^{n} x_{ki} \\ \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{1i}^2 & \sum_{i=1}^{n} x_{1i}x_{2i} & \cdots & \sum_{i=1}^{n} x_{1i}x_{ki} \\ \sum_{i=1}^{n} x_{2i} & \sum_{i=1}^{n} x_{1i}x_{2i} & \sum_{i=1}^{n} x_{2i}^2 & \cdots & \sum_{i=1}^{n} x_{2i}x_{ki} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_{1i}x_{ki} & \sum_{i=1}^{n} x_{2i}x_{ki} & \cdots & \sum_{i=1}^{n} x_{ki}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_{1i} \\ \sum_{i=1}^{n} y_i x_{2i} \\ \vdots \\ \sum_{i=1}^{n} y_i x_{ki} \end{bmatrix} \quad (14)$$

2.2 Polynomial Regression

A model is said to be polynomial when its input variable(s) has higher degree or order above 1. The models

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

and

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \quad (15)$$

are both polynomial model of second degree with one and two variables, respectively. Here are a few important things considered when fitting polynomial to a single independent variable:

- The degree of the polynomial model is kept as low as possible, preferably first or second-order. Higher-degree polynomials can lead to overfitting the data. Overfitting occurs when the model fits the training data too well and has poor performance on new data [5].
- To choose the degree of the polynomial model, there are two techniques: forward selection and backward elimination. In forward selection, models are fit in increasing degree until the *t* test for the highest order is no longer significant. In the case of backward elimination, the highest order model is fit and terms are

deleted until the highest order remaining term is *t* significant. It is worth noting that both methods will not necessarily end in the same regression model [2].

- In polynomial regression, the X-matrix can become ill-conditioned as the order increases, meaning that the $(X'X)^{-1}$ used to estimate parameters may not be accurate. This can result in considerable errors in parameter estimates. Ill-conditioning increases when the values of x are in a narrow range, leading to multicollinearity between columns of the X-matrix [5].
- Extrapolation with polynomial models is typically reliable and meaningful within a specific range in the original data. Beyond this range, the responds variable may turn at odds with the natural behaviour of the system [2].
- A hierarchical model in polynomial regression contains terms that are in hierarchy, i.e.; x, x^2, x^3, etc. This property is expected in all polynomial models because only hierarchical models are invariant under linear transformation. Nevertheless, in some cases a model may not need to be hierarchical, such as in the case of a two-factor interaction, where one term would not have been included model based on its statistical significance [5].
- Multiple variable polynomial regression involves two or more independent variables and it is similar with one variable except that in addition to the parameters that capture the polynomial effect of each regressor, we also have parameters which captures the interaction effect between parameters. The regression function is called a response surface [2].

2.2.1 Estimating the Parameters of Polynomial Regression

The OLS method is also applicable to estimate all the coefficient of the polynomial regression, if and only if it is a hierarchical model [5]. Consider the following one-input variable polynomial equation of order k;

$$y_i = \beta_0 + \beta_0 x_i + \beta_1 x_i^2 + \cdots + \beta_1 x_i^k + \varepsilon \tag{16}$$

$$i = 1,2,..n$$

Deriving the matrix for the estimation of parameters of higher degree polynomial, given the quadratic:

$$y_i = \beta_0 + \beta_1 x + \beta_2 x^2$$

Sum of squares error:

$$RSS(\beta_0, \beta_1, \beta_2) = \sum_{i=1}^{n} (\beta_0 + \beta_1 x_i + \beta_2 x_i^2 - y_i)^2 \tag{17}$$

The partial derivative of each of the three coefficients is given by:

For β_0,

$$\frac{\partial RSS(\beta_0, \beta_1, \beta_2)}{\partial \beta_0} = \sum_{i=1}^{n} 2(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 - y_i)$$

To get the minima, equate to 0:

$$0 = n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i + \beta_2 \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} y_i = n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i + \beta_2 \sum_{i=1}^{n} x_i^2 \tag{18}$$

For β_1,

$$\frac{\partial RSS(\beta_0, \beta_1, \beta_2)}{\partial \beta_1} = \sum_{i=1}^{n} 2(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 - y_i) x_i$$

$$\sum_{i=1}^{n} y_i x_i = \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 + \beta_2 \sum_{i=1}^{n} x_i^3 \tag{19}$$

For β_2,

$$\frac{\partial RRS(\beta_0, \beta_1, \beta_2)}{\partial \beta_2} = \sum_{i=1}^{n} 2(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 - y_i) x_i^2$$

$$\sum_{i=1}^{n} y_i x_i^2 = \beta_0 \sum_{i=1}^{n} x_i^2 + \beta_1 \sum_{i=1}^{n} x_i^3 + \beta_2 \sum_{i=1}^{n} x_i^4 \tag{20}$$

The above equations derived can be generalized in a matrix, so that if *k* is a given degree of the polynomial, and *n* is the number none data points, then we get;

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \cdots & \sum_{i=1}^{n} x_i^{k+1} \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 & \cdots & \sum_{i=1}^{n} x_i^{k+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{\square\square+1} & \sum_{i=1}^{n} x_i^{k+2} & \cdots & \sum_{i=1}^{n} x_i^{2k} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i x_i^2 \\ \vdots \\ \sum_{i=1}^{n} y_i x_i^k \end{bmatrix} \tag{21}$$

When solving an overdetermined system, a residual function is first created. This function involves summing the squared residuals, which then produces a parabola or paraboloid. The coefficients are then determined by locating the minimum point of the parabola/paraboloid using partial derivatives [1].

In polynomial interpolation, the goal is to construct a polynomial of degree (N-1) that passes through N given points that can be used to predict for any output of *x*. Extrapolation on the other hand is when *x* is out of the range of values in the training set. A good illustration of this is in time series predictions where we have data up to

today and we want to forecast the future outcome. Noise ϵ, is also added to this function to explains the impact of hidden variables not observed [9]

2.3 Gradient Boosted Tree Regression

This is a very sophisticated machine learning model that allows for two distinct training approaches. The first approach starts with a complex model and then fitting its parameters (such as with neural networks), and subsequently fine-tunes its parameters. The second approach is iterative, where each step trains a simple model (boosting) like decision trees, and combines them to create a more refined and accurate model. These initial simple models are known as weak classifiers or base learners. This technique leverages the concept of boosting, in which subsequent models rectify errors made by their predecessors, using it to ultimately enhancing overall predictive performance. Gradient Boosted Trees performs works well in capturing complex interactions and non-linear patterns within the data, making them well-suited for tasks like regression [19].

For the sake of simplicity, consider a regression problem with data points $D_N = \{(x_1, y_1)...,(x_N, y_N)\}$ where x_i are explanatory values and y_i are response values. The aim here is to find the prediction function, $F(x)$ using training set, such as

$$min \sum_{(x,y)\in T}(F(x) - y)^2 \tag{22}$$

ere $T = \{(x_1, y_1)\ldots,(x_N, y_N)\}$ is the test set. In boosting the goal is to construct a function $f(x)$ iteratively. It is assumed that the function $f(x)$ expresses the sum of other simple functions or weak learners $h_m(x)$

$$F(x) = \sum_{(m=1)}^{M} h_m(x) \tag{23}$$

where $h_m(x)$ is a decision tree.

The way gradient boosted tree works is the training set is taken as the input and M is set as the maximum number of iterations. We compute the mean of all the target values y_i in the first step, which become the initial approximation of function, $F(x)$.

$$F_0(x) = \frac{1}{N}\sum_{(i=1)}^{N} y_i \tag{24}$$

In the following step, for each iteration, m=1, 2..., M, compute the residuals $\widehat{y_i} = y_i - F_{m-1}(x_i)$. Then fit a decision tree $h_m(x)$ to the target $\widehat{y_i}$ by using an auxiliary training set $\{(x_1, \widehat{y_1}), \ldots, (x_N, \widehat{y_N})\}$. Finally, we add this decision trees to improve the overall model performance

$$F_m(x) = F_{m-1}(x) + \lambda_m h_m(x) \tag{25}$$

where λ_m is regularization or the learning rate derived by line search with the aim of reducing the loss function.

