Medical Insurance Cost Prediction

Sahil Mohammad

Introduction

The healthcare landscape has seen a significant cost increase over the last decade, primarily due to rising healthcare service expenses. Various factors influence healthcare costs which are covered in the dataset being examined. These factors are as follows :

- age: age of primary beneficiary
- Sex: insurance contractor gender, female, male
- BMI: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ˆ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- Children: Number of children covered by health insurance / Number of dependents
- Smoker: Smoking
- Region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- Charges: Individual medical costs billed by health insurance

The goal of this project is to analyze how these features impact healthcare charges and how can they be predicted for a sample of the population.

The associated Insurance Problem

The rising costs of healthcare services pose a significant challenge in today's society, suggesting a growing need for a good understanding of the factors that contribute to health insurance expenses. This project tries to solve the problem of predicting health insurance costs by leveraging a dataset, consisting of crucial variables. By analyzing key factors as mentioned above, we aim to connect together the various elements that influence healthcare expenditure.

The primary motivation behind this project lies in the imperative to understand the dynamics of healthcare costs. As medical expenses continue to rise, a predictive model can offer valuable insights into the underlying patterns and relationships that drive these costs. By using data science and predictive analytics, the project aims to provide a helpful model that can be used by any individual to predict the insurance cost that needs to be paid. The outcomes of this project can potentially inform policy adjustments, aid in the development of targeted medical methods, and assist individuals in making informed choices regarding their insurance coverage. As we investigate and explore the dataset, we anticipate solving the patterns that drive the insurance cost.

Dataset Exploration

For any further Machine Learning algorithms, it is important to see which features are categorical and which are numerical:

```
##         age         sex         bmi    children      smoker      region
##   "numeric" "character"   "numeric"   "numeric" "character" "character"
##     charges
##   "numeric"
```

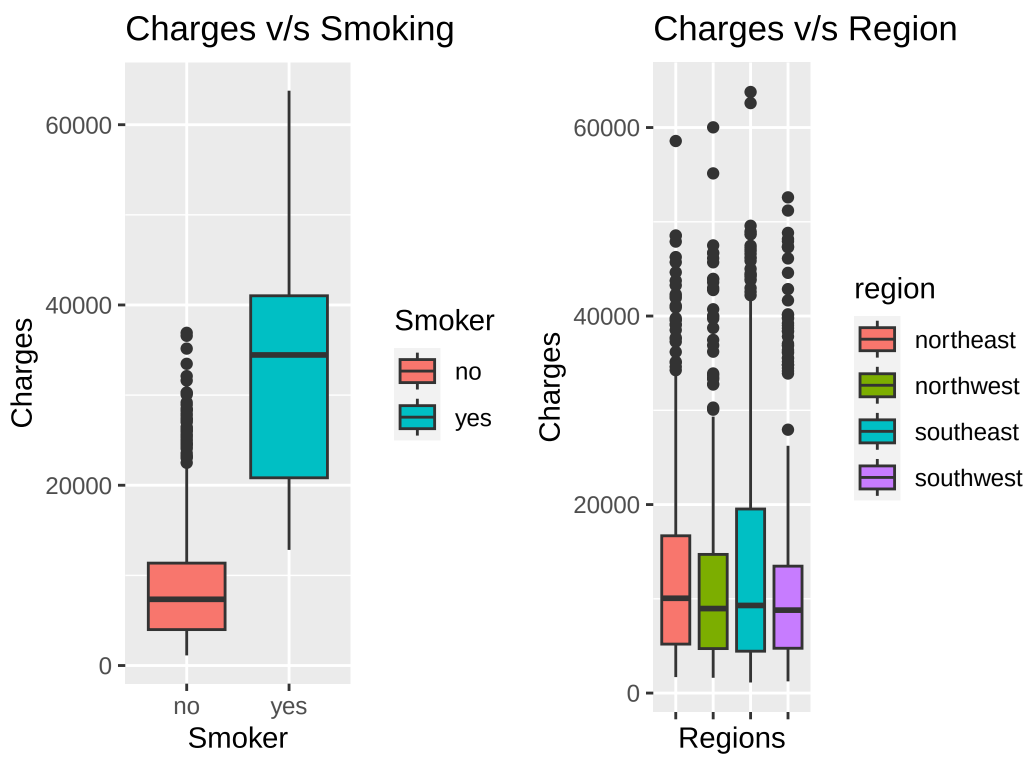
```
## [1] 1338    7
```

We check the class of each column of the dataset along with the dimensions. It is clear that the data comprises of 1,338 rows and 7 columns. Here we see that sex, smoker and region features are categorical while the others are numerical. Another important check is for missing data. Incomplete data can significantly affect the effective analysis and interpretation of data, potentially leading to misleading conclusions and wrong results. Checking for missing data is therefore a vital step in machine learning. Identifying and addressing these missing values is crucial for maintaining data integrity, ensuring model performance, and enabling informed decision-making. Thus, we check for columns that contain any missing values:

```
##      age      sex      bmi children   smoker   region  charges
##        0        0        0        0        0        0        0
```

