DS 6030
Statistical Learning
Disaster Relief Project Group 11: Part 2

Angelo Orciuoli
Christina Land
Scott Tumperi
Tyler Hobbs

Pre-Processing

The most important part of pre-processing this data was separating the classes and making a binary outcome from the classes as well. In regard to the training dataset, the classes were originally separated into five different classes in the original dataset. These classes were Blue Tarp, Rooftop, Soil, Various Non-Tarp, and Vegetation. The dataset came from the aftermath of the destruction of Haiti's earthquake in 2010, and there was a team from Rochester Institute of Technology flying an aircraft and collecting high-resolution geo-referenced imagery. Those who had been displaced from their homes had temporary blue tarps covering their homes. As such, the class for this dataset will require only focusing on the Blue Tarp class and not any of the other classes. Since the rest of the classes are not needed, the classes of Rooftop, Soil, Various Non-Tarp, and Vegetation were set to Non-Blue Tarp within the training dataset. Since this was now a binary classification of Blue Tarp and Non-Blue Tarp, these became different groups within the data, and they were factored as well. During this process, a column type was created to show that any point was Blue-Tarp or Non-Blue Tarp.

The holdout set consisted of eight different files, compared to the training set that was a single file. These holdout sets contained four files that were exclusively just the Blue-Tarp data, and there were four files of Non-Blue Tarp data. The first step in pre-processing this data was to load all of the files and concatenate them into a dataframe that included all of the holdout data. As K-nearest neighbors (KNN) was not used, the location data was removed, and a subset of this data was created with only the labels for the colors. The prior file for the holdout data had column labels of B1, B2, and B3, and after the exploratory data analysis, it was found that these labels should be Red, Green, and Blue, respectively. These column labels were changed on the dataframe to correctly support the data. The Blue-Tarp data was combined together, as was the Non-Blue-Tarp data. From this, the type and group for these newly created dataframes were found and added to each respective dataframe. Once these were created for all the Blue-Tarp holdout files and the Non-Blue-Tarp holdout files, these were all concatenated to a larger single dataframe.

Models:

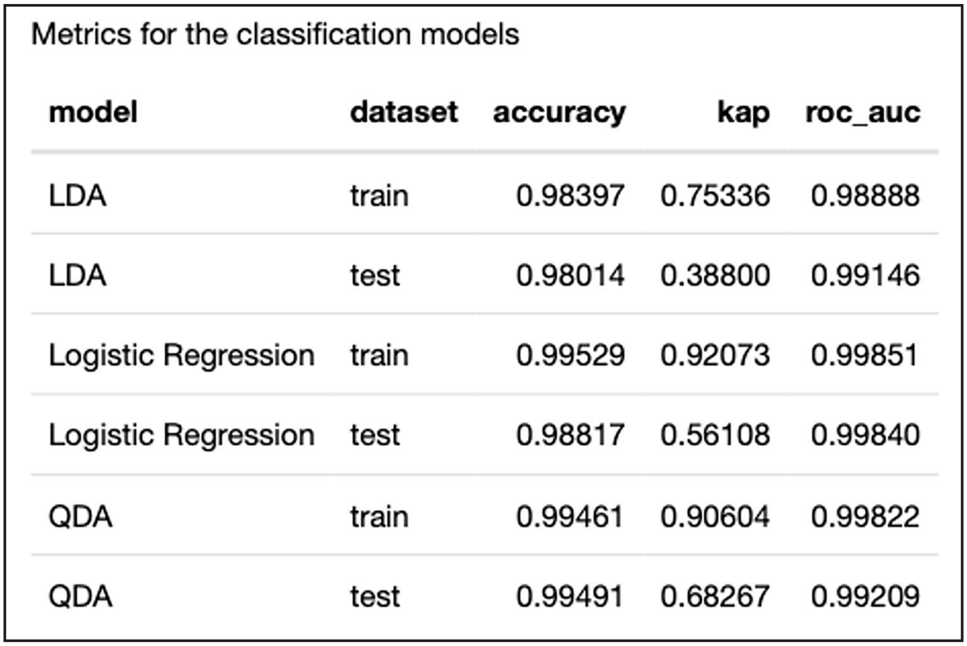
The models to be investigated were logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), ensemble method (Random Forest), ensemble method (XGBoost), and penalized logistic regression (elastic net penalty). From the initial project plan, an additional model of the ensemble method (XGBoost) was added to the model list to help better describe the data.

Logistic Regression, Linear Discriminant Analysis, and Quadratic Discriminant Analysis Models:

Training the models:

All three of these models were trained and evaluated together, as these did not have any tunable metrics, so these served as the baseline for comparison for analysis in this study. These models all followed a classification model. The engines used for these models were the glm engine for the logistic regression and the MASS engine for the linear discriminant analysis and the quadratic discriminant analysis. They were all fit to the training set. After the models were made, they followed a set of 10-fold cross-validation, and metrics are presented below.

Model Metrics from 10-fold cross-validation:



Metrics for the classification models

model	dataset	accuracy	kap	roc_auc
LDA	train	0.98397	0.75336	0.98888
LDA	test	0.98014	0.38800	0.99146
Logistic Regression	train	0.99529	0.92073	0.99851
Logistic Regression	test	0.98817	0.56108	0.99840
QDA	train	0.99461	0.90604	0.99822
QDA	test	0.99491	0.68267	0.99209

