Santatra Rafidison Rakotondrasoa
Course: Actuarial Science workshop
University of Luxembourg

FINAL PROJECT
–
Predicting the cost of health insurance

INTRODUCTION

The cost of healthcare in the United States is a major concern, affected by factors such as increasing medical costs, lack of insurance coverage, and government regulations. Predictive models using data and machine learning can help predict these costs and aid in reducing expenses. The emergence of data science and machine learning techniques has presented a potential solution for addressing this problem by enabling the prediction of healthcare costs. By leveraging large datasets of historical claims data, demographic information, and other relevant factors, advanced predictive models can be developed to forecast the costs of healthcare for various populations. A machine learning model to predict medical insurance cost* can be useful. First, for individuals, predicting insurance costs can help them make informed decisions about their coverage and budget for healthcare expenses. It can also be useful for insurance companies to predict insurance costs, set premiums and assess risk. Additionally, a model to predict insurance costs can inform policy discussions and decision-making at the government and industry levels. Predicting the cost of health insurance is a challenging task due to the complexity and variability of the healthcare system. However, machine learning algorithms can be used to analyse large datasets and identify patterns that can help predict the cost of health insurance more accurately.

DATA AND ASSOCIATED INSURANCE PROBLEM

We will be using a machine learning model implemented in the R programming language to predict the cost of health insurance for a given individual. We will use the *Medical Cost Personal Datasets* from Kaggle. The data for this project was collected from a sample of health insurance policyholders. It includes information about the policyholders such as their age, sex, body mass index (BMI), number of children covered by health insurance, smoking status, and location. The data also included the charges for each policyholder's health insurance. We will then train a machine learning model on this dataset, using algorithms to detect patterns and relationships between the input variables and the cost of health insurance.

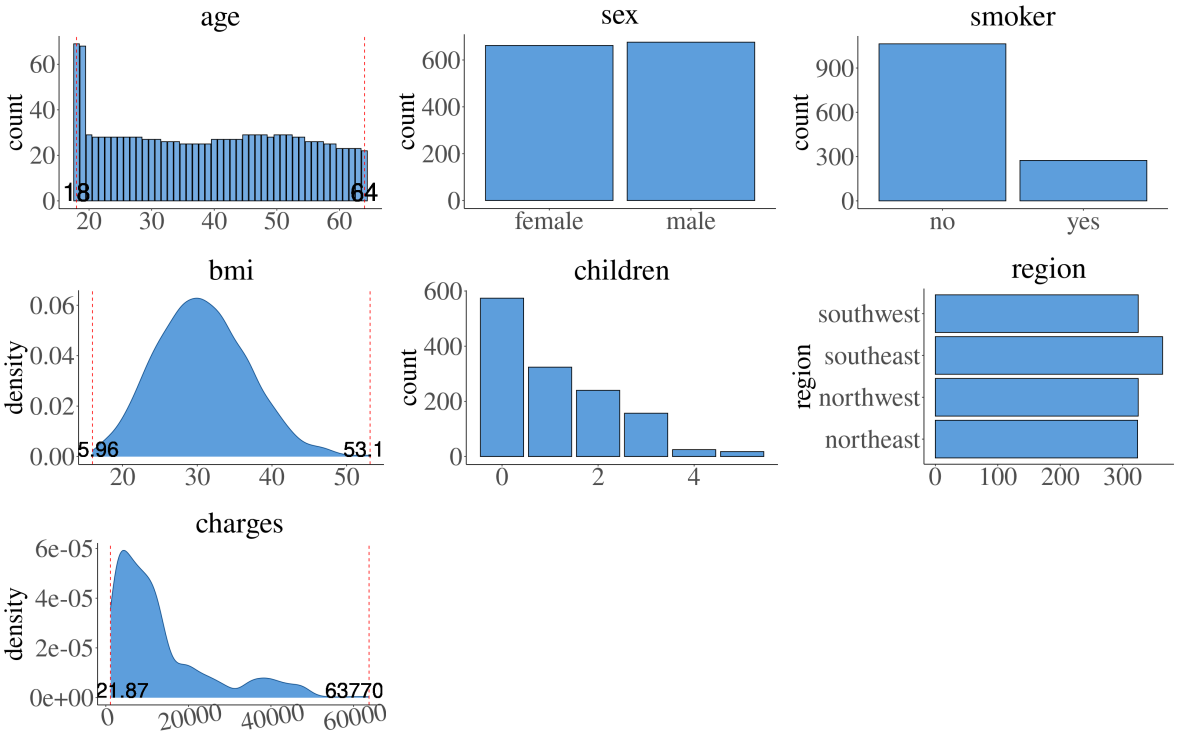
DATA EXPLORATION

- **Data vizualisation**

The following plots represent the statistical analysis of our dataset containing the information of 1338 policyholders:

*__Medical insurance cost__: amount of money that an individual or family pays for health insurance coverage.

