PSTAT 231 Final Project

Kai Wa Ho

2024-03-08

Introduction

The primary objective of this initiative is to design and implement a predictive model that accurately forecasts the health insurance premiums charged to individuals. This involves leveraging historical data and various predictive analytics techniques to estimate the cost of insurance coverage based on a range of factors, including but not limited to, individual health profiles, demographic details, and possibly lifestyle choices. The development of such a model aims to enhance the precision of premium calculations, enabling more personalized and fair pricing strategies within the health insurance sector.

Background

What is an Insurance Premium?
An insurance premium refers to the sum paid by an individual or a business for an insurance contract. This payment is made for various types of insurance coverage, including health, automotive, property, and life insurance. Once these premiums are collected, they constitute revenue for the insurance firm. However, they also entail a financial obligation on the part of the insurer, as there is a need to offer protection against any claims made under the policy. If the premium is not paid by the individual or the business, the insurance agreement can be terminated.

Goal

When an individual enrolls in an insurance plan, they are required to pay a premium to their insurance provider. This premium is essentially the cost of the insurance coverage. Policyholders have various payment options at their disposal. Some insurance companies offer the flexibility of paying premiums in installments, either on a monthly or semi-annual basis, while others might necessitate a full payment upfront before activating any coverage. The amount of the premium is influenced by multiple factors, such as the nature of the coverage, the age of the insured, the geographic location of the insured person, any previous claims made, as well as issues related to moral hazard and adverse selection, among others.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## -- Attaching packages --------------------------------------- tidymodels 1.1.1 --
##
## v broom        1.0.5     v rsample      1.2.0
## v dials        1.2.0     v tune         1.1.2
## v infer        1.0.5     v workflows    1.1.3
## v modeldata    1.3.0     v workflowsets 1.0.1
## v parsnip      1.1.1     v yardstick    1.3.0
## v recipes      1.0.9
##
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
##
##
## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
##
##
##
## Attaching package: 'discrim'
##
##
## The following object is masked from 'package:dials':
##
##     smoothness
##
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Loaded glmnet 4.1-8
##
## corrplot 0.92 loaded
##
## randomForest 4.7-1.1
##
```

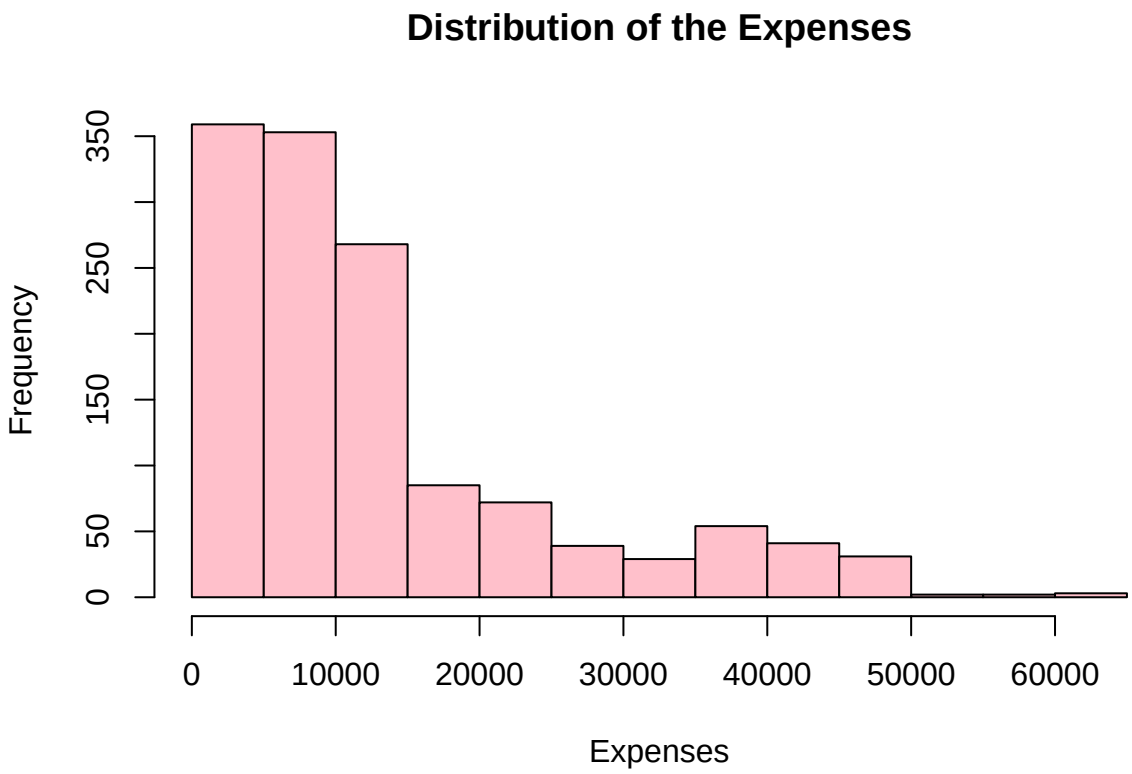
```
## Type rfNews() to see new features/changes/bug fixes.
##
##
## Attaching package: 'randomForest'
##
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
##
##
## Attaching package: 'xgboost'
##
##
## The following object is masked from 'package:dplyr':
##
##     slice
##
##
## Loading required package: rpart
##
##
## Attaching package: 'rpart'
##
##
## The following object is masked from 'package:dials':
##
##     prune
##
##
##
## Attaching package: 'vip'
##
##
## The following object is masked from 'package:utils':
##
##     vi
##
##
##
## Attaching package: 'ranger'
##
##
## The following object is masked from 'package:randomForest':
##
##     importance

