Disaster Relief Project 1

Hallie Parten

2023-03-07

Introduction

This report will detail and justify the steps taken to build five different models that will predict the existence of a blue tarp from imagery data-three different colored pixel values. The goal is to choose the highest performing model that will be employed to locate blue tarps from high-resolution geo-referenced images of Haiti. The coordinates from these images will be deployed to rescue people displaced by the 2010 earthquake before they run out of food and water. Given these conditions, a "high performing" model will be both precise and efficient. This report will describe how each model was evaluated for those two criteria.

Data Wrangling and EDA

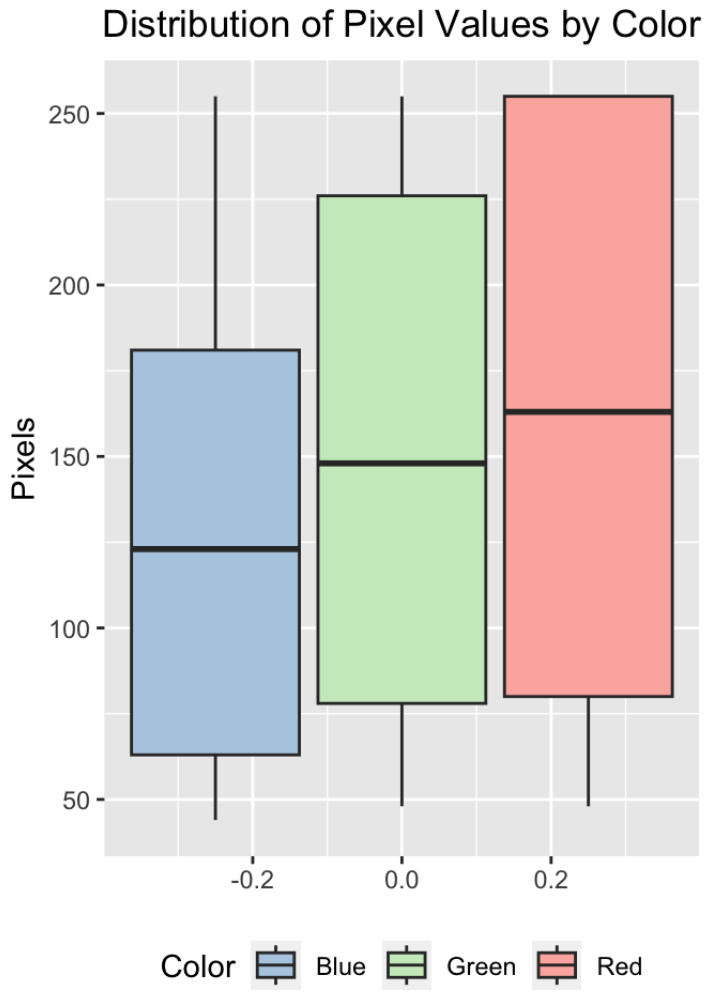
To begin, I imported the training data set of 63,241 observations and 4 variables - 3 quantitative predictor variables: `Blue` , `Red` , `Green` colored pixel values, and 1 categorical response variable: `Class` of landscape object. First, I checked the data set for any missing values and found none. Then I checked the distributions of the quantitative predictors for any extreme values (indicating data entry errors) or differences in the scales.

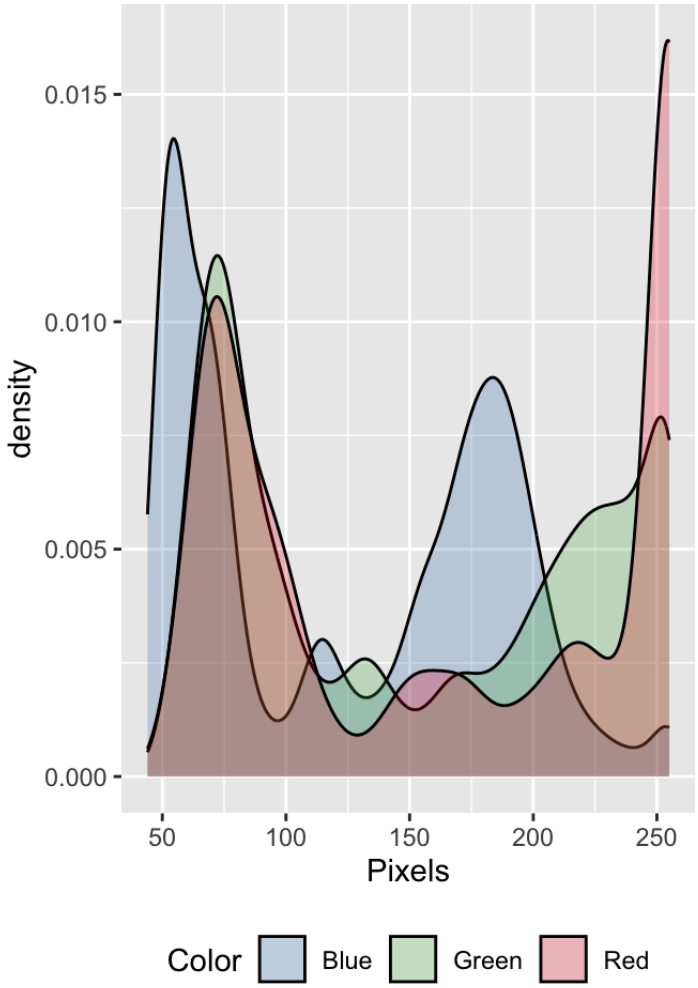
Below, we do not find any evidence of data entry errors or differences in the ranges of the predictor variables. The current combined range of all three predictor variables is from 44-255. I discuss my later decision to scale the range to a minimum interval in the model-building section of this report.

```
## [1] "NA values:"
```

```
## [1] 0
```

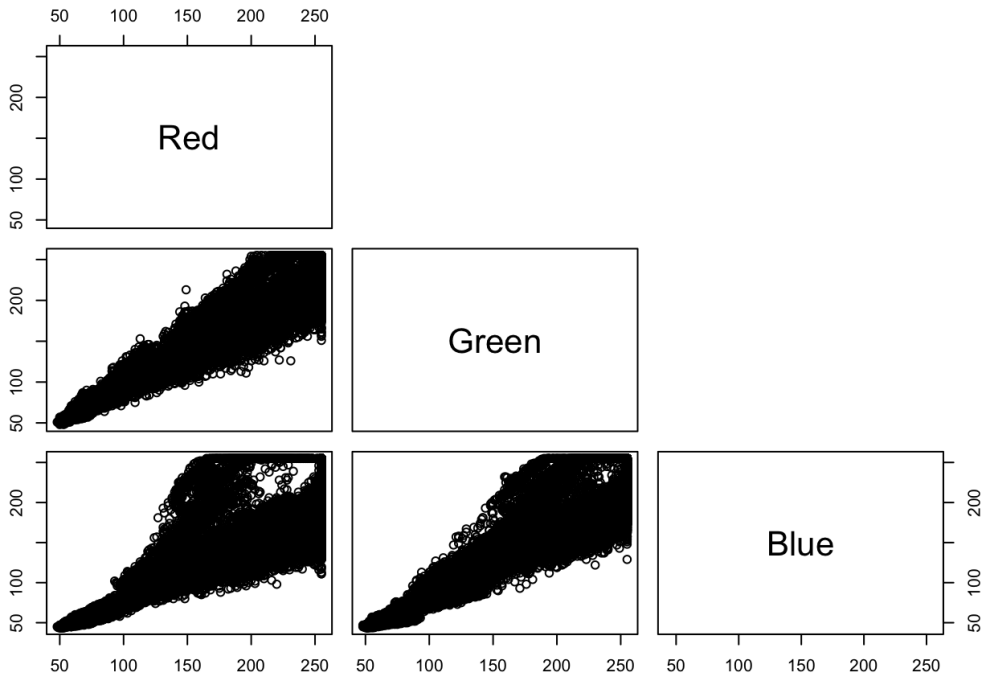
```
##     Class                Red          Green            Blue
##  Length:63241       Min.   : 48   Min.   : 48.0   Min.   : 44.0
##  Class :character   1st Qu.: 80   1st Qu.: 78.0   1st Qu.: 63.0
##  Mode  :character   Median :163   Median :148.0   Median :123.0
##                     Mean   :163   Mean   :153.7   Mean   :125.1
##                     3rd Qu.:255   3rd Qu.:226.0   3rd Qu.:181.0
##                     Max.   :255   Max.   :255.0   Max.   :255.0
```