Figure 1: Visualization of the data set



❖ **Checking for outliers**

It is important to check for outliers in a dataset as they can have a significant impact on the overall analysis and interpretation of the data. Outliers can skew statistics and lead to flawed conclusions. Therefore we want to identify and treat outliers appropriately by utilizing techniques such as box plots. We used percentiles to identify the lower and upper bounds of the dataset using percentiles. Any data point outside these bounds can be considered an outlier and should be removed. In Figure 2, we plot the box plots for the variable `*age*`, `*bmi*`, and `*charges*`, and in Figure 3, we plot the same plots after removing the outliers.

Figure 2

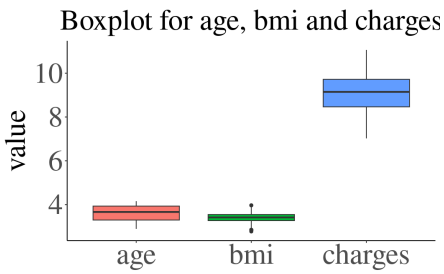
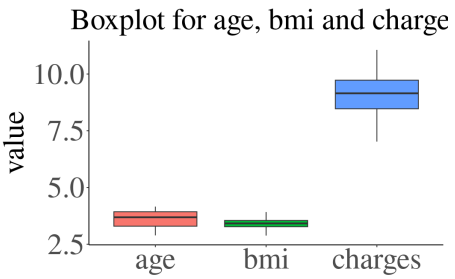


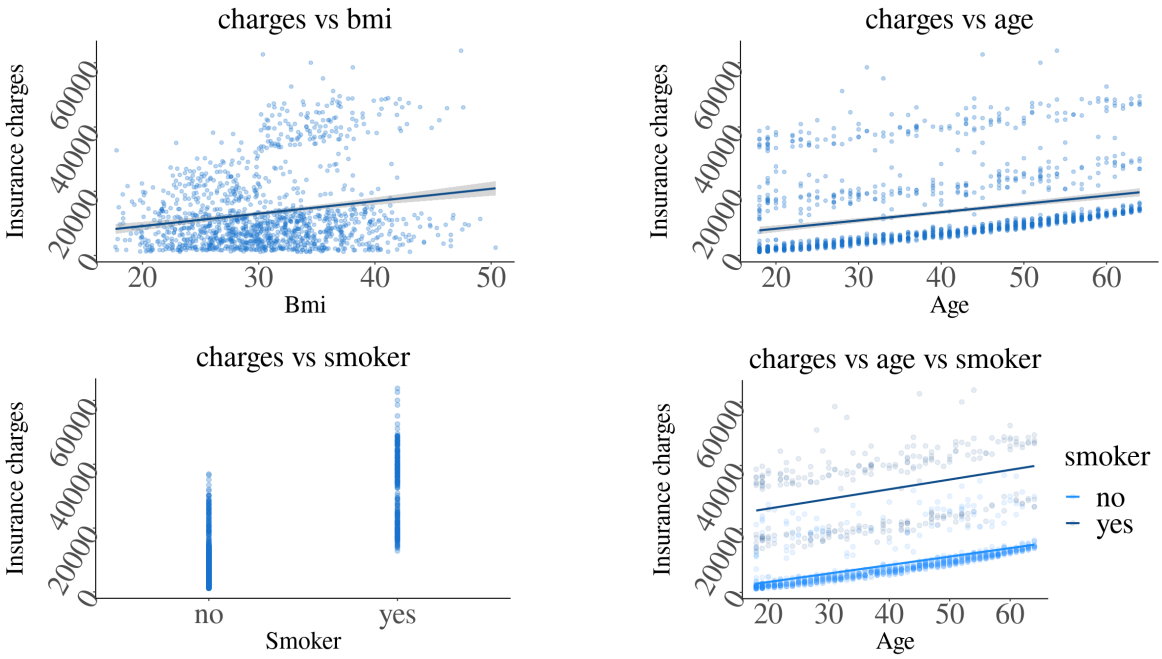
Figure 3



❖ **Correlation analysis in data set:**

Is there any relation between `*age*`, `*smoker*`, `*bmi*` and the insurance *charges*? The following figures show the scatter plots of some pairs of variables.