## Rows: 1338 Columns: 7
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): sex, smoker, region
## dbl (4): age, bmi, children, expenses
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

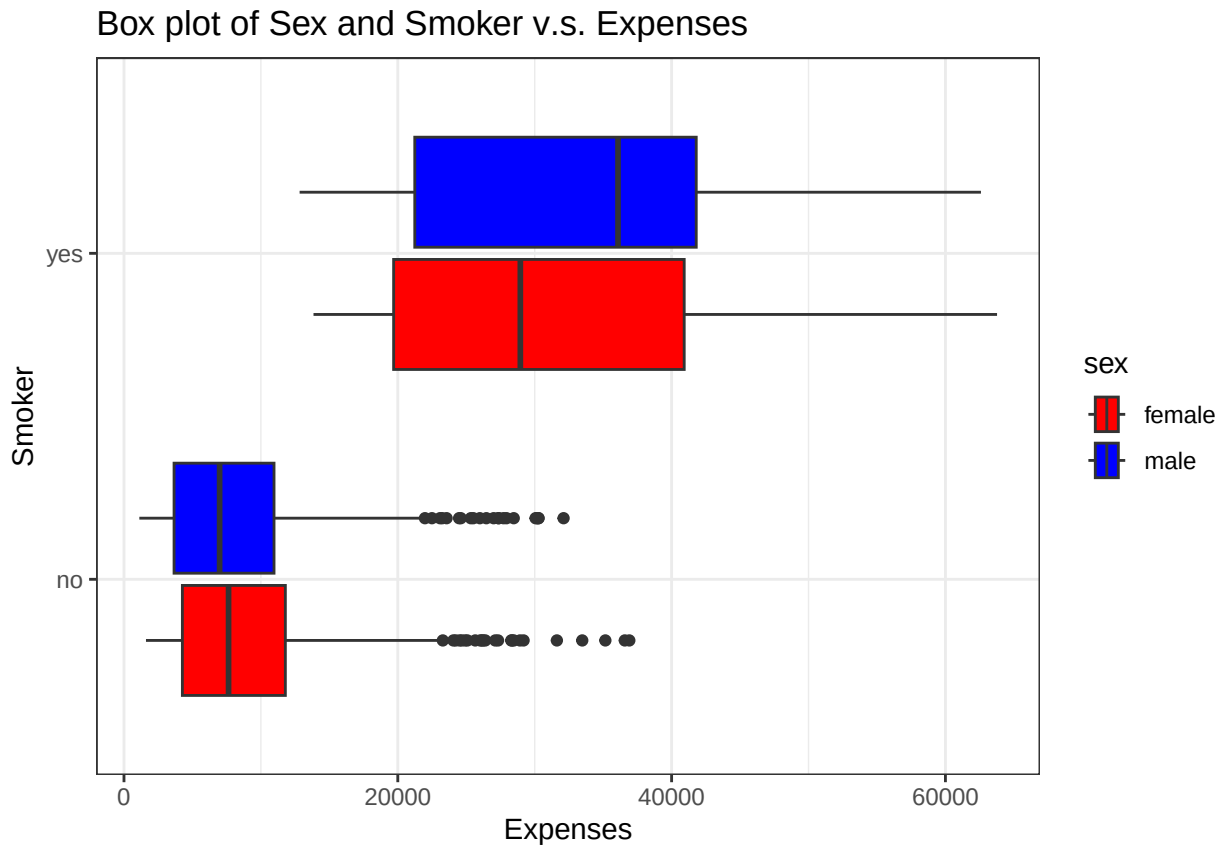
```
## # A tibble: 6 x 7
##     age sex      bmi children smoker region    expenses
##   <dbl> <chr>  <dbl>    <dbl> <chr>  <chr>        <dbl>
## 1    19 female  27.9        0 yes    southwest   16885.
## 2    18 male    33.8        1 no     southeast    1726.
## 3    28 male    33          3 no     southeast    4449.
## 4    33 male    22.7        0 no     northwest   21984.
## 5    32 male    28.9        0 no     northwest    3867.
## 6    31 female  25.7        0 no     southeast    3757.
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :16.00   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.67   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.70   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.10   Max.   :5.000
##     smoker             region             expenses
##  Length:1338        Length:1338        Min.   : 1122
##  Class :character   Class :character   1st Qu.: 4740
##  Mode  :character   Mode  :character   Median : 9382
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

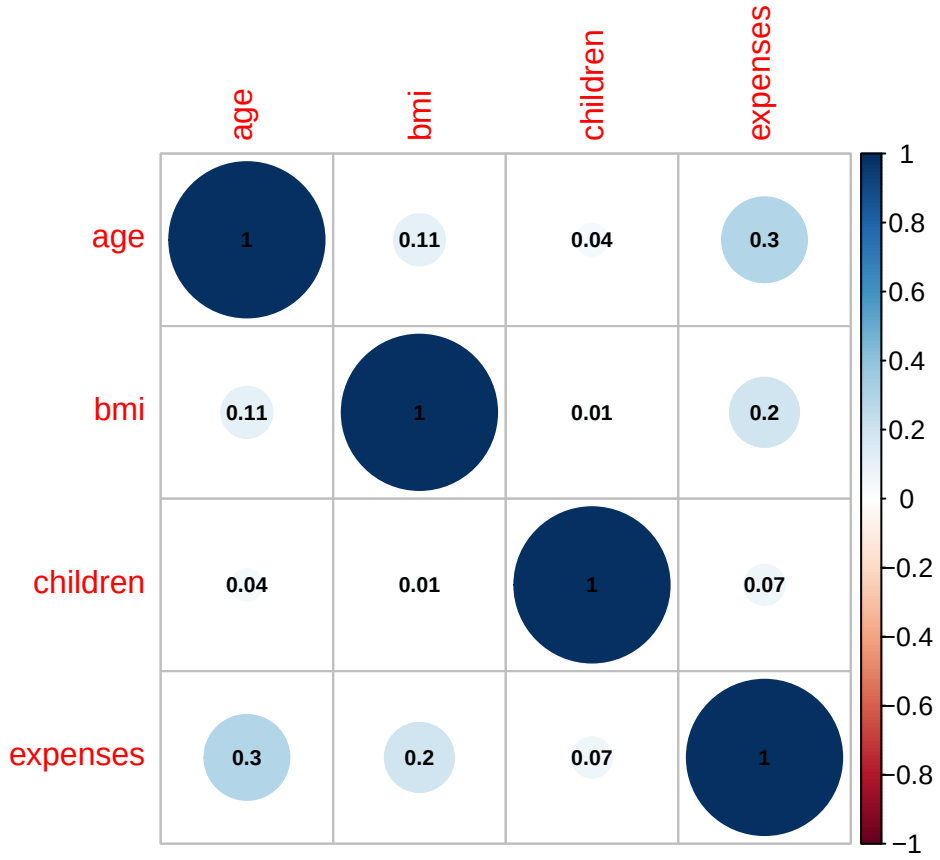
```
## No missing value.
```



The range of expenses in our dataset extends from a minimum of 0 to a maximum of 65,000, with a significant concentration of expenses occurring within the span of 1,500 to 15,000.

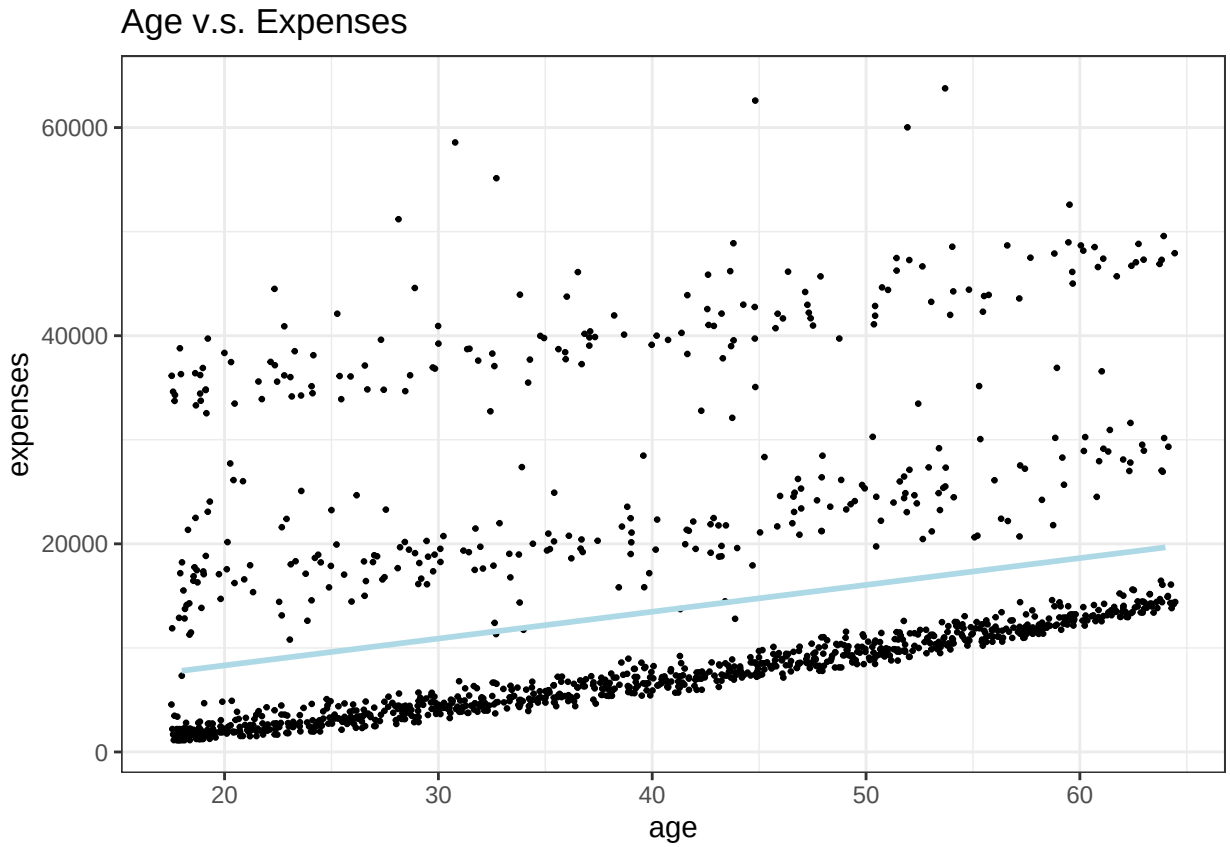


This graph clearly illustrates that smokers face considerably higher insurance premiums compared to non-smokers. Within the non-smoker category, women's premiums are slightly above those for men. Conversely, in the smoker group, men's premiums are marginally higher than women's. Additionally, the data for non-smokers include several outliers.



The plot reveals that there is a positive correlation between all the predictors and insurance costs, with age displaying the most significant positive association with expenses. Conversely, the number of children shows the weakest positive link with the costs.

