****

**CareDash Online Fake Review Identification**

By

Anil Velappan

Vicky Pang

Siwen Yang

Data Mining and Business Intelligence

Professor Niam Yaraghi

Fall 2018

Stamford, CT

**Executive Summary**

CareDash’s mission is to transform the healthcare provider review industry to make it more transparent, accessible, and inclusive for all patients. The goal is to make patients feel secure and comfortable about deciding on physicians to seek for their medical needs. CareDash offers the most trustworthy information, clean comprehensible data, and publish healthcare information to benefit all patients. In order to offer the best service for their patients, they are looking for new methods to give the best clean data to their patients. The Fall 2018 class of Data Mining and Business Intelligence will join forces to implement new methods and ideas that we’ve learned in our course to help CareDash succeed even more.

**Problem Statement**

The data that we’ve received consisted of 116,149 rows, and within this data, CareDash needs our help to identify the fake reviews. The data came from their direct website, which is an online review website of physicians. The challenge is to identify fake reviews; fake reviews may be identified in the unfiltered data by reviews that looked like they’ve been paid for, contained meaningless speech or writing, physician office and user review locations are too far apart, etc. We implemented the concepts from our course to design algorithms to identify if these aspects (and more) would result in fake reviews or not.

**Methodology Modeling for Libra Action**

In order to find the best results, we will model against two targets, which are Status Map and Libra Action. Prior to modeling, we made sure to clean and modify the data first. First, we inspected the full data, then created additional variables.

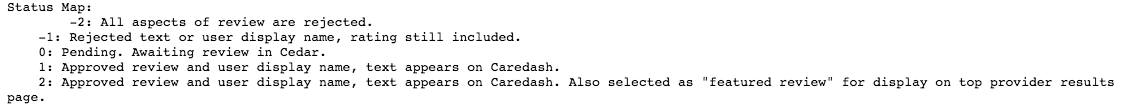
**The new variables:**

**IsUserStateDifferent** – If User State Review and State Provider are the same, it would equal to 1. If they were different, it would equal to 0.

**UserFrequency** – How many times did this user created a review?

**Same\_UserSessionSource** – If User Source is the same as Session Source, it would equal to 1. If they were different, it would equal to 0.

**We also decided to simplify the status map:**

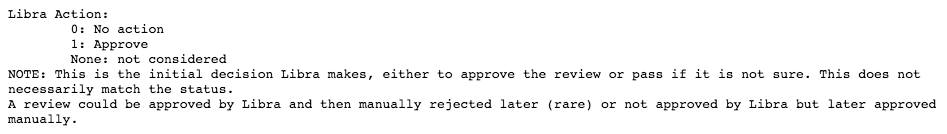
**Original Status Map:** 

**Simplified Status**

We removed 0 (Pending), because we do not have enough information. We merged -1 and -2 to 0 (Not approve), because they are both rejection status anyway. We merged 1 and 2 to 1 (Approved), because they are both approvals anyway.

For the model against Libra Action, we also created another new variable. We labeled it as **revised\_libra\_status**. We revised the Libra Action, because some of the Libra Action that were approved (1), a CareDash specialist would manually change the 1 into a 0 if there were any discrepancy with the review.