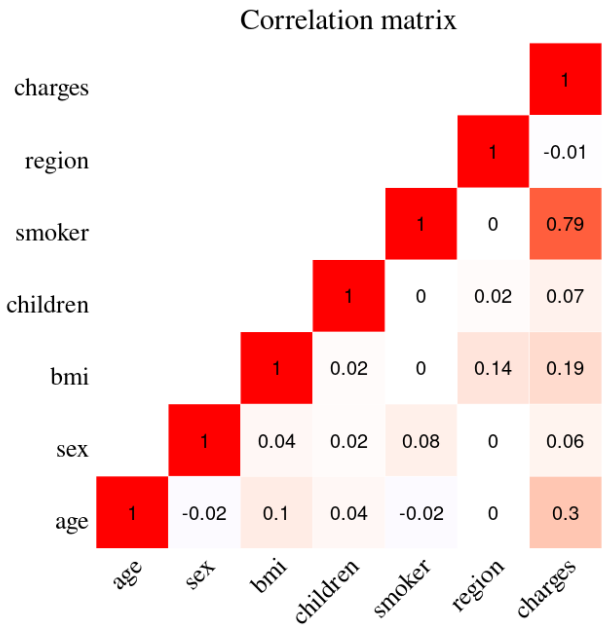
Figure 4



❖ **Correlation matrix and its statistical significance:**

Pearson's technique is the method used for computing the correlation matrix between different variables in a dataset. It is based on the Pearson correlation coefficient, which measures the linear association between two variables. The correlation matrix is a square matrix with the same number of rows and columns as the number of variables in the dataset, with each element in the matrix representing the correlation between a pair of variables.

Figure 5



In order to investigate the association between paired variables, a statistical test for correlation was employed. Specifically, one of Pearson's product moment correlation coefficient, Kendall's τ or Spearman's ρ was utilized to quantify the strength and direction of the association between the paired samples. This approach allows for the assessment of the degree of association between the paired samples and enables the identification of any linear or non-linear relationships present in the data.

Table 1: p-values of the correlation metrics

[1]: "age" [2]: "sex" [3]: "bmi" [4]: "children" [5]: "smoker" [6]: “region" [7]: “charges”

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	NA	0.51	0.00	0.14	0.40	0.91	**0.00**
[2,]	0.51	NA	0.12	0.50	0.01	0.87	0.04
[3,]	0.00	0.12	NA	0.56	0.99	0.00	**0.00**
[4,]	0.14	0.50	0.56	NA	0.86	0.51	0.02
[5,]	0.40	0.01	0.99	0.86	NA	0.97	**0.00**
[6,]	0.91	0.87	0.00	0.51	0.97	NA	0.71
[7,]	0.00	0.04	0.00	0.02	0.00	0.71	NA

- There is a weak positive relationship between `***charges***` and `***age***` that is statistically significant (coefficient of 0.3, p-value of 0.00).
- There is a strong positive relationship between `***charges***` and `***smoker***` that is statistically significant (coefficient of 0.79, p-value of 0.00).
- There is a weak positive relationship between `***charges***` and `***bmi***` that is statistically significant (coefficient of 0.19, p-value of 0.00).

FEATURE ENGINEERING

Since we have explored the characteristics of our dataset through various visualizations and statistical analyses, we can proceed with feature engineering for our machine learning models. This involves selecting and transforming relevant features from the dataset in order to improve the performance and predictive power of the models.

- **Label encoding:**

It is a method of encoding categorical data to be used in a machine learning model. It involves assigning a unique integer value to each category in the data. For example, we have some categorical variables such as sex with two categories: `female` and `male`, and smoker with categories: `yes` and `no`. We encode the data as follows: `female` = 1, `male` = 0 for the variable sex, and `yes` = 1, `no` = 0 for the variable smoker.

- **One-hot encoding:**

It creates a new binary column for each unique category in a categorical variable. For example, the categorical variable region has four categories: southeast, northeast, northwest, and southwest, then four new columns will be created for each of those categories.

- **Generating polynomial features:**

It is a technique used to increase the complexity of a model by adding polynomial terms to the input variables. This is done to capture non-linear relationships between the input variables and the output variable. By adding polynomial terms, the model can fit more complex and non-linear decision boundaries, which can improve the model's performance on the training set. This technique was not used on the Linear Models.