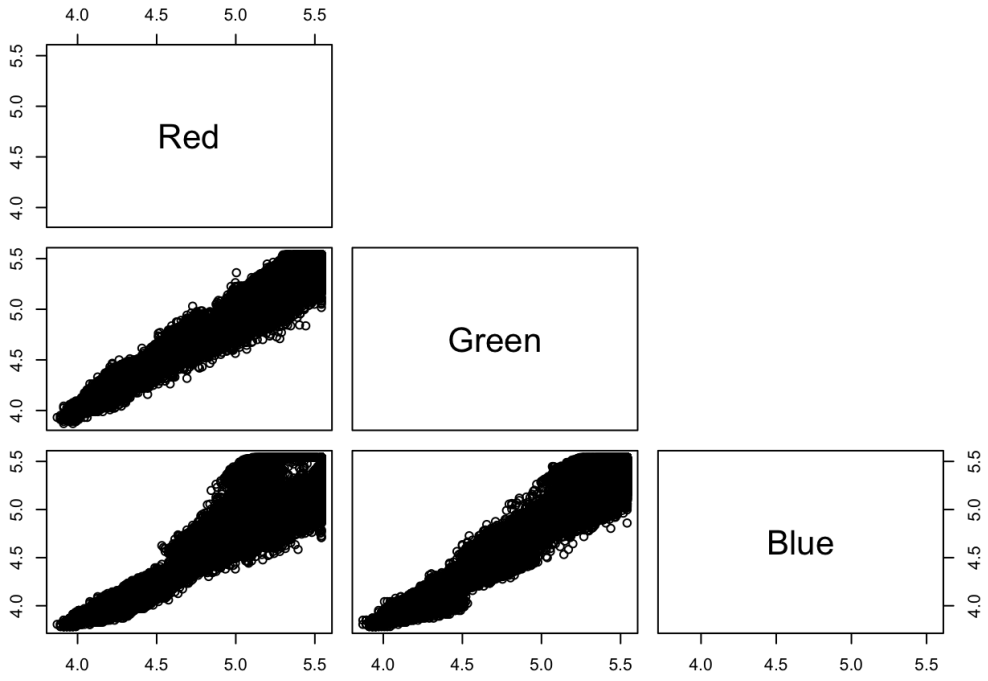
Next I looked at a pairwise scatter plot of the quantitative predictors in the data set to assess the relationships between the predictors.

Below, we see that all three predictors are strongly positvely correlated. These correlations potentially indicate shared information, similar patterns, or interaction- which this report investigates in the model building section. We also see a fanning out pattern in the pairwise plot which indicates heteroscedasticity or unequal variances in the predictors. Heteroscedasticity can lead to poor model fit, biased predictions, and therefore poor model performance which is not something that we can afford given the constraints and importance of this task.



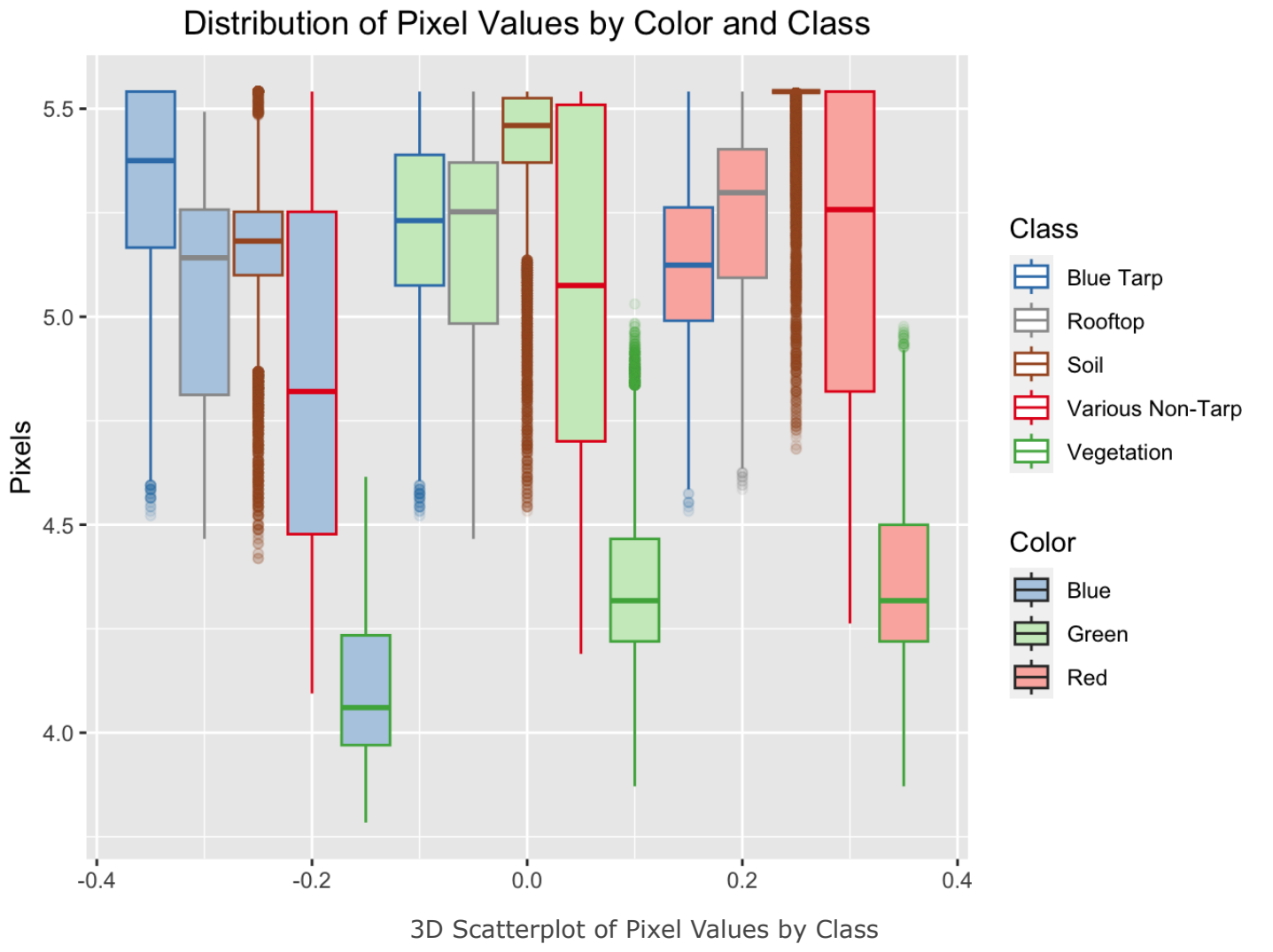
To achieve more consistent variance across the predictor variables' ranges, I applied a log transformation to each.

In the updated pairwise scatter plots below, we see that fanning out pattern has been mostly resolved and the relationships among the predictors appear more linear.



Next, I plotted the distributions of each predictor (colored pixel value) by `Class` to see if there was significant separation between classes by each predictor. I also created a 3D scatter plot to see if there was significant separation between classes by all three predictors.

Below, we see significant separation, particularly for the `Blue Tarp` and `Vegetation` class.



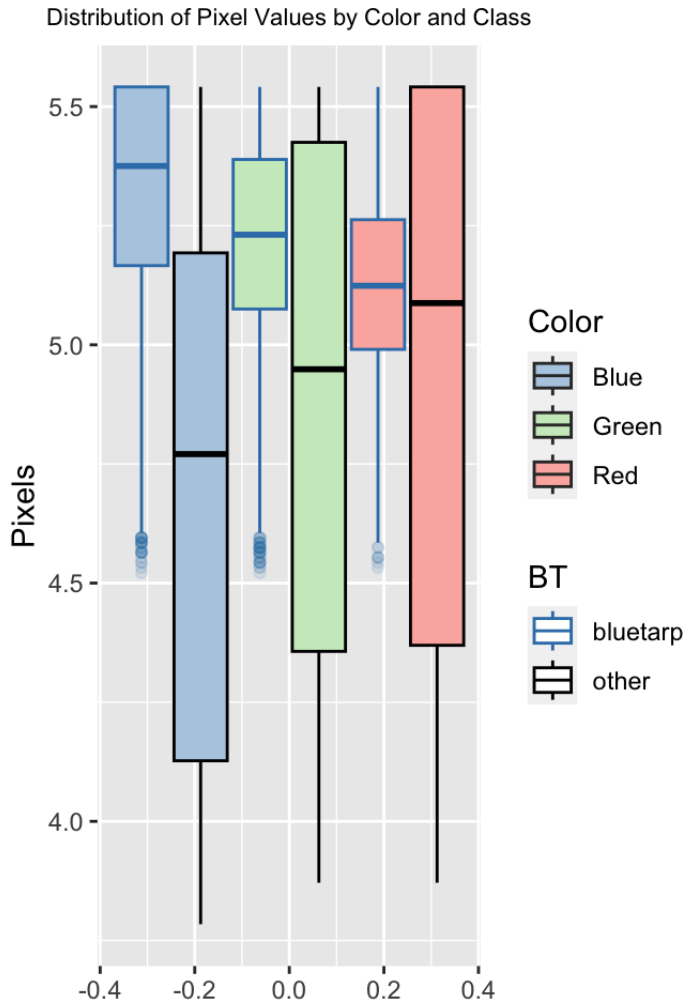
Because of the distinct separation we observe for our class of interest, `Blue Tarp`, and because the goal of this model is to predict the presence of `Blue Tarp` alone, I chose to collapse the other classes to one class called, `Other`. The training data set now contained imagery data for two classes, `bluetarp` and `other`. Below, the table shows the imbalance among the relative frequencies for the two classes within

