Predicting Extended Absence time

Rose Gogliotti

East

Group 2

Other Group 2 Members:

Pavel, Sarah, Shravya and Catherine

Table of Contents

Introduction

With this project we set out to apply a classification problem to the Absenteeism dataset. The Absenteeism dataset was generated to be used in academic research at the Universidade Nove de Julho Postgraduate Program in Informatics and Knowledge Management. The data was generated from 27 unique individuals and tracks their absenteeism in hours as well as characteristics about their employment, personal information and information on the absence. Using this data, we decided to predict if an absence would be greater than 1 day (>8 hours work). Our goal was to optimize the prediction ability on the sensitivity of the prediction of the extended absence.

Initial Preprocessing

The initial dataset had 21 variables many of which had spaces in the variable names. The names of the variables were changed for easier processing by shortening names and adding an underscore rather than a space. The variable “Son” was also changed to “children” to make the variable easier to understand in the context of what the data actually represented.

All variables in the dataset were read into R studio as numeric variables initially. This is not a correct way to represent many of the variables which are actually numeric factors. The variables are "ID", "Reason", "Month", "Day", "Seasons", "Disciplinary_failure", "Education", "Social_drinker", "Social_smoker" were converted to factors. The variables “Transportation_expense", "Distance", "Service_time", "Age", "Work_load", "Hit_target", "Children”, "Pet", "Weight", "Height", "BMI", "Absent_time" were initially left as factors.

Since we are focusing on classifying instances where an individual may miss more than a full day of work, thus the variable “Absent_time” was converted into a factor where missing

more than 8 hours was changed to a value of 1. This represents the positive in our classification approach. The values in "Absent_time" that were less than or equal 8 hours where changed to a zero result and represent the negative in our classification approach. The "Absent_time" variable before the transformation is shown in Figure 1 and the data after the transformation is shown in Figure 2.

Exploratory Data Analysis

The dataset is composed of 21 variables and 740 observations. For each numeric variable a summary was run to determine the range of the variable and a histogram was generated for a visual representation. For the factor variable a table was generated to determine the number of observations in each category and a bar graph was made to get a visual representation of their relationships. In each has the data was observed for abnormalities or unexpected values. A bivariate graph was also produced for each variable comparing it to all 20 other variables.

From this exploration we most notably saw that the data "Absent_time" variable was extremely imbalanced. Roughly 8% of the data fell into the minor class as shown in Figure 2. Imbalanced data can be problematic for classification problems. We also saw that some of the variables appeared to not have much predictive abilities. This led us to believe ultimately some variables would be dropped from the data for analysis.

Additional Preprocessing

From EDA it was determined that there were no missing values in the dataset. It was however discovered that the last three entries in the dataset had a zero reported in the Month variable for rows 738 through 740. The month variable, as explained in the initial dataset documentation, should be represented as values 1 through 12, such that 1 refers to January and 12 refers to December. Since zero does not have a meaning in that set, those three rows were omitted from

further analysis. It was also determined that the ID variable did not add predictive value to the dataset, so that variable was dropped from the dataset.

Initial Models

For initial analysis 5 models were selected: Kth Nearest Neighbor (KNN), a general linear model (GLM), a decision tree, Random Forest and SVM. These models were selected at random from techniques discussed in this class and previous classes that we believed could work with the dataset for classification. A preliminary model was developed for each of the selected models using as many defaults as available in the codes. No steps were taken to optimize the models beyond choosing initial parameters that required non-default values (e.g. k=2 for KNN). All variables were used in the initial modeling except the ID variable, that was previously removed. Each model was run in a loop 50 times with a resampled training and test set and the results were stored in a matrix.

The results of these models were then compared on the metrics of error, sensitivity, f-measure, and geometric mean. The error was used simply to see how many instances the model was classifying incorrectly. This is useful in determining overall how the model is performing but gives no details on how well the minority and majority class are being predicted. Sensitivity was selected because we ultimately are concerned with predicting when a person will be absent longer than a day. Since extended absences is the minority class and sensitivity capture how well a model predicts the minority class, this makes it a good choice. We also chose the f-measure and g-mean, which both give a more well-rounded picture of how the model is performing overall, taking into account both sensitivity and specificity.

Model Optimization

Model Selection

Boxplots were generated for each of the 4-model metrics for comparison. The error (Figure 3), sensitivity (Figure 4) , f-measure (Figure 5), G-mean (Figure 6) comparative boxplots shows that of the 5 models GLM and KNN performed the best visually in all but the error metric. In the initial models, GLM and KNN had the highest error. SVM was found to have the lowest error; however, further analysis shows that this was achieved by predicting all major class. In further analyzing the metric comparisons, we determined that KNN performs modestly better than GLM in every metric except error. As a group we discussed selecting either KNN or GLM and ultimately decided that KNN would be a better choice to optimize because it is similar, and we felt it could do well in prediction.

Preliminary Attempt: Monte Carlo Cross Validation Optimizing Error

For an initial attempt to optimize a model we were inspired by work found on the Notre Dame server by “Steve”. This model used a general linear model to predict variable importance and a Monte Carlo cross validation approach to optimize the number of variables and k-values based on error metrics. In the example, the variables were all numeric; however, our data also had factors. That approach seemed to only work with numeric data and ended up predicting the most important predictor variable was “Height” which seemed unlikely.