```
## `geom_smooth()` using formula = 'y ~ x'
```



Three distinct patterns of positive correlations are observed, potentially influenced by factors such as gender, body mass index (BMI), and smoking habits. This observation aligns with the expectation that as individuals age, their health conditions tend to deteriorate.

Model Development

Now, we're at the stage where we can begin applying models to our dataset to ascertain whether it's possible to forecast insurance premiums using the available predictors. Before proceeding, we need to prepare our data, which involves dividing it into subsets, establishing a recipe for processing, and generating partitions for k-fold cross-validation.

Before we begin modeling, the initial step involves dividing our dataset into two parts: a training set and a testing set. The training set's purpose is to train our models, while the testing set serves as a means to evaluate the models' performance on unseen data. This evaluation typically revolves around identifying the model that yields the "best" performance, often determined by the lowest Root Mean Squared Error (RMSE) in regression tasks. Employing this method ensures that we mitigate the risk of overfitting, as the model does not learn from the entirety of the data. I have opted for an 80/20 division, allocating 80% of the data for training and the remaining 20% for testing. This approach ensures that the majority of the data is utilized for training, while still reserving a sufficient portion for an effective evaluation of the model's capabilities on new data.

During our analysis, we'll consistently employ the same set of predictors, conditions, and desired outcome. To streamline this process, we plan to devise a single, comprehensive recipe that will be applicable across all our models (with minor modifications as necessary). This strategy allows each model to apply its specific techniques and methodologies using this standardized recipe.

K-Fold Cross Validation

To implement k-fold stratified cross-validation, we'll organize our training data into 5 separate segments, known as folds. In this process, each piece of the training dataset is allocated to one of the five folds. For every individual fold, a test set is created from that specific fold, while the combination of the remaining four folds serves as the training set for that iteration. This procedure results in a total of five distinct folds.

The essence of k-fold cross-validation lies in dividing the dataset into k parts, as mentioned, with each part once serving as the test set and the other k-1 parts combining to form the training set. This cycle repeats for each fold. We then fit the model to each of these training sets and evaluate it against the corresponding test sets. The performance of the model is assessed by averaging the accuracy across all test sets from each fold, providing a comprehensive measure of the model's effectiveness.

Fitting models

We've reached the stage where it's time to construct our models. Due to the extensive computation time required by these models, we've opted to save the results from each model to bypass the need for repetitive executions. For evaluating model performance, I've selected the Root Mean Squared Error (RMSE) as the primary metric. RMSE is widely recognized for its effectiveness in assessing regression models, as it quantifies the discrepancy between the predicted values and the actual values using the Euclidean distance. Consequently, a model with a lower RMSE is preferable, indicating that its predictions are closer to the actual figures. Given that RMSE is a measure of distance, normalizing our data, which was accomplished during the recipe phase, is crucial for accurate model evaluation.

Recipe

To begin, we'll initiate the modeling process by defining the model we intend to use. Next, we'll establish a workflow for this model by incorporating both the model itself and the recipe we've previously prepared. Following this, it's necessary to construct a tuning grid that outlines the parameters we plan to adjust, including the specific range for each parameter and the number of levels we'll explore. Finally, with the workflow, the k-fold cross-validation folds, and the tuning grid ready, we can proceed to the tuning phase, where we'll adjust our selected parameters to optimize the model's performance. In the final step, we gather the performance metrics for the optimized models and sort them by the mean RMSE in ascending order to identify the model with the lowest RMSE. We then select this model, noting its RMSE value, and store it in a variable for subsequent comparison purposes.

Workflow

Tuning Grid

Tunning Grid

