DS6030 Disaster Relief Project 1

Group 4: Alanna Hazlett, Etienne Jimenez, Venkat Viswanathan

Introduction

In 2010 Haiti was hit by a devastating earthquake. Many homes were destroyed and people were making shelters utilizing blue tarps. In an effort to provide assistance with food and water, the Rochester Institute of Technology was flying over the country capturing images to locate people utilizing these blue tarps. They did not have enough time or manpower to sift through these photos by hand to locate people, as they wanted to get help to them as quickly as possible. They utilized the pixel colors and the amounts of those pixels to create a model to identify where the blue tarps are. Here, we too are building a model to identify the blue tarps based on pixel data in an effort to quickly find survivors who may need assistance. Once the model is built, we could utilize it on future data, say from another natural disaster where we know people are using blue tarps, to make these same predictions. This will help speed up recovery efforts and provide support to survivors.

Data

The data we utilized to train our model was in a csv format, called HaitiPixels.csv, and had the following variables.

- Class, a categorical variable describing the landscape/architecture identified in the photograph. Comprised of Vegetation, Soil, Rooftop, Various Non-Tarp, and Blue Tarp. Blue Tarp is the category of interest for predictions.
- Red, a numeric variable for the amount of red pixels in the photograph.
- Green, a numeric variable for the amount of green pixels in the photograph.
- Blue, a numeric variable for the amount of blue pixels in the photograph.
- For building our model we created a dataframe called binary_training_data, which is comprised of the same columns as training_data, however Class was mutated into a binary category, with NotBlueTarp and BlueTarp. BlueTarp is the category of interest for predictions.

The data provided for us to make predictions on was separated into multiple txt files.

- orthovnir057_ROI_NON_Blue_Tarps.txt
- orthovnir078_ROI_Blue_Tarps.txt
- orthovnir069_ROI_NOT_Blue_Tarps.txt

- orthovnir069_ROI_Blue_Tarps.txt
- orthovnir067_ROI_NOT_Blue_Tarps.txt
- orthovnir067_ROI_Blue_Tarps.txt
- orthovnir078_ROI_NON_Blue_Tarps.txt
- orthovnir067_ROI_Blue_Tarps_data.txt

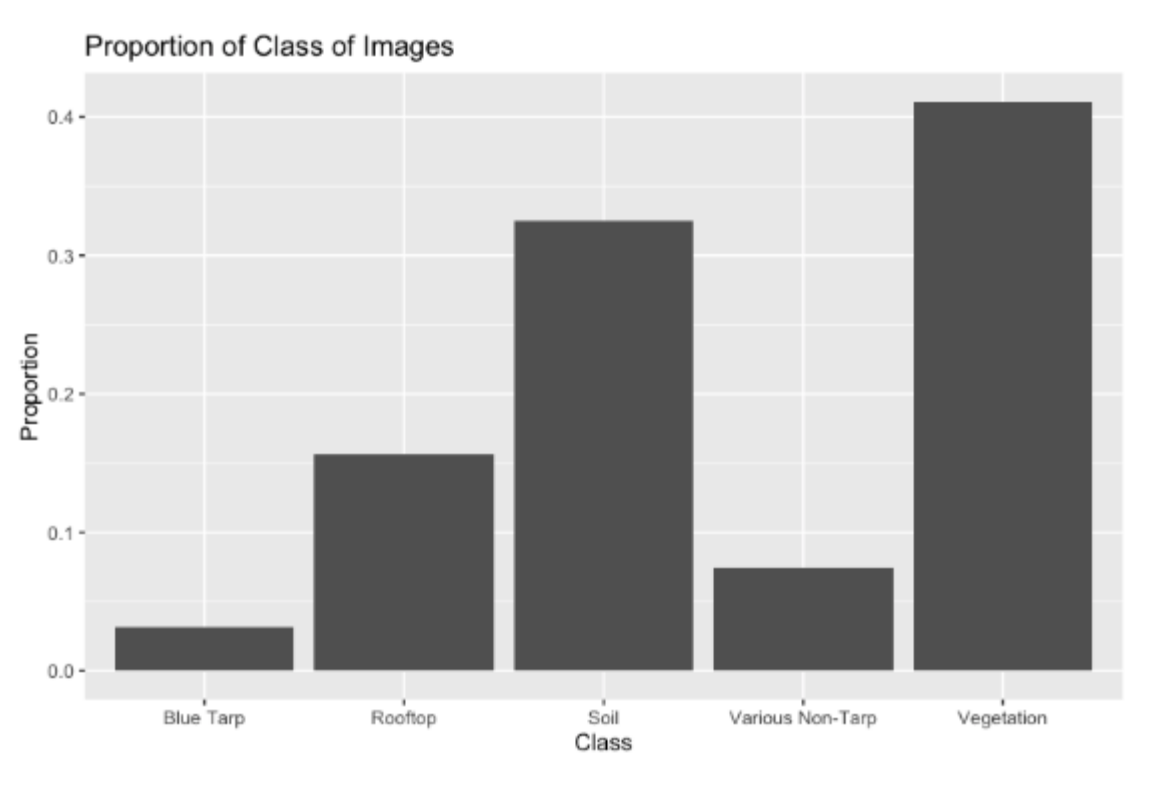
All these files except orthovnir067_ROI_Blue_Tarps_data.txt were used to create our holdout dataframe, as this file is a repeat of orthovnir067_ROI_Blue_Tarps.txt. The first 8 lines of the txt files are metadata we did not need, so they were skipped. We only retained the relevant columns we needed to make our predictions, B1, B2, and B3. These columns correspond to Red, Green, and Blue in our training data, and we used techniques to determine which column in the holdout corresponded with which column in the training data.