We ultimately modified this approach and converted to a random forest approach to predict variable importance based on Gini index scores. Figure 7 shows the predicted order of importance. The variables were then ordered in the data frame based on the Gini prediction from most to least important, so that when the cross validation ran, each time it added the next most important predicted variable. With that in place a Monte Carlo Cross Validation (CV) was run

optimizing the number of variables and the K-value on error. The Monte Carlo CV took a 2/3rd sample from the training set on each run.

This model ended up predicting high K-values always predict the same optimization. Further inspection revealed that this was because of the data imbalance. This led to the model predicting all major class which minimized the error to the percent of the data that was in the minor class. This was suboptimal and thus a different model was found to be necessary.

Second Attempt: Monte Carlo Cross Validation Optimizing Sensitivity

In our second attempt. the initial model was modified to optimize on sensitivity rather than error. We decided that this was a better metric to optimize on because sensitivity identifies the percent of the minor class that was accurately modeled. With this change the cross validation was run again. The model predicted that the optimal variable number and K-value combination was 13 variables and a k=1.

The cross validation found that with this combination a sensitivity of 27% could be achieved . When the model was run with new data though it was found to have a median sensitivity of 23%, which lower than the initial unoptimized model. This can be seen visually in Figure 8. This was suboptimal and indicates the model was overfit on the cross validated training sample. Thus, it was determined that a different optimization method should be attempted.

Final Attempt: SMOTE with Monte Carlo Cross Validation Optimizing Sensitivity

After being introduced to the SMOTE sampling technique we became curious if this technique to balance our data might help with out classification problem. To incorporate SMOTE into the model previously developed, we replaced the sub setting code previously setup to create the testing and training dataset with code that partitions the data into two equal groups. The SMOTE code was then applied to the training set that was 50% majority and 50% minority class.

This code was inspired by modeling done in “SMOTE - Supersampling Rare Events in R”. This code was then run with the Monte Carlo Cross validation technique that splits the SMOTE applied training set into 2/3 subsets and tests all possible combinations of variables numbers and k-values. Using the results from this loop we found the average sensitivity from each of the permutations of number of variables and K-values. A graph of these combinations is shown in Figure 9

The top 5 combinations were selected for continued testing to see how well the projected models fit when a different training and test set are applied. These top 5 combinations are shown in Figure 10. These models were fit into a loop that selected new testing and training data and applied the SMOTE technique. The sensitivity results from this analysis are documented in Figure 11. Each of the models performed reasonably similarly with regard to sensitivity, error, f-measure and G-mean.

In looking closer at the results of the top 5 sensitivity models we ultimately selected model number 4 which uses 14 variables and a K-value of 19. Model 4 was found to have the highest sensitivity of the models. It was also found to have the highest error. The top three models with regard to projection from Monte Carlo CV sensitivity only used 2 variables. It was determined that relying on only 2 variables was not ideal and it appeared they were overfit to the CV set more than model 4.

When comparing the optimized and initial models in Figure 12 we see that the optimized model significantly improved the sensitivity over the initial model. In fact, the median sensitivity increased from 23% to 81% . This means that the optimized model is better able to predict the minor class. On the other hand, Figure 13 shows the error rate also raised significantly. When looking at the confusion matrices, false positives metrics from a both models run with the same

test data (Figure 13) we see that the optimized model saw a decrease in the accuracy rate and specificity. Therefore, it can be concluded that while the optimized model does a better job at identifying the minor class, it comes at a cost to the other metrics. This type of optimization is acceptable when there are consequences to missing a minority result but is not ideal for many other circumstances.

Conclusions

Using the SMOTE technique combined with Monte Carlo Cross Validation on K-value and number of variables we were able to optimize a KNN model to predict when an individual will miss more than a full day of work with a high degree of sensitivity. The final model used 14 variables and a k-value of 19. Creating a more sensitive model was seen to come at a higher cost to the error rate. Future study would be needed to optimize on both parameters; however, we did not place a priority on optimizing both in our analysis.

Possible additional things to further optimize the dataset include trying an ensemble approach to the dataset. In analyzing the data some of the “Reason” factor results never miss more than 8 hours. I suspect that with a more advanced technique that divides the data prior to classification would achieve a higher degree of both accuracy and sensitivity.

References

Amunategui, Manuel “SMOTE - Supersampling Rare Events in R” no date. Available at: http://amunategui.github.io/smote/

(no last name given), Steve “Refining a k-Nearest-Neighbor classification” no date. Available at: https://www3.nd.edu/~steve/computing_with_data/17_Refining_kNN/refining_knn.html

Martiniano, A., Ferreira, R. P., Sassi, R. J., & Affonso, C. (2012). Application of a neuro fuzzy network in prediction of absenteeism at work. In Information Systems and Technologies (CISTI), 7th Iberian Conference on (pp. 1-4). IEEE.