Fortunately, we do not have NA values in any of the columns. We can proceed with further explorations.

For an unbiased ethical model, the gender of a person should not have any effect on the charges that the person has to pay for insurance. However, given the fact that the person smokes or not, might be an important feature. Let us check if that is the case.



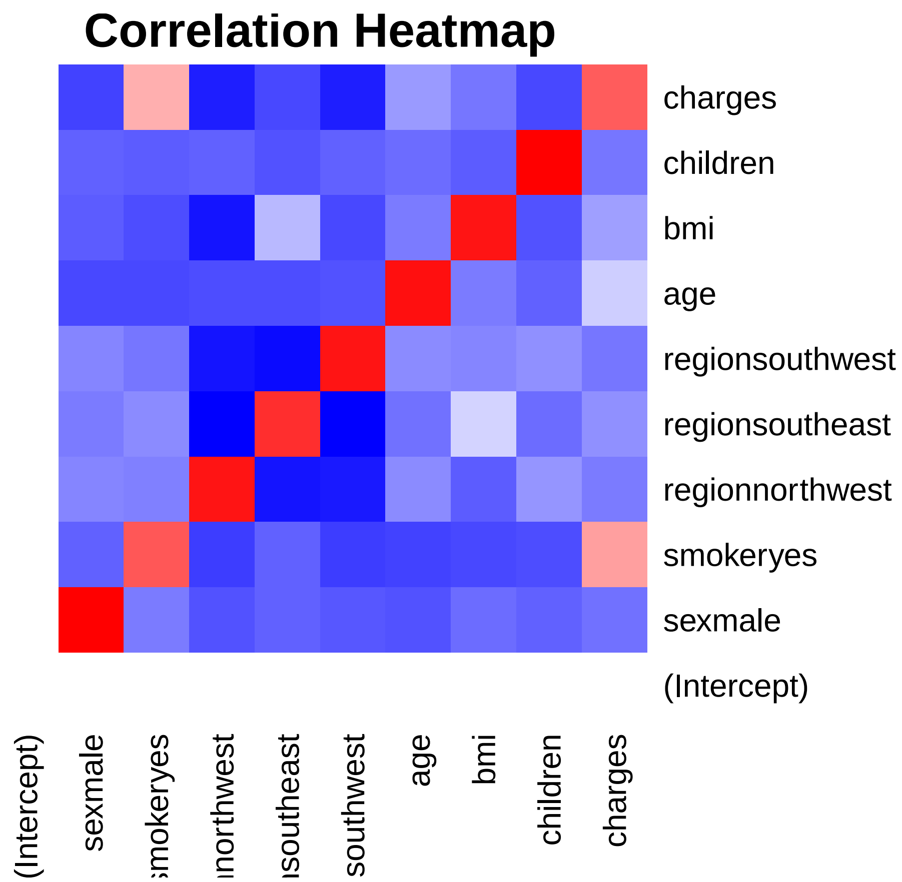
The figure above shows a boxplot of charges versus smoking status and region. From the plot, we can make the following conclusions :

- Smokers tend to have higher charges than non-smokers: This is evident from the fact that the median charge for smokers is much higher than the median charge for non-smokers in all three regions.
- There is a regional variation in charges: The median charge is highest in the Northeast and lowest in the Southwest. Also, given that the charges are highest in the Northeast, let us check it is also the region with most number of smokers, in other words, if there is a correlation between the two features.

```
## The region with the maximum number of smokers is: southeast
```