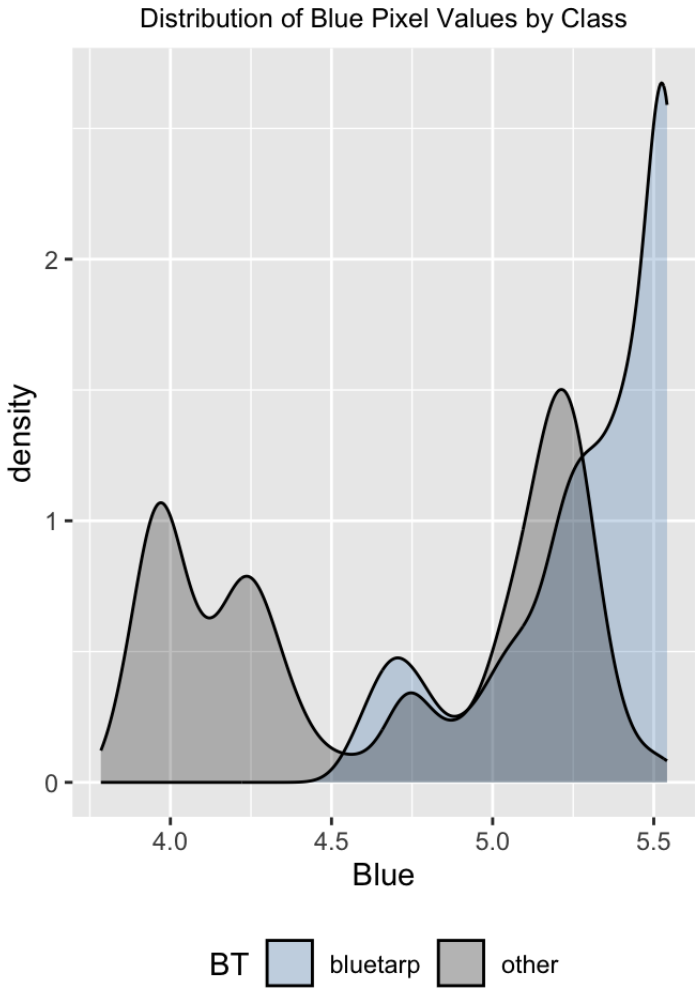
the data set. **Note: I chose to set `bluetarp` as the reference class throughout the model building process.*

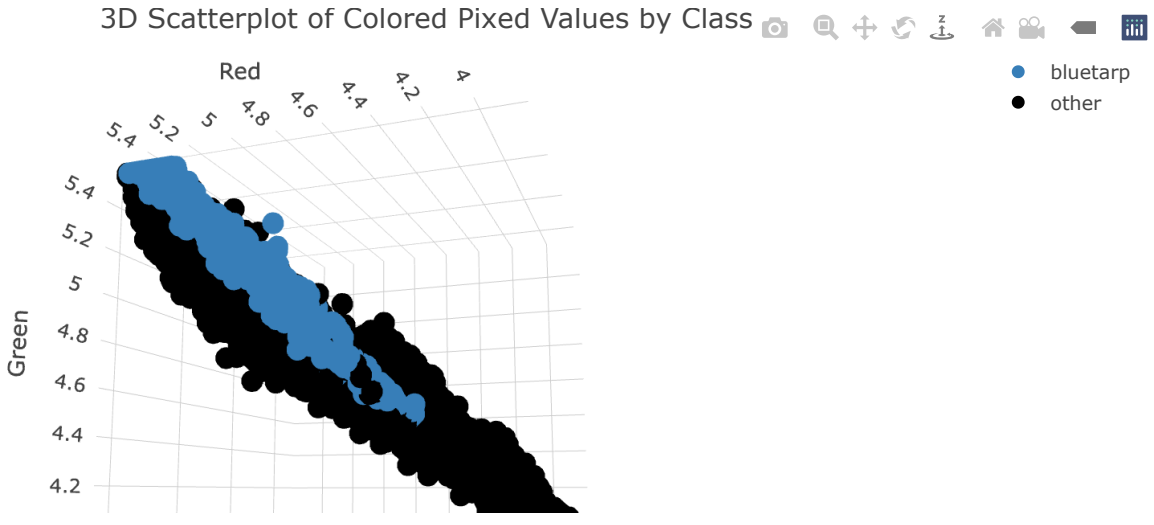
```
## [1] "Class Frequencies:"
```

```
##
## bluetarp    other
##     2022    61219
```

In the box plot below, we see that the separation looks the strongest for the `Blue` pixel predictor, which makes sense given that blue is the identifying color of the blue tarps that we are looking for. We also see the separation maintained in the density plot of `Blue` by `Class`. Lastly, we see the distinct separation of `bluetarp` is maintained in the 3D scatter plot of all three colored pixel values by `Class`. We also see that the shape of the binary class boundary appears approximately linear in the 3D scatter plot.







Model Fitting, Tuning Parameter Selection, and Evaluation

I built R functions to perform my model building with the arguments that would be necessary for all five models I planned to build. I used the `caret` package in R and set up the `trainControl` object to perform 10-fold cross-validation that optimized the model based on the area under the ROC curve. I passed three additional arguments to the `trainControl` object in my model training function, `classProbs=TRUE`, `savePredictions=TRUE`, and `summary=twoClassSummary` which allowed my model to access the out-of-fold predicted class probabilities to build an ROC curve. I also added the argument `metric='ROC'` to my `train` object which set the AUC as the metric to be used for optimizing tuning/complexity parameters.

Logistic Regression Model

The first model I built was the logistic regression model. Initially, I chose to fit an additive regression of the binary response, `BT`, on the three colored pixel values, `Red`, `Blue`, and `Green`. Secondly, I decided to create a logistic regression model that included the main effects from `Blue`, `Green`, and `Red`, and interaction terms between the `Blue` pixel values and the other colored pixel values; `Blue:Red` and `Blue:Green`. This decision was based on the high correlations between predictors we observed in the pairwise scatter plots and on my working theory that the effect of `Blue` on predicting the presence of blue tarp in an image would depend on the level of `Red` and `Green` because certain combinations would prompt the blue to be more distinguishable in the image.

Below we see the coefficients of the interactions terms are significant within a 99.99% significance level. Although the coefficient for `Red` is insignificant, I chose to leave all main effect predictors in the model due to the hierarchy principle. Additionally, the data set had sufficient sample size to justify the additional complexity and risk of over-fitting from the including the additional terms in the model.

We see below that the AUC value for the model with the interaction terms is slightly higher than the AUC for the simple, additive model. This larger AUC value implies greater separation of classes and therefore better predictive power for the interaction model. We also see that the Sensitivity, or the ability to accurately predict `bluetarp` is greater for the model with the interaction terms. Given its higher predictive potential, I chose to explore threshold selection only for the interaction model.

> For building each of my subsequent models, I decided to test these two subsets of predictors for model selection determined by the greater AUC value from cross-validation:
>
> - 3 Predictor Model/Additive Model: `Blue`, `Green`, `Red`.
> - 5 Predictor Model/Interaction Model: `Blue`, `Green`, `Red`, `Blue:Red`, `Blue:Green`.
>
> To accomodate the interaction terms, I added an the additional argument, `preProcess=c("center, "scale")` to my model-building function that would subtract the mean of the predictor's data from the predictor values and divide by the standard deviation. I chose to add this additional step to all of my models, because it would not hinder the performance estimates and it would help avoid any issues with distance calculations during the non-parametric KNN model-building.