```
## > A | warning: A correlation computation is required, but ‘estimate‘ is constant and has 0
##                standard deviation, resulting in a divide by 0 error. ‘NA‘ will be returned.

## There were issues with some computations   A: x1There were issues with some computations   A: x2There
```

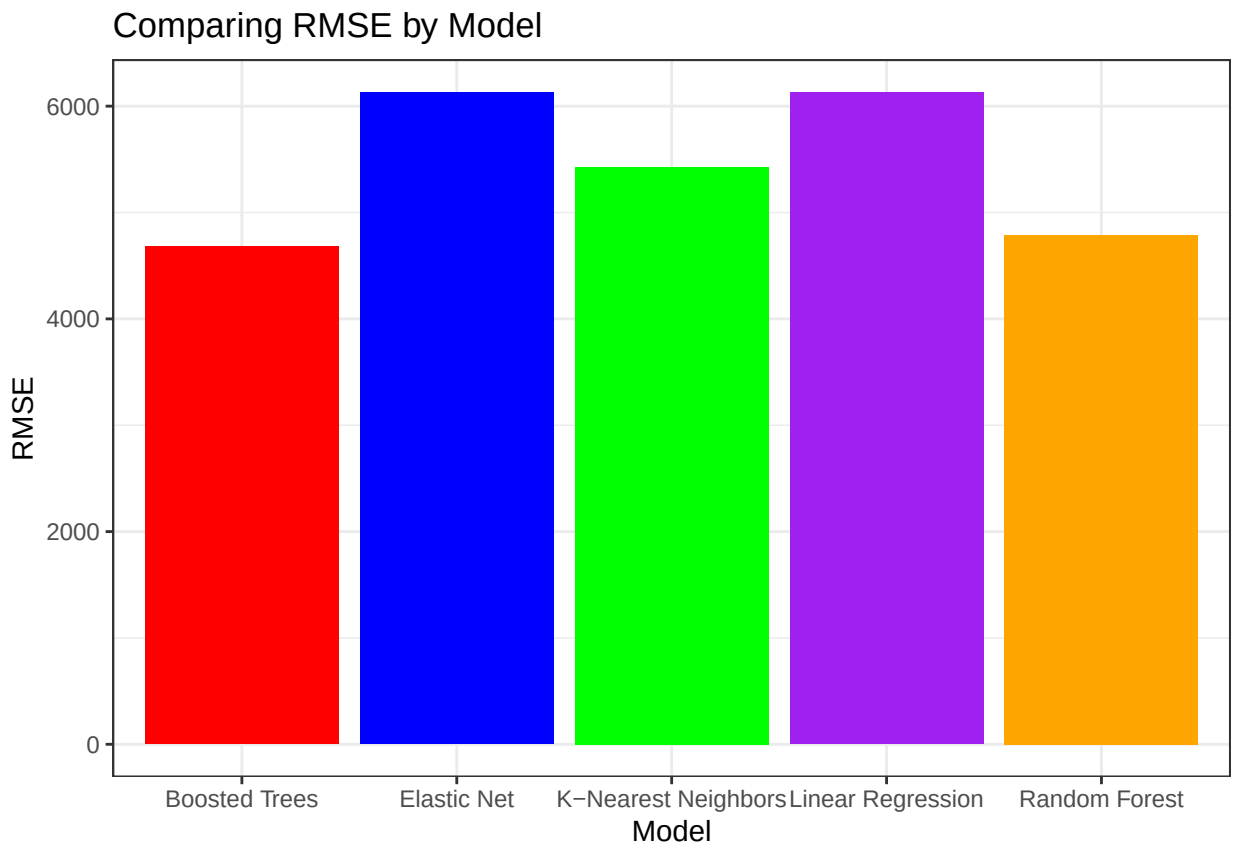
Collect Metrics