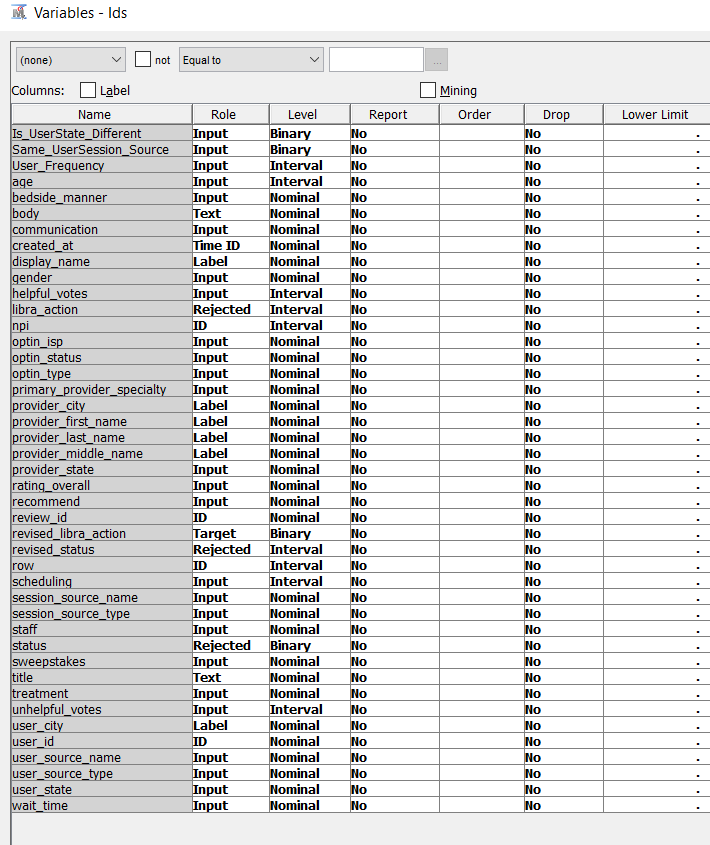
**Original Libra Action:**

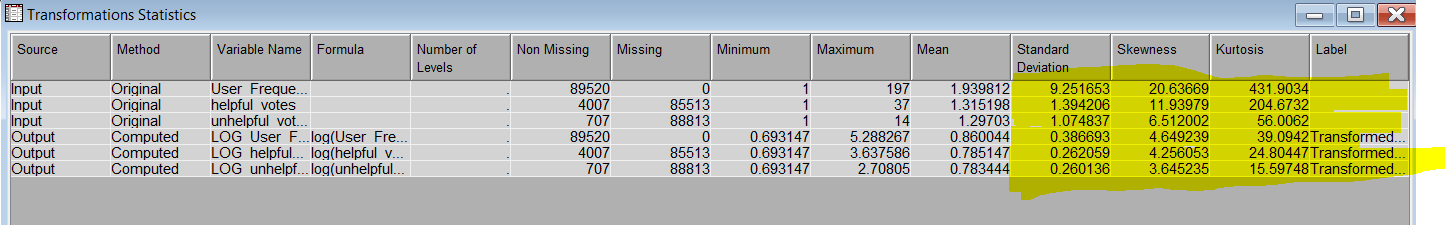


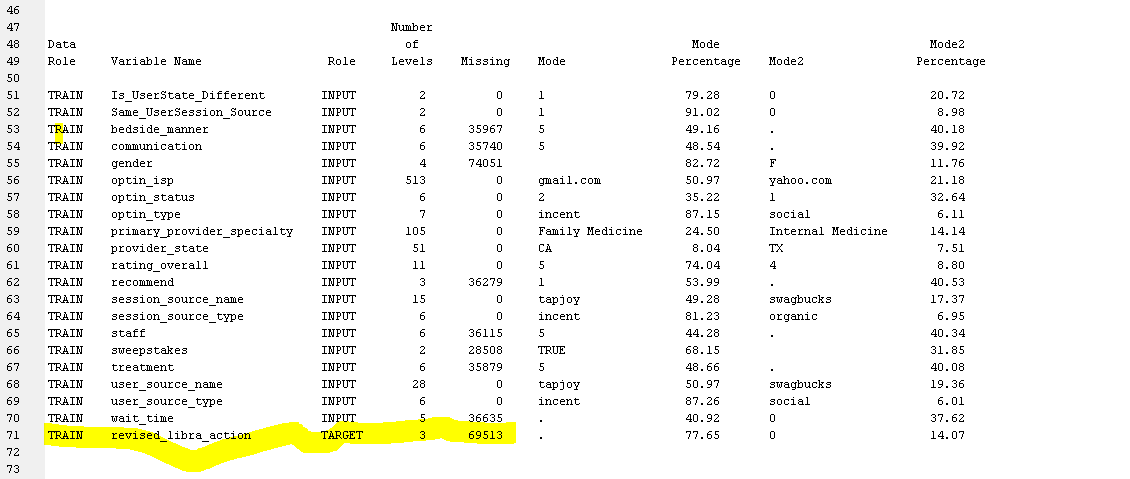
**Data Transformation**

In the Data Transformation stage, we noticed some of the interval variables like the helpful votes, unhelpful votes, and user frequency had a skewness or a non-normal distribution. We then applied the log transform to those variables. We also observed the sweepstakes variables, and they seemed to have null for false in the data. Therefore, we replaced the nulls with false.

In this step, we’ve transformed the data, and then we removed any missing values based on our observations below. During the Stats explore step, we found that there were 69,513 missing data. Keep in mind that we are modeling against the Libra Status first. After we’ve cleaned and modified the entire data, we had to split the data in Excel. We used the first 90K observations and left the rest for final testing of the model. The quality of the project would ultimately be tested based on the remaining observations (about 27K).

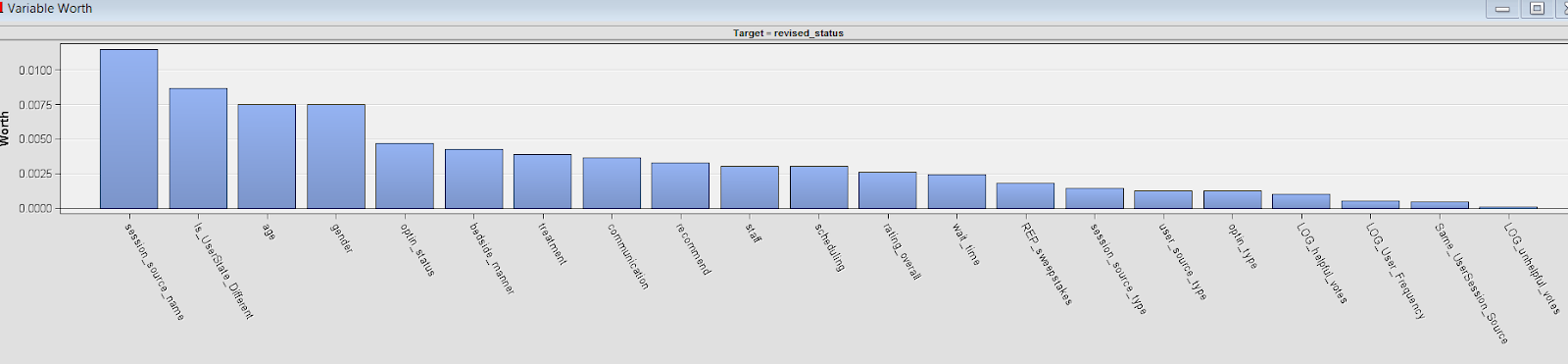






**Descriptive Analysis**

During this step, we explored the variables in the stat explore step. We wanted to see which variable were worth keeping in the analysis and also to view the missing values.

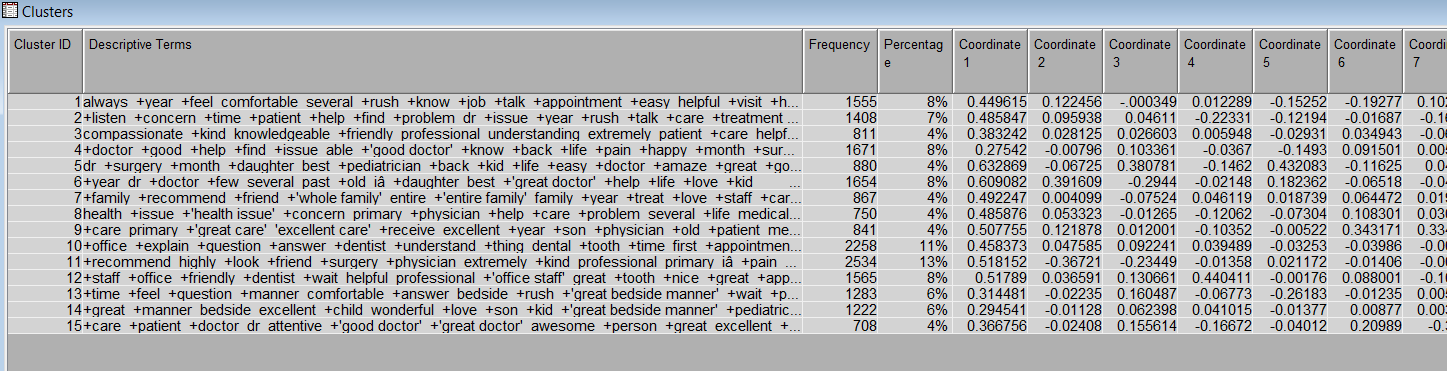


**Text Mining**

In the text mining stage, we ran the data mining techniques on the text body of the

reviews to identify the patterns or text clusters in the reviews.