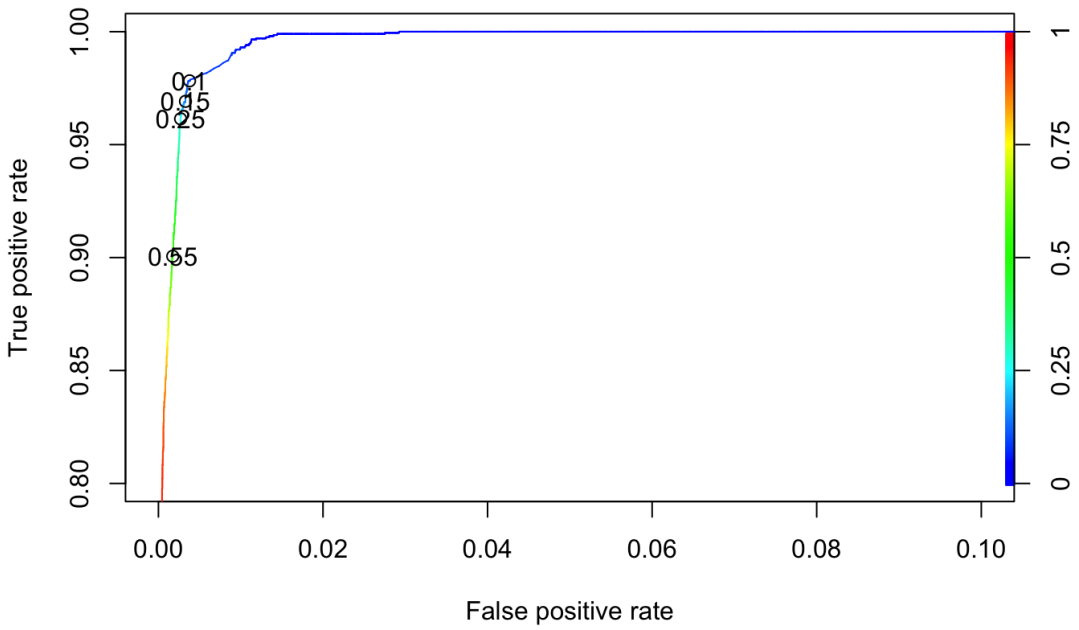
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0297   0.0004   0.0022   0.0111   3.4261
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    12.0773     0.3598  33.569  < 2e-16 ***
## Blue          -38.6521     1.8039 -21.427  < 2e-16 ***
## Red           -62.0062    11.9412  -5.193 2.07e-07 ***
## Green          69.8040    10.6078   6.580 4.69e-11 ***
## `Blue:Red`    152.3093    22.4624   6.781 1.20e-11 ***
## `Blue:Green` -122.1595    20.7010  -5.901 3.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 17901.6  on 63240  degrees of freedom
## Residual deviance:  1411.3  on 63235  degrees of freedom
## AIC: 1423.3
##
## Number of Fisher Scoring iterations: 11
```

metric	log_additive.performance	log_interact.performance
AUC	0.999179	0.999474
Sensitivity	0.904068	0.910964

Next, I reviewed the ROC curve that was created using the out-of-fold predicted probabilities and a table of other binary classification performance metrics to select the best probability threshold for my interaction model. Given the context of the prediction problem - identifying blue tarps expediently in order to provide life-saving rescue to storm-survivors in Haiti - it is important to both minimize false negatives to ensure that rescuers do not miss the opportunity to save a life and false positives to ensure that rescuers do not waste time searching where there is no one to be found. Therefore, I chose to select a threshold by maximizing the `F1` metric. `F1` is a combined measure of the model's ability to minimize false positives (high precision) and minimize false negatives (high recall/sensitivity) and is particularly useful when classes are imbalanced, as is the case here, and as such, accuracy can be misleading. The `F1` score is calculated using the following formula: $\frac{2\times(Precision\times Recall)}{Precision+Recall}$ (*note: recall and sensitivity refer to the same concept and are calculated using the same formula. This report uses sensitivity and recall interchangeably*). A value of 1 indicates that the model and threshold achieves perfect precision and recall.

The ROC curve is zoomed in to the top left corner because the model performs relatively well at most thresholds. We see in our table and graph below that `F1` is maximized in our interaction model at a threshold of 0.25. In Figure 1.1, we see that `F1` balances the counteracting effects of sensitivity and precision. In Figure 1.2, we see that a threshold of 0.25 would effectively balance the false negative rate and the false positive rate, ideally leading to an case of balanced precision of identifying all possible survivors in true blue tarps and efficiency of not wasting time/resources on false instances of blue tarps.




```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.85810	0.98945	0.99406	0.01070
0.10	0.93513	0.99565	0.97823	0.00377
0.15	0.93701	0.99583	0.96883	0.00328
0.20	0.94015	0.99606	0.96537	0.00292
0.25	0.94150	0.99617	0.96142	0.00268
0.30	0.93835	0.99600	0.95152	0.00253
0.35	0.93538	0.99584	0.94114	0.00235
0.40	0.93250	0.99568	0.93223	0.00222
0.45	0.92902	0.99549	0.92233	0.00209
0.50	0.92541	0.99530	0.91096	0.00191
0.55	0.92226	0.99515	0.90058	0.00173
0.60	0.91997	0.99503	0.89267	0.00158
0.65	0.91698	0.99489	0.88229	0.00139
0.70	0.91525	0.99481	0.87635	0.00127
0.75	0.90944	0.99451	0.86249	0.00113
0.80	0.90562	0.99432	0.85212	0.00098

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.85	0.90096	0.99410	0.83926	0.00078
0.90	0.89208	0.99366	0.81998	0.00060
0.95	0.87063	0.99258	0.78091	0.00042

Figure 1.1

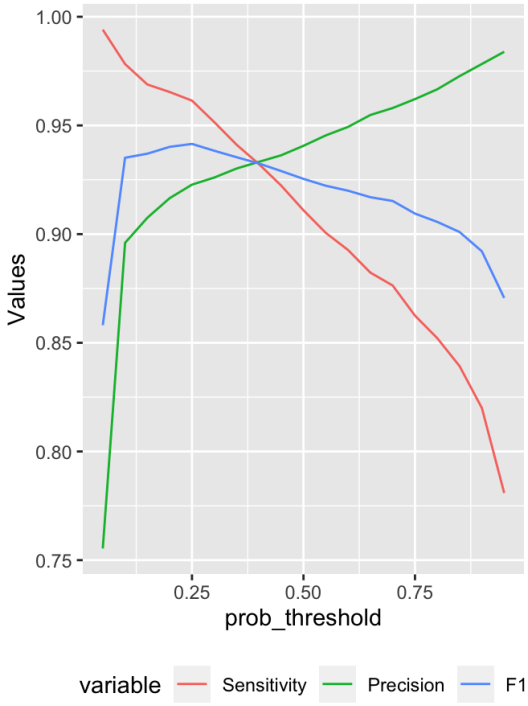
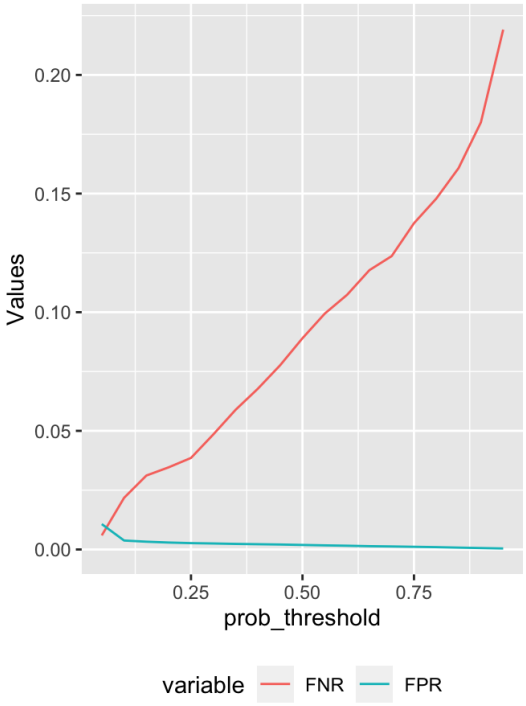


Figure 1.2



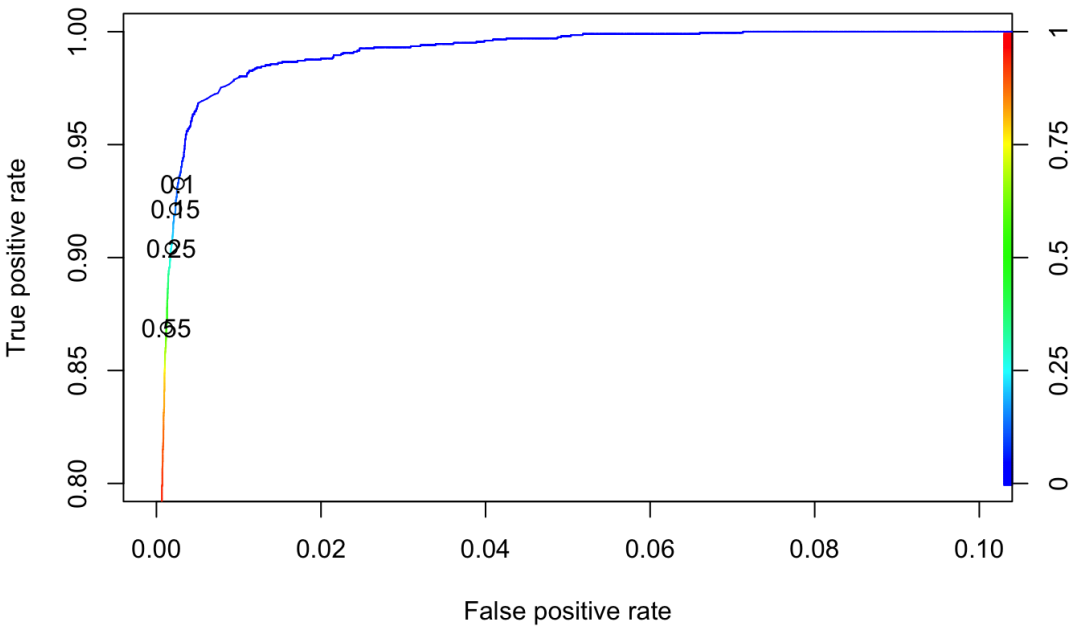
Linear Discriminant Analysis Model (LDA)

The second model I built was the LDA model. I followed the same process of fitting a three predictor (additive) model and a five predictor (interaction) model with the additional `Blue:Red` and `Blue:Green` predictors. I used the same metric of AUC to choose which model to proceed with for threshold selection.