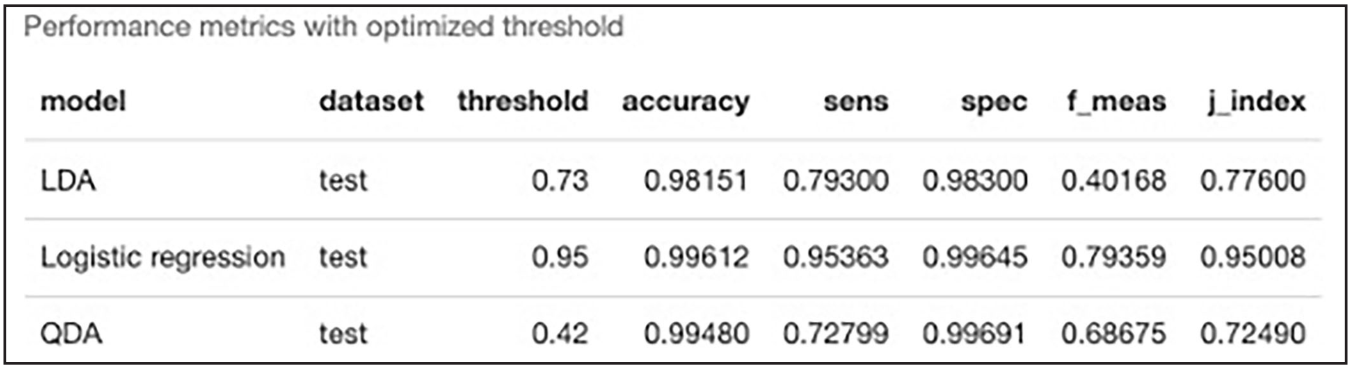
After analyzing the metrics for all three of these models on the training and the test set, the notion was that these models are nearly identical in generalizing the data. But, upon closer evaluation, it can be said that the LDA model had a marginal decrease in both the accuracy and the AUC from the ROC curve, thus it is the worst-performing model for this dataset. In addition, the logistic regression had a minimally higher accuracy metric and AUC from the ROC curve; therefore, it is the best-performing model at this point in predicting the blue tarps.

Tuning the F-measure:

Even though the models for each of the three methods were quite good, the LDA models performed significantly worse than the others in the training set. This shows that the threshold of true positives for Blue-Tarps to Non-Blue-Tarps may be incorrect. To fix this, a function was set for each model to correctly find the appropriate threshold to find the accurate number of positive cases and also to appropriately identify the false positives as well. This was done by correctly arranging a function to find the threshold at which the best F-measure for each model will be.

After running this analysis, it was found that the best thresholds were 0.95, 0.73, and 0.42 for logistic regression, LDA, and QDA, respectively. The F-measure plots are generated and placed in Appendix 1. The confusion matrices were also found for all these models for comparison, as well as the percentage of false negatives, and these were placed in Appendix 2. The metrics of all three models were then evaluated again below:

Model Metrics:



Performance metrics with optimized threshold

model	dataset	threshold	accuracy	sens	spec	f_meas	j_index
LDA	test	0.73	0.98151	0.79300	0.98300	0.40168	0.77600
Logistic regression	test	0.95	0.99612	0.95363	0.99645	0.79359	0.95008
QDA	test	0.42	0.99480	0.72799	0.99691	0.68675	0.72490

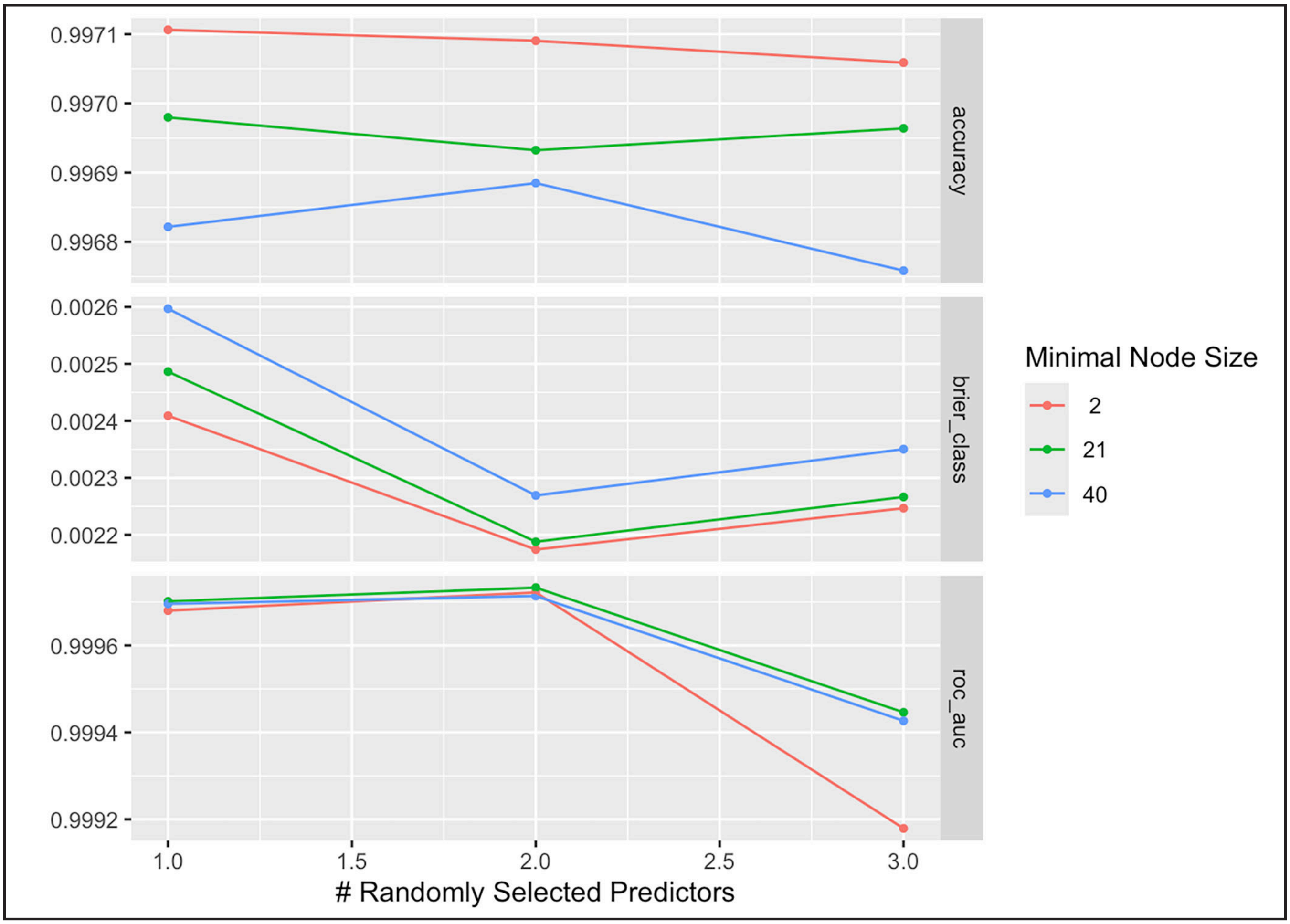
Upon evaluating the optimized thresholds for the training and test data with the previous results, it seems that these models are not the best fit for the data. This is shown when analyzing both the F-measure and the J-index variables for the three models that were generated. The F-measure is a metric that helps in classifying the true positives and also helps with the misrepresentation of true negatives as well. Then, the J-index, also known as Youden's Index, is a performance measure that helps in showing the difference in binary classification, such as in this case. This formula for Youden's Index is Sensitivity + Specificity - 1. In this case, there is much room for improvement, as a "perfect" F-measure is 1, a "good" F-measure is ≈ 0.7, and models that need to be refined are ≈ 0.5. The logistic regression and the QDA F-measures are the same as the previous metrics in that they can generalize well to the data, as their F-measures are ≈ 0.79 and ≈ 0.68, respectively. But, with the LDA model, the F-measure is significantly lower, at ≈ 0.40, showing that this model needs improvement. In addition, it shows that the LDA model is not able to classify the true positives of the Blue-Tarps accurately as well. In terms of the Youden's Index, it is shown once again that the logistic regression is the best at determining the difference between the Blue-Tarps and the Non-Blue-Tarps with the dataset. The Youden's Index for the LDA and QDA models is "good," but their ability to identify true positives is not there. These two models would need to be improved in order to be used for a model prediction, but the logistic regression would be a great candidate for prediction.