```
# Linear Regression
lm_fit <- fit_resamples(lm_workflow, resamples = insurance_folds)
lm_rmse <- collect_metrics(lm_fit) %>%
  dplyr::slice(1)

# KNN model
knn_rmse <- collect_metrics(knn_tunned) %>%
  arrange(mean) %>%
  dplyr::slice(6)

# Elastic Net
elastic_rmse <- collect_metrics(elastic_tunned) %>%
  arrange(mean) %>%
  dplyr::slice(44)

# Random Forest
rf_rmse <- collect_metrics(rf_tunned) %>%
  arrange(mean) %>%
  dplyr::slice(193)

# Boosted Trees
boosted_rmse <- collect_metrics(boosted_tunned) %>%
  arrange(mean) %>%
  dplyr::slice(126)
```

Model Result

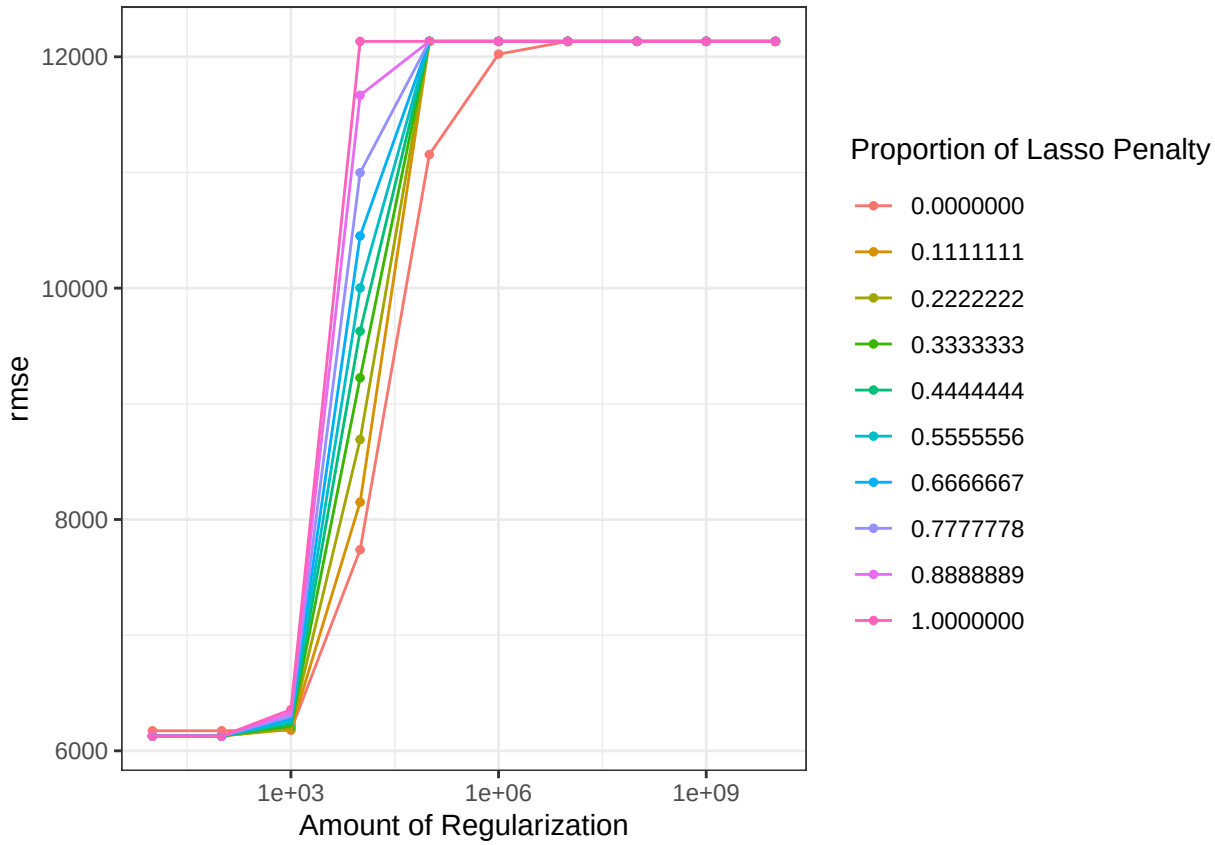
At this juncture, it's important to evaluate and contrast the outcomes from all our models to determine which have yielded the most favorable results.