Below we see that the AUC for the interaction model is slightly higher than the AUC for the three predictor (additive) model. Given this higher metric for overall prediction performance, we proceed with threshold selection for the interaction model.

metric	lda_additive.performance	lda_interact.performance
AUC	0.997605	0.999027
Sensitivity	0.877369	0.874865

In Figure 2.1 below, we see that the the LDA model with interaction terms performs best for the task of identifying blue tarps (sensitivity) at lower probability thresholds. Once again, I selected an overall threshold for the model by the `F1` metric, which was maximized at a 0.10 threshold.



prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.92497	0.99507	0.94954	0.00343
0.10	0.92685	0.99529	0.93272	0.00265
0.15	0.92529	0.99524	0.92135	0.00232
0.20	0.92431	0.99522	0.91195	0.00203
0.25	0.92347	0.99521	0.90404	0.00178
0.30	0.92183	0.99515	0.89563	0.00157
0.35	0.92178	0.99516	0.89217	0.00144
0.40	0.91829	0.99497	0.88426	0.00137
0.45	0.91644	0.99488	0.87931	0.00131
0.50	0.91416	0.99475	0.87486	0.00129
0.55	0.91169	0.99462	0.86844	0.00121
0.60	0.90990	0.99453	0.86449	0.00118
0.65	0.90810	0.99443	0.86053	0.00114
0.70	0.90606	0.99434	0.85410	0.00103
0.75	0.90421	0.99424	0.85015	0.00100
0.80	0.90041	0.99404	0.84323	0.00098
0.85	0.89326	0.99367	0.82839	0.00087
0.90	0.88299	0.99315	0.80861	0.00075
0.95	0.87053	0.99254	0.78538	0.00062

Figure 2.1

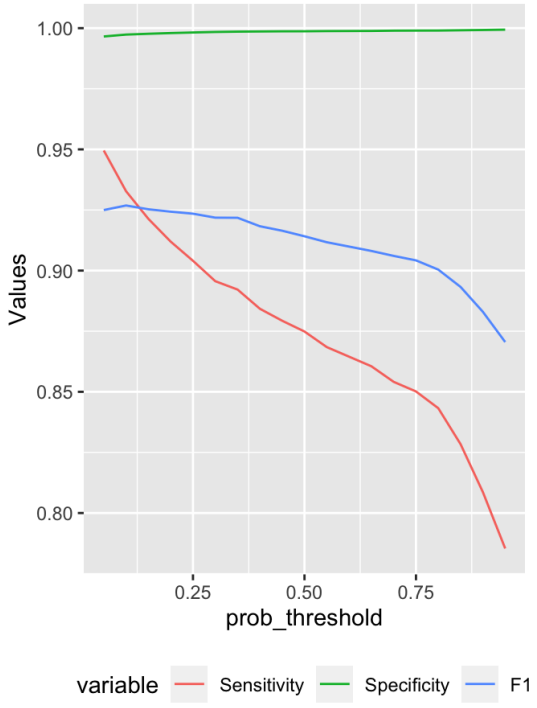
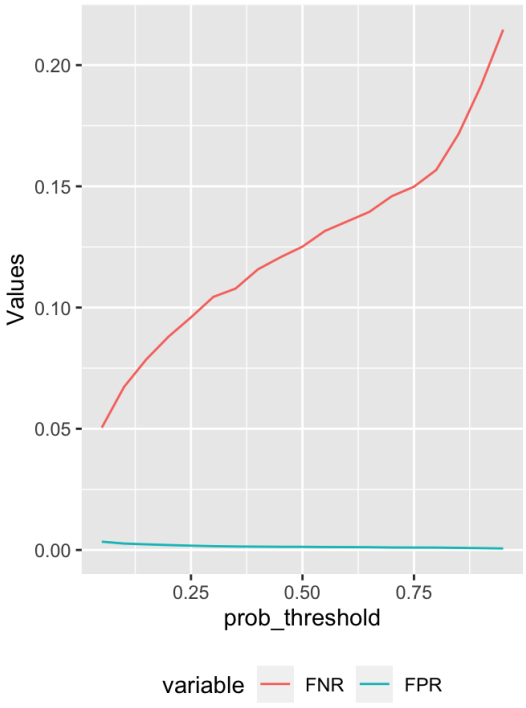


Figure 2.2



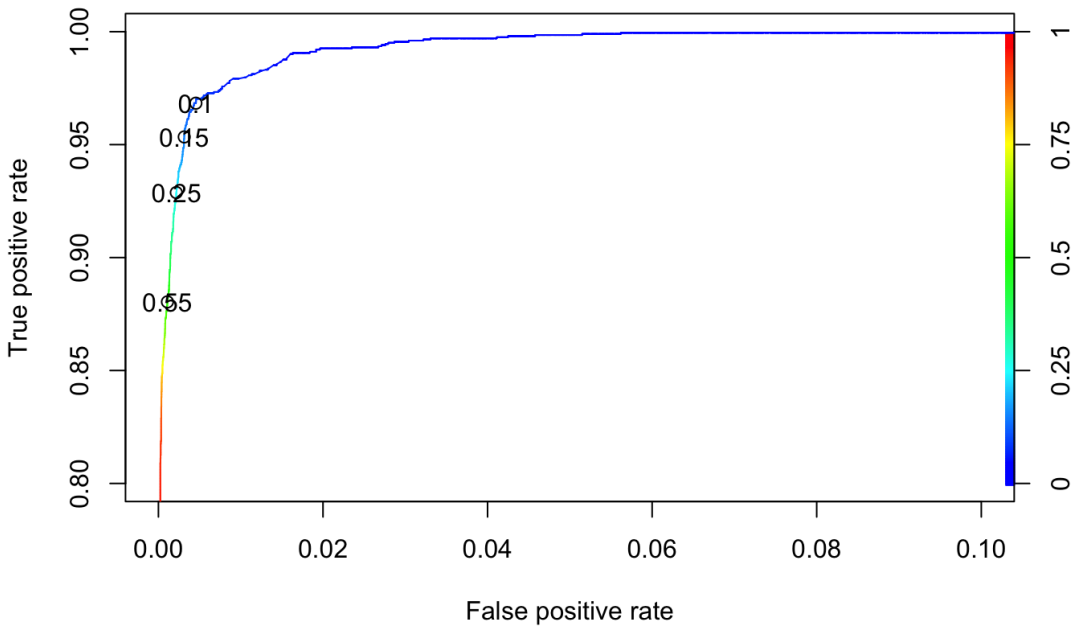
Quadratic Discriminant Analysis Model (QDA)

The third model I built was the QDA model. Again, I built one model with the three (additive) and another model with the five predictors (interaction).

Below we see that the AUC for the model with interaction terms is slightly lower than the simpler model. However, the Sensitivity, or the model's ability to accurately predict the `bluetarp` class is more than .9 higher for interaction model at this default 50% probability threshold. This is the most significant difference between model selection metrics. Therefore, we proceed with threshold investigation for both the five feature model and the three feature model.

metric	qda_additive.performance	qda_interact.performance
AUC	0.999112	0.998866
Sensitivity	0.885770	0.977745

