Disaster Relief Project Part 2:

Authors: Alexandra Ferentinos, Brendan Jalali, Pete Landers

Introduction

Natural disasters can be unpredictable and have devastating effects on communities. People can lose their livelihoods, as well as their housing in a blink of an eye. Disaster relief efforts can be a herculean task as factors such as time, budget, and manpower. These factors can dictate the speed at which displaced persons are rescued. The 2010 earthquake in Haiti proved especially difficult to provide aid as main roadways were made impassable due to land debris and road fissures. However, with advancements in aerial image capture, it became possible to collect large imagery datasets of a land mass, without having to use ground travel.

This project aims to accurately discern, from gathered geo-referenced imagery (RGB pixels) of the Haiti 2010 natural disaster, the blue quick shelters erected that symbolize displaced persons. This is done by creating ten machine-learning models and tuning them with selected parameters. Ultimately, to find the optimal model that can effectively and accurately parse the image data to identify the quick shelters.

Additionally, this project will attempt to address if there are any unforeseen issues with this methodology of disaster relief aid from a realistic standpoint. Specifically, model image discernment errors due to the environmental variables, as well as computational resource expenditure.

Overall, this project gives a conceptual overview of the application of model building, tuning, and evaluation to give future disaster relief efforts an ease in providing effective aid. As those who need aid can be identified from image data, then relief efforts can more accurately estimate the amount of people displaced as well as the amount of aid required to save lives.

Data

Column Name	**Description**
Class	Truth of the Classification transformed into Binary Response Two levels: Representing whether a Blue Tarp was present or not ● TRUE (Blue Tarp present) ● FALSE (Blue Tarp not present)
Red	Numeric RGB value associated with Red ranging from 000-255
Green	Numeric RGB value associated with Green ranging from 000-255

Blue	Numeric RGB value associated with Blue ranging from 000-255

Figure 1: Table describing project variables.

EDA

Figure 2, shows the trend of the data in a three dimensional space. Where 'Class' is to note the color type that coordinates with the structure type picked up by the overhead image, and how it was classed. The 'Class' as color is relevant as the pixel trends in relation to shelter structure. It appears that 'Blue Tarp' pixel values are centered at Blue pixel values of 150 to 250. Additionally, they are likely to be accompanied by dense green pixel values,but less frequently accompanied by higher red pixel values.

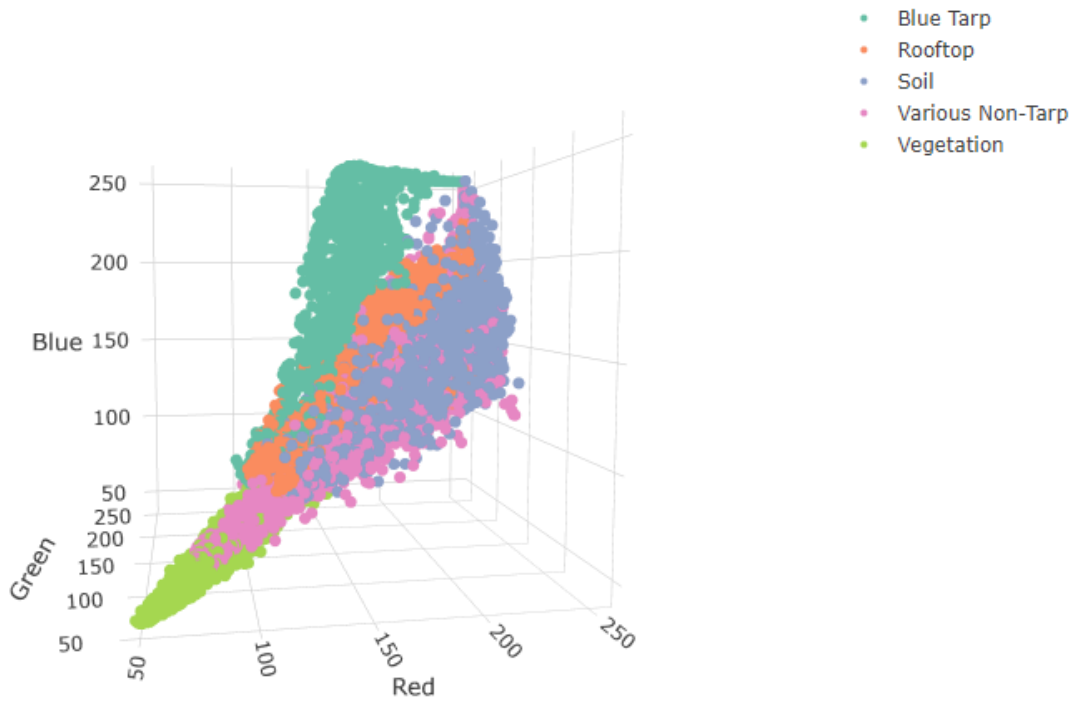

Figure 2: Plotly graphic of Haiti column data colored by class.

Figure 3 shows the correlation of the various pixel colors to the Classes of True and False. Examining the scatterplots in the correlation plot it would appear that the True classification of a Blue Tarp presents from pixel values of 200 to 250. This correlation could be due to the color value association on the color wheel of those pixel values being registered as Blue Tarp present.

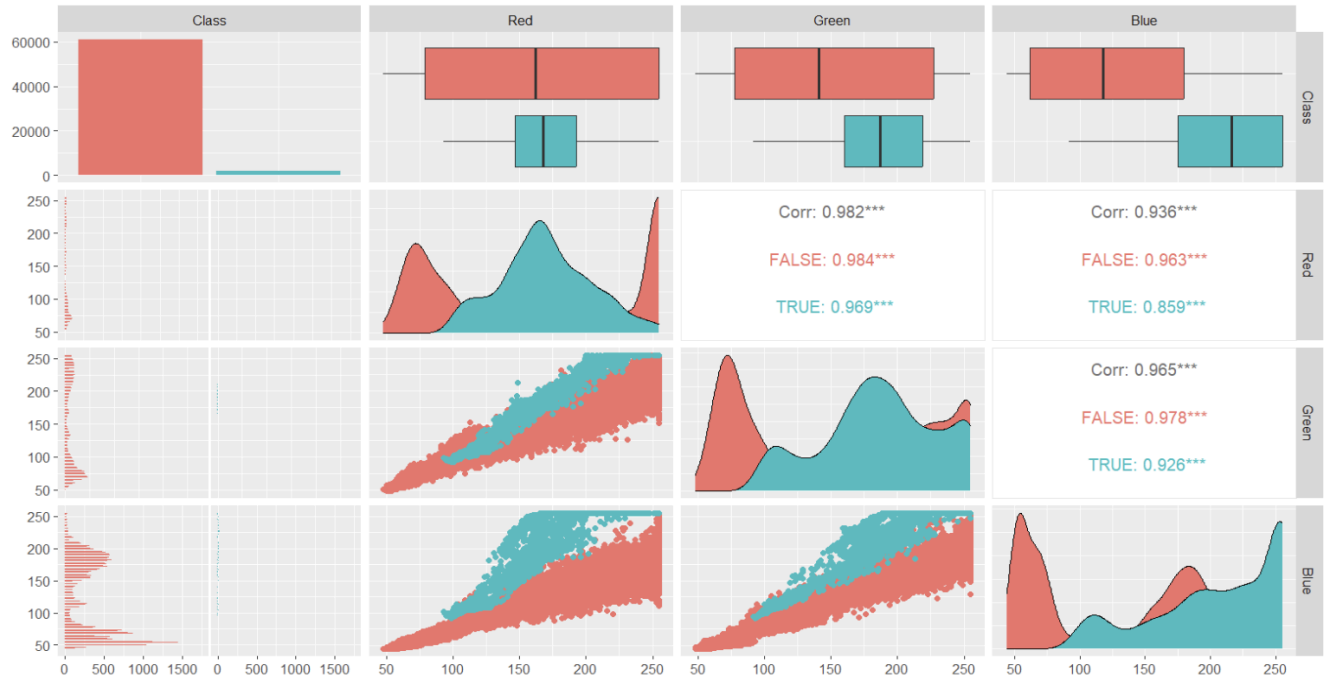

Figure 3: Correlation plot of Haiti data.