```
## # A tibble: 5 x 2
##   Model                RMSE
##   <chr>               <dbl>
## 1 Boosted Trees       4682.
## 2 Random Forest       4790.
## 3 K-Nearest Neighbors 5428.
## 4 Elastic Net         6127.
## 5 Linear Regression   6130.
```

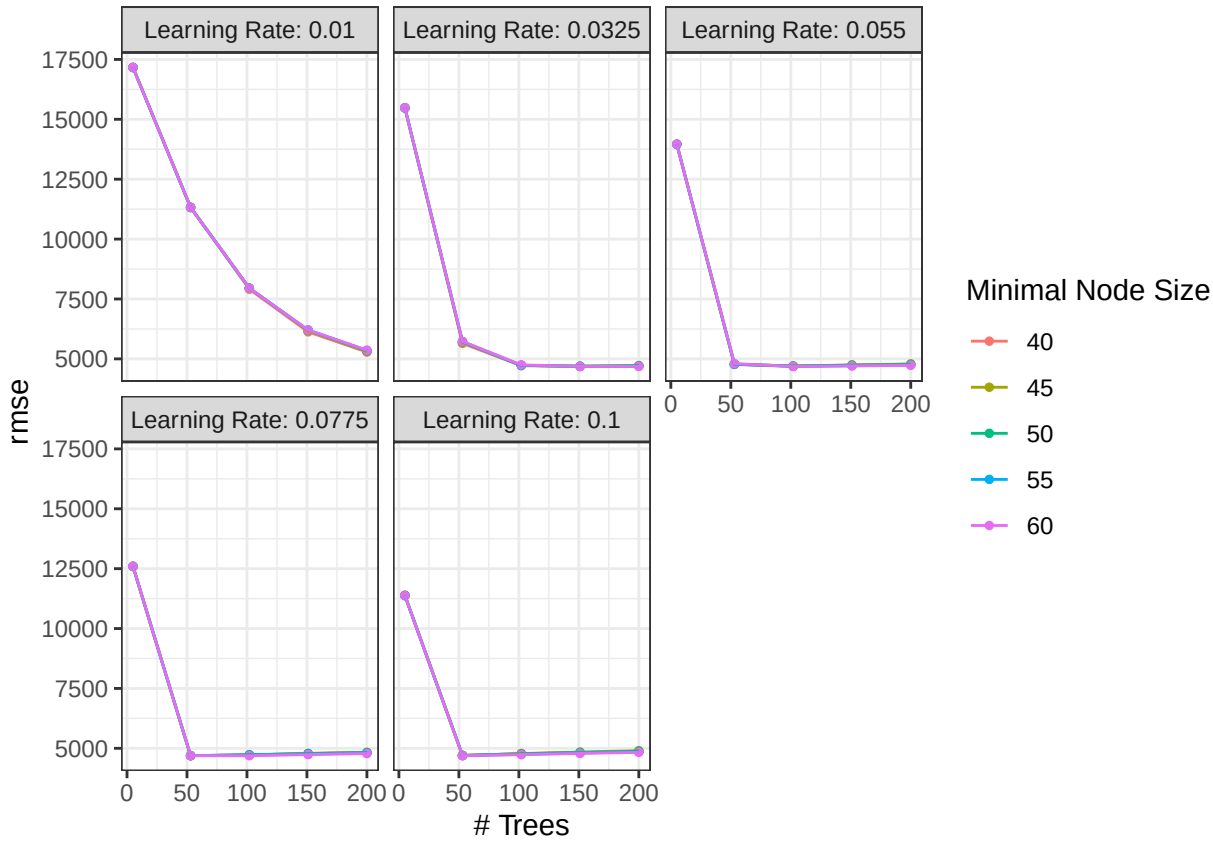


Model Autoplots

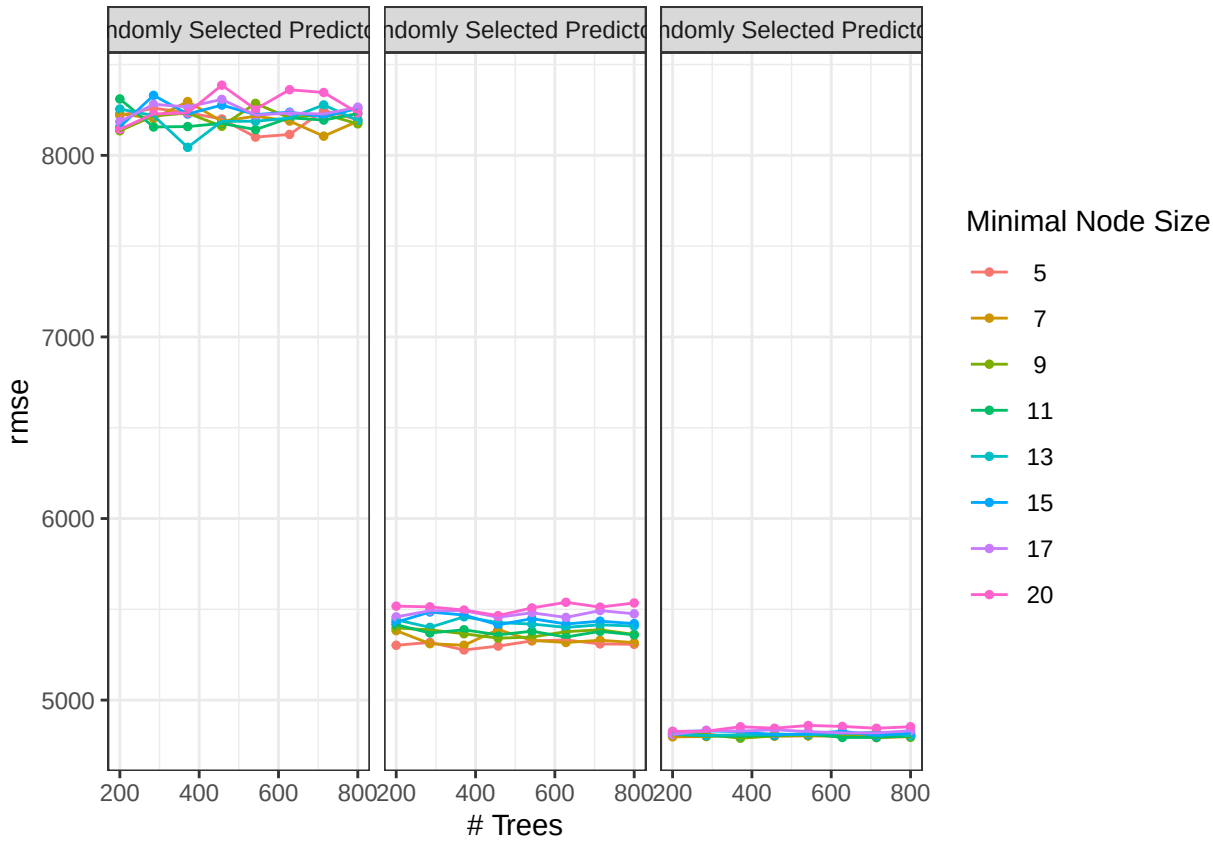
The autoplot function in R provides a graphical representation that illustrates how the tuning of each parameter influences each model's effectiveness. Through these plots, we assess model performance based on the RMSE metric, with a lower RMSE indicating superior performance by the model.



In the case of the elastic net model, we adjusted the penalty and mixture parameters across ten distinct levels. The visual analysis indicates that penalty values ranging from two-thirds to the maximum tend to yield the best results. Conversely, as the penalty term decreases, the model's performance deteriorates.



In the boosted trees model, the parameters we fine-tuned included the learning rate, the total number of trees, and the minimum node size, each at five varying levels. It's worth mentioning that prior adjustments were made to the model before setting the final parameter ranges. The model exhibited poor performance at higher learning rates. While a high learning rate enables quicker learning, it also results in less thorough training and reduces the model's ability to generalize. Consequently, we opted for a learning rate range between 0.01 and 0.1 to determine the optimal lower learning rate for enhanced performance.



Result