Description of Methodology

To perform our calculations and analysis, we employed RStudio's libraries **Tidyverse, Tidymodels, discrim, patchwork, probably, reshape2 and doParallel**. All models were trained using Tidymodels' functions, such as recipe, workflow, as well as their pre-built functions to carry out specifications, training, and fitting for logistic regression, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA).

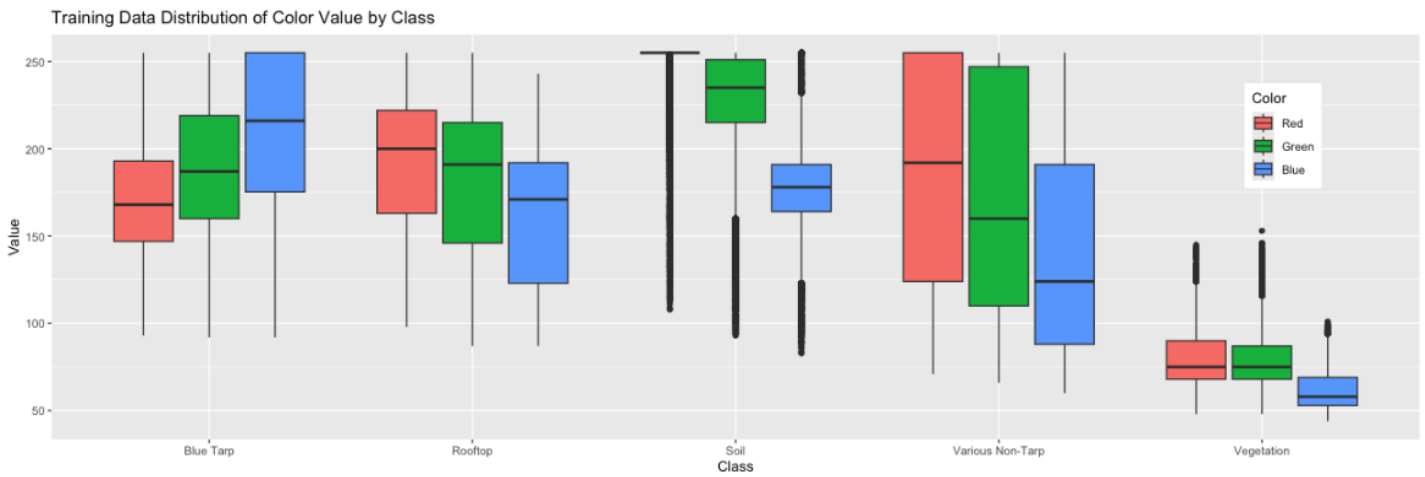
We now describe our approach from starting to check out the data's properties, to interesting visualizations and proportions we found in the data, to eventually adding model specifications and considering the training data into the model fitting. We started by carrying out some exploratory data analysis to get an initial feeling of the structure of our training data.