Figure 4 shows where the Haiti data was preprocessed to show the counts of 'TRUE' and 'FALSE'. 'FALSE' being, not 'Blue Tarp', 'TRUE' being 'Blue Tarp'.

Class	Count
FALSE (not a blue tarp)	61219
TRUE (a blue tarp)	2022

Figure 4: Table of Tarp Classification by Count.

Figure 5 shows the boxplots of the different pixel values for the various colors. The means of the FALSE class across the colors, Red and Green are similar. However, the Blue pixel values have the lowest mean amongst FALSE classifications.. The Red pixel value has the lowest mean among TRUE classifications followed by Green then Blue. As one might expect, high concentrations of Blue values correspond to TRUE classifications–akin to seeing a blue tarp.

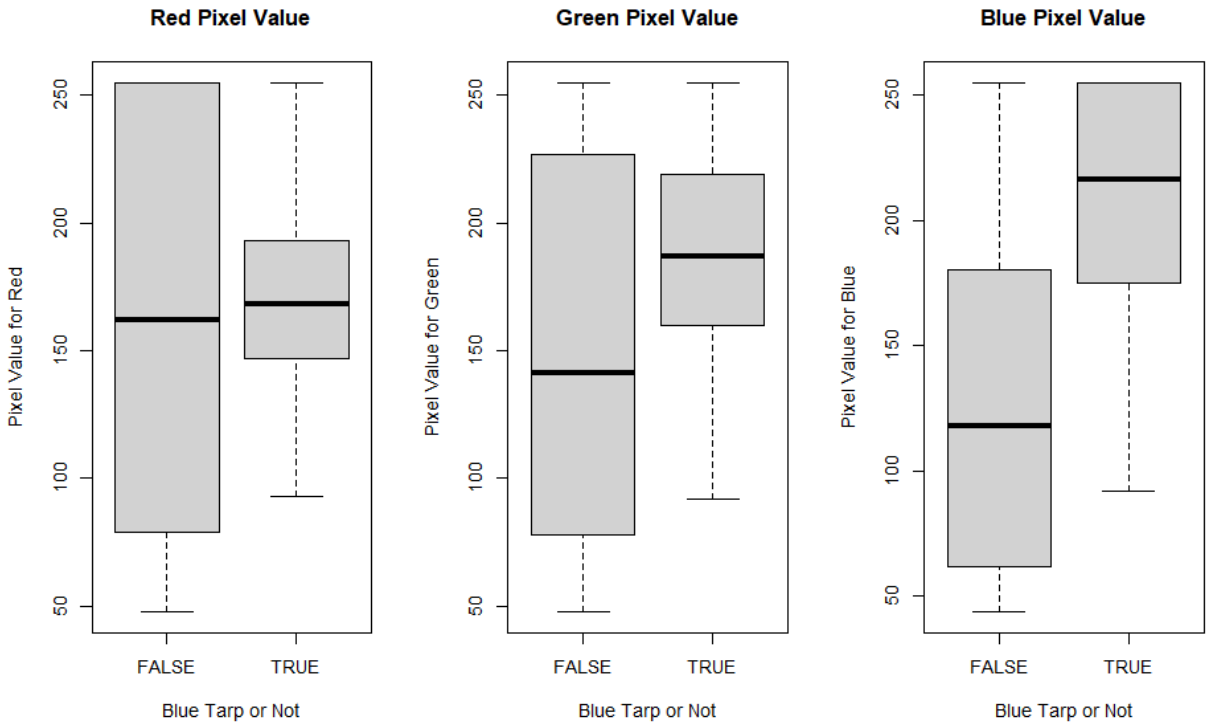

Figure 5: Boxplots of Various Pixel Values Separated by Blue Tarp or Not.

EDA (Test Column Determination)

The test data given was formatted in several separate files that needed to be compiled into one. Also, the pixel color columns were randomly labeled ‘B1’, ‘B2’, and ‘B3’ instead of ‘red’, ‘green’, or ‘blue’. Further exploration was required to determine the proper mapping to the test data columns.

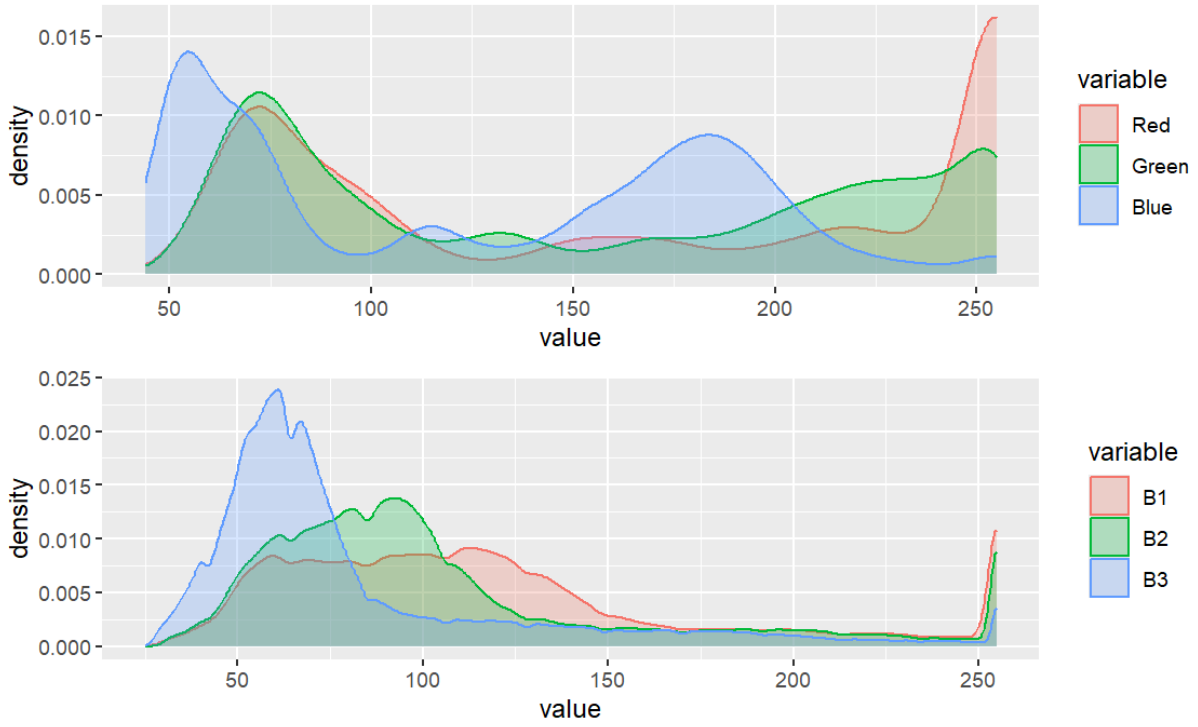

Figure 6: Density Distribution Comparison Between known RGB Columns and Unknown RGB Columns.