Three Feature QDA Model




```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.80587	0.98473	0.98764	0.01537
0.10	0.91988	0.99459	0.96835	0.00454
0.15	0.93079	0.99546	0.95303	0.00314
0.20	0.93124	0.99556	0.93967	0.00260
0.25	0.93137	0.99562	0.92879	0.00217
0.30	0.93070	0.99562	0.91940	0.00186
0.35	0.92791	0.99549	0.90704	0.00158
0.40	0.92538	0.99537	0.89862	0.00144
0.45	0.92204	0.99519	0.88923	0.00131
0.50	0.92056	0.99511	0.88577	0.00127
0.55	0.92012	0.99511	0.88033	0.00109
0.60	0.91950	0.99511	0.87292	0.00085
0.65	0.91585	0.99492	0.86451	0.00077
0.70	0.91479	0.99488	0.86055	0.00069
0.75	0.91256	0.99478	0.85215	0.00051
0.80	0.91109	0.99472	0.84721	0.00041

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.85	0.90629	0.99447	0.83780	0.00036
0.90	0.90056	0.99417	0.82742	0.00033
0.95	0.87933	0.99307	0.79032	0.00023

Figure 3.1

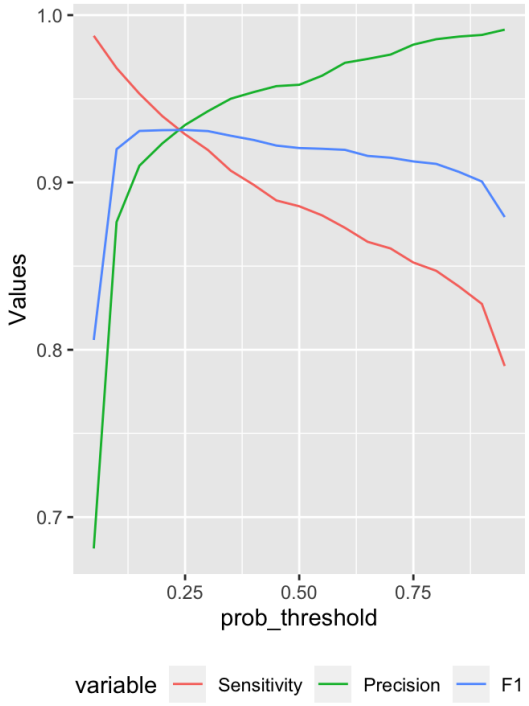
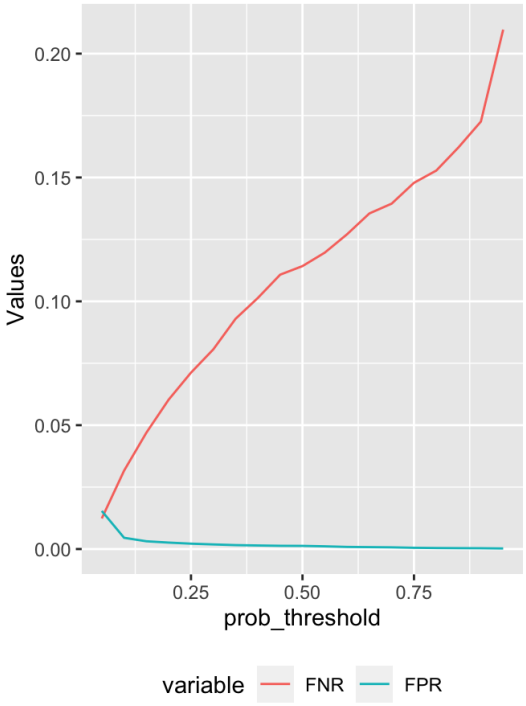
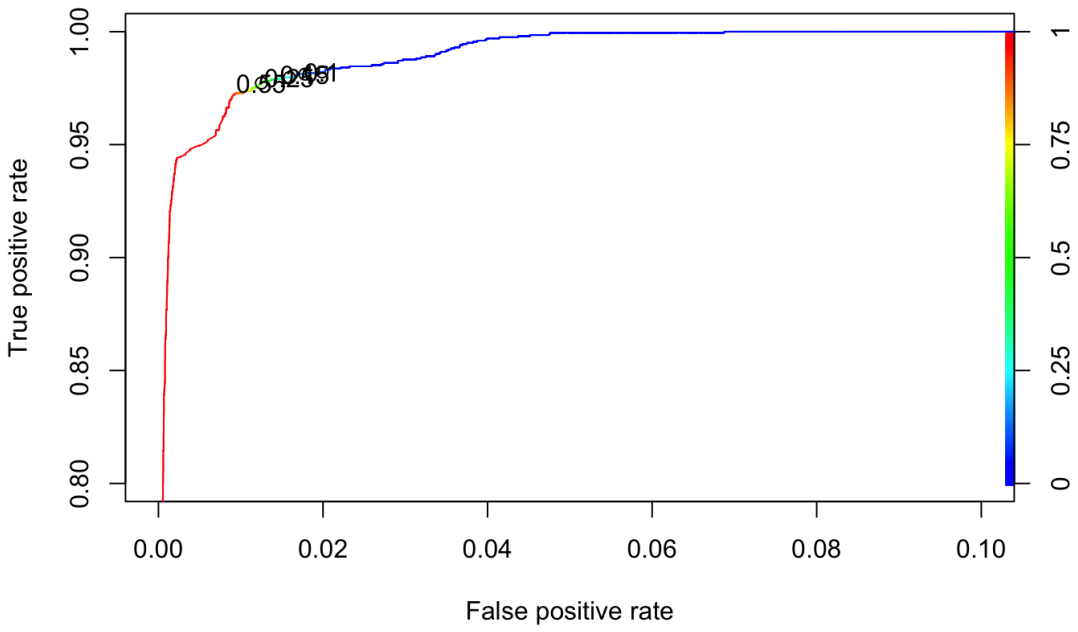


Figure 3.2



Five Feature QDA Model



prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.73449	0.97717	0.98417	0.02306
0.10	0.76097	0.98019	0.98219	0.01988
0.15	0.77954	0.98218	0.98121	0.01779
0.20	0.79010	0.98327	0.98022	0.01663
0.25	0.79662	0.98393	0.97972	0.01593
0.30	0.80574	0.98485	0.97923	0.01496
0.35	0.81241	0.98550	0.97873	0.01428
0.40	0.81656	0.98590	0.97873	0.01387
0.45	0.82084	0.98631	0.97824	0.01343
0.50	0.82600	0.98680	0.97774	0.01290
0.55	0.83018	0.98718	0.97725	0.01250
0.60	0.83220	0.98737	0.97675	0.01228
0.65	0.83723	0.98784	0.97527	0.01174
0.70	0.84167	0.98825	0.97379	0.01127
0.75	0.84482	0.98852	0.97379	0.01099
0.80	0.84998	0.98898	0.97329	0.01050
0.85	0.85428	0.98936	0.97280	0.01009
0.90	0.86216	0.99002	0.97280	0.00941
0.95	0.86887	0.99061	0.96983	0.00871

Figure 3.1

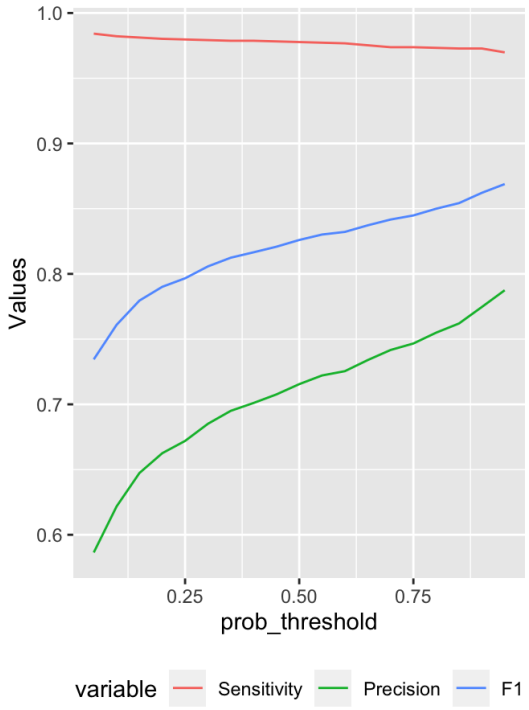
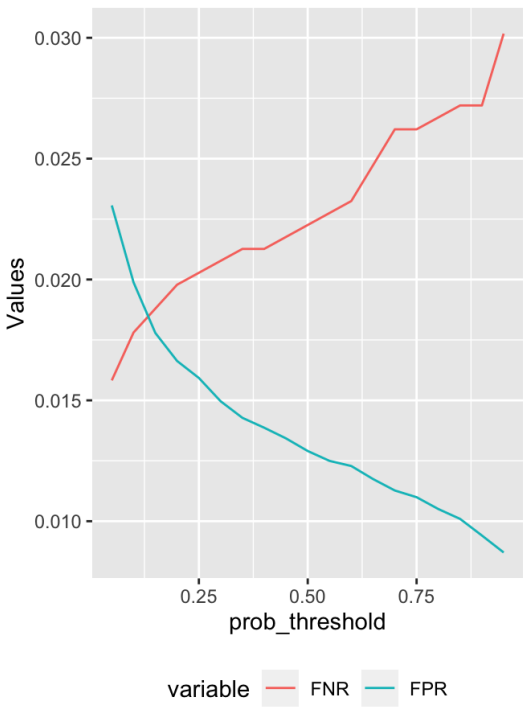


Figure 3.2

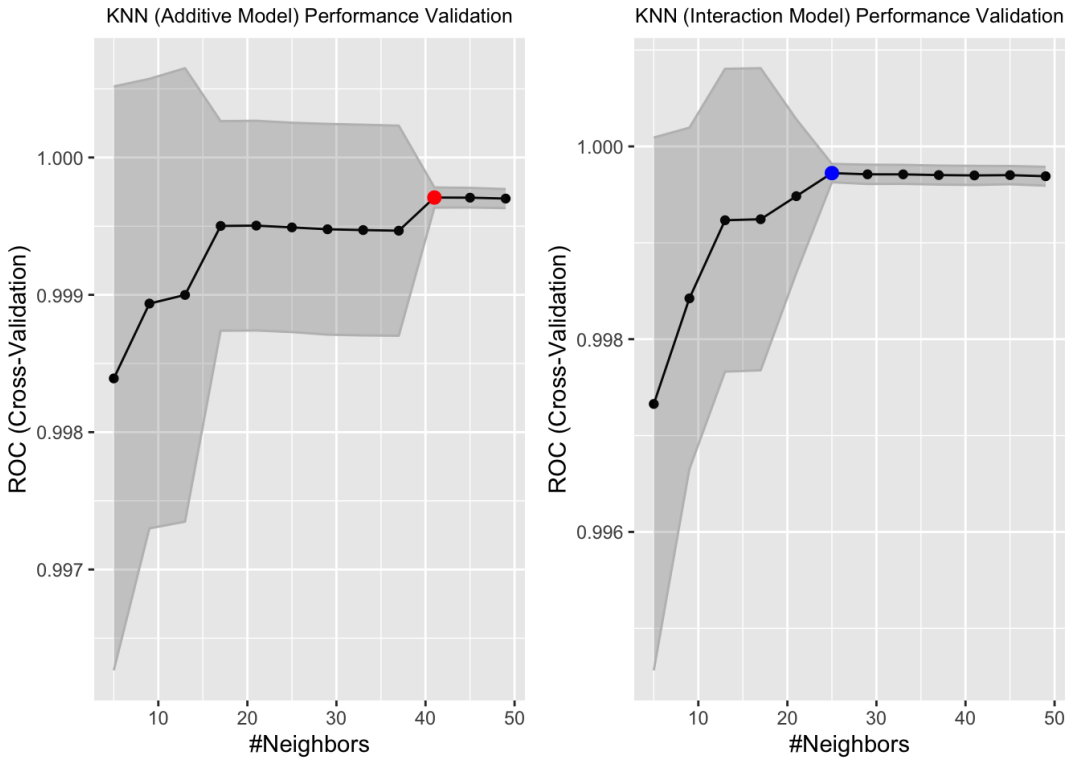


By a comparison of both QDA models, we see that the highest `F1` is found at a theshold of 0.25 fo the 3 features (additive) model. Importantly, the Sensitivity is lower at this threshold for this model. however, we keep in mind that a higher `F1` means that we have sacrified marginal performed for accurately predicting `bluetarps` to reduce the inaccuracies of predicting `bluetarps` where there are in fact none.

K-nearest Neighbor Model (KNN)

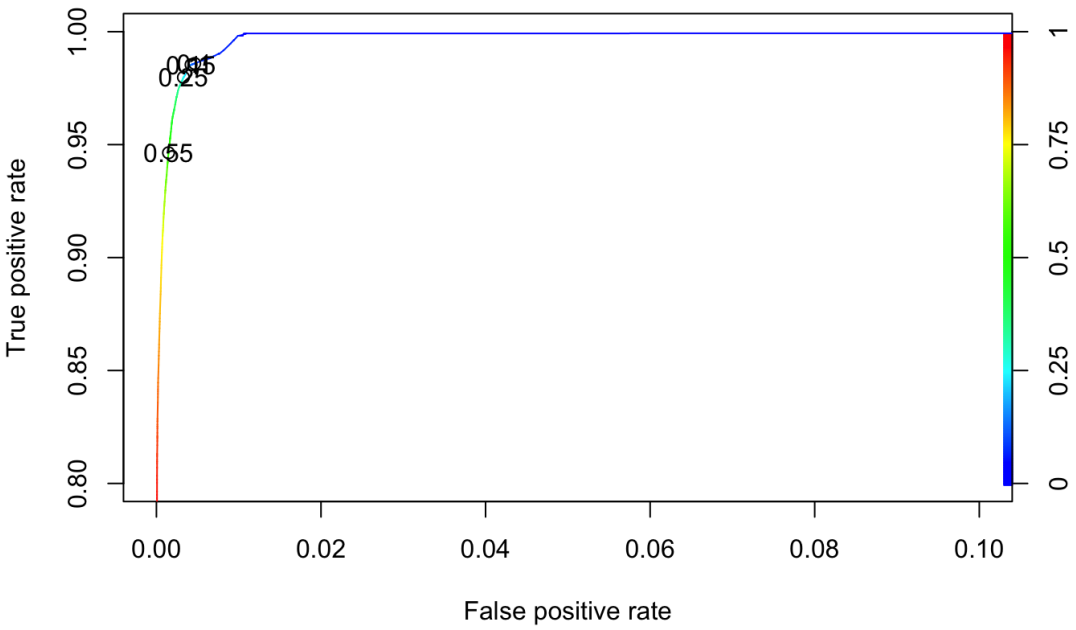
The fourth model I built was the KNN model. Again, I performed model selection for both the three predictor (additive) and five predictor model (interaction). In this instance, model selection involved tuning the parameter k, the size of the neighborhood. I also set up the tune grid to search over range of 5 to 50 by 4s for the optimal k. As mentioned earlier in this report, the argument to scale the feature values had already been added to the model-fitting function.