**As seen below, clusters were identified after trying different combinations:**



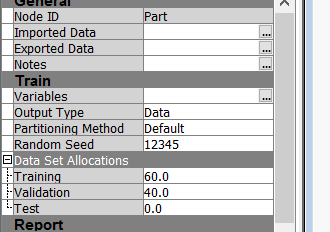
The text cluster and text cluster SVDs were then passed down as input to the different predictive

models.

**Data Partition**

In the data partitioning stage, we partitioned the data into 60% training and 40%

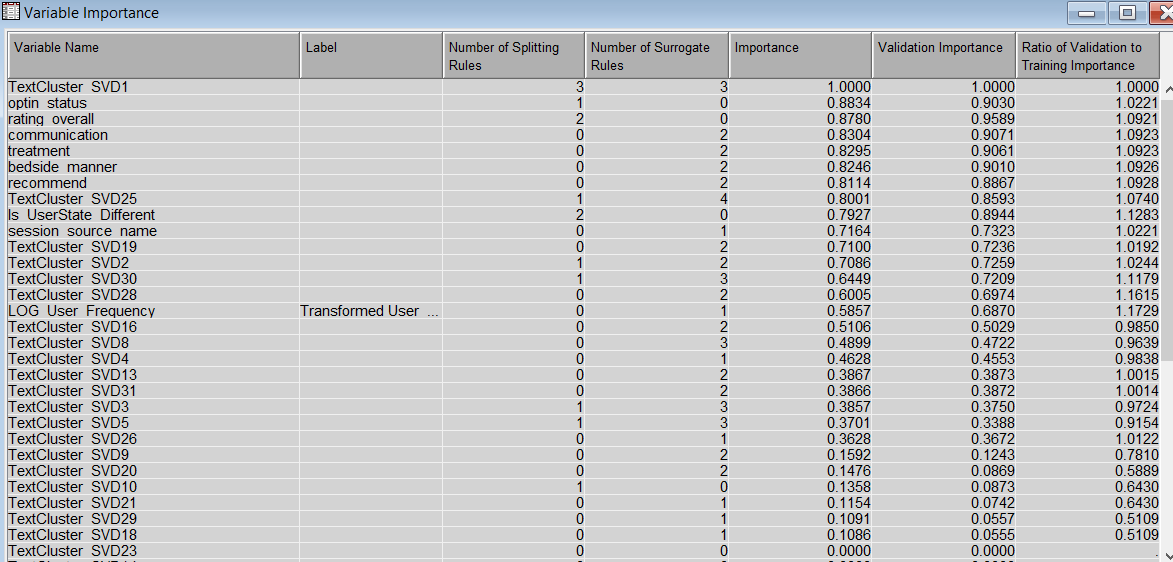
Validation.

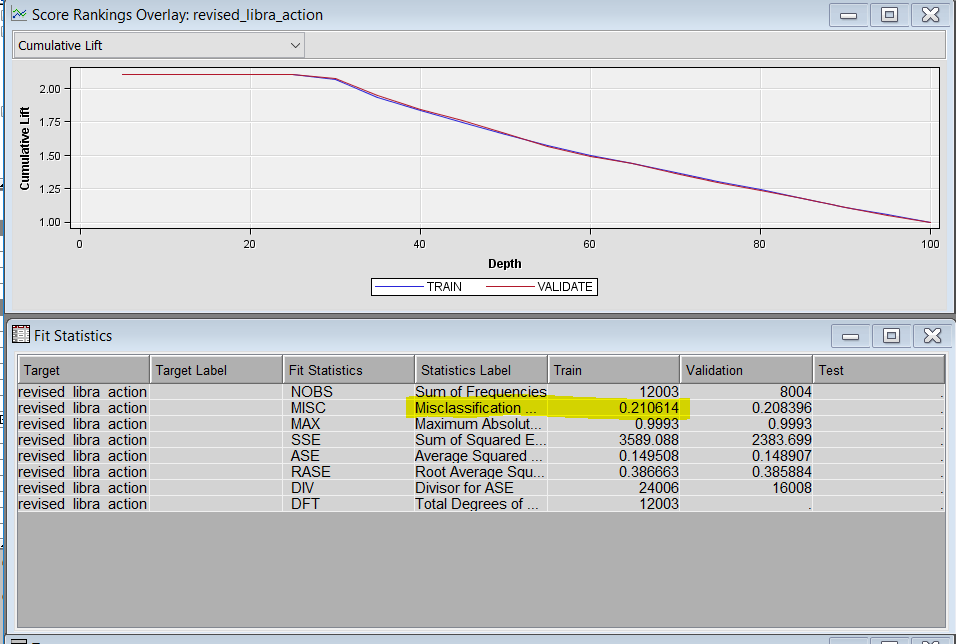


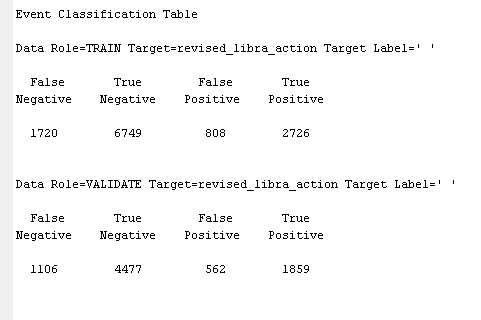
**Models**

**Decision Tree:**

We ran the decision, and the model resulted with the best misclassification rate.