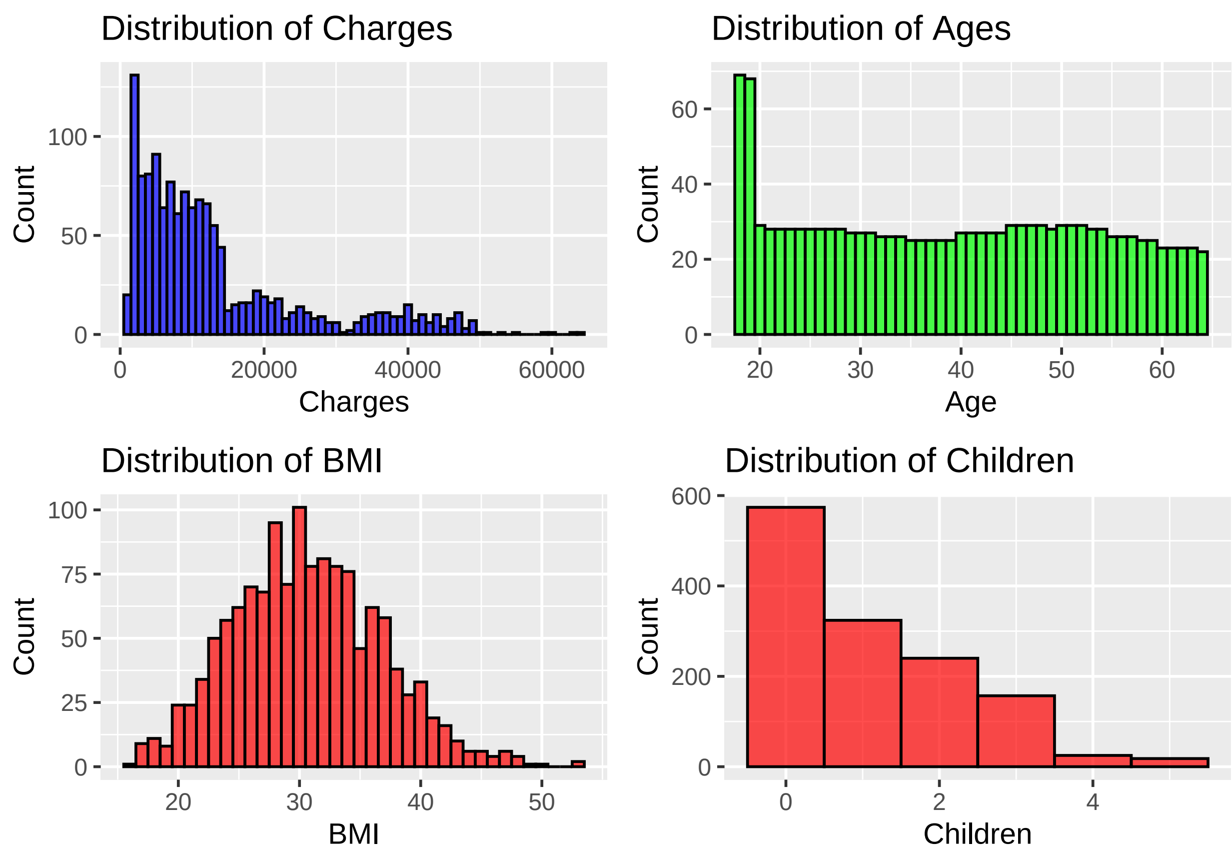
However, on running a code snippet, the results above indicate otherwise. This means there are other factors due to which Northeast has higher charges.

Let us check if these findings are reported by correlation with the help of a heatmap.



We see that there are no features that are very highly correlated. However, as expected, we see that smokers tend to pay more and we see the correlation between smoker("yes") and charges.

We now check the distribution of various features :



The distributions of features are important to see because we get an idea of the general trends.

- Regarding Charges, we see that most of the people pay below 20,000 USD. However, there are some individuals who pay above 60,000 USD. Thus, we can say that the distribution is right skewed.
- When it comes to ages, we see there are a lot of people below 20, but after that we see it is more of a uniform distribution. With roughly around 30 people from each age.
- BMI clearly follows a normal distribution with mean around 30. This is consistent with the normal distribution of BMI in the general population.
- Moreover, considering the number of children, we see that most of the people do not have any children. The distribution is clearly right-skewed. This indicates that there are more individuals with zero children than individuals with one or more children. This pattern aligns with the general population distribution, where most people do not have children.