```
## # A tibble: 1 x 9
##   trees min_n learn_rate .metric .estimator  mean     n std_err .config
##   <int> <int>      <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1   102    60      0.055 rmse    standard   4682.     5    228. Preprocessor1_M~
```

Fitting

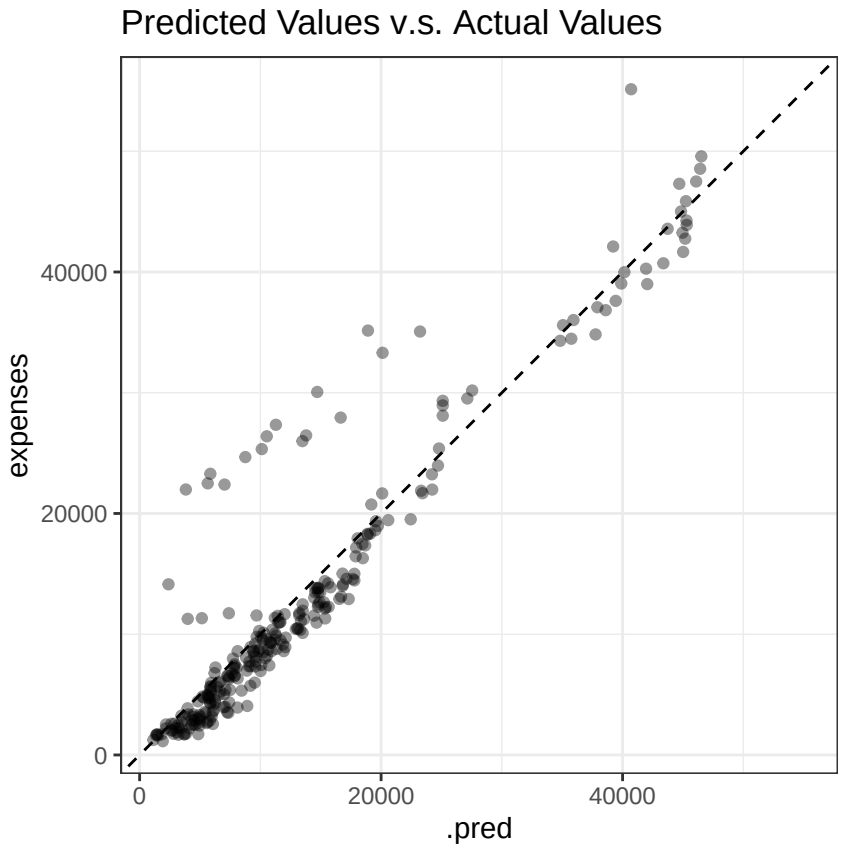
Testing

Final Model

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       4230.
```