From the start, we knew that the training data was made up of four columns: three describing the colors found in the pixels from each picture, and one column classifying the terrain of the pictures. Possible options for the Class column include Vegetation, Soil, Rooftop, Various Non-Tarp, and Blue Tarp. What we were most interested in was to have an idea of the distribution and proportions of the Blue Tarps in this data, because those are the places where we need to send aid resources. We start by plotting a bar chart with the proportions of the different Class categories:



In the above plot, we observe that the Class for most observations is different from Blue Tarp. Indeed, from the results of a proportions table, we see that the Blue Tarp observations comprise only 3% of the training data set.

We found that an overwhelming number of observations in this dataset are of the Vegetation and Soil types, and together they make up more than 70% of the observations. The proportion of observations categorized as Blue Tarp is exceedingly small, making up only 3% of the total data. We now want to give ourselves an idea of how the three color variables are distributed among each Class category and see whether there exists any kind of pattern between colors and Class. To do that, we created the following visualization:



From the above, we can draw a few remarks. First, the most abundant category in the Class variable - Vegetation - averages lower numbers than all other four categories. Second, the categories of Soil, Rooftop, and Various Non-Tarp all have higher numbers of Red and Green color pixels, and Blue has the smallest average for all three of them. Lastly, the Blue variable has the highest average and concentration for the Blue Tarp category, strongly implying that there may exist a strong correlation between an observation and a high number of the Blue variable.

As mentioned in the Data Section above, our training data consisted of four variables: the Class column and the three colors considered in the EDA section. However, since the goal of our analysis is to identify and predict where an observation could potentially contain a Blue Tarp, we mutated the Class column to point out whether the Class was a Blue Tarp or not, with this now binary column converted into factor before we continue carrying out our model training. All three remaining variables will remain numeric, and they will serve as the three predictors for our three models.

The holdout set is a single data frame consisting of the seven merged files described in the Data section above. It contains over 2 million observations, and within it we find nine columns each describing distinct characteristics of pictures taken above the affected areas in Haiti. We noticed that the first six columns contain an ID marker and descriptive information of each picture's location, while the last three columns contained three numbers which highly resembled the three colors from our training data set.

However, the labels in these three columns were just given as B1, B2, and B3. Thus, we set ourselves to find statistical patterns between the three color variables in the training set and these last three columns in the holdout set, to determine which color corresponded to which column.

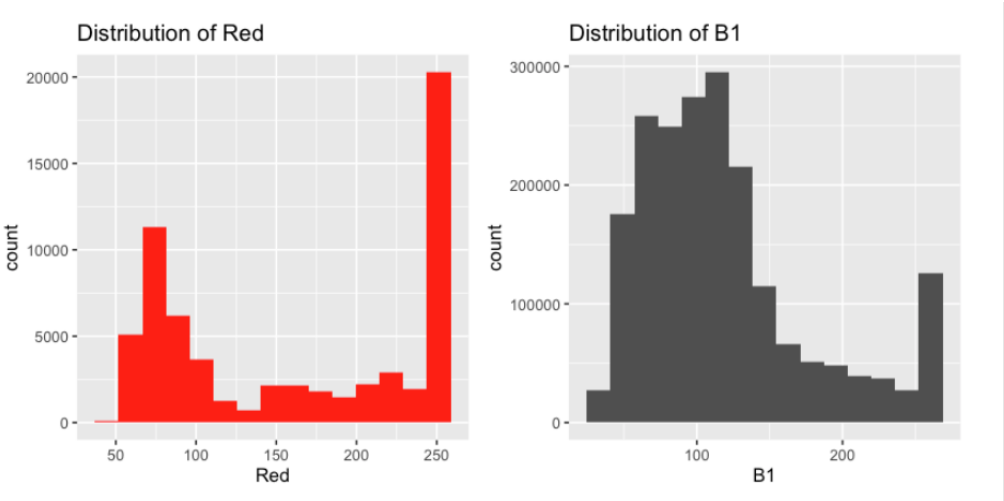
We started by figuring out how many of the training data's rows had Red as the highest number out of the three and doing the same for the Green variable. After dividing each number by the total number of rows, we found that 67.5% of the training data's observations had Red as (strictly) the highest number out of the three, with 17.2% having Green as the highest number out of the three. Blue was the highest number in only 2.5% of those observations, which is consistent with the proportion of Blue Tarp observations in the Class column.