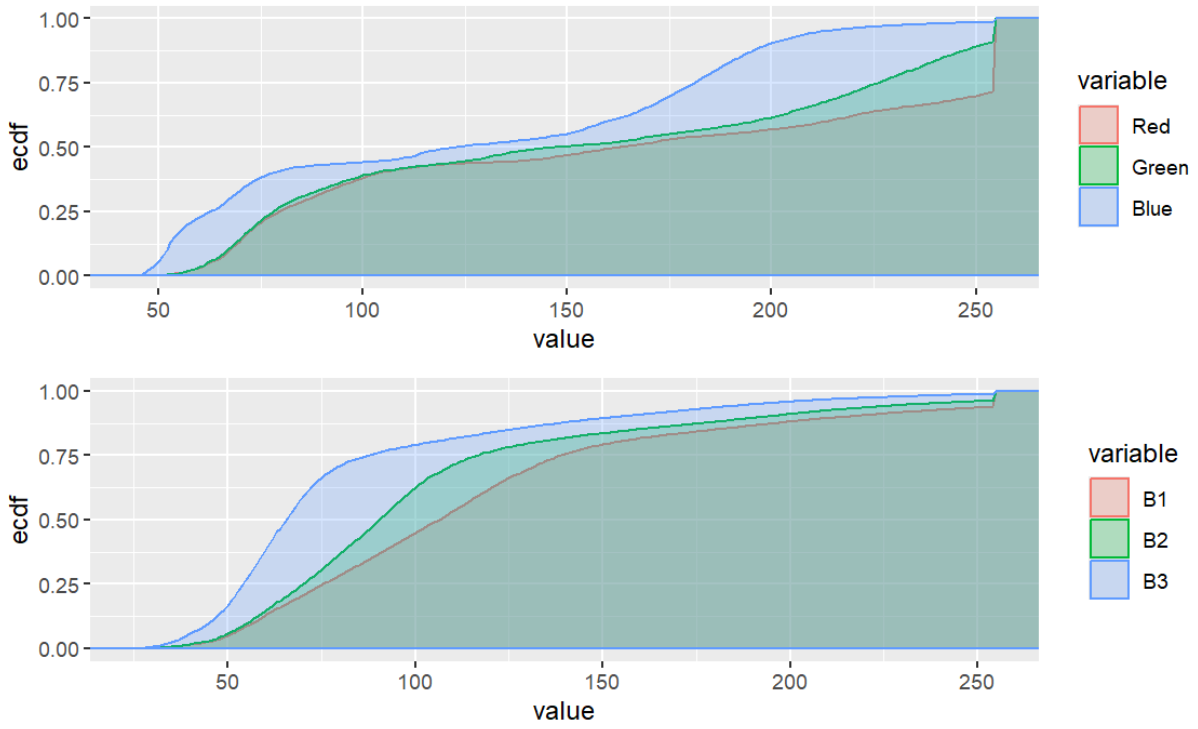

Figure 7: Cumulative Density plots of known Red, Green, and Blue Columns compared to unknown columns.

Above in Figure 6, the density plots for the 3 columns: Red, Green, and Blue from the training data are shown. Directly below that, the undetermined columns from the compiled test data can be seen. In Figure 7, the cumulative density functions of the known Red, Green, and Blue columns along with the unknown columns show similar patterns that are seen in Figure 6. Blue stays on top of Green and Red, while in the bottom plot, B3 stays on top of B2, and B1. It is shown that Green is in the middle, and so is B2. Also, Red stays below the other two cumulative density plots and this lines up with B1 in the bottom plot.

With the combination of Figure 6 and Figure 7, the determination is made that the density distributions of Red, Green, and Blue match with B1, B2, and B3, respectively.

Description of Methodology

To begin the analysis, the data must be understood and pulled together into two manageable sets, the training data and the test data. The training set was pre-existing while the test set required extensive cleaning, interpretation, and merging into one table. As discussed above, the alignment for Red, Green, and Blue on test columns were determined and thus the model building can start. The models built were the following: a logistic regression model, linear discriminant analysis model, quadratic discriminant analysis model, tuned nearest neighbor model, tuned penalized logistic regression model, tuned random forest (ranger) model, tuned boosting model, tuned linear support vector machine, tuned polynomial support vector machine, and tuned radial basis functions kernel support vector machine model. These 10 models will serve as the base for analysis.

Now, to determine the best parameters for all of the tunable models. Each tunable parameter for each model was tuned using processes described in detail in ‘Parameter Tuning and Model Selection’.

The next step was to cross-validate the models. This was done with 10-fold cross-validation. Metrics, specifically the ROC AUC, were produced for these cross-validated models. Now that a cross-validated version of all 10 models has been built and the tunable models have been tuned, computing metrics comes next. ROC curves, ROC AUC values, accuracy, sensitivity (TPR), specificity, precision, and (FPR) were all calculated for each of the models, and ROC AUC values were calculated on the test and training data for comparison.

After cross validation, thresholds were determined for each model using a maximized J-index. The J-index serves as the metric that allows for the comparison across different thresholds for each of the models. In this step, the J-index serves as a balance between false positives and false negatives. This is described and justified in more detail below.

Each model was then executed on test data and their performance was evaluated against training data and the other models in the group using ROC AUC values as well as ROC curves. Each model was also compared using accuracy, sensitivity, specificity, precision and FPR at their selected threshold.

Description of Used Software

R studio was used for the statistical computing, analysis, and graphical displays for this project. The tidyverse package was heavily used for the data visualizations and graphic displays. While, tidymodels methodology was implemented for the model specifications, tuning, and optimal model selection.

For the ensemble models the following engines were used: 'xgboost,' 'ranger,' for the boosted and random forest model building respectively. The support vector machines models engines used was the 'kernlab' engine. These engines were used as they are a necessity for constructing these models and further tuning them with the found specified optimal parameters during the autoplot analysis and best selection process.

Parameter Tuning and Model Selection

For parameter tuning ROC AUC was selected, which is the measure of the area under the Receiver Operating Characteristic curve, as a measure of selecting the best tuning parameter for the Nearest Neighbor model, and the Penalized Logistic regression. The reason ROC AUC was chosen in this case is to be able to compare the performance across a wide range of neighbor values, and ROC AUC is an excellent comparative metric that measures, "overall performance of the classifier, summarized over all possible thresholds". (Chapter 4, An Introduction to Statistical Learning).

For the nearest neighbor model, the tunable parameter is the number of neighbors, and a randomized grid of 30 values ranging from 2-100 is used to select a neighbor value for each of the models created. The optimal neighbor value is determined by selecting the best-performing option from the plot of ROC AUC against the number of nearest neighbors. Refitting the tuned model with this parameter, the next step is to look at the penalized logistic regression model.

For PLR, the tunable parameters are both penalty, ranging from -5 to -1, and mixture, ranging from 0 to 1, and once again a randomized grid is used to choose the tested mixture and penalty values for the 10 models created. The ideal values of mixture and penalty values are chosen using the maximum ROC AUC provided by tuning. Finally, refitting the models with this tuning parameter as done above.

Next, the model tuning for the random forest (ranger) model was conducted and the tuned parameter was the ‘mtry’ value ranging from 2 to 8, and the grid was randomized over 10 iterations.

Next, the model tuning for the boosting model was conducted and the tuned parameters were the ‘tree depth’ and the ‘learn rate’ parameters, both using default ranges and the grid was randomized over 10 iterations.

Next, the model tuning for the linear SVM model was conducted and the tuned parameters were the ‘margin’ and ‘cost’, both set to default ranges, and the grid was randomized over 10 iterations.

Next, the model tuning for the polynomial SVM model was conducted and the tuned parameters were the ‘margin’, ‘cost’, and ‘degree’ and they were all set to default ranges, and the grid was randomized over 10 iterations.

Lastly, the model tuning for the SVM RBF model was conducted and the tuned parameters were the ‘margin’, ‘cost’, and ‘RBF sigma’. ‘Margin’ and ‘cost’ were set to their default ranges but ‘RBF sigma’ was set to test from -4 to 0, and the grid was randomized over 10 iterations.

Passing these tuned models through the autoplot function shows the trend of ROC AUC values varied by the number of each of the individual parameters. The trend is noted by the shape/behavior of the trend of ROC AUC values across the various ranges discussed above, and this leads to the selection of the best model parameters for each of the tuned models.