The aim of the project is to use all the features that were explored above, correctly and effectively to predict the amount that the person needs to pay for a medical insurance. However, to run any machine learning algorithm, it is important to treat and convert the categorical variables into some encoded form. The output below shows the encoding of the categorical variables to a numeric form, making them suitable for use in predictive modeling for medical insurance charges:

```
## # A tibble: 6 x 7
##     age   sex   bmi children smoker region charges
##   <dbl> <dbl> <dbl>    <dbl>  <dbl>  <dbl>   <dbl>
## 1    19     1  27.9        0      1      1  16885.
## 2    18     2  33.8        1      2      2   1726.
## 3    28     2  33          3      2      2   4449.
## 4    33     2  22.7        0      2      3  21984.
## 5    32     2  28.9        0      2      3   3867.
## 6    31     1  25.7        0      2      2   3757.
```

We can clearly see that the Categorical variables are now encoded to integers. Here, sex is 1 for females and 2 for males. Smoker is 1 if "yes" and 2 if "No". Also, different regions are encoded based on different order levels.

Machine Learning Models:

The data was divided into a training and test set with a 80-20 split. These sets have been used for each of the models below.

Linear Regression:

The linear regression model is utilized here to predict medical insurance charges. The model aims to capture the linear relationship between the response variable, "Charges," and the different features which are the predictor variables. These predictors are selected based on their potential influence on insurance costs, and the model estimates coefficients for each predictor, indicating the expected change in charges associated with a one-unit change in the corresponding variable. Additionally, the model calculates an intercept term, representing the estimated charges when all predictors are zero. Model performance is assessed using metrics like R-squared, providing insight into the proportion of variance in charges explained by the predictors. Once trained, the model can be applied to new data to make predictions, offering valuable insights into the factors contributing to medical insurance costs.

The general equation for a simple linear regression model is given by:

$$Y = \beta_0 + \beta_1 X1 + \beta_2 X2 + ... + \beta_p Xp + \varepsilon$$

Where:

- Y is the response variable. β_0 is the intercept term. $X_1, X_2, \ldots, X_p$ are different predictor variables.
- $\beta_1, \beta_2, \ldots, \beta_p$ are the corresponding coefficients and ε is the error term.

GLMNET :

GLMNET, short for Generalized Linear Models with L1 and L2 Regularization, is a regression analysis technique that combines the principles of linear regression with regularization methods. It is particularly useful when dealing with high-dimensional datasets where the number of predictor variables is large. GLMNET simultaneously performs variable selection and regularization by minimizing a combination of the least squares term and penalties for the absolute values of the coefficients (L1 regularization) and their squares (L2 regularization). In this project, we utilise Cross-validation using the 'cv.glmnet' function to identify the optimal values for the regularization parameters, lambda, and alpha. Lambda controls the overall strength of regularization, while alpha determines the mix between L1 (lasso) and L2 (ridge) penalties. We then extract the best lambda value and, if applicable, the corresponding alpha. Finally the model is then used to make predictions on a separate test set, and performance metrics are computed to evaluate the model's effectiveness in predicting the response variable.

XGBOOST:

XGBoost, or Extreme Gradient Boosting, is a powerful machine learning algorithm renowned for its efficiency and effectiveness in handling diverse datasets. The algorithm sequentially builds a series of decision trees, each correcting the errors of the previous one, ultimately creating a strong ensemble model. Here we create a grid of hyperparameter values, such as the learning rate (eta), maximum depth of trees (max_depth), and other relevant parameters. XGBoost allows for the inclusion of additional hyperparameters like subsample and colsample_bytree, providing a fine-tuned control over the model's behavior. For each combination of hyperparameters, the code calculates the corresponding R-squared values, and our approach extends to evaluating other performance metrics, enabling a comprehensive understanding of the model's behavior. Finally, we select the set of hyperparameters that yield the highest R-squared. This iterative optimization process significantly contributes to refining the XGBoost model's configuration, ensuring its accuracy in predicting insurance-related variables. Beyond its performance on insurance datasets, XGBoost finds applications across various domains, such as finance, healthcare, and marketing, showcasing its versatility and effectiveness in solving complex problems and making it a widely adopted tool in the data science community.