- **Normalization:**

It adjusts the scale of the features so that they are on a similar scale, we want the values of each variable except the target variable to be scaled in the interval [0,1]. This can be especially important in models that use distance measures, such as k-nearest neighbors or support vector machines because these models can be sensitive to the scale of the features. Normalizing the data can also help reduce the time required to train the model, because the optimization algorithm may find a suitable solution more quickly when the features are on a similar scale.

MODELS

❖ **Linear Models:**

- **MM1 - Multiple Linear Model on `smoker`, `age`, and `bmi`:**

We will initially employ a Multiple Linear Model (LM) as the primary analytic technique for the regression task. To begin with, we want to fit the model with the variable `smoker`, `age`, and `bmi` since they correlate the most with the response variable `charges` (highest correlation coefficients in the correlation matrix).

- **MM2 - Multiple Linear Model on all the variables:**

The model was re-evaluated by incorporating all the remaining variables to investigate if a better fit could be achieved. By comparing the results of these two models, it can be determined whether including all the variables improves the model's predictive capability. This approach allows for a more comprehensive examination of the underlying relationships within the dataset and enables the identification of the most suitable model for the analysis.

❖ **Other machine learning models:**

The previous analysis revealed non-linear associations among the variables in the dataset. Therefore, we will use some well-known machine learning models like k-Nearest Neighbors (kNN), support vector regression (SVR), Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) which can capture more complex relationships within data. We used the {caret} R package to perform k-fold cross-validation to tune the hyperparameters, and train and test these models.

- **kNN:**

kNN for regression task predicts a continuous outcome variable based on the average value of the k-nearest data points in the training set. kNN uses a distance metric (such as Euclidean distance) to find the **k**-nearest data points and then takes the average of the output variable for those points to make a prediction. It's a non-parametric method, which means it doesn't make any assumptions about the underlying distribution of the data. It's simple and easy to understand, but the choice of k and the distance metric can affect the performance of the model.

- **RF:**

Random Forest for regression tasks is an ensemble learning method that combines the predictions of multiple decision tree models. A decision tree is built on a random subset of the data and a random subset of the features. The final prediction is made by averaging the predictions of all the decision trees. The **mtry** parameter controls the number of features that are randomly sampled at each split in the decision tree. The range of the mtry parameter depends on the number of features in the dataset. For a dataset with p features, the range of mtry is from 1 to p, where a lower value of mtry corresponds to a smaller number of features being considered at each split and a higher value of mtry corresponds to a larger number of features being considered at each split.

- **SVR:**

Support Vector Regression is a type of linear regression model that uses a technique called "Support Vector Machine" (SVM) to fit a linear model to the data. The goal of SVR is to find the best hyperplane that maximizes the margin of error while keeping the errors within a certain limit. Specifically, we used **SVMRadial** or Support Vector Machine Radial is a variant of SVR, that uses a radial basis function (RBF) kernel instead of a linear kernel. The RBF kernel allows the model to fit non-linear decision boundaries by mapping the input data into a higher dimensional space where the data becomes linearly separable. SVMRadial has several hyperparameters such as the cost **C** which is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the misclassification errors, and the parameter sigma σ determines the width of the RBF kernel and it is inversely proportional to the width of the RBF kernel. A small value of sigma leads to a smooth decision boundary and a low variance model, while a large value of sigma leads to a complex decision boundary and a high variance model.

- **XGBoost:**

It is an advanced implementation of the gradient boosting algorithm for machine learning. It is designed for both speed and performance, and it is a powerful tool for both classification and regression tasks. XGBoost builds an ensemble of decision trees, where each tree is built on the residuals of the previous tree, to minimize the loss function. The final prediction is made by averaging the predictions of all the decision trees. XGBoost is a powerful and flexible algorithm that can be fine-tuned using several hyperparameters like the number of Boosting iterations ***nrounds***, the L1 regularization term α and the L2 regularization term λ that is used to prevent overfitting (a small value corresponds to a weak regularization and allows the model to fit the data closely) and the learning rate parameter η which controls the step size of the gradient descent optimization algorithm.

ANALYSIS OF RESULTS

❖ **Performance metrics:**

We used several evaluation metrics to assess the performance of the machine learning models, including root mean squared error (RMSE), R-squared, and mean absolute error (MAE). We also plotted the predicted charges against the actual charges to visualize the model's performance.