Random Forest Model:

Training and Tuning the Model:

This was the first tunable model to be tested on the dataset; it was performed using the Ranger engine and importance to impurity. This was also a classification model, and the number of trees was set to 500, and the metrics to tune were min_n and mtry. When setting the parameters, mtry, which is the number of variables that can be randomly sampled at each tree split, was given a range of 1 to 3. The min_n, which is the number of data points in a node, was given a range of 2 to 40.

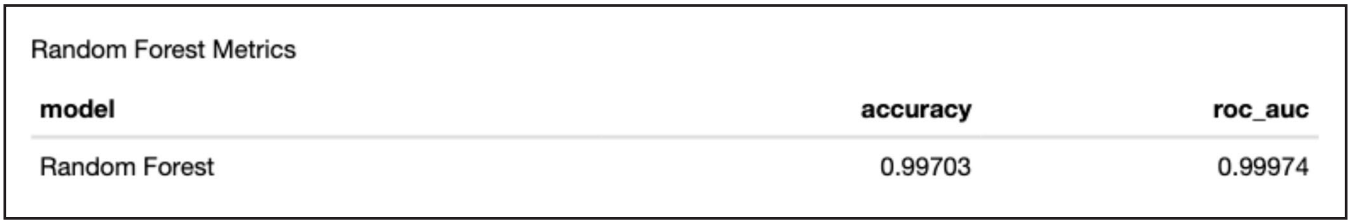
Tune Results:



Following tuning, it was determined that the best parameters for Random Forest were mtry = 2 and min_n = 21. These best parameters were put in a trained, finalized workflow and put through a set of 10-fold cross-validation. The metrics succeeding the 10-fold cross-validation are below.

Model Metrics:

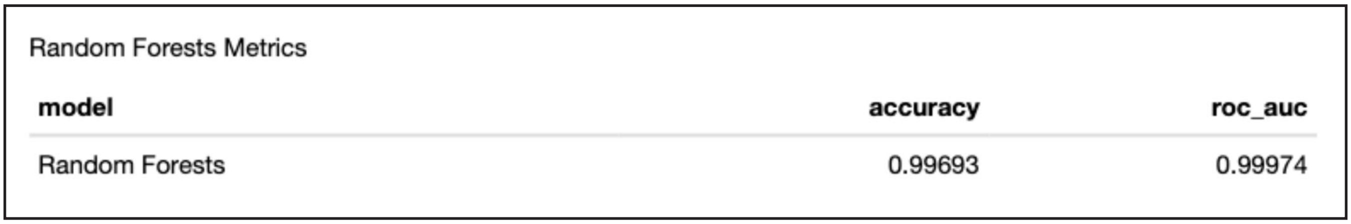
Training Set:



Random Forest Metrics

model	accuracy	roc_auc
Random Forest	0.99703	0.99974

Holdout Set:



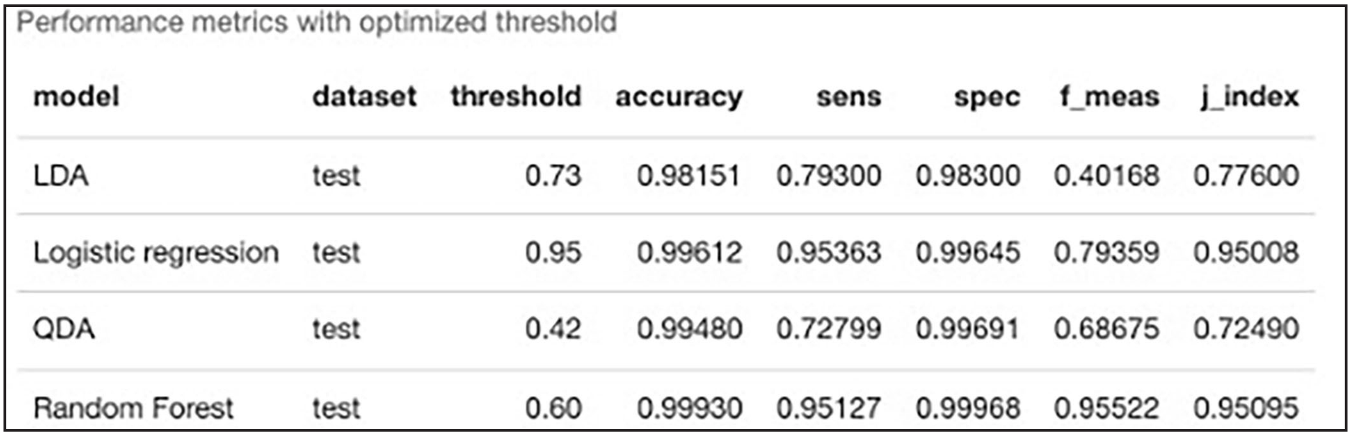
Random Forests Metrics

model	accuracy	roc_auc
Random Forests	0.99693	0.99974

After analyzing the metrics between the training and the holdout set of the Random Forest model, it can be determined that there is a marginal amount of overfitting occurring within this model. This is due to the accuracy being slightly larger in the training set than in the holdout set, resulting in the data being fit too well to the training set. As a result, this would be good for generalizing the data, but since it is overfitting the training set, a better model should be found to find the best proportion of true positives of blue tarps.

Tuning the F-measure:

The Random Forest model was also tuned with the F-measure, and it was found that the best threshold for this case study was 0.6. The plot for the tuning of the F-measure is placed in Appendix 1. In Appendix 2, the confusion matrix for the Random Forest model was generated, as well as the percentage of false negatives of blue tarps. Below is the comparison of the updated model metrics comparing the optimal threshold of all the models that have been used thus far.



Performance metrics with optimized threshold

model	dataset	threshold	accuracy	sens	spec	f_meas	j_index
LDA	test	0.73	0.98151	0.79300	0.98300	0.40168	0.77600
Logistic regression	test	0.95	0.99612	0.95363	0.99645	0.79359	0.95008
QDA	test	0.42	0.99480	0.72799	0.99691	0.68675	0.72490
Random Forest	test	0.60	0.99930	0.95127	0.99968	0.95522	0.95095

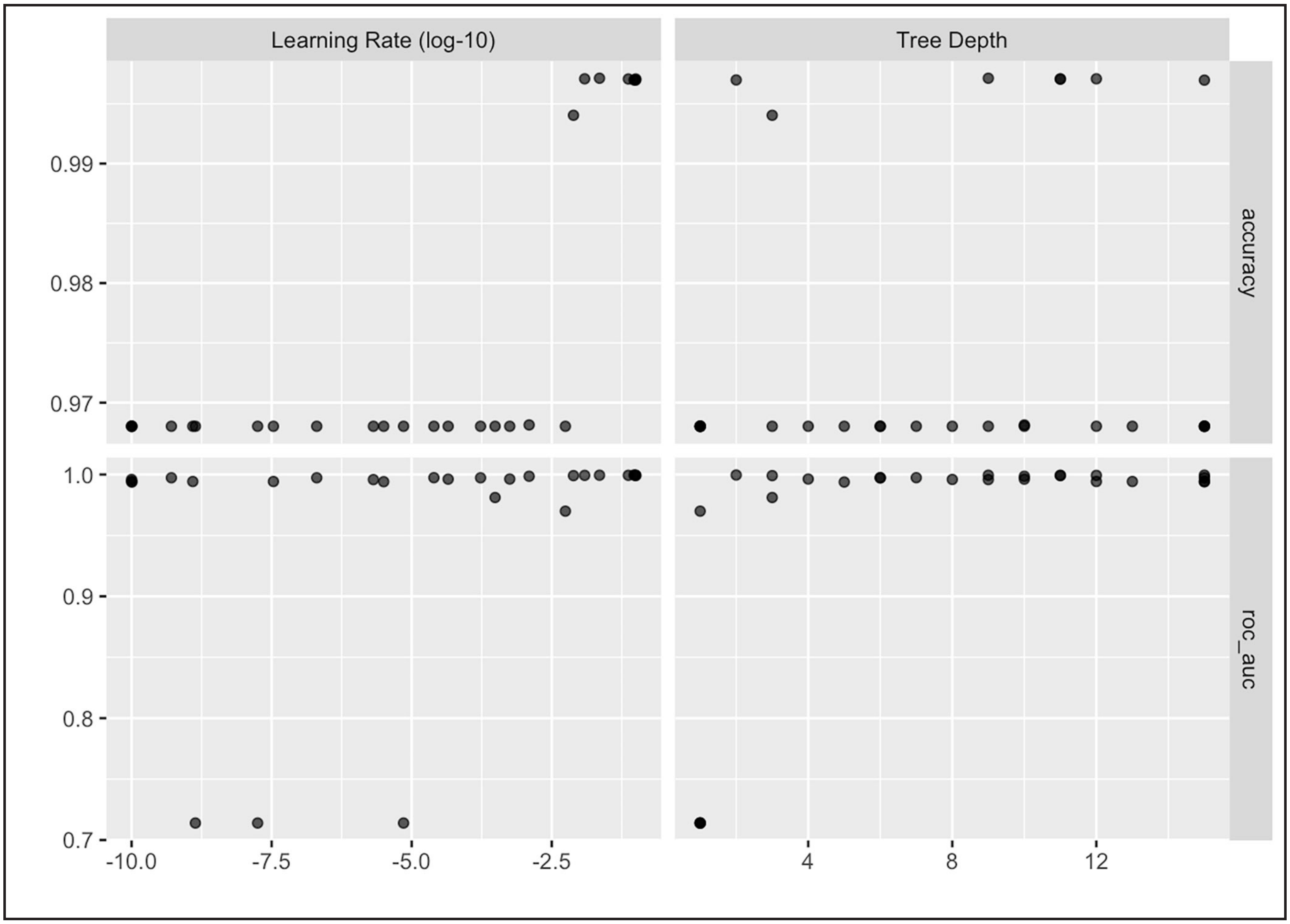
From this comparison of the optimal thresholds, the Random Forest model is a near "perfect" model is which it classifies the true positives at ≈95% accuracy from the sensitivity and the True negatives at ≈99% accuracy, leading it to be the best model thus far. This is solidified by the values in the F-measure and the Youden index as well.