Figures

Figure 1. Absent_time pre-processing histogram

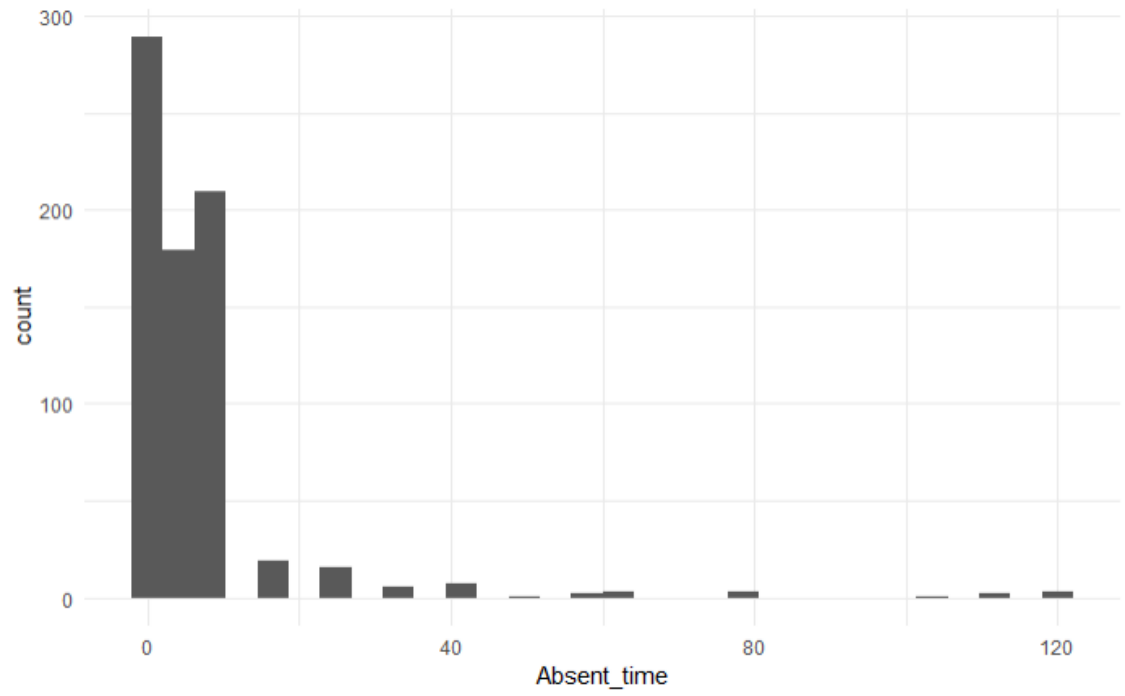

Figure 2 Absent_time bar graph showing distribution

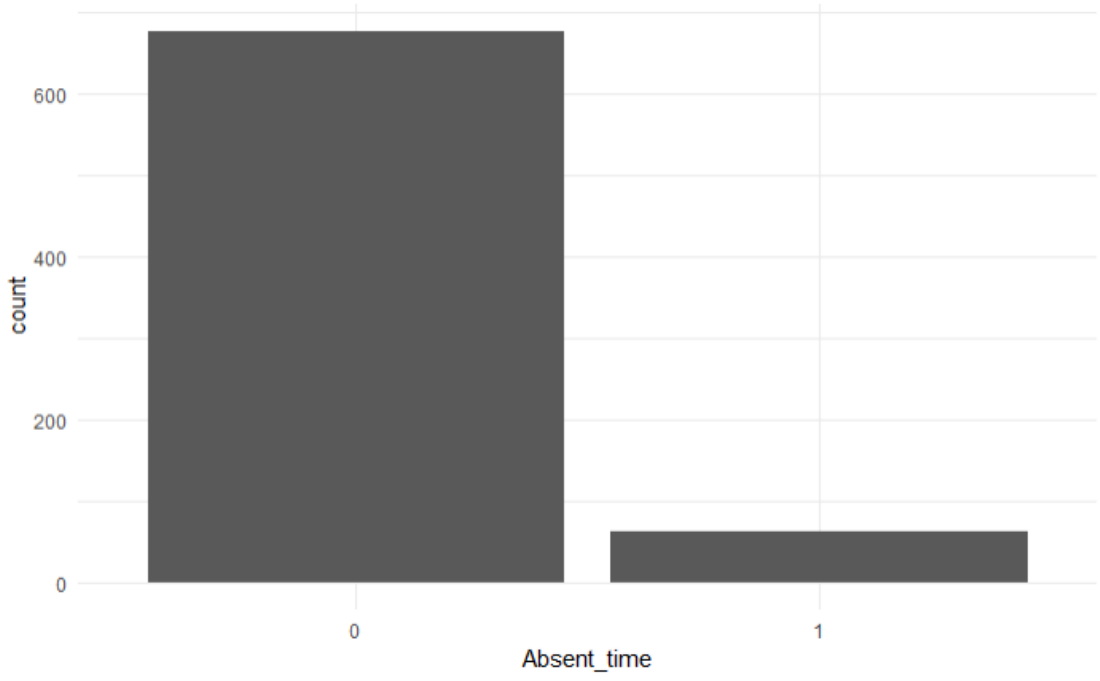

Figure 3

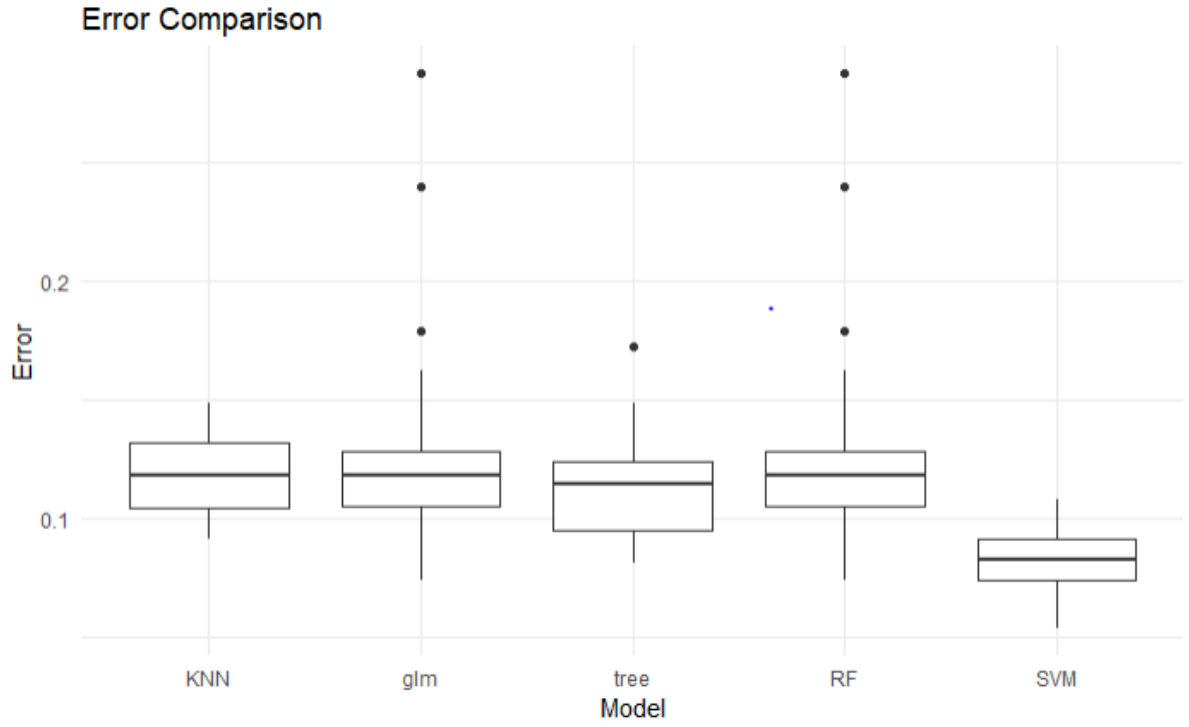

Figure 4

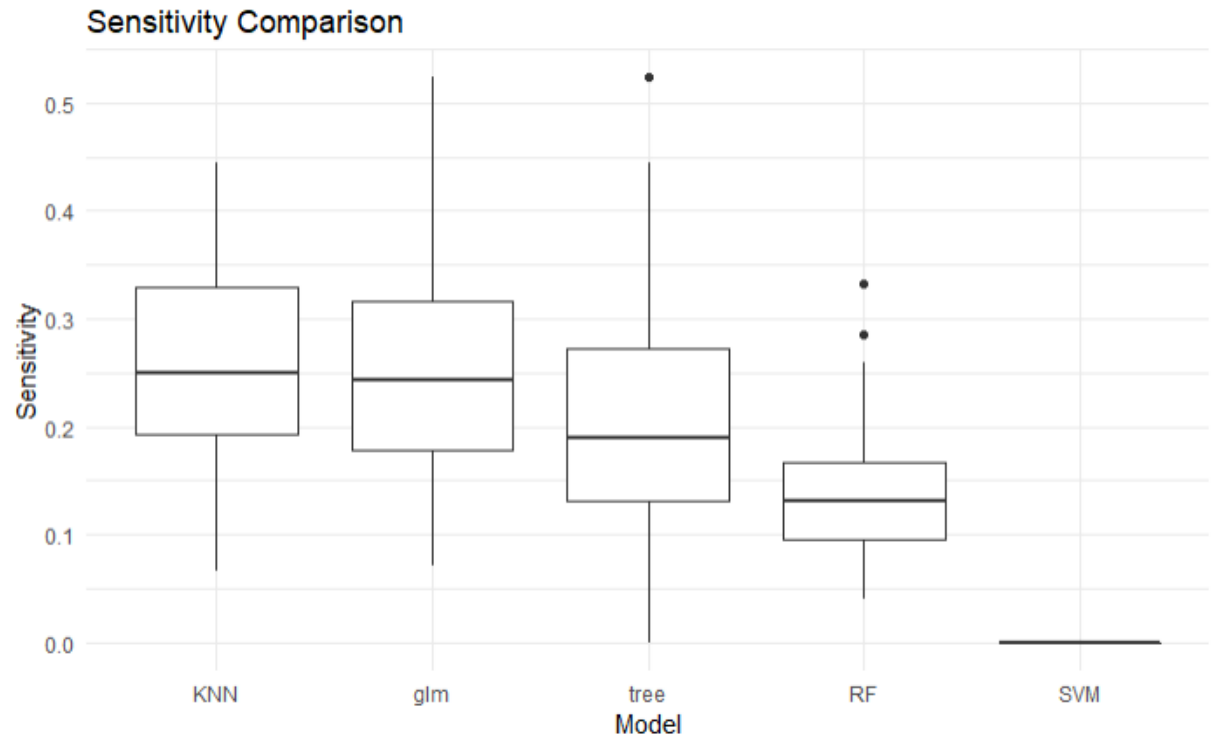

Figure 5