2.4 Logistic Regression

Logistic Regression, although not specifically implemented in this study, is another vital regression analysis technique; also referred to as covariates and a categorical dependent or response variable [3]. There are two types of logistic regression models; binary regression and multinomial logistic regression. Binary logistic regression is mainly used when the dependent variable is binary or dichotomous, example: 1 or 0, true or false, yes or no, etc, and the independent variable is either categorical and/or continuous. In the case where the dependent variable has more than two categories, then it is a multinomial logistic regression [3][6]. In the logistic regression, the relationship between the target variable and the independent variable can be represented by a sigmoid curve [2]. The analysis is based on the probabilities, odds

and odds ratio. In the case of binary logistic model, the odds of an event are the ratio of the probability that an event will occur to the probability it will not occur. Given the response variable Y and the explanatory variable X, let $\pi(x) = P(Y = 1|X = x) = 1 - P(Y = 0|X = x)$, the logistic regression model is given such that [6]:

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta X)} \tag{26}$$

Also, the logit (log odds) has a linear relationship given by [6]

$$logit\big(\pi(x)\big) = \ln(odds) = \ln\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \beta\chi \tag{27}$$

In the case of multiple explanatory variables, which can be a combination of quantitative and qualitative, the regression function for $\pi(x) = P(Y = 1)$ at values $x = (x_1, \ldots, x_k)$ of k the regression model will be [6]:

$$logit\,[\pi(x)] = \ln\left(\frac{p}{1 - p}\right) = \alpha + \beta_1\chi_1 + \cdots + \beta_k\chi_k \tag{28}$$

Where p is the probability that y = 1, for k explanatory variables. The left side of the equation is known as logit, or the log-odds. There is no error term in this model. The alternative equation for finding $\pi(x)$ is given by [6]:

$$\pi(x) = \frac{\exp(\alpha + \beta_1\chi_1 + \cdots + \beta_k\chi_k)}{1 + \exp(\alpha + \beta_1\chi_1 + \cdots + \beta_k\chi_k)} = \frac{1}{1 + e^{-(\alpha + \beta_1\chi_1 + \cdots + \beta_k\chi_k)}} \tag{29}$$

Given quantitative predictors, indicator variables are used for each category which are a suitable format that can be incorporated into the regression equation [6]. The coefficient β_j represents the impact of χ_j on the log odds Y =1, while taking into account the other predictors χ_k. For instance, $\exp(\beta_j)$ is the factor by which the odd of a 1-unit increase in χ_j, while keeping fixed the other predictors χ_k [6].

Logistic regression technique is used when the size of data is large and there is almost equal occurrence of values to come in target variable. There should also be no multicollinearity [6].

2.5 Pros and Cons of Regression Analysis

Pros:

- Regression works very well when the data has a relationship; linear or quadratic.
- It is easy to implement, interpret and train efficiently.
- It can effectively manage overfitting by making use of either dimensionality reduction techniques, regularization or cross validation.
- Extrapolation can be made even beyond the given dataset.

Cons:

- It relies on the assumption there is a relationship between the response and explanatory variables.
- It can be sensitive to noise and may overfit in case of high correlation.
- Regression models are very sensitive to outliers which can significantly affect the performance.
- It is exposed to multicollinearity which can affect the stability of the model. This should be avoided.

Regression analysis on large data set is mainly carried out using computer software, such as Python, R, MATLAB, SPSS, EViews, etc. For this research Python is employed.

Chapter 3

MACHINE LEARNING FUNDERMENTALS

Machine Learning (ML) is a type of artificial intelligence which employs algorithms and statistical techniques to processes and identify patterns in raw data. The primary objective of ML is to enable computer systems to learn and improve from experience without requiring explicit programming or intervention from humans [9].

Python is the most popular programming language for machine learning because it is easy to use, open source, and has a wide array of libraries and tools for data manipulation, modeling, and evaluation. Machine learning with Python involves using the Python programming language and associated libraries to build and train models that can learn patterns in data, make predictions, and make decisions. The most popularly used Python libraries for machine learning which will also be employed in this thesis include; NumPy, Pandas, Scikit-learn, TensorFlow [10].

3.1 Machine Learning Libraries in Python

- NumPy: The name is the short for numerical Python and the library is made up of array objects. Using NumPy, we are able to carry out the following operations in Python: mathematical and logical operations on array, Fourier transformation, and last but not the least, operations related with linear algebra [10].
- Pandas: This is another very important library for machine learning. It is primarily used for handling data; data manipulation, processing and analysis. Using Pandas

for data processing we can effectively load, prepare, manipulate, model, and perform analyses on the data [10].

- Matplotlib: Matplotlib is such a vital library in Python for ML data visualization; used to create visualizations and plots in 2D and 3D. It provides a wide range of tools for creating graphs, charts, histograms, scatterplots, and more, and is widely used in the scientific community for visualizing data and communicating research findings [10].
- Seaborn: It is another useful Python library, for data visualization that provides a high-level interface for creating informative and attractive statistical graphics. The library offers a variety of plotting functions for different types of data, such as scatter plots, line plots, bar plots, histograms, and heatmaps. Seaborn is built to work smoothly with Pandas data frames and NumPy arrays, and other data analysis libraries [10].
- Scikit-learn: This also another useful Python library machine learning. Some key features that make it very essential are that It is built on top of NumPy, Matplotlib and SciPy, It can be accessed by everyone and can be used in different contexts, and finally, a wide variety of ML algorithms can be implemented using it. [10]

Depending on the platform one is using, these libraries can be installed/imported for use by just a single line of code. In Jupiter Notebook, the packages are imported into Python script by the following line of codes [10]:

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.model_selection import train_test_split, GridSearchCV
        from sklearn.linear_model import LinearRegression
```

Figure 1: Lines of Code for Various ML Python Libraries

Prominent machine learning techniques include supervised learning, unsupervised learning, semi-supervised learning and reinforced learning [10].

Supervised learning: involves training a model on a labeled dataset; one where each data point is associated with an output or response variable. The goal of the model is to learn a mapping from input variables to the output variables and using it to make predictions on new or unseen data. To illustrate, consider:

X: the set of input or independent variables, and;

Y: the corresponding output variable.

Here, the goal is to employ an algorithm for learning the mapping function connecting input variables to the output variable: $Y=f(x)$. The primary aim is to create an accurate approximation of this mapping function, such that when presented with new input data (X), the algorithm can make reliable predictions for the output variable (Y). The reason this is called "supervised" is that, the learning process is being guided by human supervisor. Examples of supervised algorithms include Linear Regression, Decision Trees, Random Forest and K-Nearest Neighbors (KNN) [9][10]. In machine learning regression model, given inputs and output that are quantitative, it seeks to learn from training set and formulate a numeric function that captions its relationship. In polynomial interpolation, the goal is to find a polynomial of degree (N-1) that passes through N given points that can be used to predict for any output of x.

Unsupervised machine learning algorithm; entails training a model on an unlabeled dataset, where there is no target or output variable to be predicted. Unlike supervised learning Algorithms, unsupervised learning operates independently of a supervisor guiding the learning process. Instead, it extracts patterns and insights from data, even when input data is unlabeled. This approach is useful when it is not feasible to use pre-labeled training data, as is the case with supervised learning [10].

To illustrate, suppose we have a set of input variables, denoted by x, but we do not have any corresponding output variables. In this case, unsupervised learning algorithms can be used to discover interesting relationships and patterns within the data, without any need for pre-existing labels or guidance. Examples of unsupervised learning algorithms include K-nearest neighbors, K-means clustering, dimensionality reduction, and anomaly detection. Unsupervised learning is also very useful in exploratory data analysis, data visualization, and feature engineering [10].

Semi-supervised learning; a type of machine learning methods that lies somewhere between fully supervised and fully unsupervised approaches. These methods rely on a combination of small amounts of labeled data, as in supervised learning, and large amounts of unlabeled data, as in unsupervised learning. The different approaches that can be used to implement semi-supervised learning.

One approach is first to construct a supervised model using the small available small labeled dataset. Subsequently, an unsupervised model is applied to the extensive unlabeled data to generate additional labeled samples. These samples are then incorporated into the model training, and the process can be repeated in iterations.

The other approach involves using unsupervised techniques to cluster similar data samples, and then annotating these identified groups. The combination of this information is then been used to train the model. This approach requires some additional effort, but can potentially yield more accurate results [10].

- Reinforcement learning; a type of machine learning method where there is an agent who is to be trained over time so that it will be able to interact with a specific environment. This is different from other learning methods and is not commonly used. The agent follows a set of strategies for interacting with the environment and takes actions based on the current state of the environment after observing it. The main steps involved in reinforcement learning [10]:
    1. preparing the agent with initial strategies,
    2. observing the environment and its current state,
    3. selecting an optimal policy based on the current state and performing an action,
    4. receiving a reward or penalty based on the action taken,
    5. updating strategies if needed, and
    6. repeating steps 2-5 until the agent learns and adopts optimal policies.

Chapter 4

DATA ANALYSIS

The dataset used in this thesis is gotten from Kaggle. Kaggle is an online platform for data science and machine learning that provides datasets and offers a community for data science enthusiasts to collaborate and learn from one another. Kaggle was founded in 2010 and acquired by Google in 2017. This dataset consists of 1338 entries related to insurance, with insurance charges given against the following attributes: Age, Sex, BMI (body mass index), Number of Children, Smoker, and Region [12]. The attributes of the dataset are a combination of both numerical and categorical variables. This dataset doesn't contain any missing or undefined values [11]. This dataset was used to forecast the health insurance premium. The entire methodology carried out in this thesis follows the steps shown in Figure below:

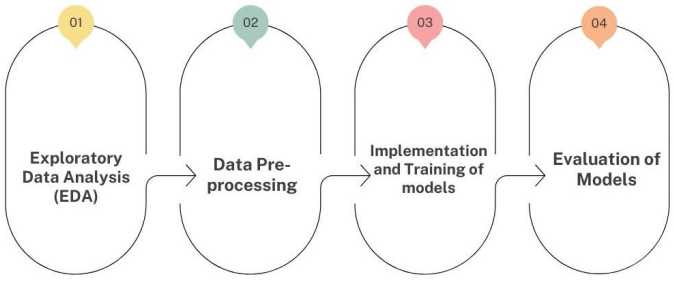

Figure 2: Stages of the Machine Learning Process

4.1 Exploratory Data Analysis (EDA)

In Exploratory Data Analysis (EDA), the main statistical characteristics of the dataset are explored and summarized, with the aim of understanding its structure and

identifying relevant patterns, relationships and trends that may be present. EDA is performed on the dataset before applying any formal ML algorithms to it. This is to guide the selection of appropriate technique and attributes for the models [10]. EDA is mainly cross-classified in two ways; graphical method and non-graphical method. Non-graphical methods involve the computing the summary statistics which includes the mean, standard deviation, correlation, skewness, outliers, of all variables in the data. Graphical method visually summarizes the data in a diagrammatic fashion for easy comprehension and comparison. [13].

The statistical summary of the dataset was assessed, which provided details such as the count, mean, standard deviation, and other related statistics pertaining to the columns in the dataset.