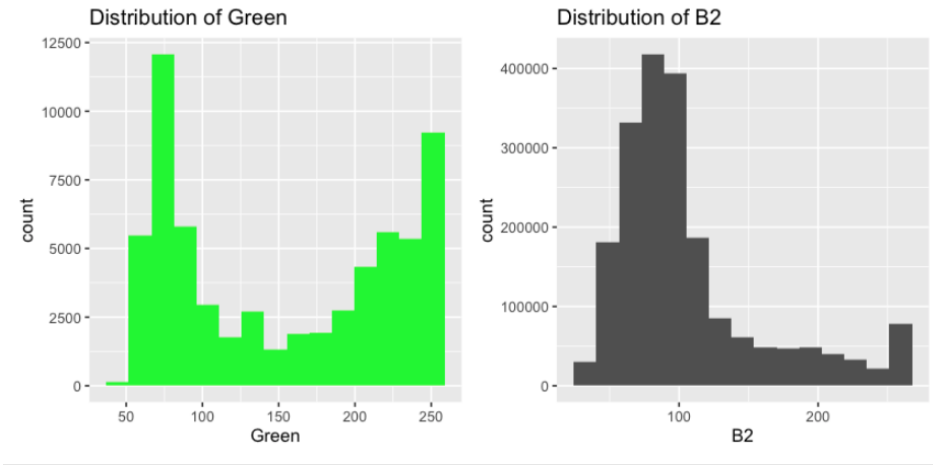
Moving into the holdout set, we ran identical code on the three columns B1, B2, and B3. Interestingly, we found that the proportion of observations where B1 was strictly the highest out of the three was 76.04%, for B2 this number was 15.3%, while for B3 it was only 1.2%. With this statistical data in hand, we are confident to mark the value of B1 as that of Red, as it was by far the largest out of the three and with a highly similar proportion as the biggest number in the training data set. We mark the value of B2 as Green, also with an almost identical proportion as in the training set and higher than the last number. This implies we mark the final value of B3 as Blue.

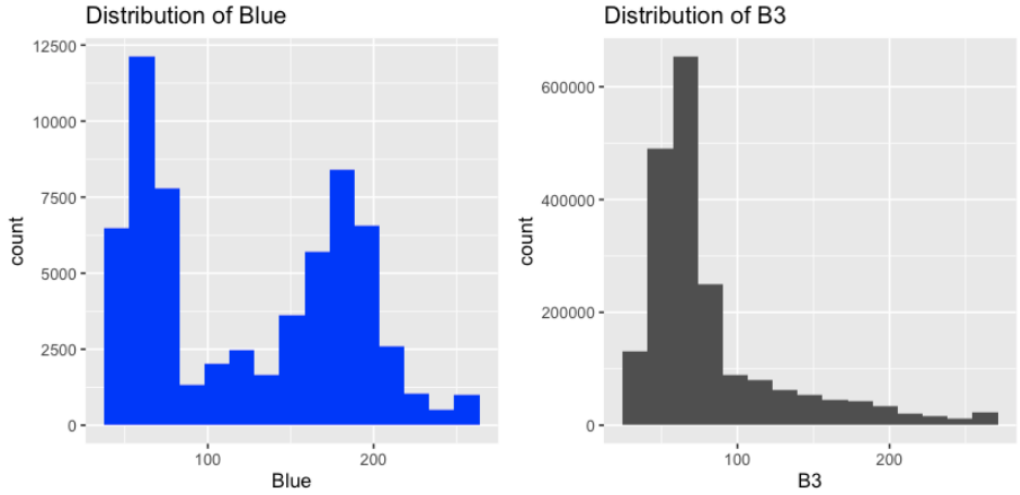
In addition to the proportions, we utilize the distribution of the pixel colors. B1 appears similar in distribution to Red, as they share a large count near 250, and also having the most density around 50-125 pixels.