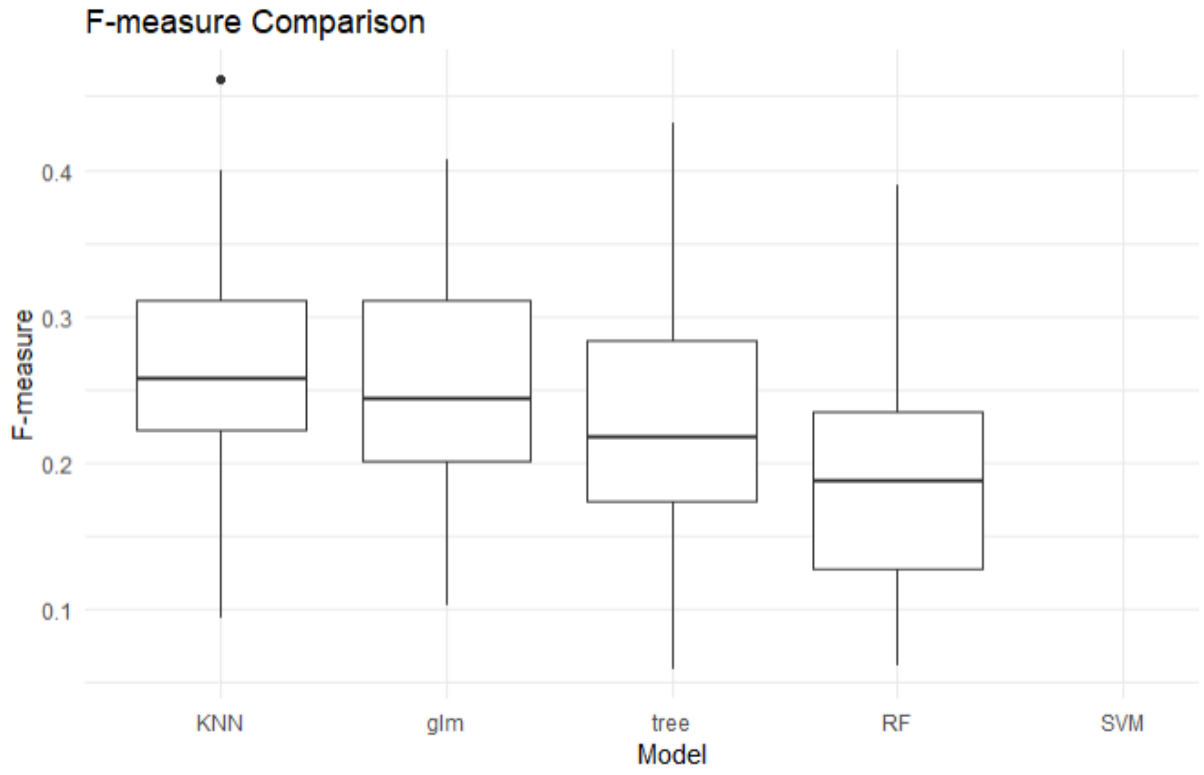

Figure 6

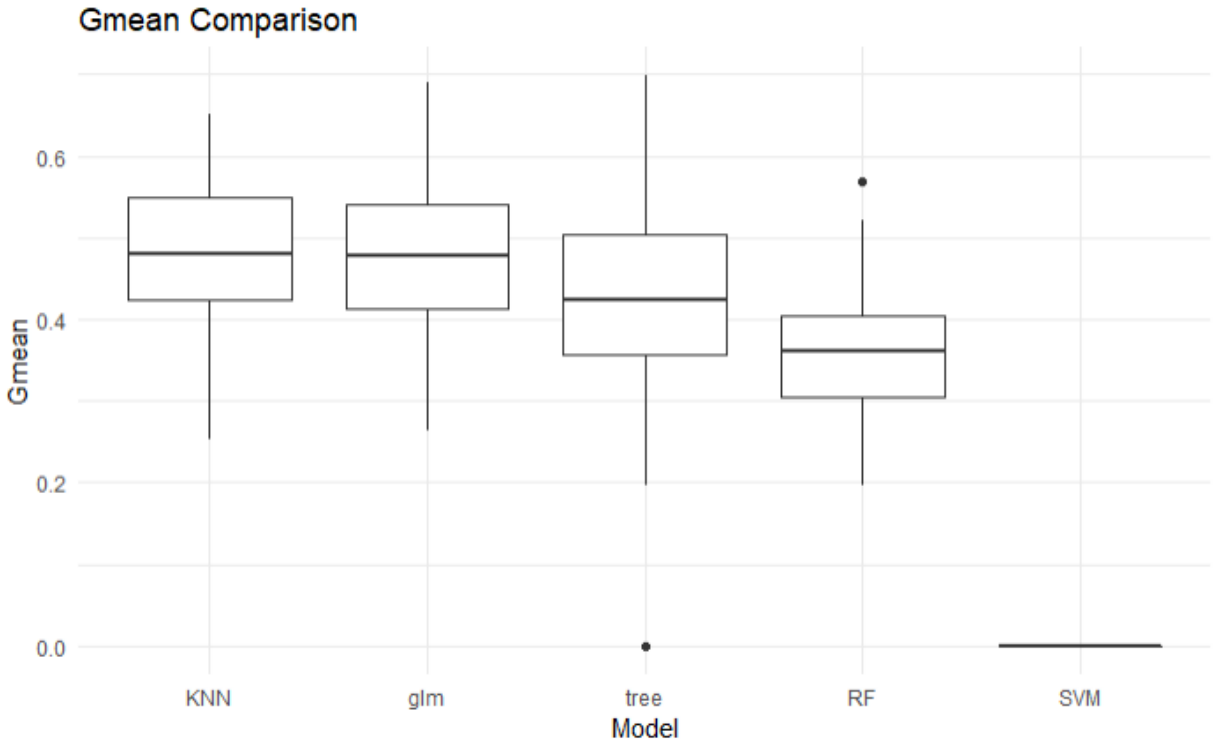

Figure 7. Order of Importance predicted by Gini Index

```
 [1] "Reason"                "Work_load"             "Month"                 "Day"
 [5] "Hit_target"            "Distance"              "Height"                "Weight"