$\hat{Y}_i$: the predicted values.

Y_i : the actual values.

- **RMSE:**

Root Mean Squared Error (RMSE) is a metric used to evaluate the performance of a model. It is calculated by taking the square root of the average of the squared differences between the predicted values and the actual values. Lower value of RMSE indicates a better fit of the model.

$$RMSE = \sqrt{\frac{1}{n}\sum(\hat{Y}_i - Y_i)^2}$$

- **R-squared**:

R-squared is a statistical measure used to evaluate the performance of a model. It ranges from 0 to 1, with a value of 1 indicating a perfect fit and a value of 0 indicating no fit. It measures the proportion of the variance in the dependent variable that is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum(\hat{Y}_i - Y_i)^2}{\sum(\hat{Y}_i - \bar{Y}_i)^2}$$

- **MAE:**

It measures the average magnitude of the errors in a set of predictions, without considering their direction. It is calculated as the sum of the absolute differences between the predicted values and the actual values, divided by the number of observations.

$$MAE = \frac{1}{n}\sum |\hat{Y}_i - Y_i|$$

- **Results:**

We performed **grid search** and **cross-validation** for the hyperparameter tuning on each of the non-linear models to optimize the hyperparameters and identify the best model. At the end, we then evaluated the performance of the models:

Table 2: Performance comparison for all the models

Models	Train set			Test set			
	RMSE	R-squared	MAE	RMSE	R-squared	MAE	Hyperparameters
MM1	6129.66	0.64	4241.86	5994.48	0.67	4360.65	-
MM2	6093.39	0.64	4205.57	5942.73	0.67	4318.94	-
k-NN	4594.62	0.81	2733.39	5638.22	0.74	3539.88	k=4
RF	**2638.10**	**0.94**	**1394.42**	4719.03	**0.83**	2707.27	mtry=4
SVMRadial	4734.85	0.80	2441.28	**4710.65**	0.82	2590.82	C=1, σ=0.7
XGBoost	3405.94	0.90	1764.15	4749.20	**0.83**	**2510.87**	α=2.5, λ=2.4, η=0.001, *nrounds*=12

We observed that XGboost and SVMRadial are the best models where as Random Forest is overfitting on the train data. The following figures present the plot comparing the predictions and the true values (Figure 6) and the residual plot (Figure 7) for XGboost on the test data.

Figure 6:

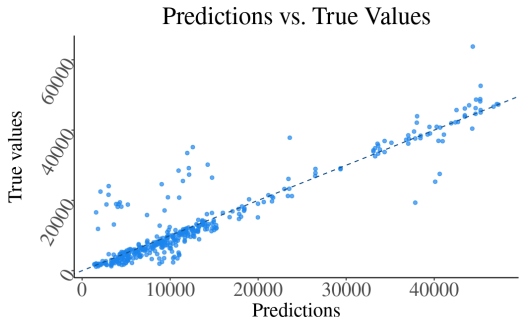
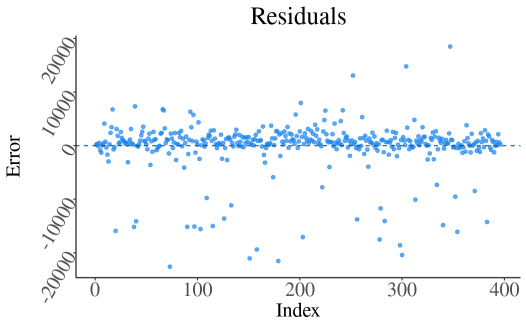


Figure 7:



CONCLUSION

The goal of this project was to predict insurance costs using a machine learning regression model. Predicting insurance costs can be beneficial for individuals and insurance companies. It can help individuals make informed decisions about their coverage and budget for healthcare expenses. For insurance companies, it can help them to set premiums and assess risk. Additionally, it can inform policy discussions and decision-making at the government and industry levels.

The machine learning models performed well in predicting health insurance charges despite being trained on a limited dataset. The XGboost and SVMRadial models had the best performance on the test data. However, all of the models had a good performance, and the choice of the model may depend on the specific needs and goals of the user.

REFERENCES

[1] {ggplot2} package documentation:
https://www.rdocumentation.org/packages/ggplot2/versions/3.4.0

[2] {caret} package documentation:
https://topepo.github.io/caret/

[3] Medical Cost Personal Datasets:
https://www.kaggle.com/datasets/mirichoi0218/insurance