XGBoost Model:

Training and Tuning the Model:

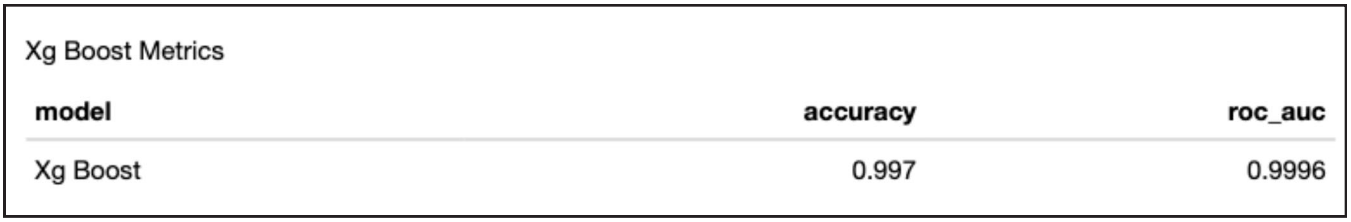
This is the second tunable model that was chosen for this data, and this was once again a classification model. It used the lightgbm engine, and within this engine, the number of trees chosen was 500, and the tunable metrics that were chosen were tree_depth and learn_rate. Below is the plot of the tune results.

Tune Results:



After tuning for XGBoost on the data, the optimal tree_depth was found to be 3, and the best learn_rate was 0.0977998. These best parameters were put in a trained, finalized workflow and put through a set of 10-fold cross-validation. The tuned model was then evaluated to see its predictive abilities.

Model Metrics:



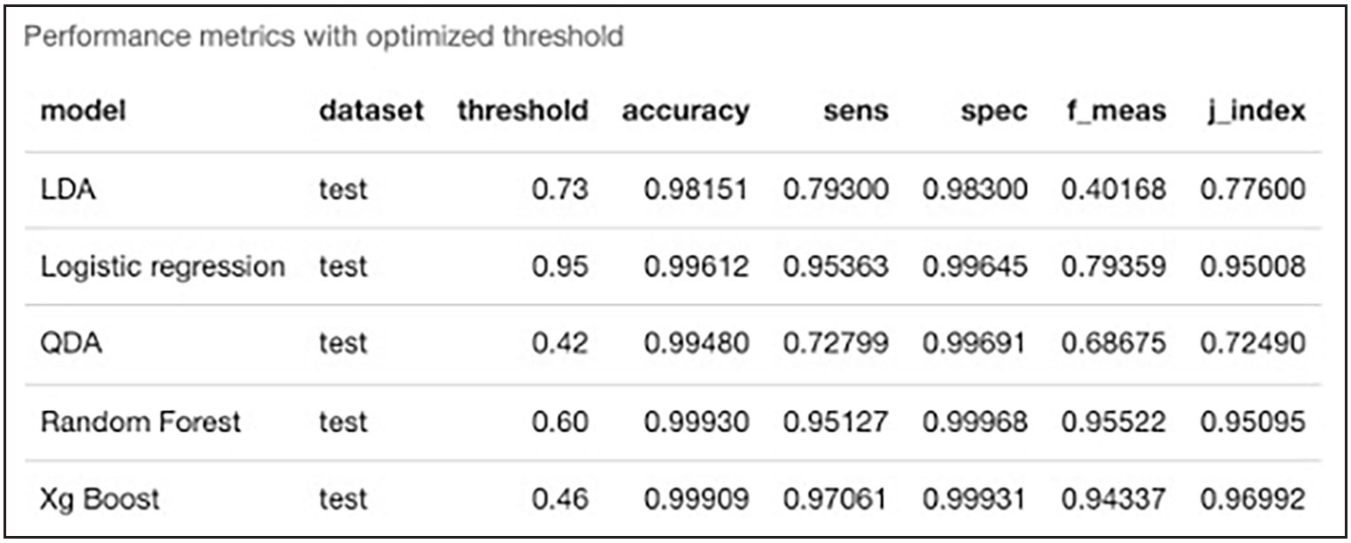
Xg Boost Metrics

model	accuracy	roc_auc
Xg Boost	0.997	0.9996

After evaluating the 10-fold cross-validation metrics from the tuned XGBoost model, it can be seen that this is the best model that has been made for generalizing the data thus far. It is similar to the Random Forest model with the holdout metrics, but the Random Forest model overfit in the training set. The XGBoost model has a marginally smaller AUC from the ROC curve, but the accuracy is higher than the Random Forest tuned model, showing that it can generalize and predict the amount of true positive blue tarps better in the dataset.

Tuning the F-measure:

The XGBoost model was also tuned with the F-measure, and the optimal value was found to be at 0.46. The plot for the XGBoost F-measure is in Appendix 1. The confusion matrix to see the difference in True positives and True negatives was also made, and this is in Appendix 2, as well as the percentage of Blue Tarps. Below is the comparison of the updated model metrics comparing the optimal threshold of all the models that have been used thus far.