**Gradient Boosting Tree:**

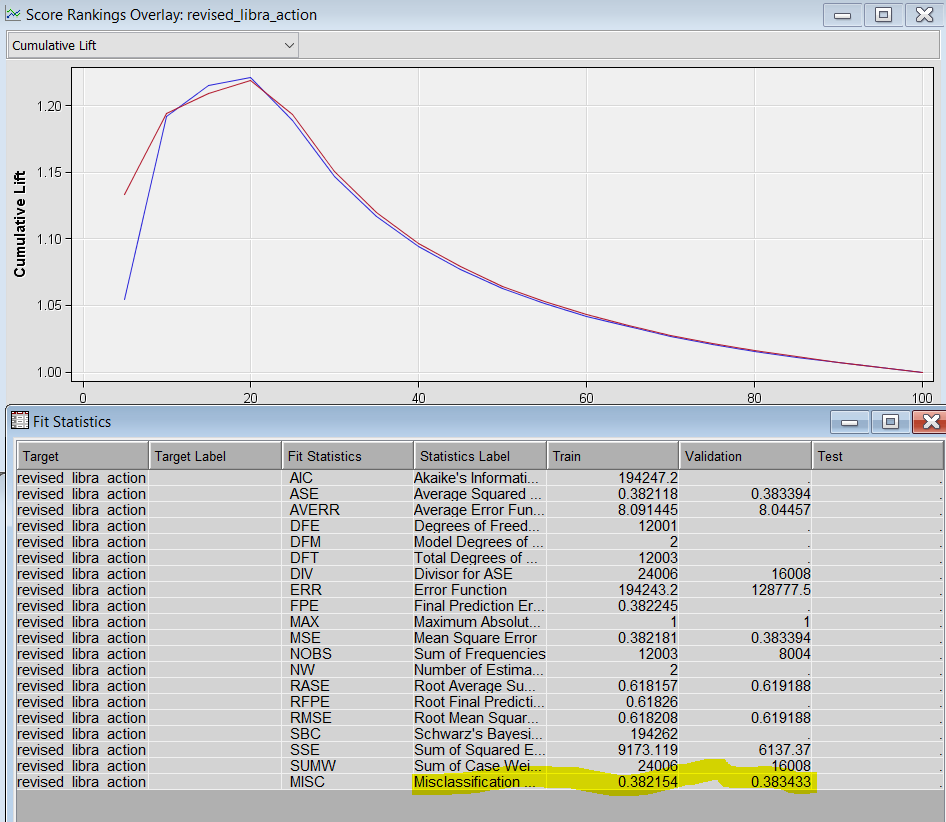
As seen below, we ran gradient boosting tree on the data but did not find much

improvement in the misclassification rate.



**Regression Model:**

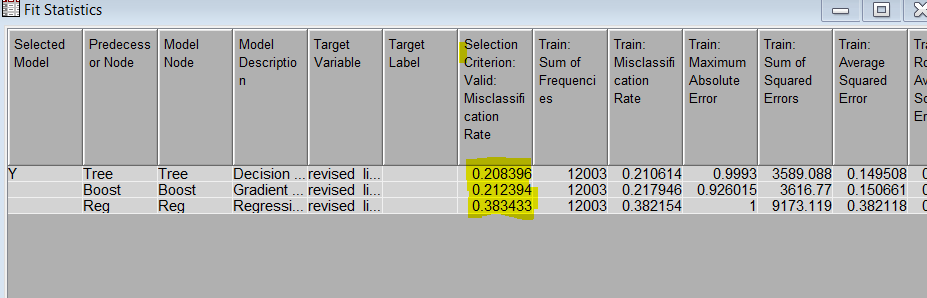
We ran the regression model, and below are the results of the regression model with logistic regression. We also didn’t see any improvement in misclassification. In fact, misclassification rate increased.

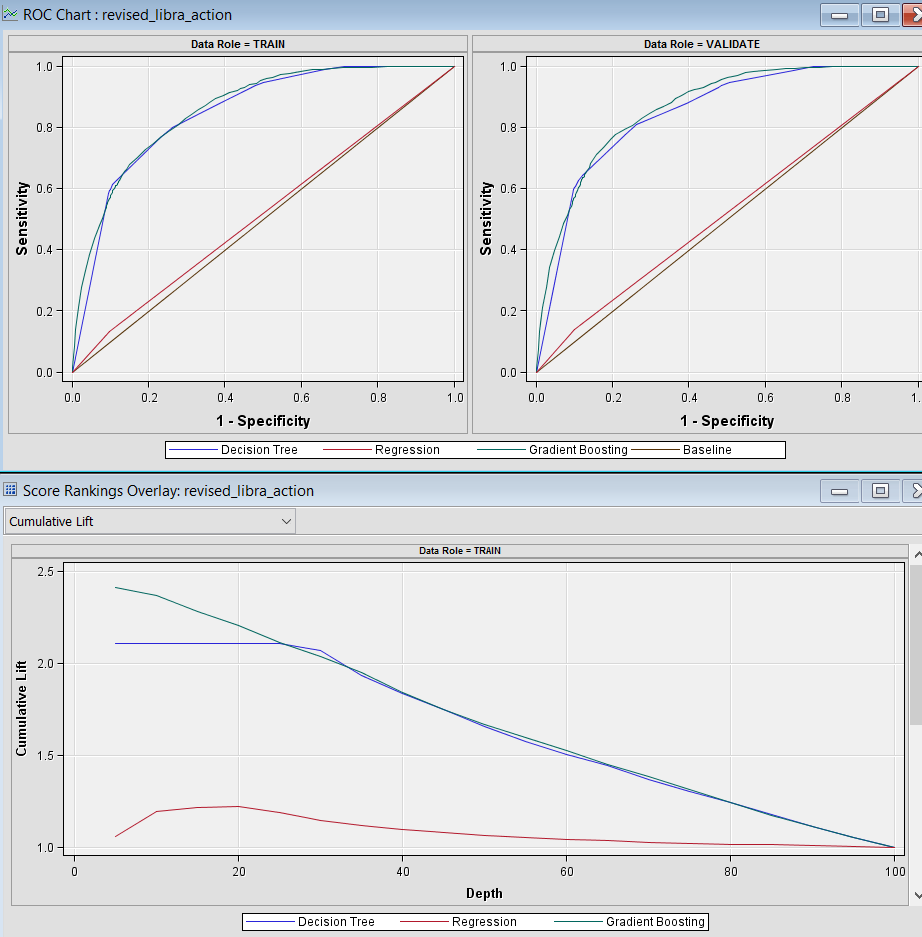


**Best Model from Libra Action**

After running the three models, we compared them, and observed that the decision tree

model performed the best based on the misclassification rate.