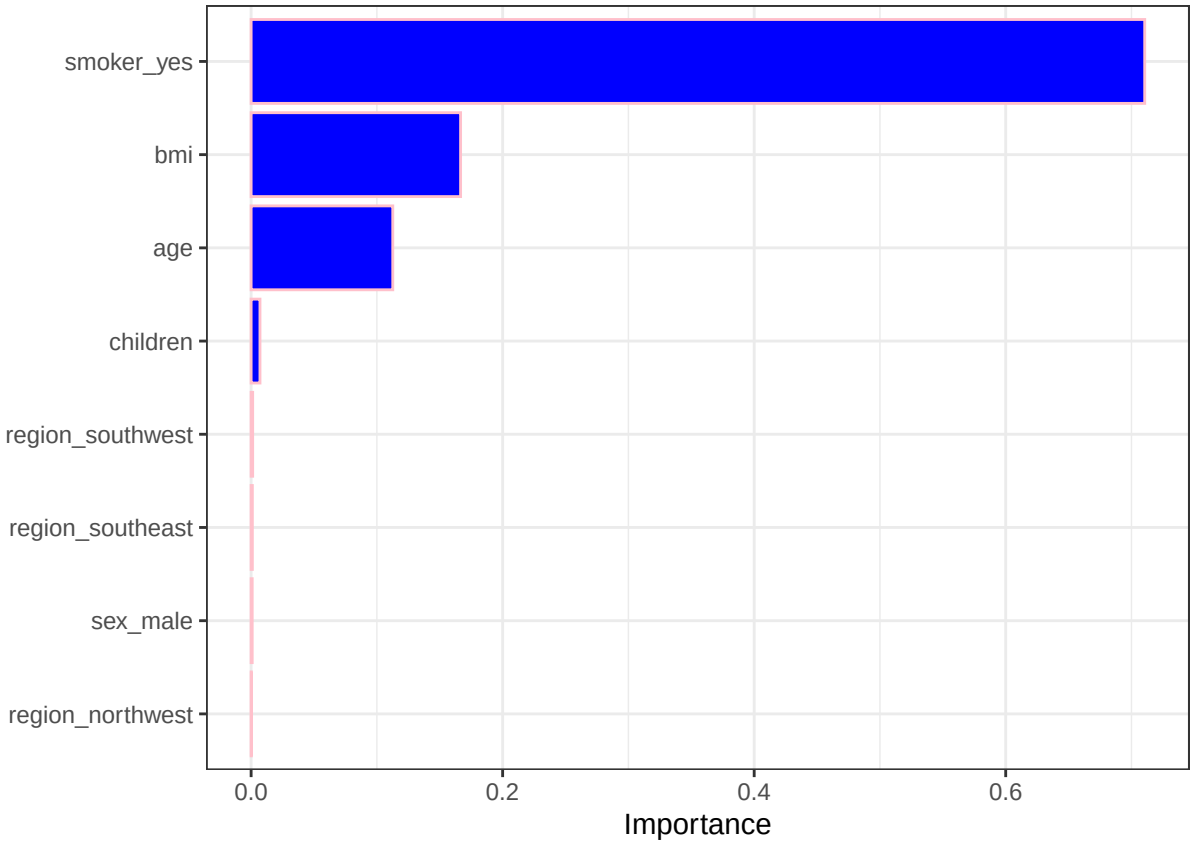
Interestingly, our Boosted Trees model showed improved performance on the testing set compared to the cross-validation folds, achieving an RMSE of 4557.3. Despite this improvement, the RMSE remains relatively high, which could be attributed to the limited number of predictors used and the wide range of values in our outcome variable, charges.

Visualization of predicted value v.s. actual value



Should each observation be predicted with perfect accuracy, the dots on the plot would align to create a straight line. Observing the plot, we notice that a majority of the dots indeed align with the line, suggesting that the model was quite effective in forecasting insurance premium rates. It's also encouraging to note that the model did not generate any negative predictions for insurance charges. Nonetheless, a number of dots are positioned above the line, indicating underestimation of some insurance charges. In summary, the Boosted Trees model performed commendably in predicting premium charges, yet there remains scope for further refinement.

Importancy of Variables



This analysis confirms that the predictor 'smoker' plays the most significant role in determining the outcome, aligning with our initial expectations. While other variables also contribute to the predictive accuracy, 'smoker_yes' emerges as the overwhelmingly dominant factor. This observation is logical, considering smokers are more likely to suffer from a range of chronic conditions, leading to higher medical expenses. On the other hand, the predictors 'sex' and 'region' appear to have negligible influence on the prediction of expenses in our model.

Conclusion

Throughout this project, we embarked on a journey of data exploration and analysis, examining various variables with the aim of developing a reliable model to forecast insurance premium costs based on factors like gender, age, smoking status, body mass index (BMI), the number of children, and geographical region. After rigorous evaluation and computation, we identified the Boosted Trees model as the most effective in predicting insurance premiums, though it is acknowledged that the model is not without its limitations and there is scope for enhancement.

Looking ahead, exploring the potential of neural networks presents an intriguing avenue for further research. Such an approach would demand increased computational resources and could extend the analysis to a broader set of predictors. Moreover, leveraging a neural network to analyze Instagram images to detect specific colors associated with beautiful sunsets—like purple, red, orange, pink, and yellow—offers a novel application. Training this network on thousands of images could enable it to recognize these colors in new images, potentially improving the model's performance in terms of accuracy and reducing the Root Mean Squared Error (RMSE) even further. An additional step I'm considering is to incorporate more region-related

predictors into the model, hypothesizing that geographical factors could significantly influence premium costs. Ultimately, it is envisioned that this enhanced model could serve as a benchmark for insurance companies to evaluate their premium pricing models.

This project has been a valuable platform for honing my skills in machine learning and data analysis. Delving into this dataset ignited a deeper interest in understanding the nuances of my data and in discovering the most suitable predictive model. As an Actuarial Science major, this project holds profound personal significance, given its alignment with my future career aspirations in insurance premium calculation, rate making, and risk assessment. This endeavor has not only enriched my academic journey but also solidified my passion for actuarial science, inspiring me to continue pursuing opportunities where I can apply and expand my knowledge in this field.

Sources

The data is obtained from Kaggle, by VENKAT MURALI
→link: https:// www.kaggle.com/ code/ venky12347/insurance-premium/input