Performance metrics with optimized threshold

model	dataset	threshold	accuracy	sens	spec	f_meas	j_index
LDA	test	0.73	0.98151	0.79300	0.98300	0.40168	0.77600
Logistic regression	test	0.95	0.99612	0.95363	0.99645	0.79359	0.95008
QDA	test	0.42	0.99480	0.72799	0.99691	0.68675	0.72490
Random Forest	test	0.60	0.99930	0.95127	0.99968	0.95522	0.95095
Xg Boost	test	0.46	0.99909	0.97061	0.99931	0.94337	0.96992

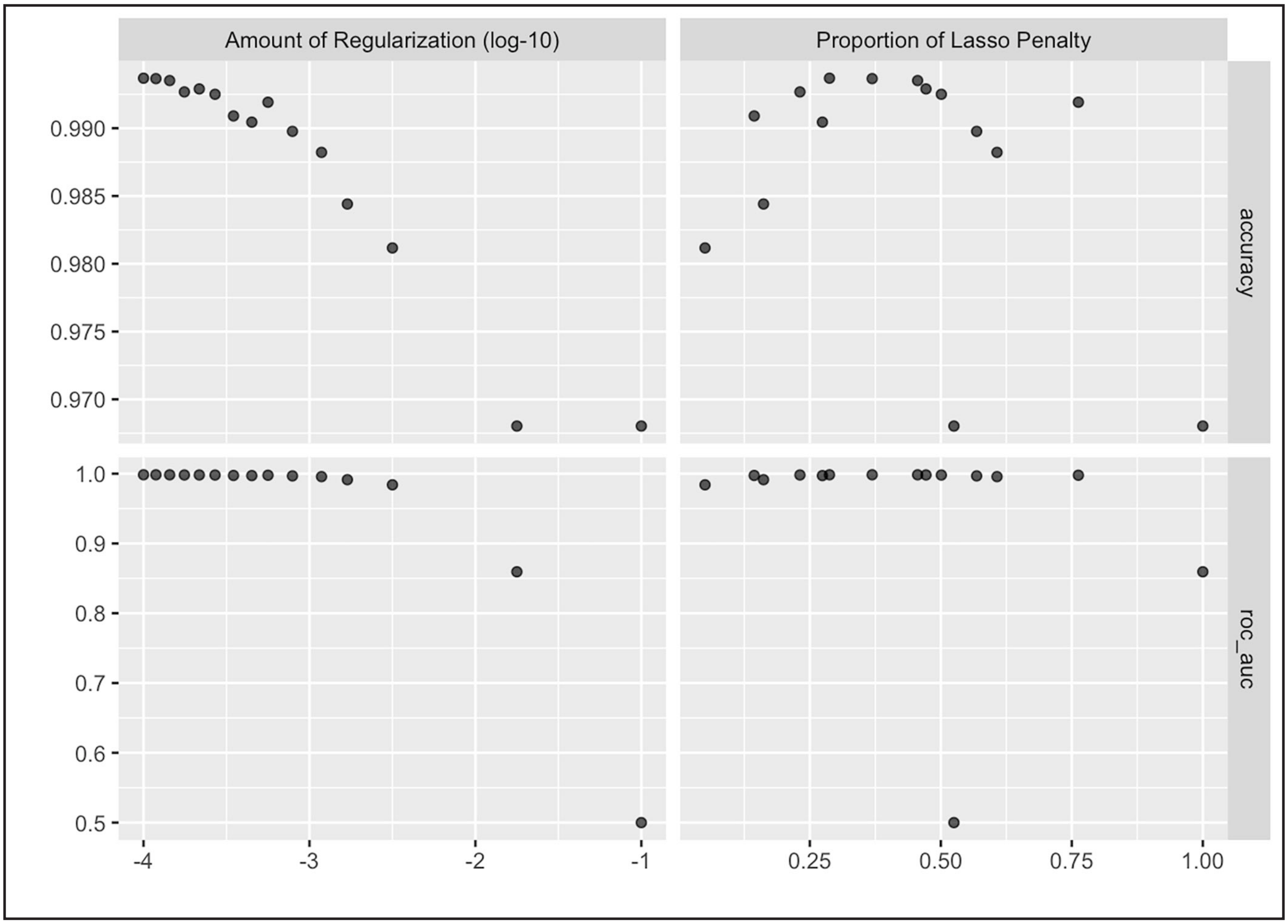
After comparing all the models at the optimal threshold, it can be determined that the XGBoost model does the best at classifying the true positives and the true negatives. This is done by how it has the highest sensitivity, and there is a marginally small drop (≈0.0003) from the Random Forest to the XGBoost but, when comparing the increase for the sensitivity and the increase for the Youden index it makes it better for generalizing the Blue-Tarps.

Penalized Logistic Regression:

Training and Tuning the Model:

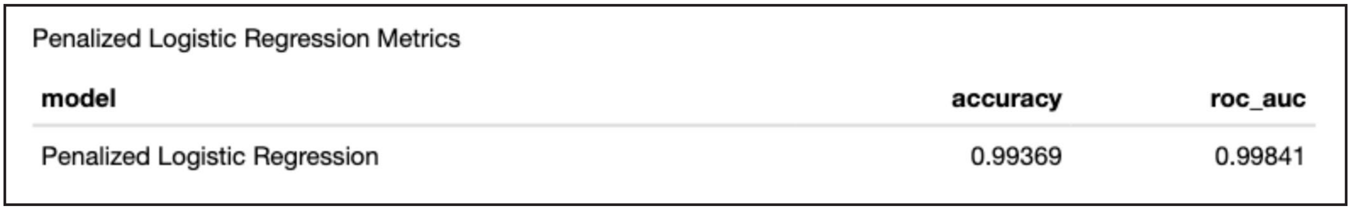
This was the final model that was made in this case study. It was also a classification model like the other models, and within the penalized logistic regression model, it used the glmnet engine, and the tunable parameters that were used were penalty and mixture. The penalty term determined how much shrinkage the model had toward zero. In this case the penalty was set to a range of –4 to –1. Then, with the mixture, a value of 0 was determined to be a pure lasso model, a value of 1 was a pure ridge regression model, and a value of 0.5 was a "true" elastic net model. The goal in this case was to find a model with an elastic net model, but as the classes were imbalanced, there was likely a sway towards either a lasso or a ridge regression by tuning the mixture parameter.