```
In [9]: insurance.describe()
```

Out[9]:

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

Figure 3: Summary Statistics of Dataset

There are two types of data visualization techniques; single-variable or “univariate” and multivariate plot. In univariate plot, each variable is visualized and understood independently. Univariate plots used in this analysis include histogram, density plot, pie chart and box and whisker plot. Multi-variable or multivariate methods on the other hand, we can plot and see the interaction between two or more variables at a time to

explore their relationships. Multivariate plot used here include correlation matrix plot and scattered matrix plot [13][10]. EDA is presented below.

- **Charges:**

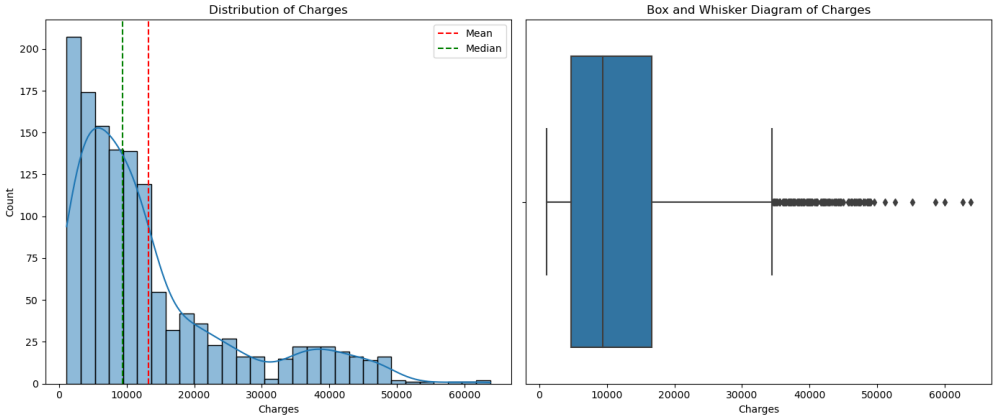

Figure 4: Histogram and Box Plot of Charges Attribute

Table 1: Distribution of Data Values in Charges Attribute

Maximum Charges	63770.43
Minimum Charges	1121.87
Mean	13279.12
Median	9386.16
Skewness	1.51
Total number of outliers	139

We have on the left in the figure above is the histogram with a class number of 30 and on the left the whisker and box plot of the distribution of charges. It can be seen that on average, clients have insurance costs of around $13,279, but there are also clients whose insurance costs exceed $60,000, and these clients are outliers that cause the distribution of the charges column to be right-skewed with a skewness value of 1.5.

- **Age:**

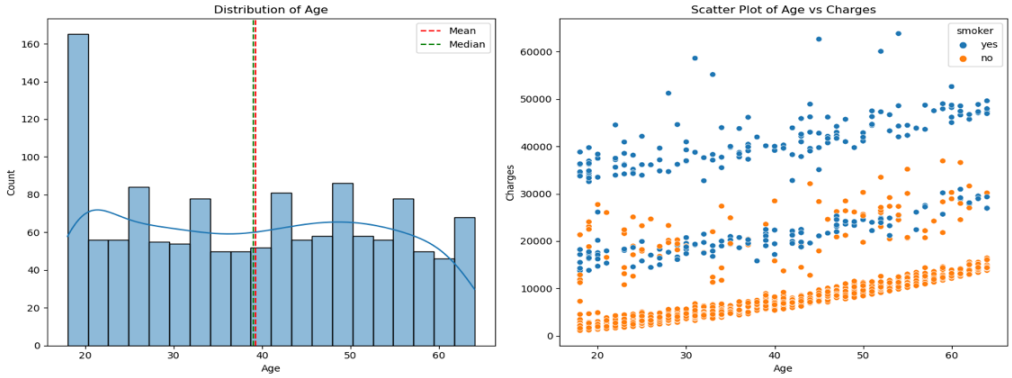

Figure 5: Histogram and Scattered Plot of Age Attribute

Table 2: Distribution of Data Values in Age Attribute

Skewness	0.054780773126998195
Mean	39
Median	39.0
Minimum Age	18
Maximum Age	64

The figure above we see the distribution of ages following a symmetric pattern, with a skewness value close to 0 and the mean and median values being similar. The scatter plot shows that there is a positive correlation between a client's age and their charges, meaning that the cost increases as the client gets older. Additionally, smokers tend to have higher charges than non-smokers.

- **Sex:**

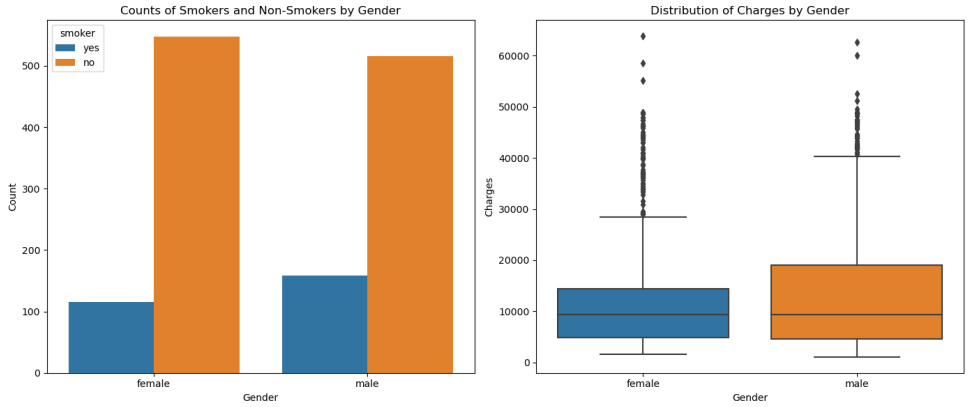

Figure 6: Bar Chart and Box Plot of Sex Attribute

Table 3: Distribution of Data Values in Sex Attribute

	Smoker			**Mean Charges**
Sex	**No**	**Yes**	**Total**	
Female	547	115	662	125700
Male	516	159	675	13975
Total	1063	274	1337	

The figure above on the right shows the counts of gender with the ration of smoker and none smoker and on the left, the box and whisker plot on the left showing the gender distribution against the charges. The number of male and female clients is not significantly different, as well as the ration of smoker are similar. But when looking at the box plot, there is a significant difference in the range of insurance costs, with the average insurance cost for male clients being around $13,975 and for female clients being around $12,570. This may be due to the fact that there are more male clients who smoke compared to female clients.

- **Body Mass Index (BMI):**

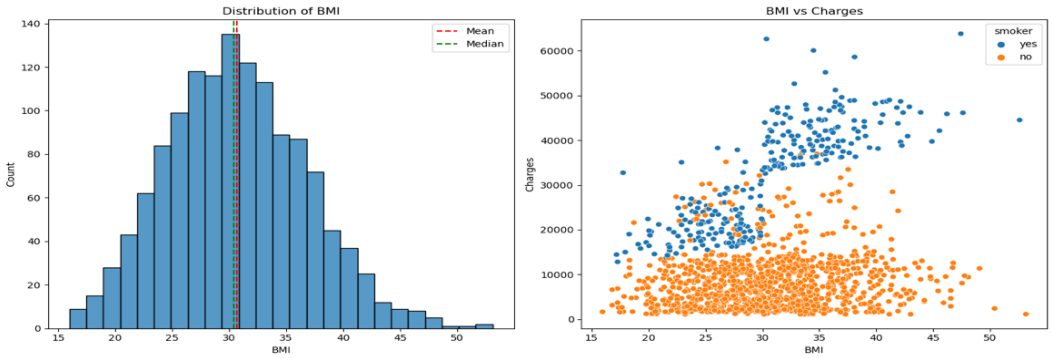

Figure 7: Bar Chart and Box and Scattered Plot of BMI Attribute

Table 4: Number of Outliers for Each Number of Children

Skewness of BMI	0.28
Mean of BMI	30.66
Median of BMI	30.40

From the plots above, it can be observed the BMI distribution follows a bell-shaped normal curve with minimal outliers. The scatter plot illustrates a mild positive correlation between the BMI and charges columns. If the client does not smoke, the increase in insurance cost due to BMI is not significant. However, for smokers, their insurance cost increases as their BMI increases.

- **Children:**

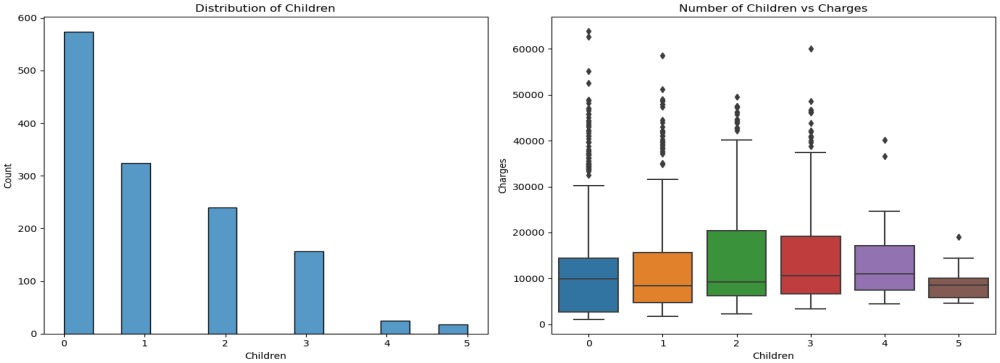

Figure 8: Bar Chart and Box Plot for Children Attribute

Table 5: Number of Outliers for Each Number of Children

Children	Number of Outliers
0	63
1	32
2	19
3	16
4	2
5	1

From the plots above, the histogram shows a left skewness and based on the box plot between the children and charges columns, there is a weak positive relationship between these two attributes. The outliers of charges also reflect when plotted against the number of children.

- **Smoker:**

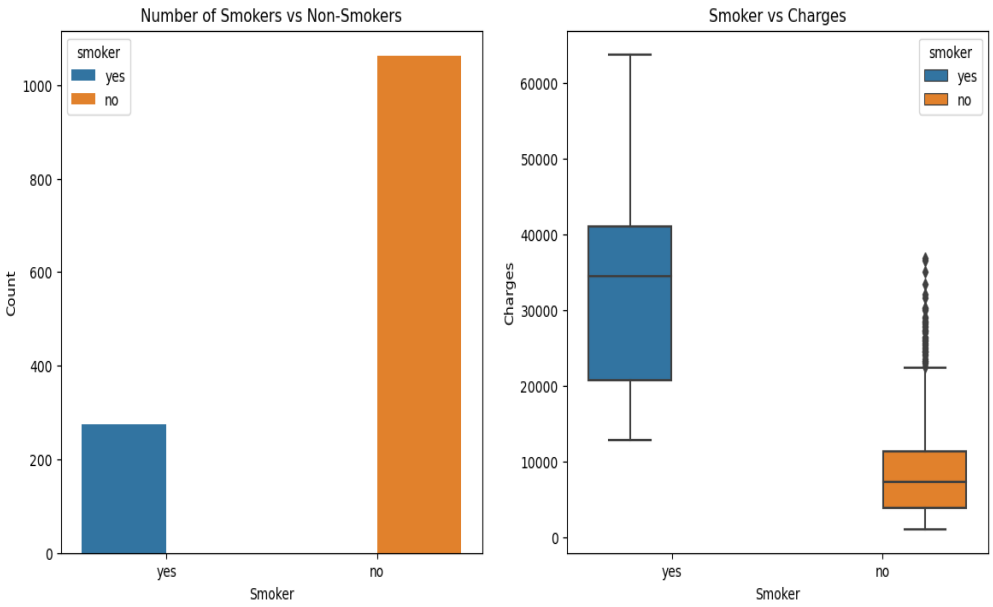

Figure 9: Bar Chart and Box Plot for Smokers Attribute

Table 6: Distribution of Data Values in Smokers Attribute

Smoker	Count	Average Charges	Outliers
No	1063	8440.660306508935	46
Yes	274	32050.23183153285	0

From above, it can be observed that, 274 people are smokers, this is equivalent to around 20% of the insurance clients. Although this number is relatively small, smokers have to pay a significantly higher cost of approximately $32,050 compared to non-smokers who only have to pay around $8,441.