 [9] "Age"                   "Transportation_expense" "Service_time"          "Pet"
[13] "BMI"                   "Seasons"               "Children"              "Education"
[17] "Disciplinary_failure"  "Social_drinker"        "Social_smoker"
```

Figure 8. Comparison of 2nd Optimized Model vs Initial

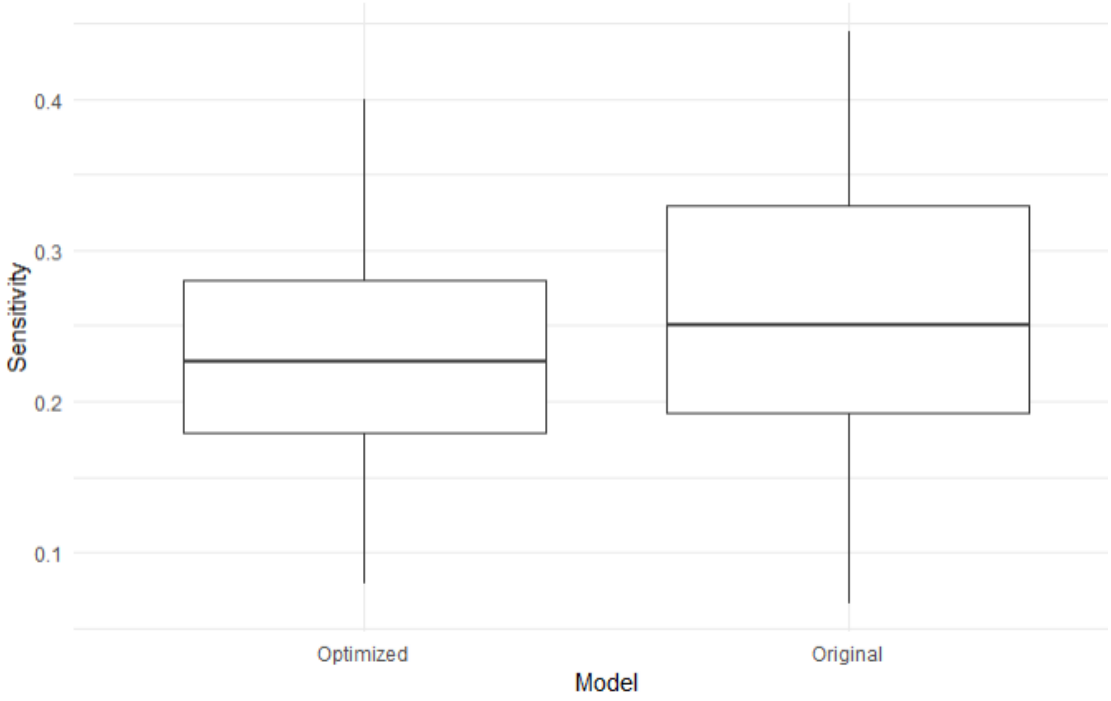

Figure 9. Visualization of combinations for optimization

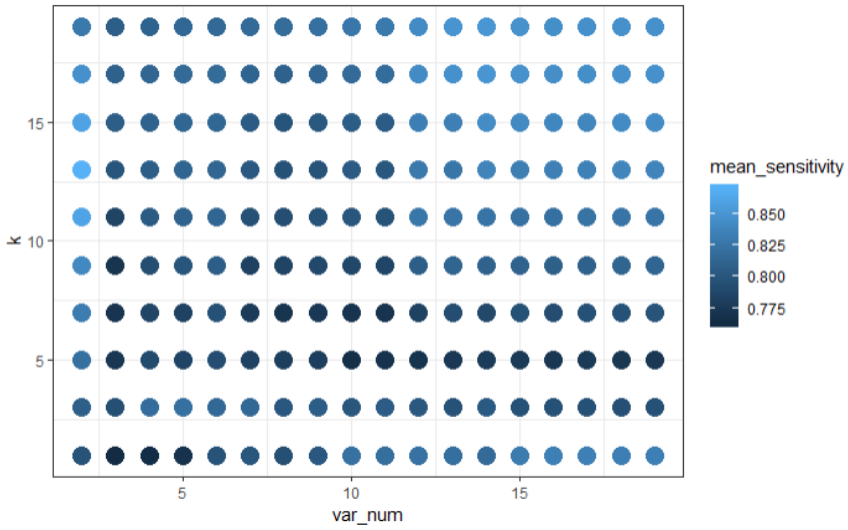

Figure 10. Ordered mean Sensitivity Combinations

var_num <dbl>	k <dbl>	mean_sensitivity <dbl>
2	13	0.8706989
2	11	0.8610326
2	15	0.8601818
14	19	0.8508015
14	17	0.8501362