B2 appears similar to Green in distribution. B2 compared to B3 has a more significant count near 250. We propose green is more likely to retain higher pixel values than blue due to its presence in nature. B2's highest density is a higher pixel value than that of B3, which aligns with Green and Blue respectively.

B3 appears similar to Blue in distribution. For Blue, most density is lower than Red or Green, and this holds true for B3 compared to B1 or B2.







The classification models for logistic regression, LDA, and QDA were all trained following the workflow process from the Tidymodels package: we prepare the regression formula, and we pass it together with the training data set to the recipe function. We define the specifications for all three models before finally defining the workflows for each, by adding the recipe defined above as well as the specifications for logistic regression, LDA, and QDA.

As we prepare for cross-validation, we also determine the number of subsets on which to perform the cross-validation and stratified sampling on the response variable, to ensure an even distribution on all the training and testing subsets. This is all done using the **resamples()** function. We also prepare the metrics we want to determine for the cross-validation, which for this analysis include the **ROC and the AUC, accuracy, kappa, and J-**

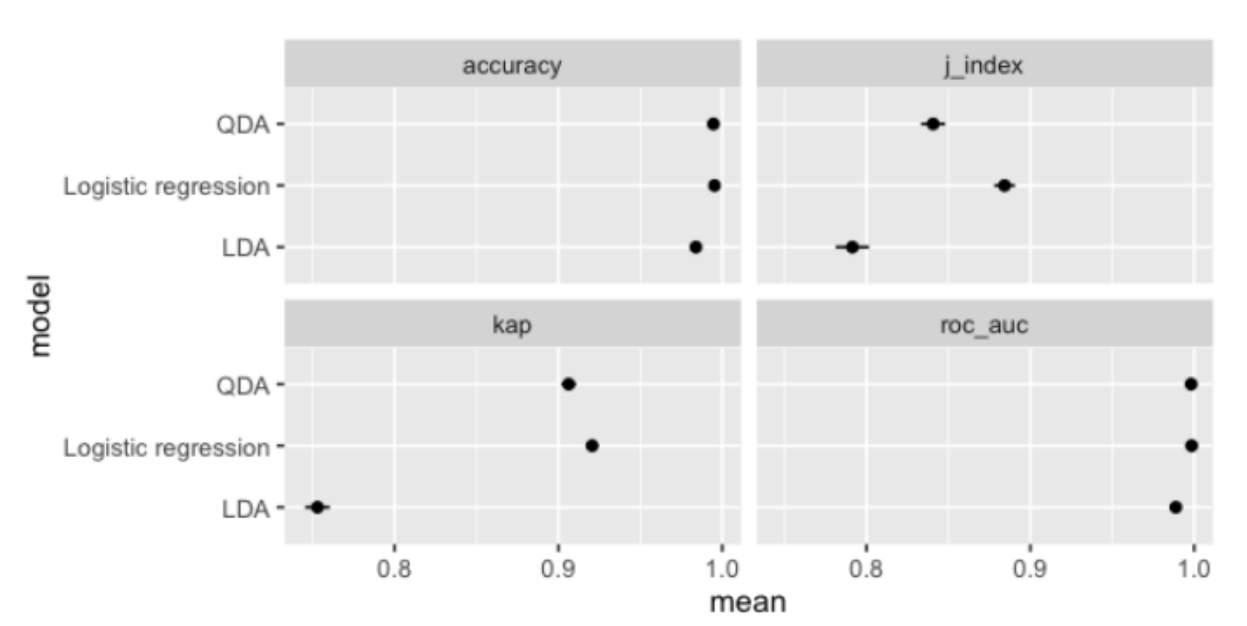
index. Lastly, we use the **control_resamples()** function to ensure we save the results we draw from the cross-validation.