- **Region:**

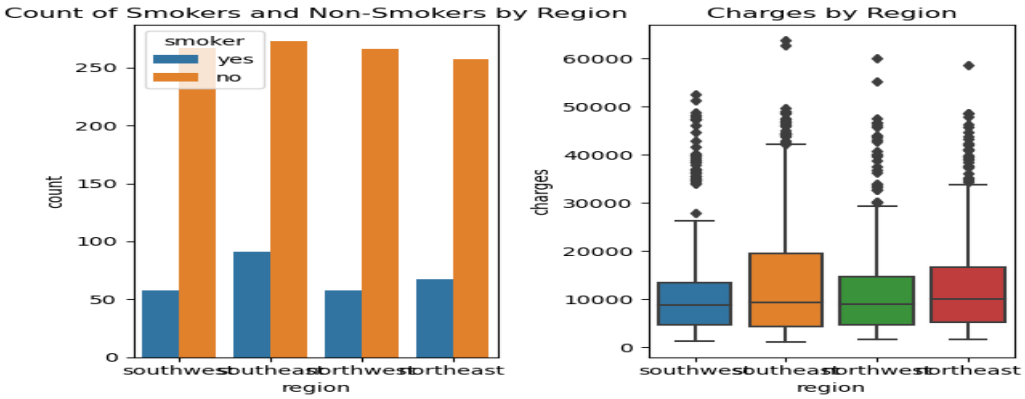

Figure 10: Bar Chart and Box Plot for Region Attribute

Table 7: Distribution of Data Values in Region Attribute

Region	Counts	Average Charge	Outliers count
southeast	364	14735.411438	26
southwest	325	12346.937377	38
northwest	324	12450.840844	29
northeast	324	13406.384516	29

From the above plot and table, it can be observed that the count and average charges for each region are relatively similar except for southeast region that is noticeably higher. The higher average charges for southeast region can be accounted for by the

fact that it has more smokers in this region and also a higher insurance cost for this region. Below is a multivariate exploration plots of the dataset.

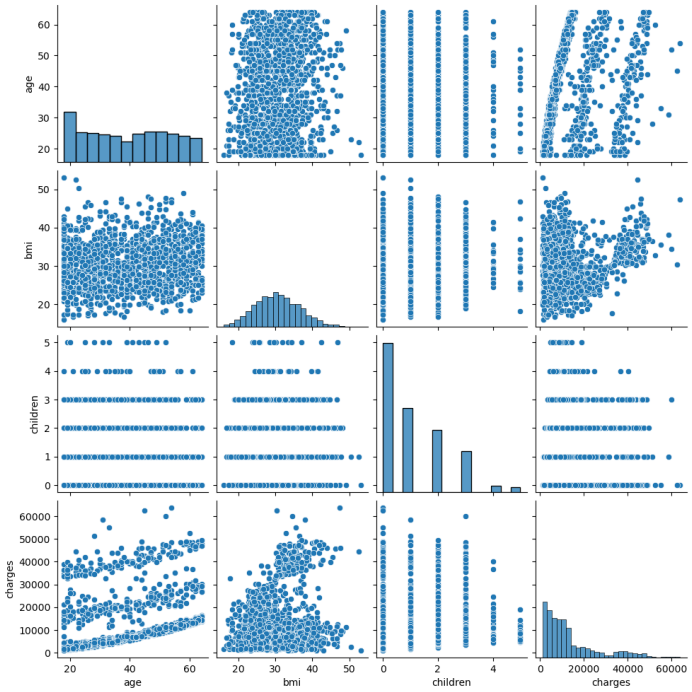

Figure 11: Pairplot Chart for Entire Dataset

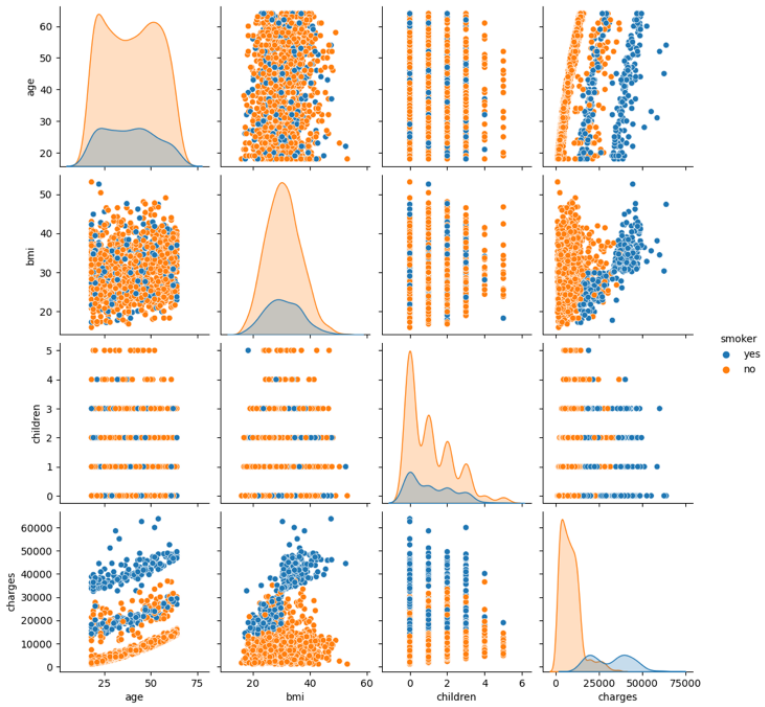

Figure 12: Pairplot Chart Differentiating Smokers and None-smokers

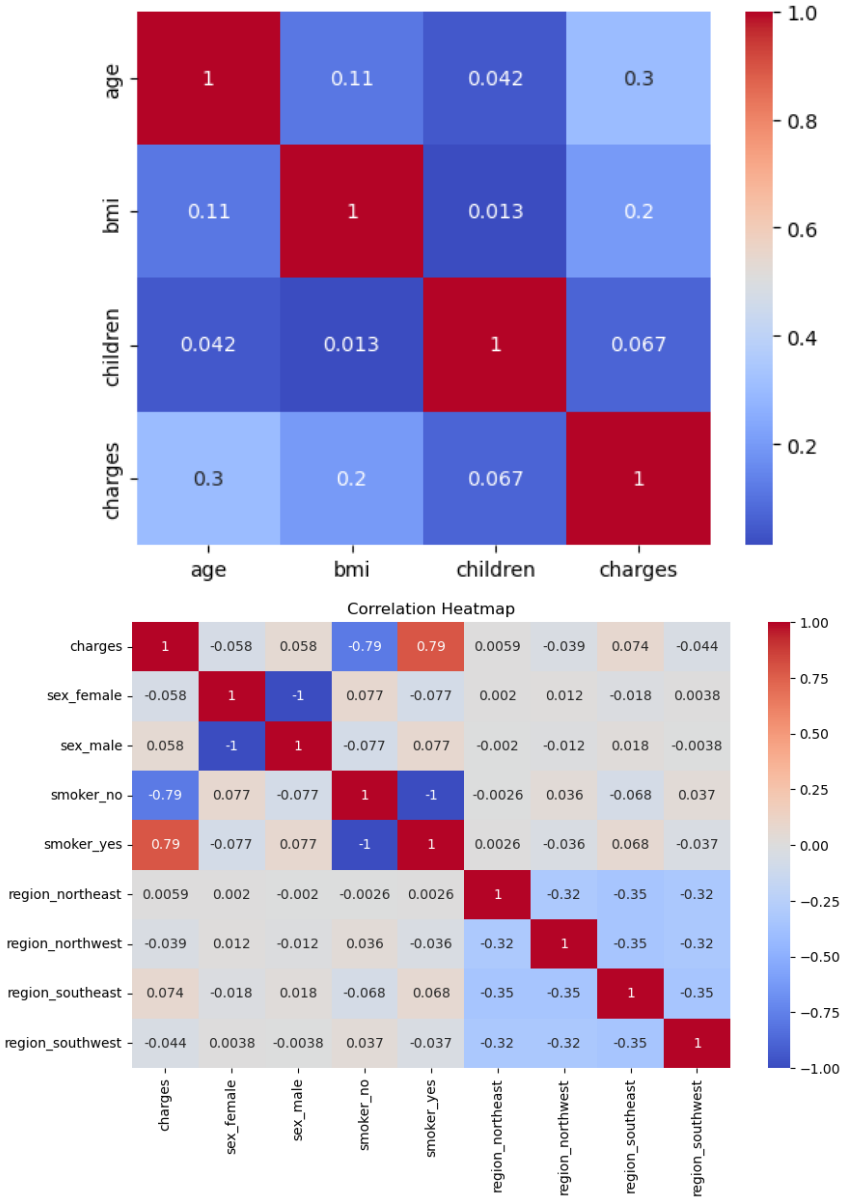

Figure 13: Correlation Matrix

From the above pairplot and correlation plot, we can clearly see the relationships between each of the variables. The pair plot only includes quantitative variables, while the second pairplot differentiates smokers from non-smokers. It is evident from these plots that the number of children has no correlation with the target variable, implying that it is not an important variable in the determination of insurance charges. Therefore, we will drop this variable from our analysis.

4.2 Data Pre-processing and Model Definition

Data pre-processing is the next very important step that is carried out in this analysis. Most classic statistical theories assume that the data is in the correct form and ready to be analyzed and so pay more emphasize on modeling, prediction, and statistical inferences [14]. Original data in most case is impure, meaning it contains inconsistent or omitted values, biased meaning certain parts of the population are poorly represented, or the variables maybe in inappropriate form for analysis. Thus, analyzing this data will open one up to erroneous results and potentially misleading decisions [14]. In data pre-processing stage, we seek to modify the data to adjust for the impact of outliers and also to transform inappropriate datatypes based on the knowledge gained from exploratory data analysis to ensure it is meaningful before feed it to the ML algorithms.

- **Handling Outliers**

From EDA, the box and whisker plots for all other explanatory variable indicated there exist a significant number of outliers in the target variable which is charges. The outliers are examined closely to against the categorical variables for any possible explain for their existence in the context of our data source.

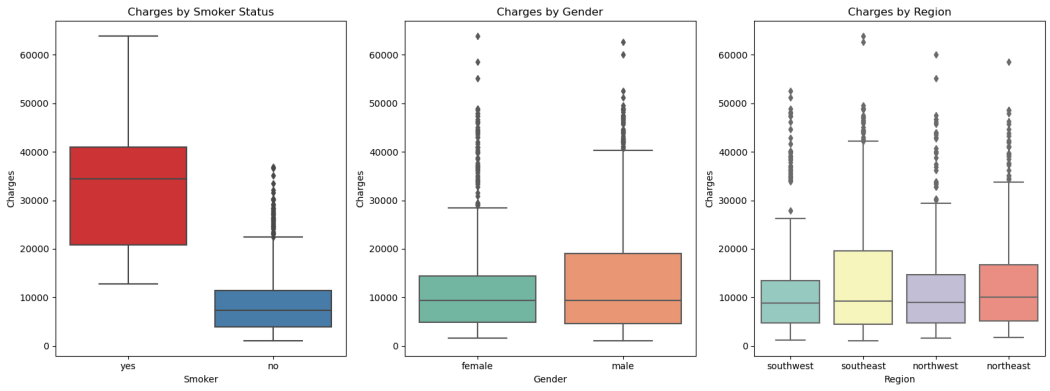

Figure 14: Box Plot for Smokers, Genders and Region Respectively

Number of outliers: 139, Percentage of outliers: 10.388639760837071%.

From the last two plots above, we can see that all categories in the variable have similar counts and they contains different number outliers but not much differences. These differences cannot be accounted for by any explanation. For the first plot, there is a significant difference in the plots; the smoker box and whisker plot have no outliers but its values begin notably higher and goes way further than that of none smokes. This can be explained by the fact that smokers are charged for health insurances more than none smokers for health insurances because they are at high risk. The outliers make up about 10% of our data and most of them are smokers and because the outliers are not random but has a meaningful presence in the data, will not be deleted.

- **Feature Selection**

The strength of a machine learning model depends greatly on the data features used for training. Including irrelevant features will negatively impact model. Conversely, using mainly relevant feature enhances accuracy [10]. Features selection involves selecting the most appropriate features to the desired output based on certain criteria. Feature selection is important because it reduces overfeeding, improves model accuracy and reduces training time for the ML model [10]. From correlation analysis in EDA, it is clear seen that not all of the features are relevant to the regression. There are some with approximately no correlation with the target variable charges, and are therefore dropped from the data. The following will be used age, bmi and smoker.