The results of the model tuning via 10-fold cross-validation show that for the three feature 0.999709. The results for the 5 feature (interaction) model with 5 features tuned an optimal k equal to 25 and a slightly greater AUC score of 0.999723. We proceed with the interaction model for threshold selection.



models_knn	k	ROC	Sens
Additive Model	41	0.999709	0.948086
Interaction Model	25	0.999723	0.951534

Below we see a probability threshold of 0.55 gives us the greatest `F1` score for this KNN model with interaction terms.



prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.86615	0.99012	0.99851	0.01016

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.10	0.92712	0.99503	0.98665	0.00469
0.15	0.93229	0.99541	0.98616	0.00428
0.20	0.93958	0.99595	0.98368	0.00364
0.25	0.94298	0.99621	0.98072	0.00328
0.30	0.94575	0.99641	0.97775	0.00297
0.35	0.94824	0.99660	0.97379	0.00265
0.40	0.94980	0.99674	0.96341	0.00216
0.45	0.95087	0.99684	0.95697	0.00185
0.50	0.95032	0.99682	0.95153	0.00168
0.55	0.95134	0.99690	0.94807	0.00149
0.60	0.95005	0.99685	0.93769	0.00119
0.65	0.94947	0.99684	0.93126	0.00100
0.70	0.94768	0.99674	0.92434	0.00087
0.75	0.94041	0.99633	0.90751	0.00074
0.80	0.93064	0.99583	0.88031	0.00036
0.85	0.92029	0.99527	0.85756	0.00018
0.90	0.90561	0.99450	0.82985	0.00007
0.95	0.89223	0.99380	0.80662	0.00002

Figure 4.1

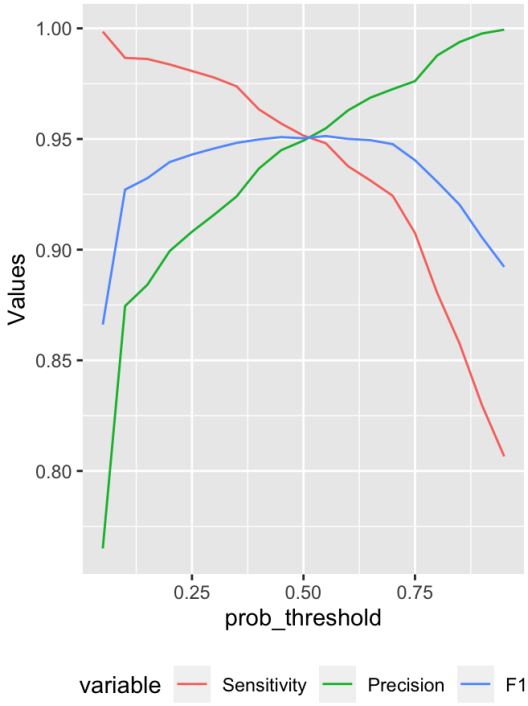
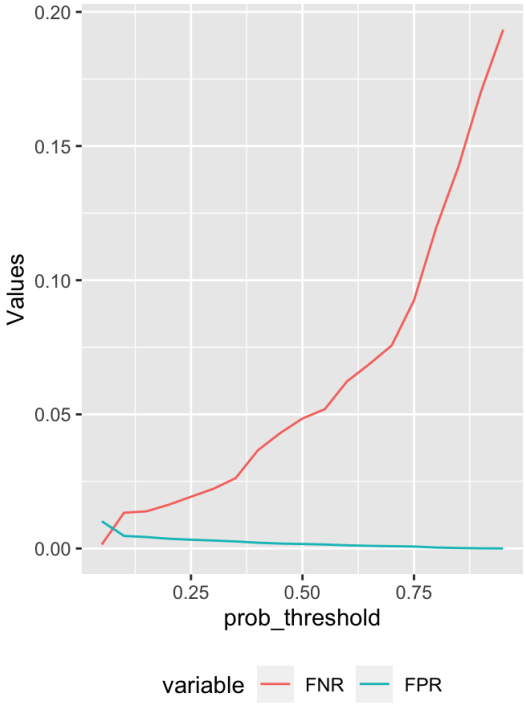


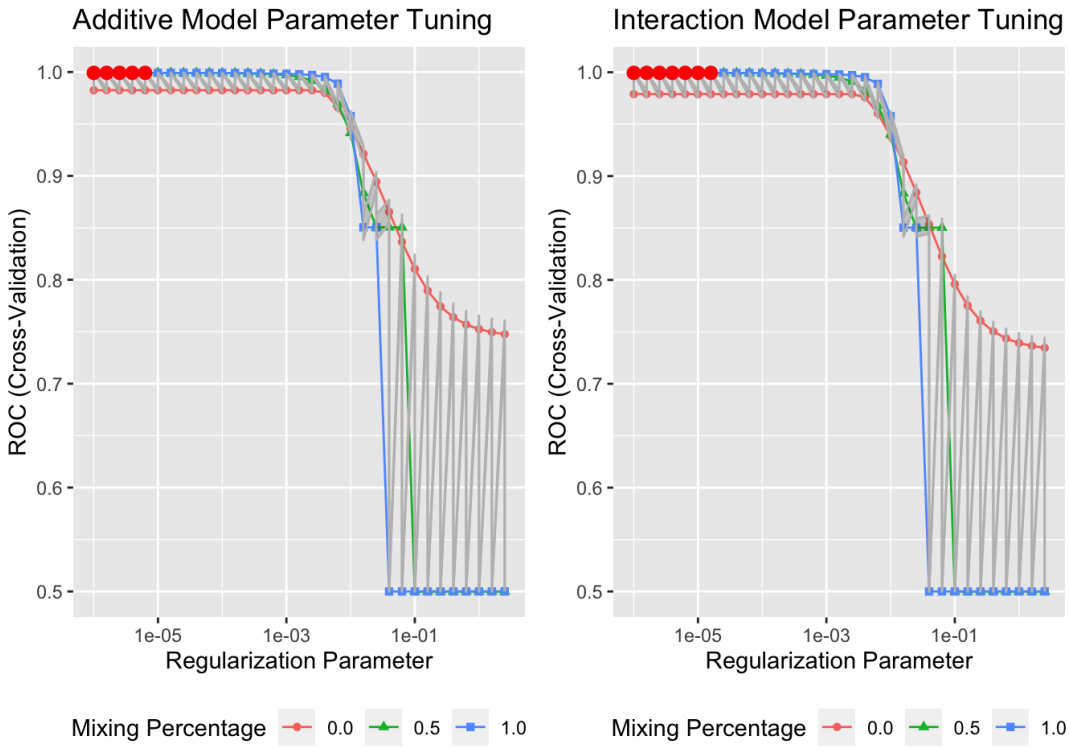
Figure 4.2



Penalized Logistic Regression Model

The fifth and final model that I built was the penalized Logistic regression model. I chose to perform elastic net and tune both the α, mixing percentage, and λ, regularization parameter, in the penalty. I set up the tune grid to search for an optimal α and λ over a range of (0, 0.5, 1) and (10e-6 to 10e0.4) by 10e0.2, respectively. The model-fitting function was set to choose the optimal tuning parameters using the maximized AUC. I ran the model selection for both the three feature model and the five feature model.

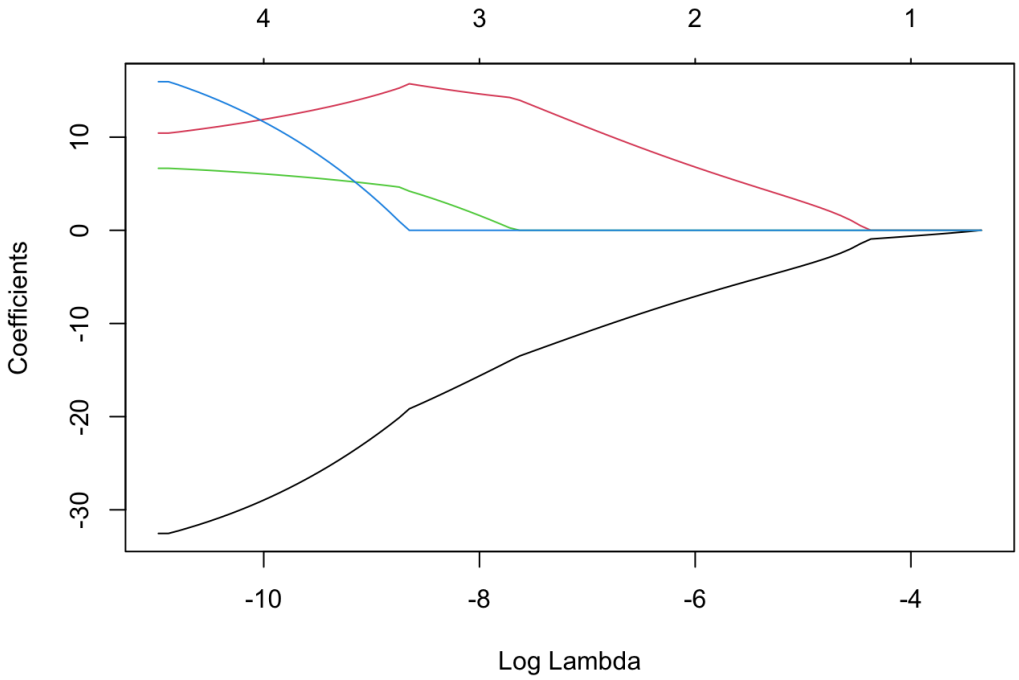
Below, the results of the tuning and selection via 10-fold cross validation for both models show that the optimal α equals 1. The optimal lambda value for the additive model was 6.0e-6 with an AUC of 0.999168. The optimal lambda value for the interaction model was 1.6e-05 with an AUC of 0.999397. We explore threshold selection and the effect of the penalty on the interaction model below.



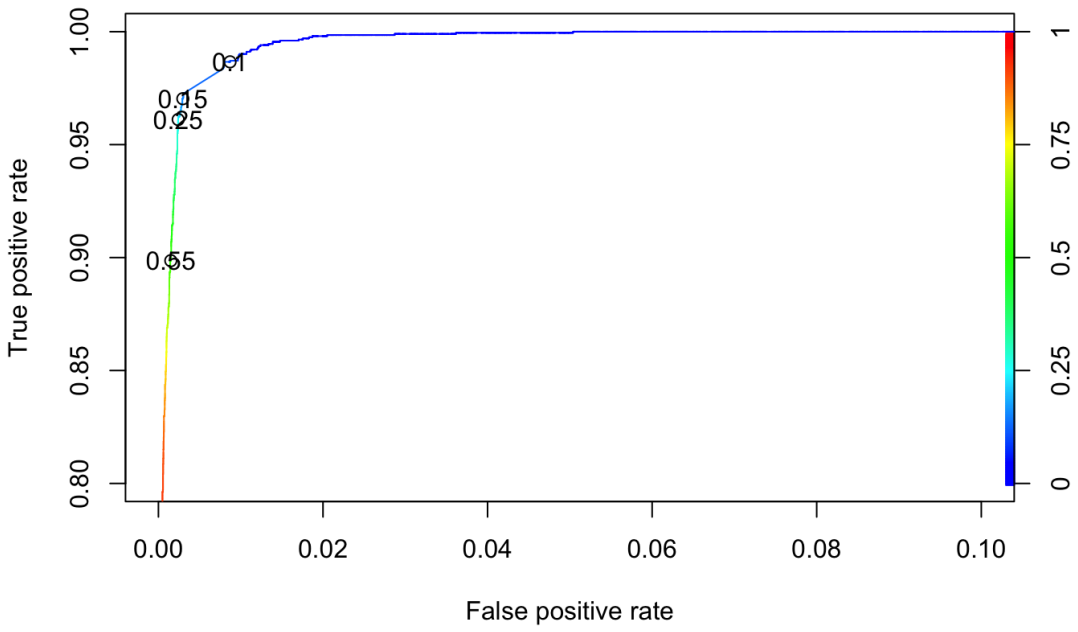
models_enp	alpha	lambda	ROC	Sens
Additive Model	1	6.0e-06	0.999168	0.900110
Interaction Model	1	1.6e-05	0.999397	0.907987

An elastic net model with an α = 1 is equivalent to LASSO regression. While the lambda value of both models was small, almost zero, we can see by comparing the coefficient output from our regular logistic and penalized logistic regression that the penalty had an impact. We see that the coefficients for the predictors have been shrunk; the coefficient for `Blue:Green` was set exactly to zero.