We now complete the cross-validation step by initializing the **fit_resamples()** function for the logistic regression, LDA and QDA workflows. Besides the workflows, we also need to pass the above-defined resamples, metrics, and control functions to them, and now our cross-validation results are ready.

To access the results from our cross-validation, we display the metrics using the **collect_metrics()** in an elegant kable table. We also created a visualization of these results for all three models in a single plot. The results were as follows:

Cross-validation performance metrics

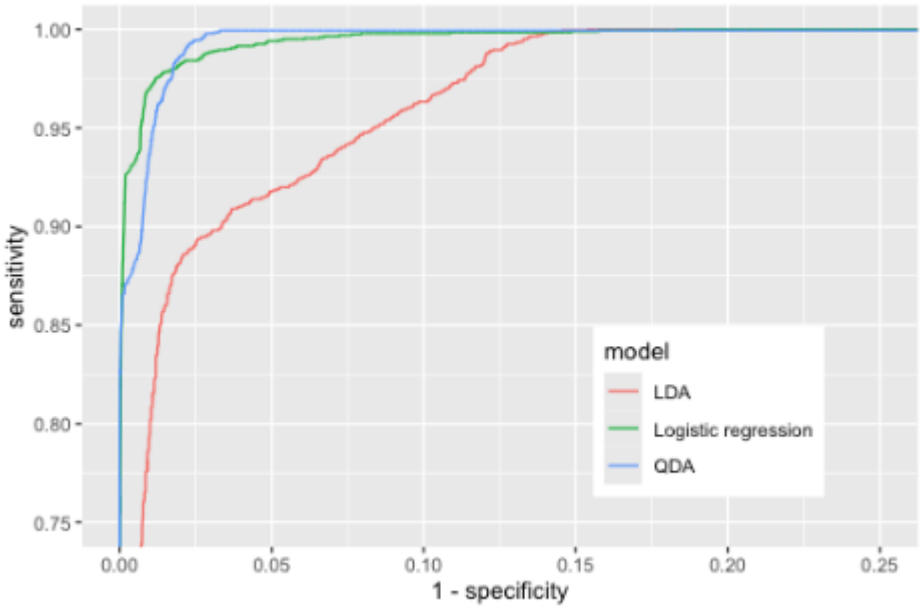
model	accuracy	j_index	kap	roc_auc
Logistic regression	0.995	0.884	0.921	0.998
LDA	0.984	0.791	0.753	0.989
QDA	0.995	0.841	0.906	0.998



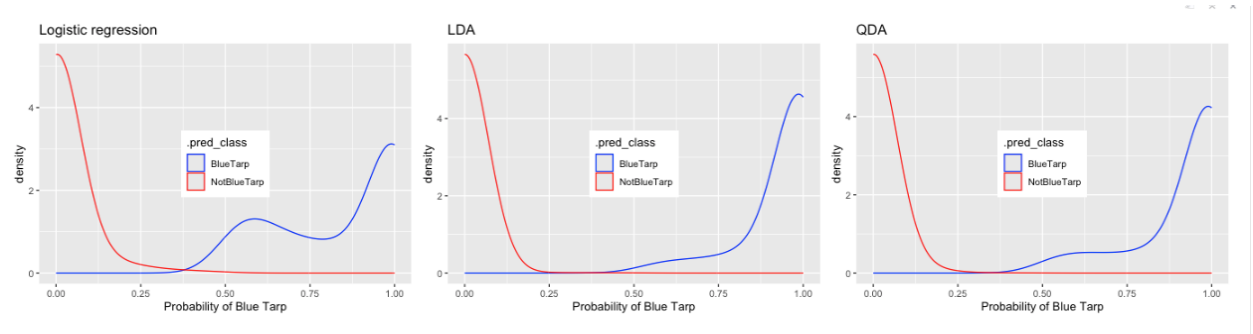
From the above, we start by pointing out that the accuracy for all three models is already very high and close to perfection. The metrics only start diverging once we consider the kappa index and the J-index, both of which show logistic regression and QDA to be superior

to LDA by a significant margin. Out of the two, we note that logistic regression performs slightly better than QDA across all metrics.