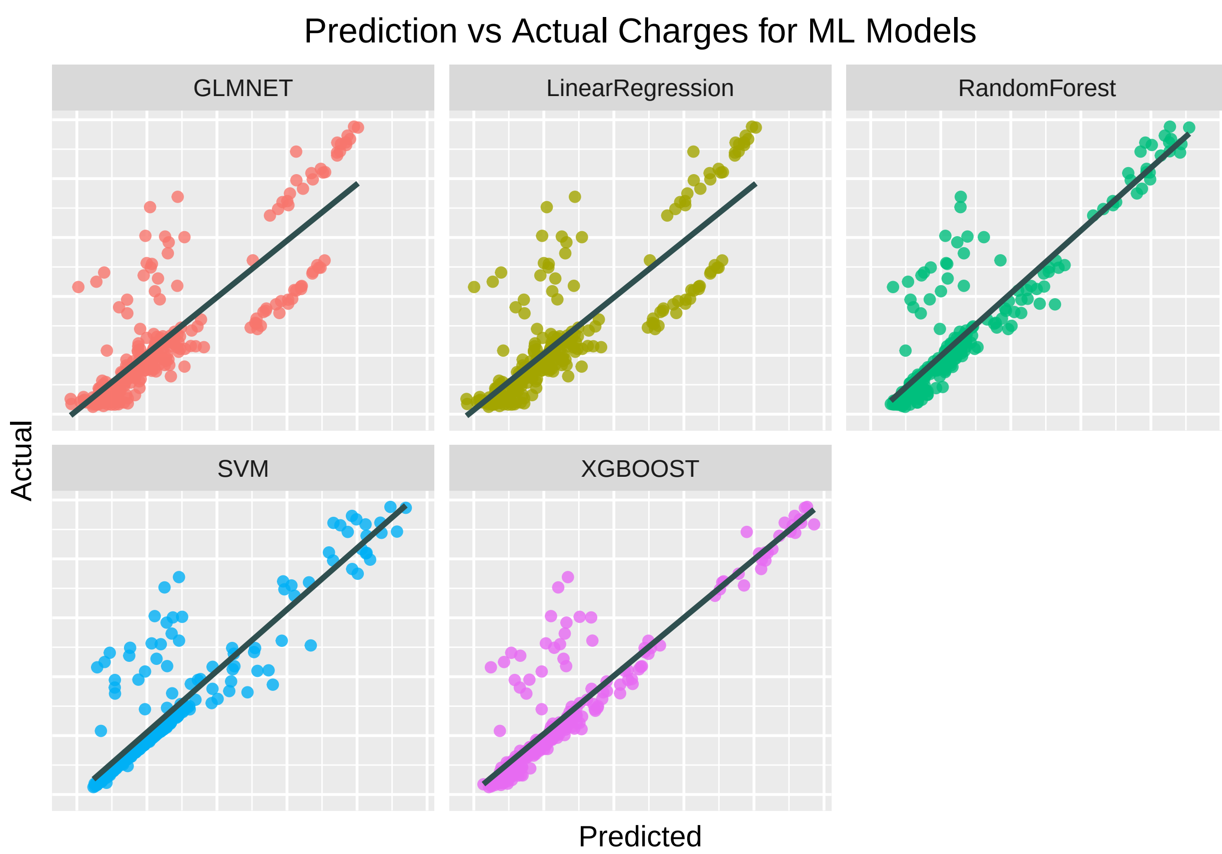
Random Forest:

Random Forest is an ensemble learning algorithm widely used for both classification and regression tasks. It builds multiple decision trees during training and combines their predictions to enhance overall accuracy and robustness. The key concept lies in introducing randomness during both the construction of individual trees and the selection of features used for splitting nodes. This randomness helps reduce overfitting and improves the model's generalization performance. Random Forest introduces the concept of out-of-bag error, where each tree is trained on a subset of the data, leaving out about one-third of the observations. The omitted samples can then be used to estimate the model's performance without the need for a separate validation set, providing an internal validation method. The final prediction is often obtained by averaging or taking a majority vote of the predictions made by individual trees. R provides the 'randomForest' package, which as the name suggests, provides the functionality to implement a Random Forest model.

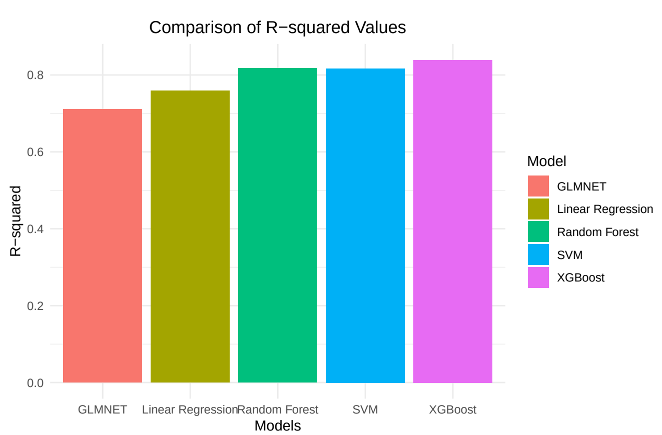
Support Vector Machine:

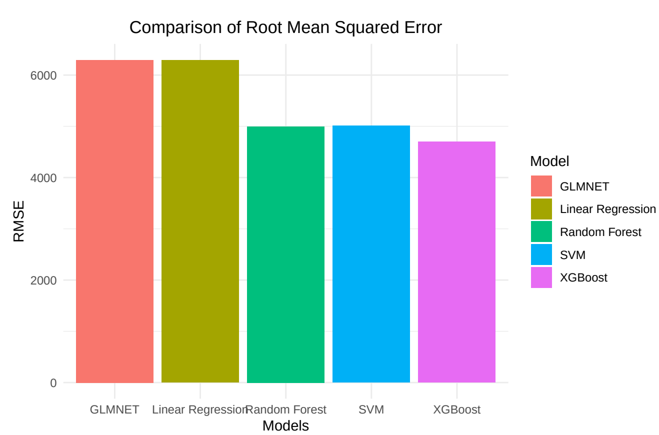
Support Vector Machines (SVM) is a powerful supervised learning algorithm that operates by identifying a hyperplane within the feature space that optimally separates different classes in classification tasks or predicts the target variable in regression scenarios. The SVM model generated through this process is then employed to make predictions on the test set. To evaluate the accuracy of these predictions, metrics such as root mean squared error (RMSE) and R-squared are calculated. SVMs show remarkable effectiveness in handling non-linear relationships within the data, providing a valuable tool for capturing complex patterns. Their robustness in high-dimensional spaces makes SVMs well-suited for diverse machine learning tasks. In this analysis, the 'e1071' package in R is leveraged to implement a Support Vector Machine specifically tailored for regression, showcasing the algorithm's adaptability across a wide range of learning scenarios and in finding meaningful insights from complex datasets.

ANALYSIS OF RESULTS :



In the comprehensive examination of machine learning model performances, the scatter plots and associated fitted regression lines offer valuable insights. Notably, GLMNET and Linear Regression emerge as less effective models, as evident by a noticeable divergence of points from the regression line, indicating suboptimal predictions. On the contrary, Random Forests, Support Vector Machines (SVM), and XGBoost showcase commendable predictive capabilities, with the observed points closely adhering to the regression line. Of particular interest is the remarkable similarity in performance between Random Forests and SVM, suggesting a comparable proficiency in predicting actual values. While on further precise observation, it appears that XGBoost was the winner. In the case of SVM and Random Forests, we see some points being away from the line around the middle region, which is not the case with XGBoost. This visual analysis showcases the effectiveness of different models, providing a subtle understanding of their predictive power and highlighting the strengths and weaknesses in each model.





Based on the above barplots, the XGBoost model has the highest R-squared value, followed by SVM and then Random Forest. This suggests that the XGBoost model is able to explain the most variation in the target variable. GLMNET and Linear Regression are both good regression models, but they are not as powerful as the other three. GLMNET is able to handle non-linear relationships between features and the target variable, while Linear Regression assumes that the relationships are linear. XGBoost is able to handle both linear and non-linear relationships, as well as complex interactions between features.

Another thing to note is that the XGBoost model has the lowest RMSE value, which is another metric for evaluating model performance. RMSE measures the average difference between the predicted and actual values, and a lower RMSE value indicates that the model is making more accurate predictions. There was a close competition between Random Forests and SVM, in both cases of R-squared and RMSE. Overall, the XGBoost model is the best model for the data, as it has the highest R-squared value and the lowest RMSE value. It is a powerful and versatile model that is able to handle a wide range of regression problems.

CONCLUSION :

After performing several exploratory data analysis, we tested various machine learning models to see how well they could make predictions on the insurance charges based on the various other features. We used different algorithms like GLMNET, Linear Regression, Random Forests, Support Vector Machines (SVM), and XGBoost. We measured how accurate the predictions were using metrics like mean squared error and R-squared values.

After looking at our results, it's clear that some models did better than others. GLMNET and Linear Regression didn't perform as well as we hoped since they had trouble capturing the patterns in the data. On the other hand, Random Forests, SVM, and XGBoost performed well, showing that they're effective in making predictions.

We also learned that XGBoost is pretty versatile. It can handle different types of relationships in the data, which is a useful feature. Also, since RMSE was lowest and R-squared was highest for XGBoost, we conclude that the best model out of the ones that were used in the project is XGBoost, for this particular dataset. Overall, this project has equipped us with valuable insights on selecting effective tools for predicting medical insurance costs, enhancing our preparedness for similar challenges ahead.