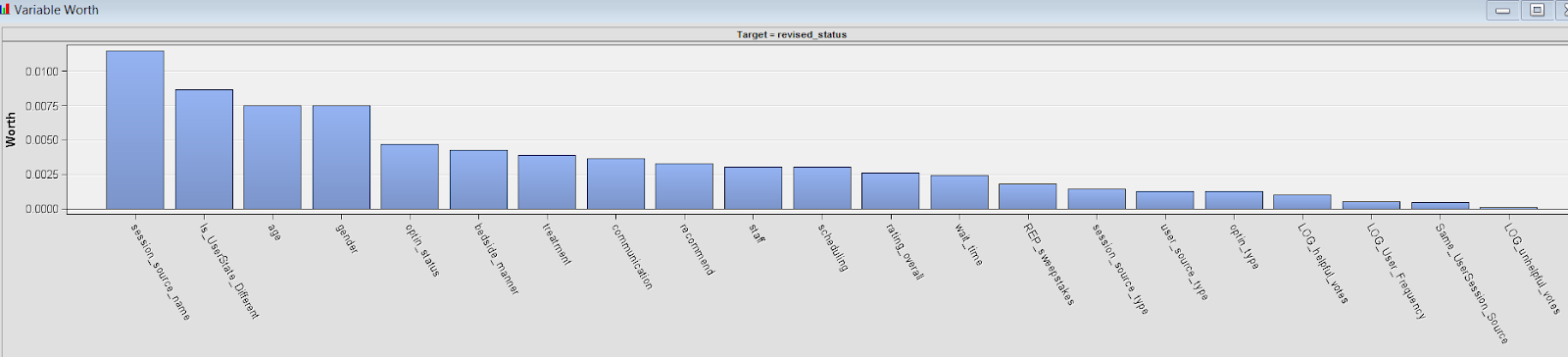




**Methodology Modelling for Status Map**

For the Status Map model, we did the log data transformations and replacement just like we did in the Libra Action modeling. The main difference is that there are no missing values for the target variables, and the modelling also would only run on all 90k rows.

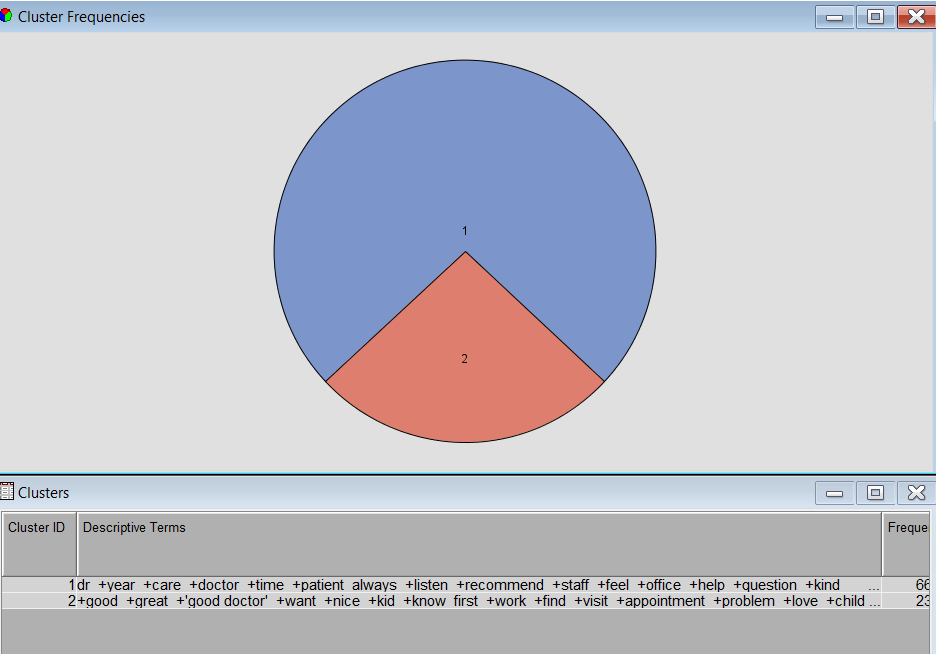
**Descriptive Analysis:**



**Text Mining**

In the text mining step in this model, it identified only 2 clusters after running on the 90k

rows. We tried forcing more clusters but the distance between the clusters were not significant.

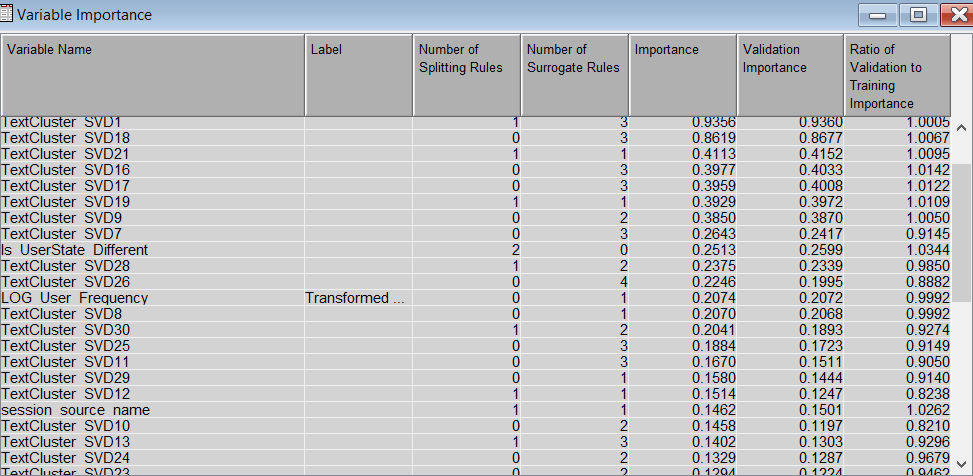


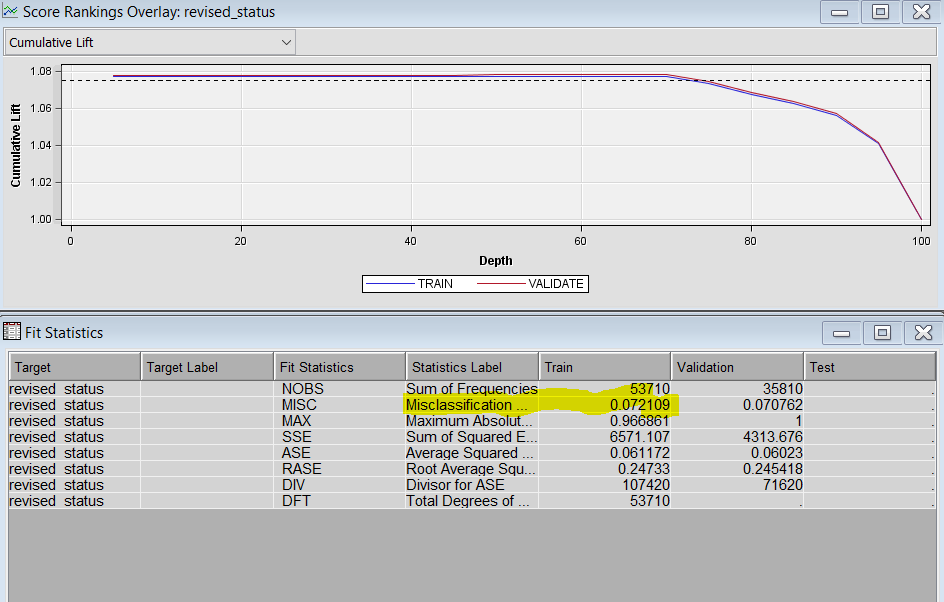
**Models**

**Decision Tree:**

In the decision tree, we observed the misclassification rate getting much better against the

status\_map target.