Tune Results:



After the tune results for the penalized logistic regression, it can be seen that the model best suits an elastic net model that is biased towards a ridge regression. This is due to the fact that the best tunable parameters that were found were penalty = 0.0001 and mixture = 0.288. These best parameters were put in a trained, finalized workflow and put through a set of 10-fold cross-validation. The tuned model was then evaluated to see its predictive abilities.

Model Metrics:



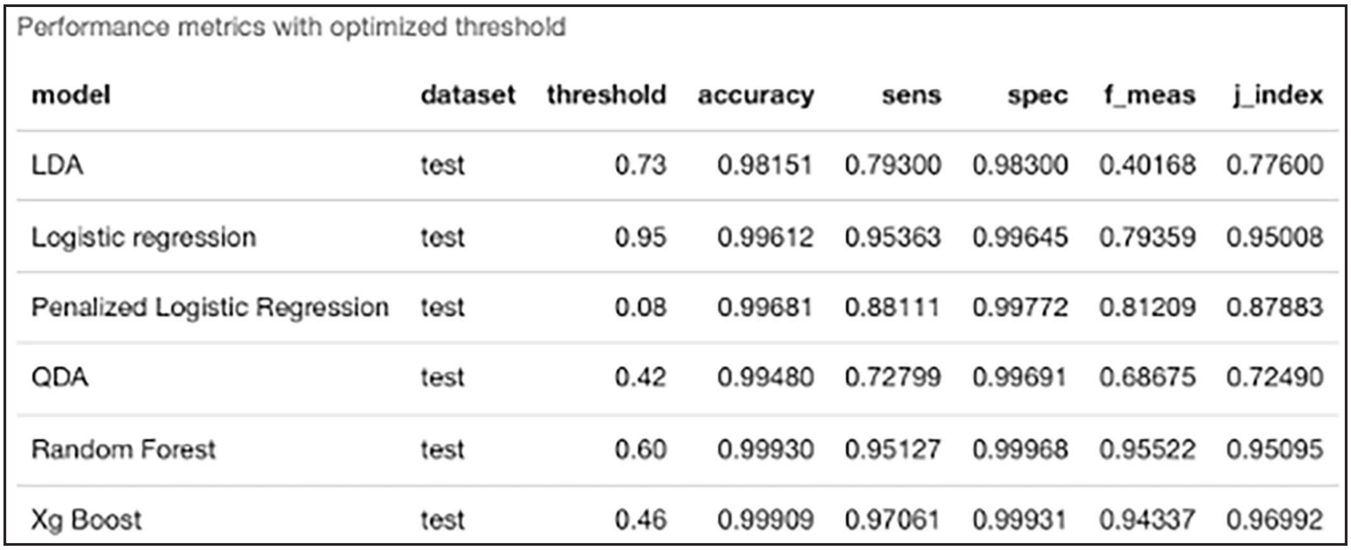
Penalized Logistic Regression Metrics

model	accuracy	roc_auc
Penalized Logistic Regression	0.99369	0.99841

This was one of the best models for the dataset, but in comparison to the other tuned models that were created within this case study, there are better models to use in predicting the number of true positives of blue tarps. In this case, the model was underfitting and not generalizing to all of the data as a result of the ridge-like regression. Therefore, the elastic net was one of the best models, but the Random Forest and XGBoost models should be used for predictions for better accuracy.

Tuning the F-measure:

The Penalized Logistic Regression was also tuned with the F-measure, and the optimal threshold was found to be at 0.08. The plot for the threshold tuning of the F-measure of Penalized Logistic Regression is in Appendix 1. The confusion matrix was also made to determine the differences in Blue-Tarps and Non-Blue-Tarps, and this is in Appendix 2, along with the percentages of false negatives of Blue-Tarps. Below is the comparison of the updated model metrics comparing the optimal threshold of all the models that have been used thus far.



Performance metrics with optimized threshold

model	dataset	threshold	accuracy	sens	spec	f_meas	j_index
LDA	test	0.73	0.98151	0.79300	0.98300	0.40168	0.77600
Logistic regression	test	0.95	0.99612	0.95363	0.99645	0.79359	0.95008
Penalized Logistic Regression	test	0.08	0.99681	0.88111	0.99772	0.81209	0.87883
QDA	test	0.42	0.99480	0.72799	0.99691	0.68675	0.72490
Random Forest	test	0.60	0.99930	0.95127	0.99968	0.95522	0.95095
Xg Boost	test	0.46	0.99909	0.97061	0.99931	0.94337	0.96992

From comparing the models at the optimal threshold, it can be determined that the Penalized Logistic Regression suffers from misclassification of the Blue-Tarps. This is shown quite well with how the specificity is at ≈99% like all the other models, but the sensitivity is only at ≈88%, leading to a decrease from the best models that were made of Random Forest and XGBoost. This is also shown with the F-measure, with a value of ≈81%, which is in between the “good” to “perfect” range, but in terms of the other models generated, there are better models that can be used in generalizing upon the Blue-Tarps.