Another crucial metric from the above table and visualization is the ROC and associated AUC. All values for the three models are very close to one another, so we give ourselves a better idea of the differences between them by creating a visualization containing the above values for all three models.



We can better understand how our models are predicting data by running some preliminary tests on their performance on the holdout data. Before moving on, we employ the **augment()** function to the models we have trained so far and visualize how well they are able to predict Blue Tarps in the following plot:



From the above graph, we note that all three models have a very high density of points assigned as NotBlueTarp when their probabilities of this having classification are less than 25%. For logistic regression, we start seeing a considerable number of points assigned as Blue Tarp for those who have probabilities as low as 50%. We identify a higher density of points in the range between 50% and 75% than in LDA and QDA. For the latter two, most of the points positively identified as Blue Tarp had remarkably high probabilities, with the vast majority having over 75% to 80% chance.

The final question to answer is how we determine the threshold with which we will train our final models. We do this by considering the best J-index metric for each model and evaluating the model's performance for this assigned value. We decided to implement the J-index rather than the accuracy because we are working with an imbalanced data set, and we see that the high accuracy levels for all three models across all threshold levels may not singularly point out where the best performance is found.

We can find the highest J-index value by augmenting the training set, that is adding columns containing the values and probabilities predicted by the model for each observation, adding a sequence of thresholds from 0.01 to 0.5, and extracting the best J-index performance out of all the selected thresholds.

Results

We determine the threshold for each model by **augment()** function and applying thresholds from 0.01 to 0.5 with an increment of 0.001. These lower-half values were chosen to provide more fine-grained difference between the three algorithms, and to push the boundaries to find the ideal cutoff point which maximizes the models' performance. This set of values provides the best J-Index and related threshold selection information. The below table describes the results.

Thresholds

...1	.threshold	j_index
Logistic Regression	0.051	0.96373
LDA	0.010	0.86493
QDA	0.019	0.97005

As we see, the model with the best J-Index value is the one for QDA, followed closely by logistic regression. LDA remains the worst-performing model out of the three. We then proceed to determine the accuracy, specificity, J-Index and ROC-AUC for each model at the determined thresholds.

Threshold Metrics

model	dataset	accuracy	sensitivity	specificity	j_index	roc_auc
Logistic	train	0.98711	0.97626	0.98747	0.96373	0.99851
LDA	train	0.97619	0.88576	0.97917	0.86493	0.98888
QDA	train	0.97657	0.99407	0.97599	0.97005	0.99822

We can see here that it appears that QDA is second best with respect to accuracy, it still has a J-Index that is slightly more than the Logistic model and much more than LDA model.

LDA performs the worst on training data. If we compare Logistic and QDA it is not easy to determine which one is the best as some metrics are better with respect to Logistic and others with respect to QDA. Also, this also reflects the fact that the dataset is imbalanced.

We also look at the proportion of Blue Tarp identified in Holdout. This will help us identify the performance of each of the algorithms. From our EDA, we calculated the proportions of the possible number of observations that could potentially be classified as BlueTarp, and further compared those instances with the results obtained by testing our refined models in the holdout set. If we can identify how many blue tarps are identified by each, we would know how they perform and what algorithm we can recommend for such real-life scenarios.

For Logistic regression we get a 0.103 proportion of holdout points classified as Blue Tarp; for LDA we get 0.038, while for QDA we get 0.039. Logistic seems to have overpredicted or overclassified the Blue Tarps leading to this higher value. This could be one reason why Logistic regression seems to have higher proportion than QDA. Also, we noticed that accuracy, model performance, and ROC/AUC values in cross validation and threshold metrics could be impacted due to imbalance in both the training and holdout datasets.