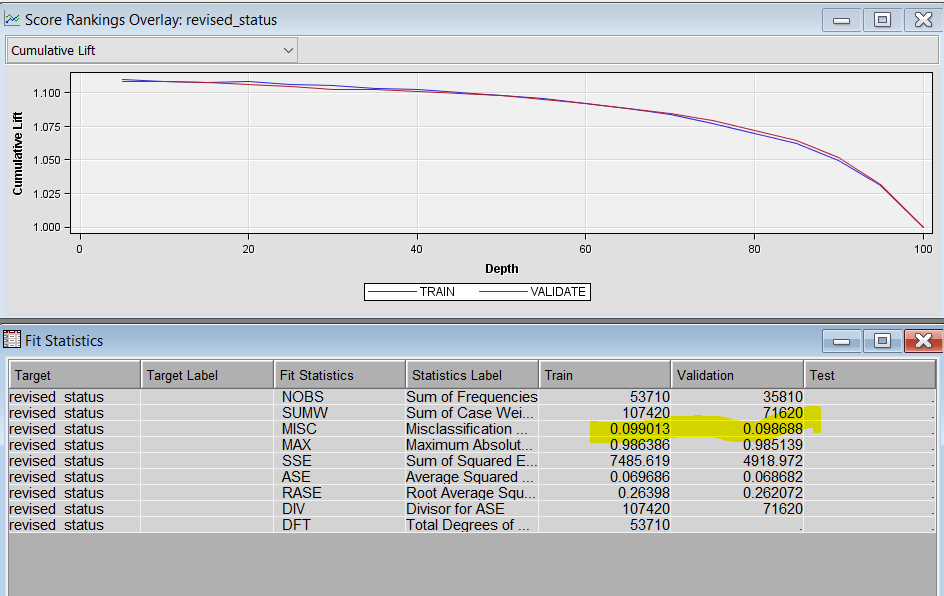




**Gradient Boosting Tree:**

In the gradient Boosting tee model, we observed that it did not improve much in the

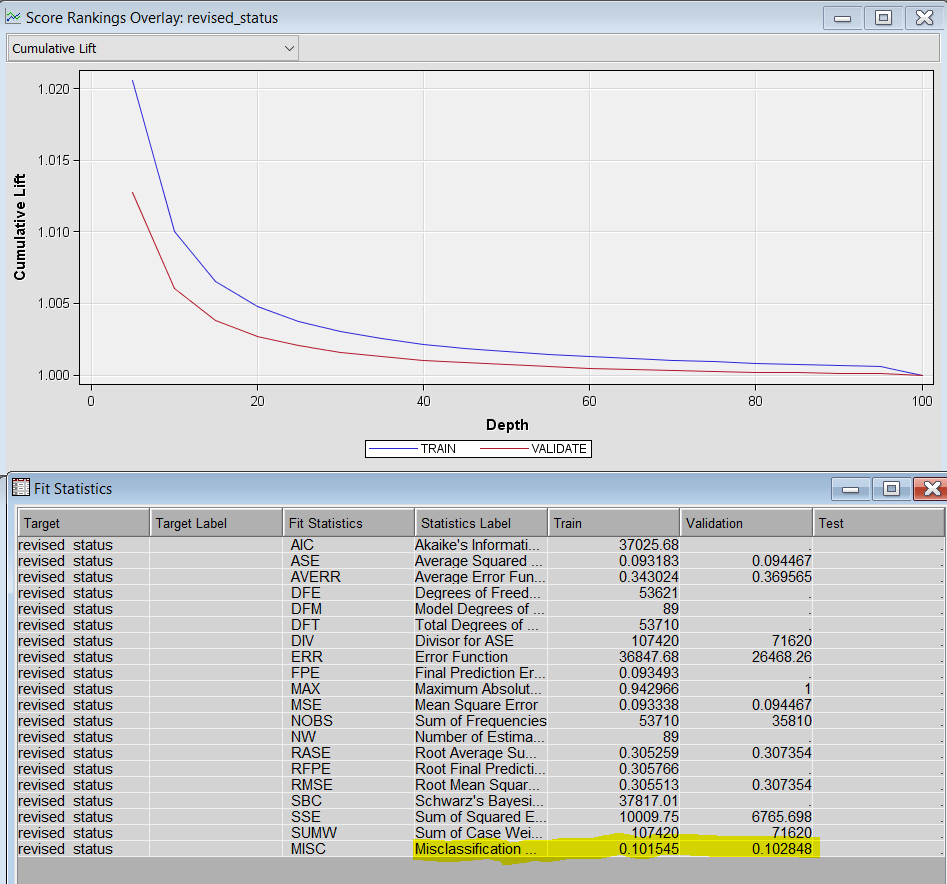
misclassification rate.