Model Validation

For the model validation, cross-validation methods were used. 10 defined folds across each model were used in order to check that the model performance was satisfactory and to detect if overfitting occurred. In this case cross-validation methods were selected because the goal was to find the model performance, and bootstrapping would be inappropriate in this case because bootstrapping produces the distribution of model parameters. Cross-validation is the optimal choice of model validation in this as it can gauge the presence of overfitting and another metric for model performance comparison.

Threshold Selection

For the type of threshold selection, the J-index was used as it ensures a trade-off balanced with consideration of both false positives and false negatives in the model evaluation. By assigning equal weight to these misclassification errors, tests with the same J-index value will demonstrate a consistent proportion of total misclassified results. The threshold selection was modified for the 2010 Haiti data to assume the nuance between simulated data versus real-world

applied action. To tackle this nuance, there was a thought about maximizing the sensitivity metric. This would in turn avoid false negatives at all costs or in other words, never misclassify a Blue tarp as anything other than a Blue Tarp. This would ensure that everyone who needed aid would receive it. However, in a real-world scenario where resources are limited, it would be best to maximize the J-index. Therefore, the model chosen by the maximized J-index applies to real-world relief efforts and takes into account that the resources that the government would be willing to provide are limited and makes a more realistic evaluation of resource allocation in that sense.

Performance Metrics and Model Performance Evaluation

As described in the tuning parameters methodology, the ROC AUC metric was selected as it is a cumulative metric representing the model's ability to classify the response across the thresholds. The ROC AUC is inherently a comparative metric. In the comparison of the ten created models selected values were quite high in the test and training data. The ROC AUC value is a beneficial metric in determining the model performance, but if it is supported by metrics such as accuracy, sensitivity (true positive rate), specificity, precision, and false positive rate, we can conclude in context which model performs the best. In this scenario as described above, it is extremely important to balance both the finite aid resources that the government can provide and the livelihood of those in the quick shelters. That is why ROC AUC, in tandem with our other metrics, allows us to determine which model performs the best in context.

Results

The data were provided in multiple parts that did not require direct splitting. After which, we developed 10 models using logistic regression (LR), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbor (KNN), penalized logistic regression (PLR), random forest, boosting, SVM linear, SVM poly, and SVM RBF. Each model was then evaluated using 10-fold cross-validation. KNN, PLR, SVM linear, SVM poly, and SVM RBF all required tuning steps in order to achieve better results than default values.

Next, the results of the parameter tuning described above are shown. The metric used to determine the chosen parameter values for all of the below was the ROC AUC.

In Figure 8, the ideal ROC AUC stabilizes at k=47.

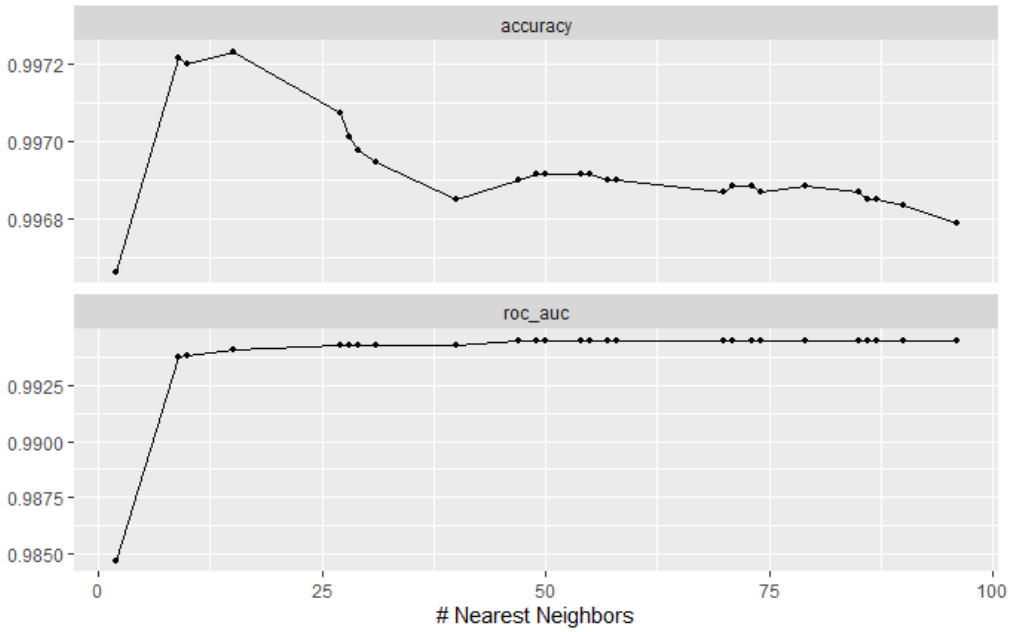

Figure 8: ROC AUC and Accuracy values varied across KNN values.

In Figure 9, the ideal penalty was 0.00158 and the ideal mixture was 0.578.

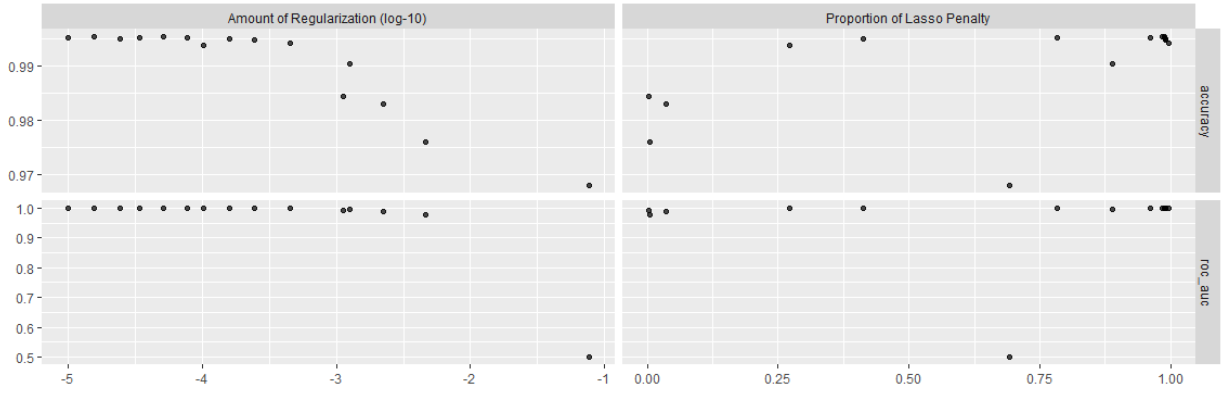

Figure 9: ROC AUC and Accuracy varied by ‘penalty’ and ‘mixture’ of PLR.

In Figure 10, the ideal mtry was 2 and the ideal min_n was 9.