Conclusions

We first consider how this data formulation enables us to address this problem using predictive modeling tools. From the beginning, we know that this is a classification problem. We are interested in solving this problem by predicting just for the small subset of the response variable within the enormous size of the dataset. Also, we notice from box plots above, the pixel colors can show some patterns or correlation between the landscape of Haiti and the differently colored pixels found in our data sets. These aspects make it so we can apply our three classification models.

The provided training data is also optimal for modeling in the aspect that there are a lot of observations in comparison to the number of predictors. One thing to keep in mind is the imbalance that has happened in both datasets may influence the way we create our models and the conclusions we arrive at.

During the training process, it was unclear which model was performing the best and if the data was suited to one prediction method. The choice was between Logistic Regression

and Quadratic Discriminant Analysis, as Linear Discriminant Analysis performed worse across the board.

From cross-validation it appears that the best performing model was Logistic Regression, as it had the same accuracy and ROC-AUC as QDA and better values for J index and kappa. When looking closer at the ROC from cross-validation of the models, we saw LDA was underperforming, and Logisitic Regression and QDA were similar. The ROC of Logisitic had a higher specificity than QDA, but the ROC of QDA had a higher sensitivity. Given that their AUCs were the same then making a choice purely on ROC would be a matter of which type of error are you more comfortable making.

QDA having a higher sensitivity (True Positive Rate) indicates that it is more likely to correctly classify the blue tarps when they truly are blue tarps. This means in comparison to Logisitic regression, it would be more likely to miss some blue tarps than QDA. Logistic regression had a higher specificity (True Negative Rate), which indicates that it is more likely to correctly classify the not blue tarps when they are truly not blue tarps. Equivalently, when comparing with QDA, it would be more likely to miss some not blue tarps than Logisitic regression. At this point, cross-validation points to QDA being the better model choice, as it is generally well performing and is more likely to correctly identify blue tarps than Logistic Regression.

However, adjusting our model thresholds in an effort to maximize our J index produced more mixed results. LDA was still the worst performing. Logisitic and QDA were more difficult to determine which was best, as Logistic had better accuracy, specificity, and ROC AUC and QDA had better sensitivity, and J index. So, similarly, we could base the decision on the type of error which we are more comfortable accepting, and we also know that J index considers the effect of both. These two ideas are what help support our choice of QDA. While our confidence in our decision wavered from time to time, we feel confident in our final conclusion that QDA is the best model for this predictive problem.

After careful consideration in training our models and determining which one works the best, we are highly confident that the QDA model we chose will help effectively save human lives. We know that our QDA model has a high degree of correctly predicting those places where help is needed. After all, the threshold chosen for this model is the one that will maximize the total number of positives, both true positives and false positives. Doing so will ensure that the number of people who may unescapably be left unattended is as low as possible.

Every day that passes matters, and every location is just as important as all the others. This is why we argue our model will benefit the highest number of people and save the lives of those in these times of distress. This is one step into providing help to survivors, given that there are other logistics goals and questions to solve, such as the path of helicopters to follow and the most optimal way of distributing resources to the people in need.

After considering all our results, we finally conclude that the best-performing model out of the three is QDA. The performance of the cross-validation metrics for QDA indicated that this model has high accuracy, J-index and ROC-AUC values on the different selected thresholds and the full training set. We can observe how QDA's performance on its most ideal J-index metric shows the strengths of this model in the training phase of our analysis.

Another advantage of choosing QDA is that, regardless of the nature of Bayes' decision boundary, as the number of observations in the training set increases, the potential variance and the overfitting of the model's predictions is expected to decrease, as the model gains stability and with it comes an increased reliability in its predictive performance given that we have a large testing set.

Coming into the predictions, the conclusions made by the QDA model highlight the low proportion of observations where blue tarps could be found overall. From the exploratory data analysis alone, we are expecting less than 3.2% of the total landscapes to be classified as having the presence of blue tarps.

On the holdout set, less than 2.7% of all observations satisfy the expected requirement to be classified this way. In practice, QDA predicted that 3.9% of the total observations would have the presence of blue tarps. This is a more sensible prediction than the one made by logistic regression, which claims that over 10% of all the 2 million observations in the holdout set will have the presence of blue tarps.