For the classification problem, our underlying question was if we could predict the Risk Profile of an individual based on Premium Amount, Credit Score, and Age. Usually, the premium amount would be determined based on their risk profile, among other variables, but let us assume that the risk profile column in the data was unintentionally purged and we were tasked with reallocating everyone to the proper risk profile category based on these input variables. Other variables were available, including several categorical variables that could prove as good predictors to determine the individual’s risk, but we decided on these three numerical variables. Credit score and age usually play into the initial determination of risk profile, and premium is generally an output from the risk profile, so it made sense to choose these variables as they were on “both sides” of the risk profile.

Our first model was a Support Vector Machine (SVM). For this SVM…(Vincenzo)

We then moved from industry to flora…from machines to trees. Our first model in this arena was a basic Classification/decision Tree (CART). We started off with the default settings of the decision tree model as seen in the figure below.

A white background with black text

Description automatically generated

Unfortunately, this most basic tree provided us with little useful information as its accuracy was identical to that of the No Information Rate (31.11% in-sample, 30.92% out-of-sample (see App.C.1-2)). The model allocated all observations to category 3 of the risk profile. The very high P-value of 0.5019 and 0.5042 in the in-sample and out-of-sample model, respectively, and the Kappa values of 0, suggest a lack of statistical significance.

We then decided to auto-tune the basic tree model and noticed that the regular grid parameters decided on a tree depth of 1, which was the same as the basic CART, so we changed it to the random grid which provided us with the below parameters.

A black text on a white background

Description automatically generated

However, changing the cost complexity, tree depth, and minimum number of observations for a split still only gave us an in-sample accuracy of 31.08% and an out-of-sample accuracy of 31.21%. Recall the basic CART model assigned a risk profile of 3 to every observation, in this automatically tuned model, it assigned either a 1 or a 3. Improvement, but minimal.

Moving to the Bagging model, we ran the tune and it resulted in the following specifications as seen below.

A white background with black text

Description automatically generated

With a greater tree depth, we expected our accuracy to improve, and it did, but barely: 35.78% in sample, and 31.31% out-of-sample. On the plus side, the model was finally allocating the observations to all four categories of risk profile (0 to 3).

Next, we estimated a tuned Random Forest Model. We ran a quick variable importance and noticed the following. The premium amount is usually a result of the risk profile, so it makes sense for it to be of most importance.

A graph with a number of text

Description automatically generated with medium confidence

However, our confusion matrices yielded some interesting results. Providing for some brief excitement, the in-sample accuracy skyrocketed to 82.25%, a very small P-value (<2.2e-16), and a large Kappa value of 0.7565. The out-of-sample accuracy metrics brought us back to reality and somehow gave only a 29.30%, which is the first and only time it was lower than the No Information Rate. Additionally, the P-value was just under 1.

Finally, we ran the Gradient Boosted Model which resulted in an in-sample accuracy of 33.03% and an out-of-sample accuracy of 32.47%. In each of our models, the out-of-sample accuracy was less than the in-sample accuracy, which makes sense, but no one model stood out as the “best” for predicting risk profile. Below is a figure compiling all our accuracies.

|  |  |  |
| --- | --- | --- |
| Model | In-sample | Out-of-Sample |
| NIR | 31.11% | 30.92% |
| CART | 31.11% | 30.92% |
| Tuned CART | 31.65% | 31.21% |
| Bagged Forest | 35.78% | 31.31% |
| Random Forest | 82.25% | 29.30% |
| Gradient Boosted | 33.03% | 32.47% |

Although the Random Forest model gave us the highest in-sample accuracy, it also gave us the lowest out-of-sample accuracy. The highest out-of-sample accuracy came from the gradient boosted model. For this reason, we resorted to using it as our model for the validation set. Running the validation set on the tuned gradient boosted model resulted in an accuracy of only 32.45%, luckily higher than its No Information Rate of 31.32%, but still very low and with a P-value of only 0.01489.