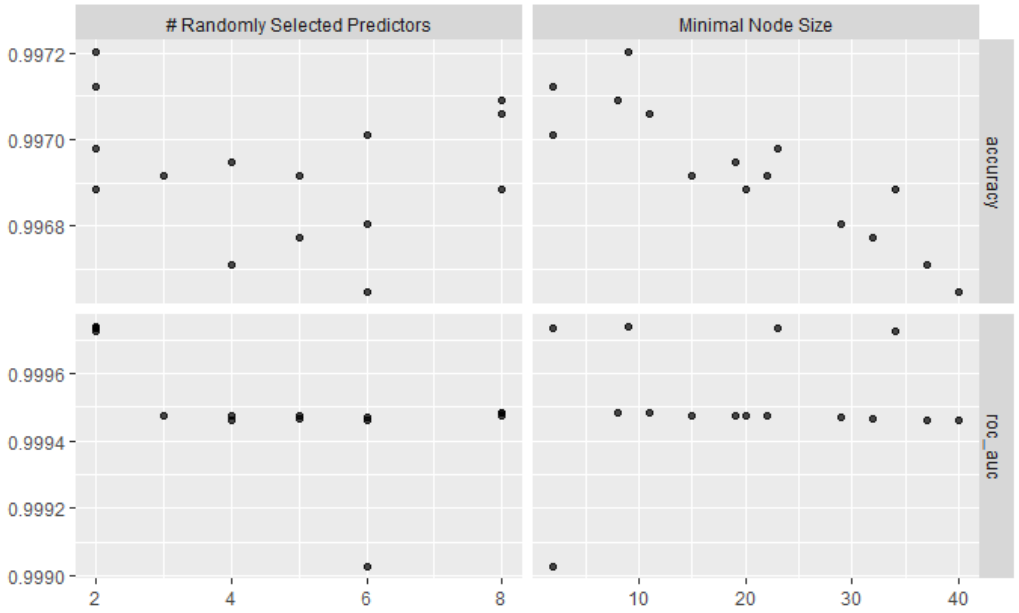

Figure 10: ROC AUC and Accuracy varied by ‘mtry’ and ‘min_n’ of Random Forest.

In Figure 11, the ideal tree depth was 8 and the ideal learn rate was 0.0098.

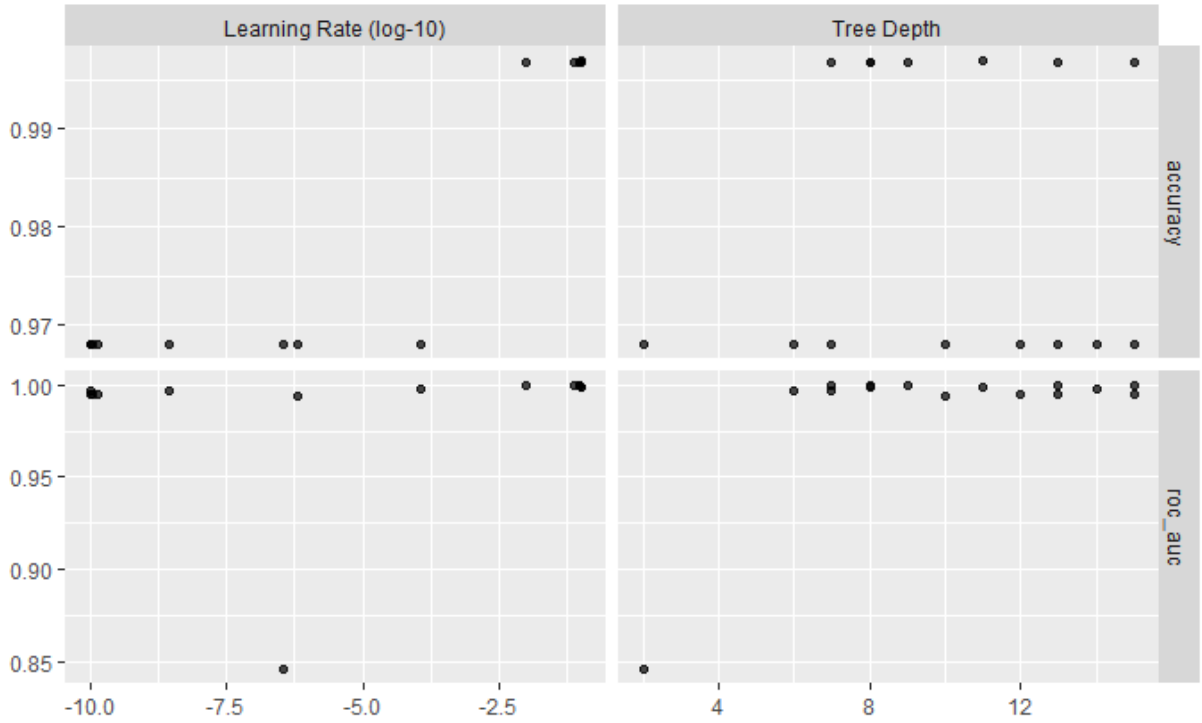

Figure 11: ROC AUC and Accuracy varied by ‘tree depth' and 'learn rate’ of Boosting

In Figure 12, the ideal cost was 31.99 and the ideal margin was 0.084.

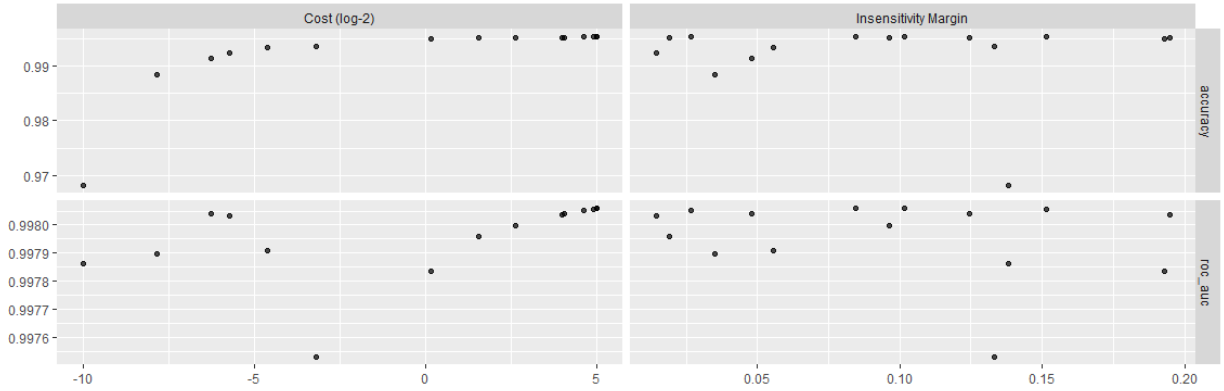

Figure 12: ROC AUC and Accuracy varied by ‘margin’ and ‘cost’ of SVM Linear

In Figure 13, the ideal cost was 0.064, the ideal margin was 0.106, and the ideal degree was 3.

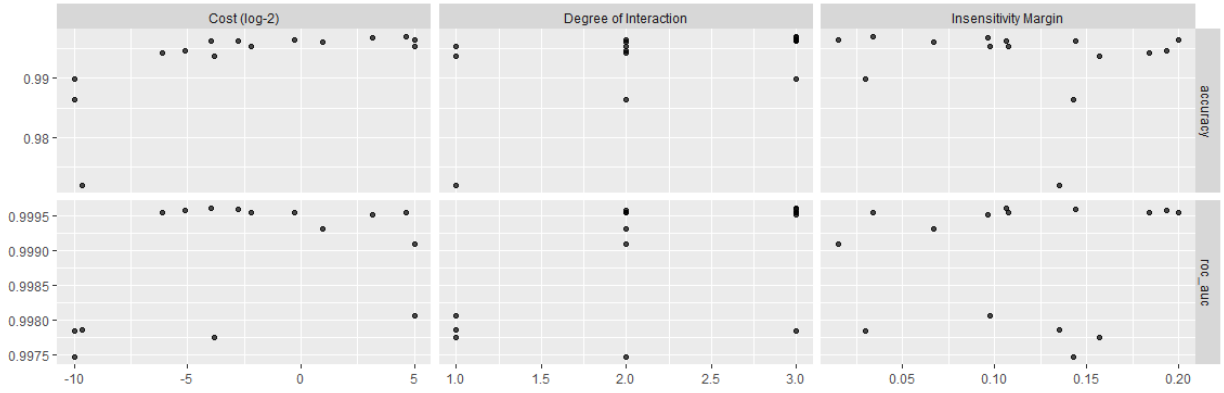

Figure 13: ROC AUC and Accuracy varied by ‘margin’, ‘cost’, and ‘degree’ of SVM Poly.

In Figure 14, the ideal cost was 31.91, the ideal margin was 0.157, and the ideal RBF sigma was 0.986.