```
In [87]: # Drop the features children, sex and region from data
         insurance = insurance.drop(['children', 'sex', 'region'], axis=1)
```

Figure 15: Line of Code to Drop Insignificant Variable from Dataset

- **Splitting Dataset**

In data splitting, we partitioned the data into training set, validation set and testing set in the ration of 7:2:1. That is, 70% of the data will be training set, 20% will be validation set and 10% will be testing set. Our model will then be fitted to the training set to get the coefficients. The validation set is employed for unbiased model evaluation during hyperparameter tuning. Final prediction is made using the test set [14]. The essence of splitting is to avoid a biased performance in prediction. This is also to avoid overfitting or underfitting. Overfitting happens when a model is very complex that it is learning both from the existing relationship and the noise in the data while in underfeeding the model is weak to capture the relationships in the data, for instance, depicting nonlinearity as a linear model [14].

```
In [133]: # Split data into training and temporary sets (70% - 30%)
X_train_temp, X_test, y_train_temp, y_test = train_test_split(
    insurance.drop('charges', axis=1),  # Features
    insurance['charges'],  # Target variable
    test_size=0.3,  # 30% for temporary set
    random_state=42  # For reproducibility
)

# Split temporary set into validation and test sets (20% - 10%)
X_val, X_test, y_val, y_test = train_test_split(
    X_test, y_test,  # Features and target from temporary set
    test_size=1/3,  # 1/3 of the temporary set for validation set
    random_state=42  # For reproducibility
)

# Print out sizes of the resulting sets
print("Training set size:", len(X_train_temp))
print("Validation set size:", len(X_val))
print("Testing set size:", len(X_test))

Training set size: 936
Validation set size: 268
Testing set size: 134
```

Figure 16: Lines of Code to Split Dataset

- **Features Transformation**

Manually handling each feature and carrying out transformation or scaling individually may lead to errors and data leakage, fall short of encapsulation, and can

be very difficult to manage and deploy. A robust and more efficient way this is handled in an end-to-end machine learning workflow, ensuring consistency and better model performance is by implementing a Pipeline. In machine learning, pipelines are a way of combining different data transformers and estimators chaining them into a single unit and allowing for a streamlined automated workflow for data preprocessing and model training [16][10]. The preprocessing pipeline is built using three packages imported from the scikit-learn library: pipeline, PCA, and ColumnTransformer.

StandardScalar(): It is use to standardize all numerical features in our data ensuring that all the features have a mean of zero and a unit standard deviation [10].
OneHotEncoder(): This is used to encode categorical features in our dataset by converting them into a binary vector in which each category becomes a separate binary feature [10].

Dimensionality reduction with PCA() employs linear projection technique to transform correlated features in the dataset in higher dimension into series of uncorrelated features in lower dimension space while retaining important information [15][10].