Figure 11

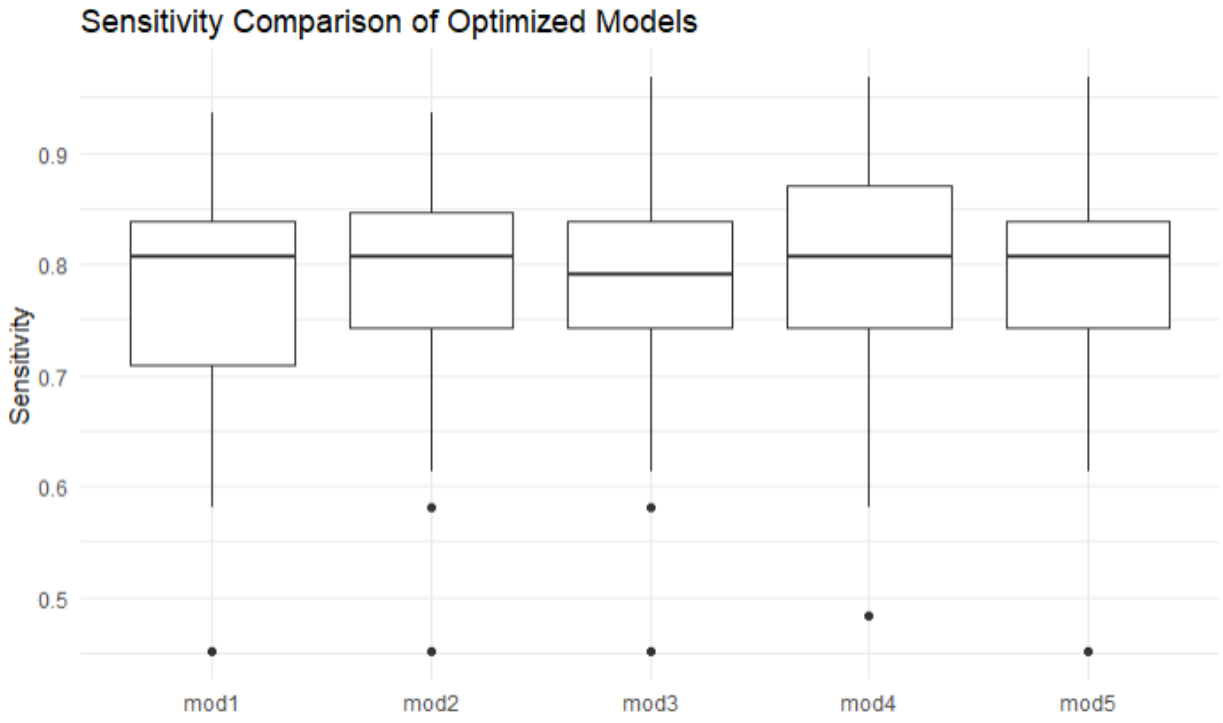

Figure 12.