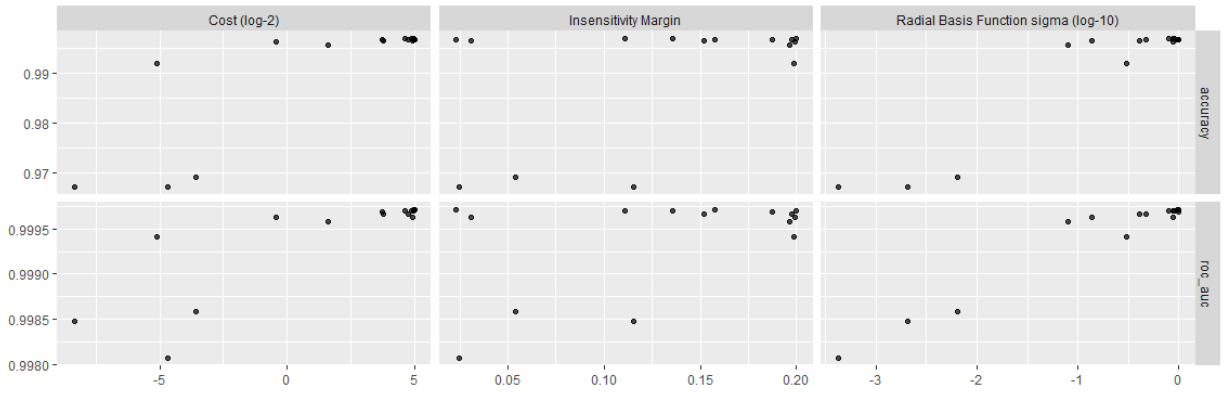

Figure 14: ROC AUC and Accuracy varied by 'margin', 'cost', and 'RBF sigma' of SVM RBF.

With each model tuned as needed, ROC plots were assessed to determine prediction thresholds. In each ROC plot, a J-index exists such that misclassified information will be equal across any test at that index.

As shown in each of the graphs below in Figures 15 and 16, the ROC curves are heavily left skewed, having maxima near 0 specificity. The optimal J-index exists at the max distance between the ROC and the diagonal dashed line (random classifier).

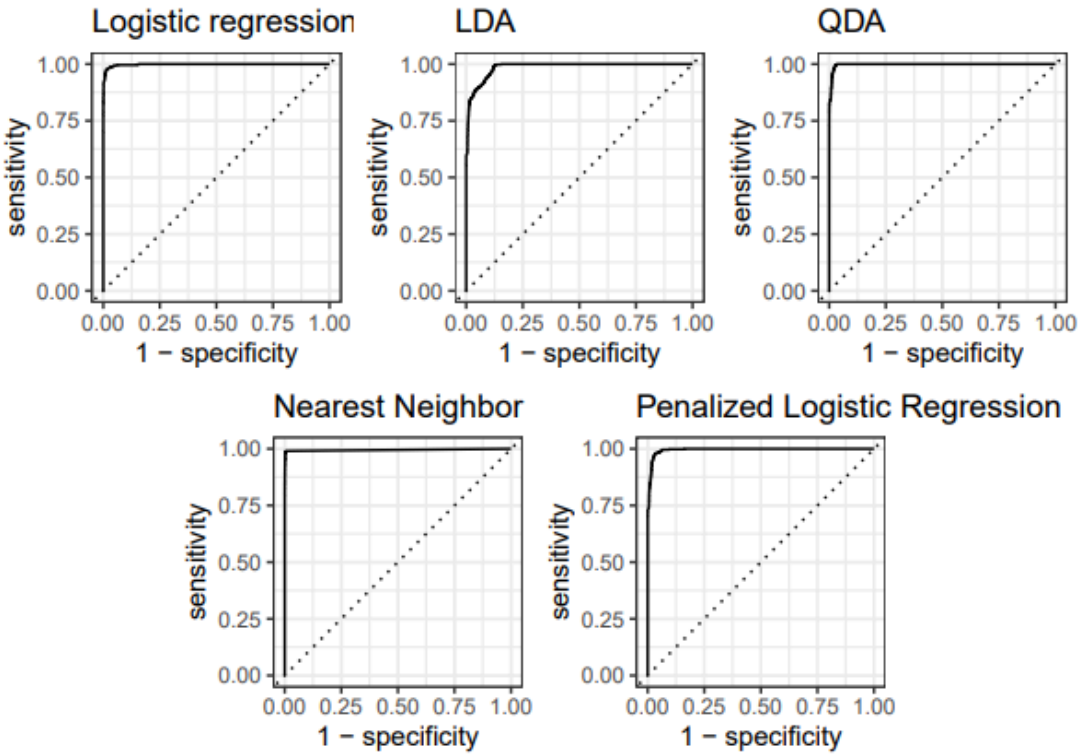

Figure 15: ROC curves of the first 5 models.

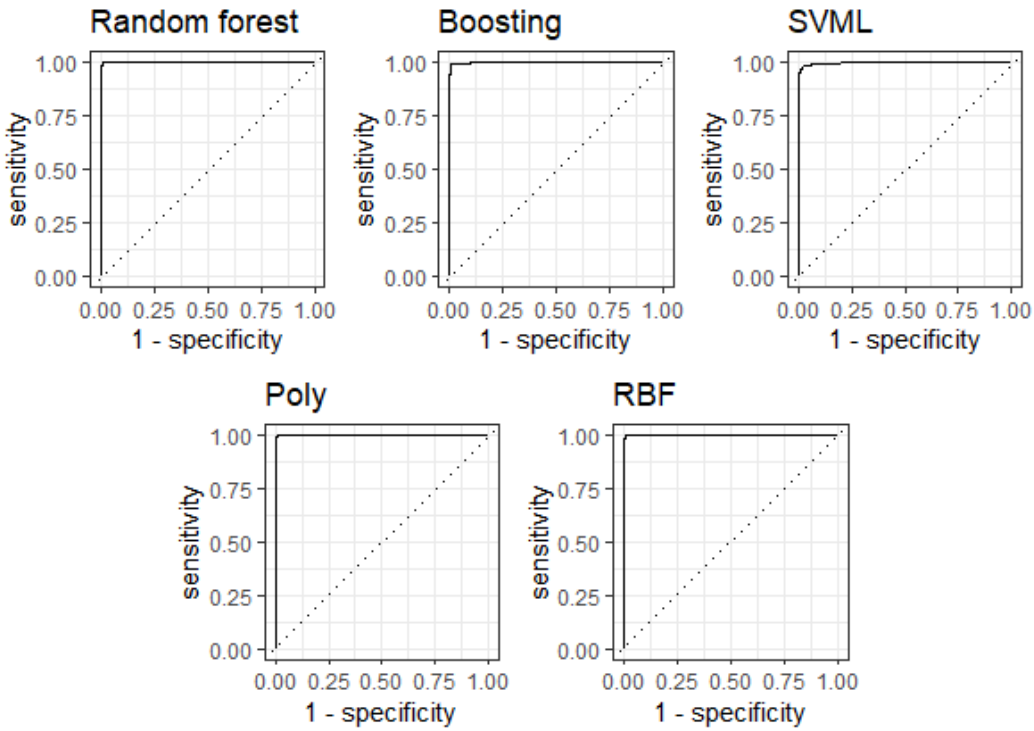

Figure 16: ROC Curves of the last 5 models.

Below in the plots in Figures 17 and 18 show plots of the J-index vs a range of threshold values, and the information pertinent in these plots is the maximized J-index plot. The vertical line corresponds to the threshold value that maximizes the J-index, and as such was our chosen threshold for each of the models.