```
In [139]: # Numeric Features Scaling
num_pipe = Pipeline([('scaling', StandardScaler())])

# Categorical Features Encoding
cat_pipe = Pipeline([('encode', OneHotEncoder(handle_unknown='ignore'

# Dimensionality Reduction using PCA
pca = PCA(n_components=0.95)

# Preprocessing Pipeline
preprocess = ColumnTransformer([
    ('num', num_pipe, X_train_temp.select_dtypes(
        include=['float64', 'int64']).columns),
    ('cat', cat_pipe, X_train_temp.select_dtypes(
        include=['object']).columns)
])

# Create the final pipeline
pipeline = Pipeline([
    ('preprocess', preprocess),
    ('pca', pca)
])

# Fit and transform the data
X_train_processed = pipeline.fit_transform(X_train)
X_test_processed = pipeline.transform(X_test)
```

Figure 17: Lines of Code for Dimensionality Reduction and Pipeline

4.3 Model Implementation

For our analysis, three regression models are implemented namely;

- **Linear**; the normal Linear Regression model.
- **Poly**; Linear Regression model with Polynomial Features, tuned using grid search with 5-fold cross-validation.
- **XGBR**; Extreme Gradient Boosting Regressor model, tuned using grid search with 5-fold cross validation.

Grid search with 5-fold cross-validation is used for optimizing performance of the model by systematically testing various combinations of hyperparameter values. Grid search also identifies the best settings that maximize model performance, while the 5-fold cross-validation ensures robust evaluation by dividing the dataset into five subsets. The model is trained and evaluated multiple times, allowing for a more accurate estimation on new data and reducing the influence of random variations.

4.4 Evaluation of Models

In this section we examine and compare the performance and the effectiveness of each of the trained models. Here, we evaluate the performance metrics for each of the model and compare to see which model has a higher accuracy in making prediction and the goodness of fit. The metrics of interest to our analysis include

- Mean Absolute Error (MAE); computes the mean absolute difference of the predicted output values and the actual output values. It shows the average magnitude of the error.
- Mean Squared Error (MSE); computes the mean squared difference between predicted output values and the actual output values.
- Root Mean Squared Error (RMSE); It takes the square root of the MSE, providing a measurement in the same unit as the output variable. This is more interpretable than MSE.
- R-squared (R^2) Score; measures the proportion of variation in the target variable that can be explained by the model. It ranges from 0 to 1 where an R^2 of 1 indicates perfect fit.

After creating the Preprocessing pipeline, the algorithms; Linear regression, polynomial regression and Extreme gradient boost regression were trained individually on the training set, and predictions made on the test set. For each of the models, the regression coefficient and the performance metrics were evaluated as well.

- **Linear Regression;** The regression coefficients for this model was determined using the OLS method. The coefficient of the of the regression along with other statistics relevant to test for its significance presented in the table below. We can

see that their P-values are <0.05, implying that the resulting values from the trained model are significant. The table below shows the intercept and coefficient from the regression:

OLS Regression Results

Dep. Variable:	charges	R-squared:	0.739
Model:	OLS	Adj. R-squared:	0.739
Method:	Least Squares	F-statistic:	881.8
Date:	Tue, 18 Jul 2023	Prob (F-statistic):	1.26e-271
Time:	18:55:01	Log-Likelihood:	-9498.1
No. Observations:	936	AIC:	1.900e+04
Df Residuals:	932	BIC:	1.902e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.338e+04	202.340	66.122	0.000	1.3e+04	1.38e+04
x1	3605.8304	192.321	18.749	0.000	3228.398	3983.263
x2	-705.9293	213.980	-3.299	0.001	-1125.869	-285.990
x3	1.686e+04	352.875	47.782	0.000	1.62e+04	1.76e+04

Omnibus:	222.518	Durbin-Watson:	2.064
Prob(Omnibus):	0.000	Jarque-Bera (JB):	528.851
Skew:	1.266	Prob(JB):	1.45e-115
Kurtosis:	5.675	Cond. No.	1.83

Figure 18: Results from OLS Linear Regression

- **Polynomial Regression;** the model was trained and the regression coefficient were determined using OLS method as well and present below. `'para_grid_polynomial'` is a dictionary in scikilearn that specifies the hyperparameter grid for the model. The hyperparameter tuned is `'polynomial_features__degree'`, and it tests with the values [2, 3, 4]. The pipeline then runs with these different degrees of polynomial features, and the best degree (which came out to be 3) automatically adopted based on cross-validated performance. With the degree of 3, We observe that the main performance measure, R- squared of train set is 0.83356, of the validation set is 0.86951 and of the test set is 0.83107. The lack of any significant discrepancy in these values implies that there no problem of overfitting and the model performs well with new

data. The problem with the model of degree 3 is its complexity; it contains 19 features, 7 of which are none significant. Below is the regression statistics;

Intercept: 8940.397856229367

OLS Regression Results

Dep. Variable:	charges	R-squared:	0.834
Model:	OLS	Adj. R-squared:	0.831
Method:	Least Squares	F-statistic:	307.2
Date:	Fri, 21 Jul 2023	Prob (F-statistic):	0.00
Time:	14:52:32	Log-Likelihood:	-9288.4
No. Observations:	936	AIC:	1.861e+04
Df Residuals:	920	BIC:	1.869e+04
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	8940.3979	277.126	32.261	0.000	8396.524	9484.271
x1	3564.6044	359.995	9.902	0.000	2858.098	4271.111
x2	-1286.9918	371.601	-3.463	0.001	-2016.275	-557.708
x3	4876.1886	210.482	23.167	0.000	4463.108	5289.269
x4	177.3459	133.227	1.331	0.183	-84.119	438.811
x5	-657.2178	244.759	-2.685	0.007	-1137.569	-176.867
x6	1590.0609	179.981	8.835	0.000	1236.841	1943.281
x7	184.4034	166.275	1.109	0.268	-141.918	510.725
x8	3455.3375	201.543	17.144	0.000	3059.800	3850.876
x9	6741.1668	145.461	46.344	0.000	6455.693	7026.640
x10	-145.4866	110.178	-1.320	0.187	-361.717	70.743
x11	-260.6307	261.467	-0.997	0.319	-773.772	252.511
x12	-434.5693	239.216	-1.817	0.070	-904.041	34.903
x13	-198.9940	271.942	-0.732	0.465	-732.693	334.705
x14	-153.6607	375.960	-0.409	0.683	-891.499	584.178
x15	2456.3969	144.724	16.973	0.000	2172.369	2740.425
x16	-66.4878	120.513	-0.552	0.581	-303.000	170.024
x17	-432.4085	263.640	-1.640	0.101	-949.813	84.996
x18	2464.8184	163.484	15.077	0.000	2143.974	2785.663
x19	7208.9119	180.647	39.906	0.000	6854.385	7563.439

Omnibus:	488.807	Durbin-Watson:	2.031
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2712.258
Skew:	2.436	Prob(JB):	0.00
Kurtosis:	9.768	Cond. No.	2.07e+16

Figure 19: Results from OLS Polynomial Regression with Degree 3

We trained the same polynomial model but with a degree 2 and with obtain approximately same and less complex with 9 features with only two non-significant features. The performance measure, R-squared for training set came out 0.830037, for validation set 0.869791 and for test set 0.830149. here are the features of the model: x1= age, x2 = bmi, x3 = smoker, x4 = age^2, x5 = agebmi, x6 = agesmoker, x7 = bmi^2, x8 = bmismoker and x9 = $smoker^2$.