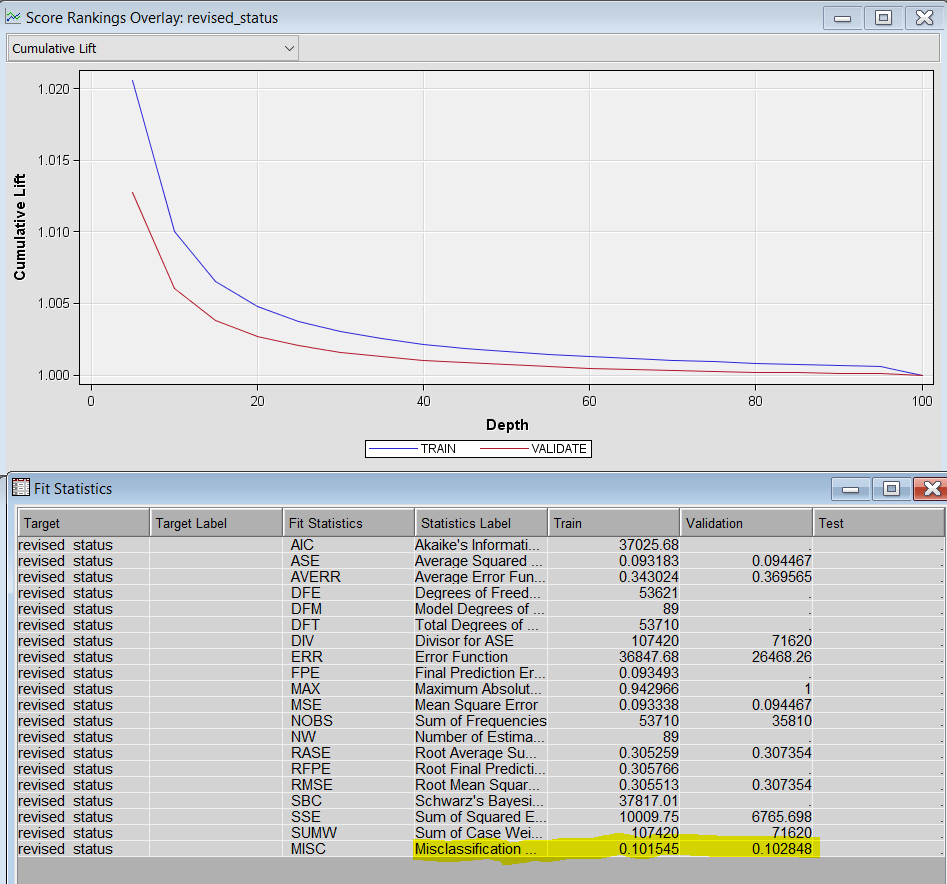


**Regression:**

We can see the model performing good but the misclassification rate is not as good as the

decision trees.