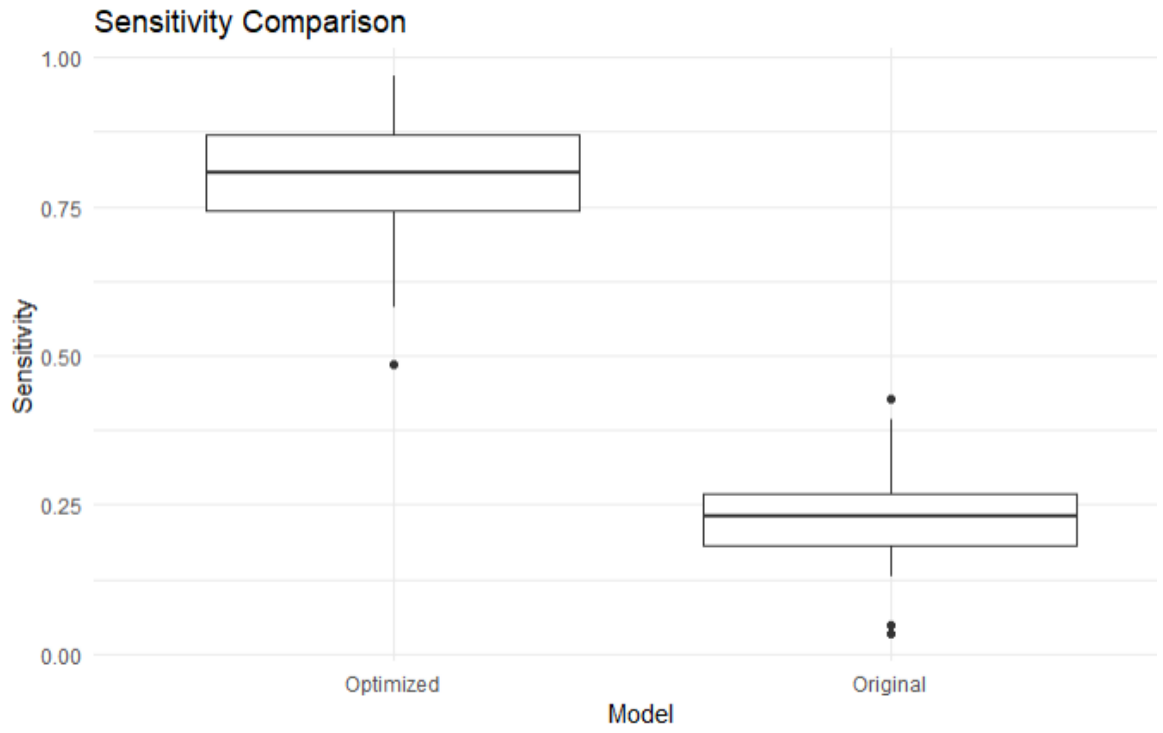

Figure 13

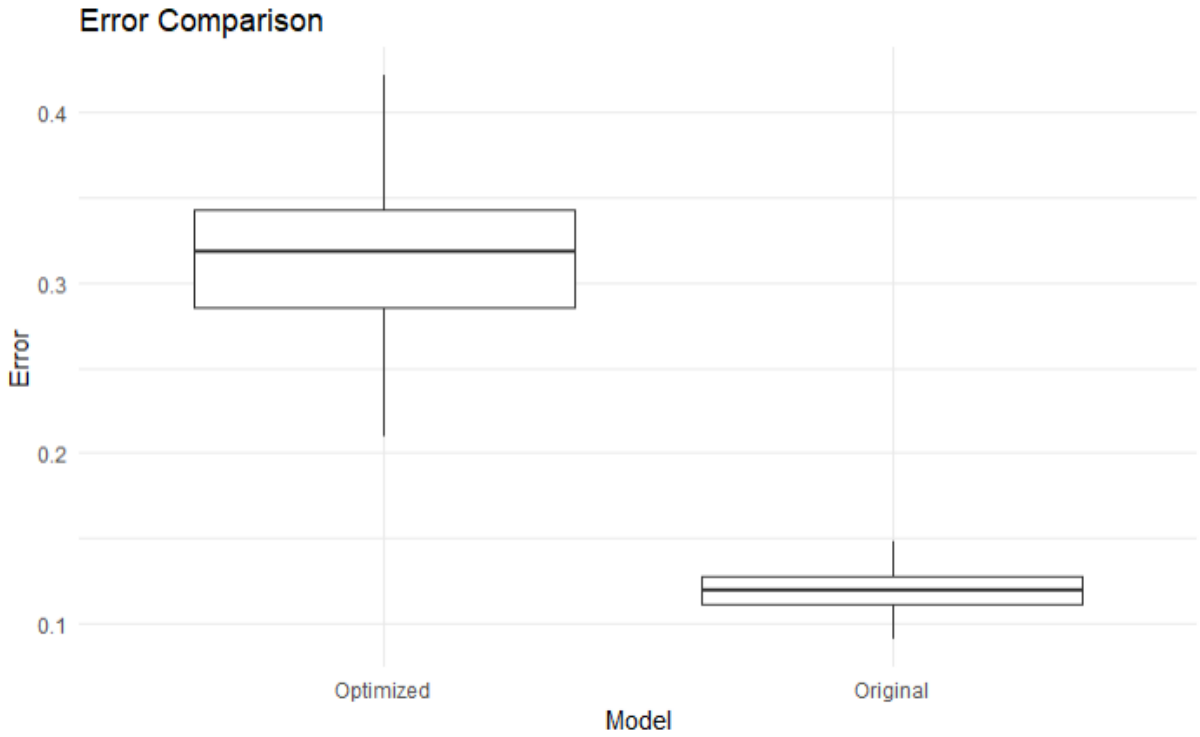

Figure 14

Optimized Model

```
Confusion Matrix and Statistics


knn   0   1
  0 232  10
  1 105  21

               Accuracy : 0.6875
                 95% CI : (0.6374, 0.7345)
    No Information Rate : 0.9158
    P-Value [Acc > NIR] : 1

                  Kappa : 0.153
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.67742
            Specificity : 0.68843
         Pos Pred Value : 0.16667
         Neg Pred Value : 0.95868
             Prevalence : 0.08424
         Detection Rate : 0.05707
   Detection Prevalence : 0.34239
      Balanced Accuracy : 0.68292

       'Positive' Class : 1
```

Initial Model

```
Confusion Matrix and Statistics


knn   0   1
  0 320  20
  1  17  11

               Accuracy : 0.8995
                 95% CI : (0.8641, 0.9282)
    No Information Rate : 0.9158
    P-Value [Acc > NIR] : 0.8868

                  Kappa : 0.3184
 Mcnemar's Test P-Value : 0.7423

            Sensitivity : 0.35484
            Specificity : 0.94955
         Pos Pred Value : 0.39286
         Neg Pred Value : 0.94118
             Prevalence : 0.08424
         Detection Rate : 0.02989
   Detection Prevalence : 0.07609
      Balanced Accuracy : 0.65220

       'Positive' Class : 1
```