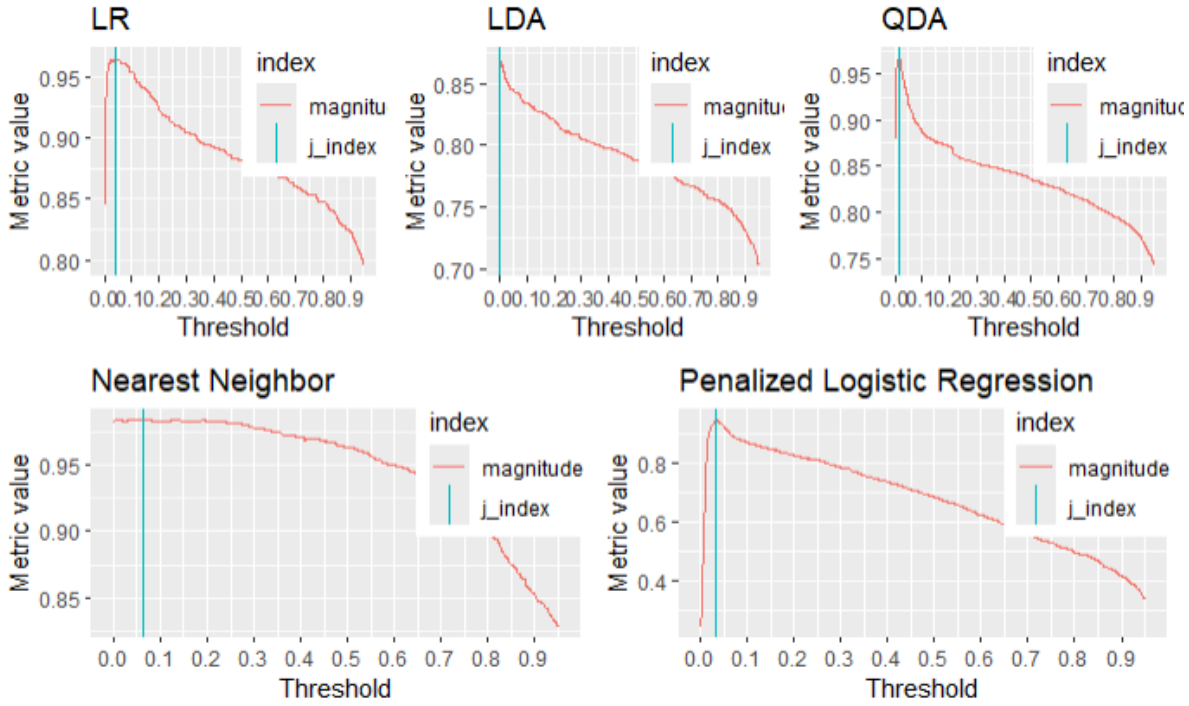

Figure 17: J-index thresholds of the first 5 models.

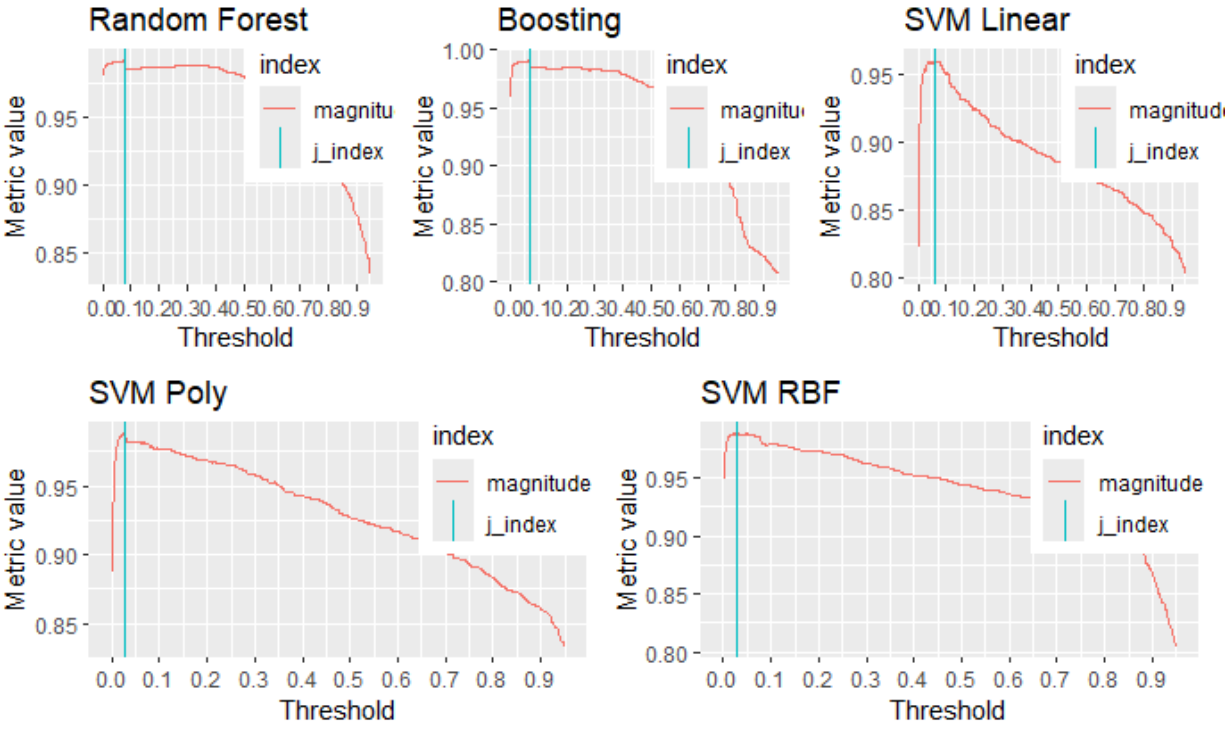

Figure 18: J-index thresholds of the last 5 models.

Model was then evaluated at the threshold for their accuracy, sensitivity, specificity, precision, and FPR using the 'probably' library.

Model	**LR**	**LDA**	**QDA**	**KNN**	**PLR**
Thresholds	0.051	0.003	0.019	0.132	0.047
Model	**Random Forest**	**Boosted**	**SVM Linear**	**SVM Poly**	**SVM RBF**
Threshold	0.08	0.07	0.061	0.026	0.032

Figure 19: Table of threshold values determined at each maximum J-index.

In order to compare model performance, a metric calculation was performed across each model on both test and training data, along with the cross-validated results comparing ROC AUC values.

Model	**LR**	**LDA**	**QDA**	**KNN**	**PLR**
ROC AUC (CV)	0.999	0.989	0.998	0.995	0.999
ROC AUC (Train)	0.999	0.989	0.998	0.994	0.999
ROC AUC (Test)	0.999	0.992	0.991	0.965	1.0
Model	**Random Forest**	**Boosted**	**SVM Linear**	**SVM Poly**	**SVM RBF**
ROC AUC (CV)	1.0	0.999	0.998	1.0	1.0
ROC AUC (Train)	1.0	1.0	0.998	1.0	1.0
ROC AUC (Test)	0.980	0.974	0.999	1.0	0.982

Figure 20: ROC AUC values of the 10 models regarding CV and test/train datasets.

Each model's performance on the test data is greatly different from training data, so there is not a noticeable sign of over or underfitting. Accuracy, sensitivity, specificity, precision, and FPR were then calculated for each model at their ideal threshold.

Model	**LR**	**LDA**	**QDA**	**KNN**	**PLR**
Threshold	0.037	0.002	0.014	0.065	0.044

Accuracy	0.985	0.958	0.971	0.996	0.963
Sensitivity (TPR)	0.979	0.910	0.996	0.988	0.955
Specificity	0.986	0.960	0.971	0.996	0.963
Precision	0.693	0.428	0.528	0.892	0.462
False Positive Rate	0.0143	0.0402	0.0294	0.00397	0.0367
Model	**Random Forest**	**Boosted**	**SVM Linear**	**SVM Poly**	**SVM RBF**
Threshold	0.08	0.07	0.061	0.026	0.032
Accuracy	0.992	0.994	0.986	0.989	0.990
Sensitivity (TPR)	1	1	0.973	1	.999
Specificity	0.992	0.994	0.986	0.989	0.990
Precision	0.795	0.844	0.699	0.748	0.763
False Positive Rate	0.00849	0.00609	0.0138	0.0111	0.0103

Figure 21: Table of performance metrics of the 5 models.