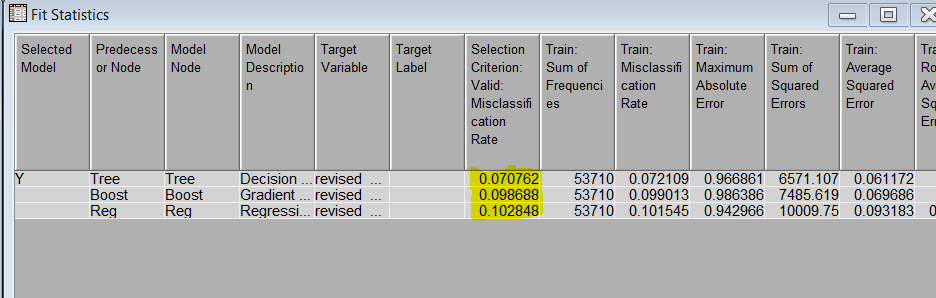


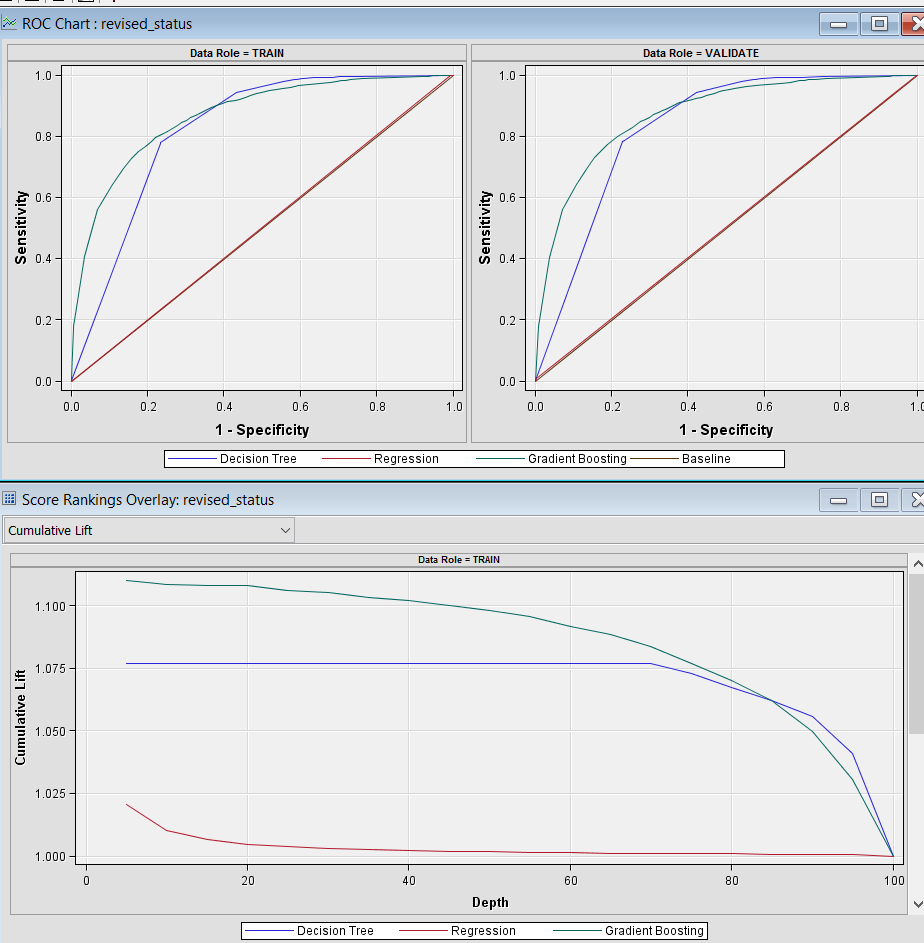


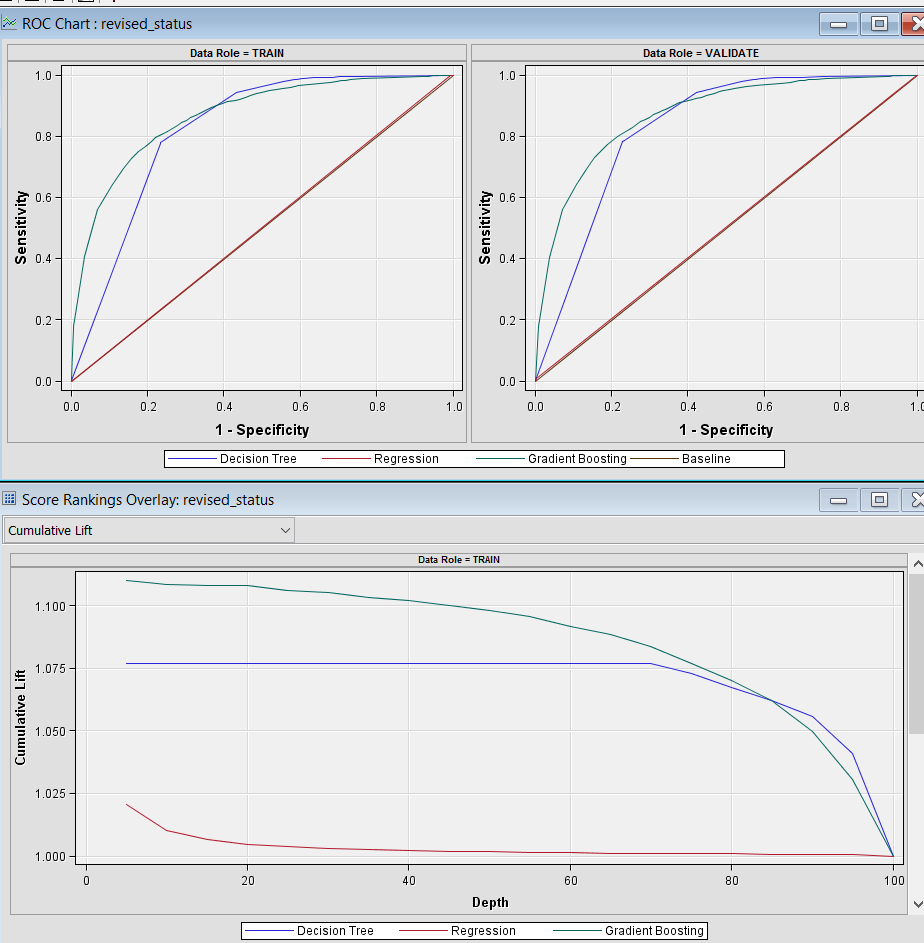
**Best Model for Status Map**

After running the three models, we compared them, and observed that the decision tree

model performed the best based on the misclassification rate.



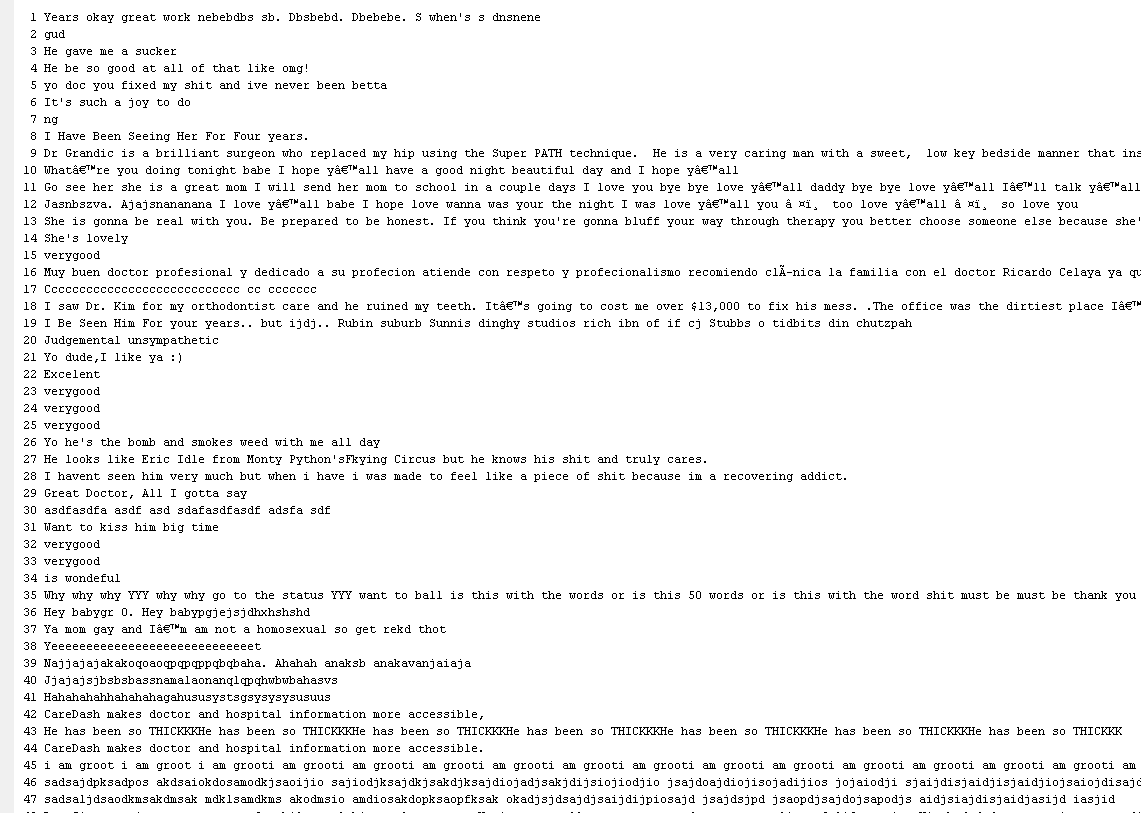




**Scoring the Score dataset**

In our final stage, we scored the data with the remaining 21k rows. Below is the

screenshot of the fake reviews identified sorted by the probabilities.



**Conclusion**

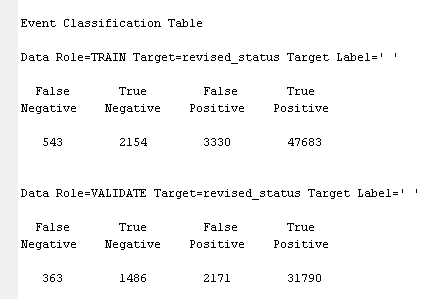
In conclusion, we’ve ran both Status Map and Libra Action Models. Based on the results, we found that the Status Map Decision Tree Model outperformed the Libra Action Model.

**Misclassification Rate:**

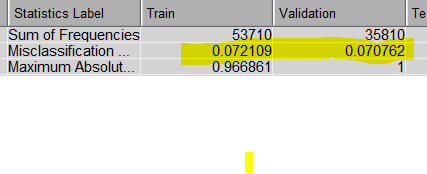
Libra  = 11060/20007 = **0.55280**

Our Best Model(Tree Model) = 6407/89520 = **0.07157**

**Our Event Classification Table:**



**Our Decision Misclassification Rate:**



As seen below, out of the 89,520 observations, Libra wasn’t able to take action on the 69,513 observations. Out of 89,520, Libra was only able to classify 20,007. The Decision Tree Model was able to classify 89,520.

|  |  |  |
| --- | --- | --- |
|  | Libra | Status Map  Decision Tree Model |
| null | 69513 | 0 |
| l | 7446 | 84974 |
| 0 | 12561 | 4546 |
| Total | 89520 | 89520 |

**Libra Event Classification Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| False positive | True Positive | False Negative | True Negative |
| 35 | 7411 | 11025 | 1536 |