Random Forest is an ‘ensemble’ model that fits based on majority voting from numerous decision trees which corrects for overfitting. As Random Forest is a non-parametric algorithm, it requires little data preparation beforehand. Variables can be ranked according to importance based on Gini index, though how a variable affects final output is less interpretable than logistic regression.

The first iteration of training the Random Forest Classifier yielded 88.3% overall Accuracy, but a low Sensitivity of 27.96%, meaning it could correctly predict fewer than one-third of the cases where death occurred. We hypothesized that the low sensitivity might be due to an imbalance within the dataset (deaths occurred in <5% of all cases), and that undersampling the majority class (no death) or oversampling the minority class (death) might increase accuracy.

Using the **caret** package, the Random Forest model was trained additionally using the following sampling techniques: down (simple random undersampling an equivalent number of cases where patients did not die), ROSE (Random Over-Sampling Examples), and SMOTE (Synthetic Minority Oversampling Technique). The results are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | original | down | ROSE | SMOTE |
| Accuracy | 88.29% | 81.10% | 91.39% | 84.06% |
| Specificity | 96.70% | 79.16% | 90.33% | 84.99% |
| Sensitivity | 27.96% | 83.05% | 92.45% | 82.82% |
| Bal. Accuracy | 62.44% | 81.11% | 91.39% | 83.90% |

The different sampling methods to balance the data all resulted in an increase in sensitivity as hoped, though overall accuracy fell for down-sampling and SMOTE due to a decrease in detecting true negative cases. However, the ROSE method greatly increased sensitivity while preserving specificity, thus leading to a higher overall accuracy and much higher balanced accuracy.

Gradient boosting is a method of converting weak learner into strong learners. The model begins by training a decision tree with equal weight, and then we will increase the weights of those observations that are difficult to classify and lower the weights for easy ones to create a new tree. Our model is therefore combination of tree 1 and tree 2. Gradient boosting identifies the shortcomings by using gradients in the loss function (y=ax+b+e, e needs a special mention as it is the error term). The loss function is a measure indication how good are model’s coefficients are at fitting the underlying data.

We used the package “gbm” to train the gbm model and used cross-validation method to determine the best iteration. There are several ways we can choose for distribution of our response variable, for example-“bernoulli” (logistic regression for 0-1 outcome), “gaussian”(squared errors), “tdist” (t-distribution loss), we used bernoulli for our gbm model. The overall accuracy for this model is 88.67%.