Future Steps:

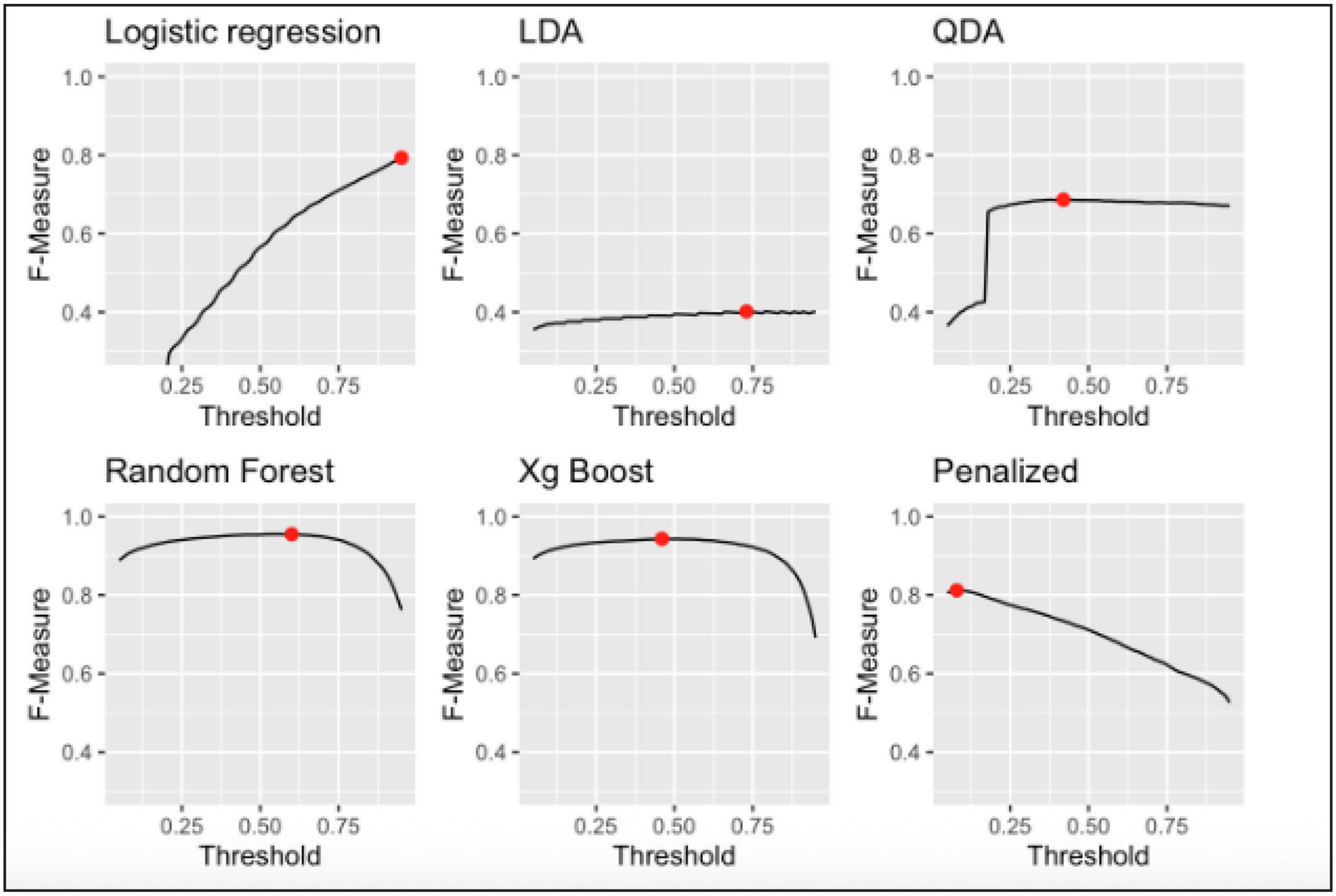
In the time ahead, there is no need to explore additional models but to explore different tunable parameters within the models that have been chosen thus far. This is especially possible with the XGBoost model, as there were several parameters that were not chosen for tuning. The tuning parameters that were not chosen were mtry, min_n, and loss_reduction. If the optimal models are not achieved, then support vector machine models can be made for the data, which would include models from the linear kernel, the polynomial kernel, and the radial basis kernel.

One of the main issues with the tuned models was the Random Forest model, as this was one of the best models because it had overfitting in the training set. To reduce this, the number of trees that are in the model could be reduced, and the overfitting should decrease as well.

Another issue that occurred was during tuning the penalized logistic regression, that the model followed a ridge-like regression by tuning the mixture parameter. To better optimize this, the mixture parameter could be set to 0.5 so it is a "true" elastic net logistic regression, and the data will then not be underfit as well.

To find the best final model for this case study, the most optimal way to find the best model is to find the model with the highest sensitivity and the highest specificity. By identifying the model with the highest in these two metrics, it will allow us to find the highest proportion of true positives of blue tarps and, additionally, the highest proportion of true negatives of blue tarps as well. In the end, this case study was trying to find the people that were displaced and who are using blue tarps, and by identifying the correct true positives and true negatives, such people could be found with such a model.

Appendix 1


Logistic regression
LDA
QDA
Random Forest
Xg Boost
Penalized
F-Measure
Threshold
1.0
0.8
0.6
0.4
0.25
0.50
0.75

Appendix 2

Logistic Regression					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp		Percentages	
Blue Tarp	14991	7069		95.36%	0.35%
Non-Blue Tarp	729	1985834		4.64%	99.65%
	15720	1992903			

LDA:					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp			
Blue Tarp	12466	33883		79.30%	1.70%
Non-Blue Tarp	3254	1959020		20.70%	98.30%
	15720	1992903			

QDA:					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp			
Blue Tarp	11444	6164		72.80%	0.31%
Non-Blue Tarp	4276	1986739		27.20%	99.69%
	15720	1992903			

Random Forest:					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp			
Blue Tarp	14954	636		95.13%	0.03%
Non-Blue Tarp	766	1992267		4.87%	99.97%
	15720	1992903			

XG Boost:					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp			
Blue Tarp	15258	1370		97.06%	0.07%
Non-Blue Tarp	462	1991533		2.94%	99.93%
	15720	1992903			

Penalized Logistic Regression:					
	Truth				
Prediction	Blue Tarp	Non-Blue Tarp			
Blue Tarp	13851	4541		88.11%	0.23%
Non-Blue Tarp	1869	1988362		11.89%	99.77%
	15720	1992903			

False Negative