```
Intercept: 9860.78919956422
                            OLS Regression Results
==============================================================================
Dep. Variable:                charges   R-squared:                       0.831
Model:                            OLS   Adj. R-squared:                  0.829
Method:                 Least Squares   F-statistic:                     568.1
Date:                Fri, 21 Jul 2023   Prob (F-statistic):               0.00
Time:                        14:45:39   Log-Likelihood:                -9296.7
No. Observations:                 936   AIC:                         1.861e+04
Df Residuals:                     927   BIC:                         1.866e+04
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       9860.7892    263.440     37.431      0.000    9343.781    1.04e+04
x1          3836.4423    157.084     24.423      0.000    3528.161    4144.724
x2          -975.7070    177.953     -5.483      0.000   -1324.944    -626.470
x3          8555.6545    181.819     47.056      0.000    8198.831    8912.478
x4            75.1678    130.164      0.577      0.564    -180.283     330.618
x5          -879.1806    226.422     -3.883      0.000   -1323.540    -434.821
x6          3857.3072    273.096     14.124      0.000    3321.348    4393.266
x7           122.7205    148.026      0.829      0.407    -167.785     413.226
x8          4937.1615    293.500     16.822      0.000    4361.160    5513.163
x9          1.011e+04    143.570     70.447      0.000    9832.322    1.04e+04
==============================================================================
Omnibus:                      482.866   Durbin-Watson:                   2.041
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2631.620
Skew:                           2.407   Prob(JB):                         0.00
Kurtosis:                       9.656   Cond. No.                     6.36e+15
==============================================================================
```

Figure 20: Results from OLS Polynomial Regression with Degree 2

- **XGBoosting Regression;** Grid search with cross validation of 5 was performed to optimize hyperparameters. This was done by `'param_grid_xgbr'` a dictionary from scikitlearn library specifying the hyperparameter grid to be searched during cross-validation. In this case, the grid includes two hyperparameters: `'n_estimators'` and `'max_depth'`. The values specified in the lists [100, 200, 300] and [3, 4, 5] respectively were tested during the grid search process to find the best combination. The model was trained to predict the target variable and it consisted of 100 trees, each with a maximum debt of 3. The XGBoost Regressor results were as well quite good both for the training and validation sets. The performance metrics, R – squared for the training set was 0.949807, for the validation set 0.846999 and for testing set 0.809378.

The table below shows the summary performance for the validation set of the three models implemented.

Table 8: Performance Metrics for Validation Set for All the Models

Metrics	**Linear**	**Polynomial (deg 2)**	**XGBR**
MAE	4054.12495	2656.60558	2926.04353
MSE	32398941.584	19749371.012	23206382.601
RMSE	5692.0068	4444.02644	4817.3003
R – Squared	0.78639	0.86979	0.84700
Adjusted R – Squared	0.78396	0.86831	0.84526

The table above, shows a comparison the performance metrics of the models in making more accurate predictions. It comes out clear that the polynomial model overall performs best. It has the lowest MAE, MSE, and RMSE, indicating better accuracy in predicting the target variable. Also, the R-squared value for this model was closest to 1, implying that the features in the model explain a significant portion (i.e., 87%) of the variance in the target variable. and it is closely followed by the XGBR model which also performs pretty well.

Chapter 5

CONCLUSION

In this study, the dataset used consists of client data from an insurance company with the aim of using machine learning model to predict the insurance cost a client will pay. From the data analysis process carried out we come to several take aways and conclusions:

- Through Exploratory Data Analysis (EDA), it was found that the average insurance cost paid by clients is $13,279, which is heavily influenced by whether a client is a smoker or not.
- Smokers make up only about 20% of the clients but have significantly higher insurance costs, with a mean of $32,050, while non-smokers pay an average of $8,441.
- The insurance cost is also affected by other factors such as age and body mass index (BMI). Older clients or clients with higher BMIs tend to have higher insurance costs. The average age of the clients is 39 years, although most of them are in their twenties, and the average BMI in the dataset is around 31 kg/m^2.
- The Sex, Number of children and Region attributes each had a correlation of approximately zero with Charges. This implies features play almost no role in the determination of insurance cost and so were dropped out of the dataset.
- Outliers were present in all features and are consistent in size. They were retained because they are meaningful in the context of the dataset.

- Based on the EDA results, the three regression models were created: Linear Regression, Polynomial Regression, and Extreme Gradient Boosting Regressor (XGBR). All models were trained using features that can influence the "charges" column, namely age, BMI, and smoker status.
- Evaluation of models is based on the performance metrics; the polynomial model preforms best in making accurate predictions, followed very closely by the XGBR model and the linear model also still performs impressively well.

Some of Limitation of this Research include:

- Firstly, the analysis is limited to a specific dataset with limited variables which may not be a representation of the general population and/or account for all the factors influencing insurance charges.
- Secondly, unmeasured variables could impact insurance premiums that are not accounted for by our models.
- Lastly, the study focuses mainly on regression models and does not explore other advanced machine learning algorithms or statistical techniques that may yield mor intriguing results.

This study is relevant for several reasons which are:

- The analysis of health insurance dataset provides insights into the factors influencing insurance premium and also how these factors relate and interact with each other.
- The developed regression models contribute to our current understanding insurance data and strive for a more accurate prediction based on many different attribute and parameters.

- The research highlights the significance of exploratory data analysis, data preprocessing and model development in the entire data analysis process for better predictions.
- Machine learning, and AI in general, is highly beneficial in the health insurance sector due to its capability of analyzing vast amounts of data fast and efficiently, resulting in streamlined operations and cost savings for both policyholders and insurers.
- By automating repetitive tasks, AI allows insurance professionals to focus on enhancing the policyholder's experience, benefiting patients, hospitals, physicians, and insurance providers alike.
- ML's ability to process historical data contributes to cognitive computing, addressing various challenges in healthcare applications and systems.

This analysis shows the potential of ML in forecasting health insurance premiums and highlights the need for further exploration and comprehensive investigation in this domain.

REFERENCES

[1] Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2012). Probability & Statistics for Engineers & Scientists (pp. 389-396).

[2] Douglas, M. C., Montgomery, D. C., & Runger, G. C. (2012). Introduction to Linear Regression Analysis (5th ed.) (pp. 2, 224-253, 423). Wiley.

[3] Hosmer, D. W., Jr., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. (pp. 1, 31) Wiley.

[4] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

[5] Shalabh. (2019). Polynomial Regression Models. Lecture Notes. Department of Mathematics and Statistics, Indian Institute of Technology Kanpur.

[6] Agresti, A. (2013). Categorical Data Analysis (3rd ed.) (pp. 163,182-183). Wiley.

[7] Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Regression Models. (pp. 48-53) McGraw-Hill Education.

[8] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Introduction to Statistical Learning. (pp. 237-244) Springer.

[9] Alpaydın, E. (2014). Introduction to Machine Learning (2nd ed.). (pp. 3, 34-36) MIT Press.

[10] Tutorialspoint (2019). Machine learning with Python. (pp. 3, 12-16, 17-19, 51-) www.tutorialspoint.com

[11] Kaggle website. https://www.kaggle.com/

[12] Kaggle (2020). US Health Insurance Dataset https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset

[13] Seltman, H. J. (2018). Experimental Design and Analysis. (pp. 61-99)

[14] Shepperd, M. (2022). CS5702 Modern Data Book. https://bookdown.org/martin_shepperd/ModernDataBook/

[15] Mirko Stojiljković. Split Your Dataset With scikit-learn's train_test_split() https://realpython.com/train-test-split-python-data/

[16] Pipelines and composite estimators. https://scikit-learn.org/stable/modules/compose.html#pipelines-and-composite-estimators

[17] Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. (pp. 102) O'Reilly.

[18] Artificial Intelligence and Health Insurance
https://www.rgare.com/knowledge-center/article/a.i.-and-health-insurance#

[19] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189–1232.

[20] Pauly, M. V., & Kunreuther, H. (2013). An Introduction to the Economics of Risk and Insurance. University of Pennsylvania Press.

[21] PWC. (2018). InsurTech's Potential to Revolutionize Traditional Underwriting. Retrieved from
https://www.pwc.com/us/en/industries/insurance/library/insurtech-underwriting.html

[22] Cutler, D. M., & Zeckhauser, R. J. (2000). The Anatomy of Health Insurance. Handbook of Health Economics, 1, 563-643.