Considering the efficacy of each model, it is hard to distinguish whether one model is objectively better than another. One could argue marginal gains from using one model from another, but ultimately, the difference is not significant enough to warrant a clear-cut stance. While trying to provide the most amount of aid to displaced peoples, it may be beneficial to look at the model that maximizes sensitivity in which case either Random Forest, Boosted, or SVM Poly should be selected as they provide the 'best' results. Of the subset, Boosted appears to perform the best having a lower FPR and higher accuracy as well as precision.

Conclusion

Optimal Model Selection

While some algorithms performed exceptionally well in cross-validation, their performance slightly decreased on the hold-out data (Random Forest, Boosted, and SVM RBF).

This can be attributed to differences in the distribution of data between the training and test sets, as well as potential overfitting to the training data relative to the other models.

Of the models, LR, PLR, SVM Linear and SVM Poly performed the best across the cross-validation and test dataset. Notably, SVM Poly performed at 1.0 AUC across CV, training, and test sets.

In summary, PLR and SVM Poly emerge as the best-performing algorithms on the hold-out data, demonstrating their ability to generalize well to unseen instances. However, it's essential to consider the context of the problem, computational efficiency, and interpretability when selecting the final model for deployment.

Model Justification

In the context of the problem, the ideal model for providing aid in the event of a natural disaster errs on the side of overestimation. Ultimately, human lives should not be lost due to the lack of relief aid and resources. It is better to project on the side of needing more resources than to be strictly budgeted and require more. With that in mind, a model that produces more false-positive results would be preferred over one that produces more false-negative results. This is achieved by choosing the model that has the highest model sensitivity of the models produced. According to Figure 21, one could choose Random Forest, Boosting, or SVM Poly as each model has a 1.0 sensitivity. Arguments could be made for SVM RBF as well as the sensitivity is .999.

Notably, Boosted appears to have the greatest accuracy at 99.4%, but other models do not trail far behind. Random Forest had an accuracy of 99.2%, SVM Poly was at 98.9%, and SVM RBF was 99.0%.

Depending on the available physical resources, aid could be provided liberally while using SVM Poly to classify the tarps as the FPR is highest amongst the best models. A body looking to be more conservative with aid could use Boosted as the FPR is minimal.

Recommendation & Rationale

Considering strictly model performance and the need for aid, it is suggested that SVM Poly is used. Although it is computationally intensive, it would provide a liberal classification of blue tarps to assure that the least number of human lives are lost due to a lack of provided aid while not significantly affecting model performance.

Scale of true positives to total number of rows in set

An extremely important note that has come up in the work of this project is the rates of actual blue tarps in comparison to non blue tarp classifications. There is such a large difference

in scale of how few blue tarp classifications there were that it is hard to say that the models are performing well when looking at metrics in Figure 21. An accuracy of 96.8% (ignoring all other metrics) would be the same as guessing all non-blue tarps, as only 3.2% of the training data has true blue tarp classifications. Another way of saying this is that a model that just guesses false every time would still be accurate 96.8% of the time. This needs to be taken into consideration very heavily when looking at the metrics as it directly affects the conclusions that determine the best model.

It is also important to discuss that this is extremely important in the context that this work is being completed. The aid provided to Haiti would be zero using the 'all falses' model example yet, as has been said, that model is 96.8% accurate. It would be absolutely unacceptable to send Haiti no aid as a result of this analysis so there needs to be further analysis in addition to the metrics in Figure 21.

The potential solution to this problem is to look at the accuracy of the models in context with the other metrics. Figure 20 provides the ROC AUC values of all of the models and this allows for comparison from model to model of its predictive ability. As can be seen in Figure 20, the models all perform extremely well. Using this information on top of the information in Figure 20 the conclusion can be drawn that the model with the highest ROC AUC value on the test and cross-validated, and the highest sensitivity in Figure 21, which is why we have above given the recommendation of using the SVM polynomial model.

Suggestions for Improvement

Several minor improvements could significantly impact the quality of data modeling efforts. Improvements include: restricting the time of day each photo is taken, improving camera quality, and filtering images of liminal space.

By restricting the time of day each photo is taken, each image is shifted by a similar amount due to the presence or lack of light. One would expect images taken at night to be more blue-shifted than images taken during the day, which could have an effect on the efficacy of the model.

Additionally, improving the camera quality could also have an impact on the model's performance. Higher quality cameras are able to capture higher quality images, resulting in better pixel color diversity. The model would potentially be able to distinguish the blue of a tarp better than the blue of light reflecting off of a stream of water.

Lastly, a large portion of the data ingested by the model is unoccupied land. This creates a large disparity between true guesses and false guesses where a model guessing false every time would be correct 96.8% of the time. Eliminating images that do not contain any blue tarps would enable models to be biased less toward estimating false all the time.

Practical use of the Models for Computational Efficiency

While working on this project the nuance of optimal model selection in a simulation vs. practical has been acknowledged as an interesting topic to explore. Real-world application can be a complex decision based trade off, as such many parameters may be considered to justify model selection. Specifically, asking the question, “While this model performs the best for demo testing, is it the best to be launched for live application?” This becomes an even more critical question when this is factored with the situation of providing aid to those in a natural disaster situation. The aftermath of a natural disaster can devastate communities rapidly, as such, time becomes a critical factor in saving the most lives. Therefore, in the context of the findings from this project, it may not be the best choice to choose the most optimal model from this perspective, as it may not consider other critical extraneous factors.

Looking at the models that performed the best from Figure 21 the ensemble models along with the SVM poly model have the highest sensitivity of identifying Blue Tarp presence and they do so accurately. So, from a practical standpoint these models would be considered for live application to disaster relief aid. However, it is important to consider how fast and efficiently aid can be distributed to those in need. So, how fast and accurately can Blue Tarp shelters be identified?

Since the data has such a small amount of Blue Tarps, it is better to use a model that is tuned for sensitivity as it would be the most computationally efficient in providing results that accurately assess the amount of Blue Tarps present. In this case it would be the PLR model. While this model may not be the optimal of all models, it would be the better choice as it can more rapidly identify the Blue Tarp locations, so those providing relief can locate them and provide aid as quickly as possible.

Applications to Help Save Human Lives

The work done in this project has the potential to be quite effective in saving human lives as it can assess if a quick shelter is present accurately, garner its location, and parse the data in an efficient amount of time. However, the application of this project would have to be integrated with pre-existing relief effort systems and protocols, which could be quite difficult.

As an example, the output of the models needs to be readable for the emergency response teams, who can then relay that information to those in the disaster relief zone. This needs to be done in a way that those actively involved in the relief efforts can understand the information that is being passed along to them. So, this may involve another protocol application on the end of the emergency response teams that would then have to be adopted and standardized across relief efforts.

Speaking of standardization, the work here will have to be scalable to large land areas. Haiti is a smaller land location therefore, there may be hindrance in the computing strength in say a larger scale land issue, such a tsunami or hurricane that has the potential to devastate a large land area.

Finally, applying this work to other types of natural elemental disasters; such as volcanic eruptions, sand storms, and landslides. Those elemental disasters stir up particulates in the air that could distort the image quality and data received from the geo-referenced imagery. As such, this work could be helpful in saving human lives if it was properly integrated with pre-existing natural disaster relief efforts, or perhaps new standards could be created to streamline the adoption of this methodology.