```
##             coefficients   logistic  penalized
## (Intercept)  (Intercept)   12.07731  11.428652
## Blue                Blue  -38.65214 -32.536381
## Red                  Red  -62.00621  10.433983
## Green              Green   69.80401   6.654858
## Blue:Red       `Blue:Red`  152.30929  15.953075
## Blue:Green   `Blue:Green` -122.15952   0.000000
```



Below we see a probability threshold of 0.25 gives us the greatest `F1` score for this LASSO model with interaction terms.



prob_threshold	F1	Accuracy	Sensitivity	FPR
0.05	0.85206	0.98898	0.99109	0.01109
0.10	0.87607	0.99105	0.98714	0.00882
0.15	0.94293	0.99624	0.97080	0.00292
0.20	0.94338	0.99630	0.96289	0.00260

prob_threshold	F1	Accuracy	Sensitivity	FPR
0.25	0.94431	0.99638	0.95894	0.00238
0.30	0.94074	0.99617	0.94953	0.00229
0.35	0.93786	0.99602	0.94014	0.00214
0.40	0.93401	0.99581	0.92728	0.00193
0.45	0.93099	0.99565	0.91787	0.00178
0.50	0.92796	0.99549	0.90799	0.00162
0.55	0.92439	0.99530	0.89810	0.00149
0.60	0.92210	0.99519	0.89019	0.00134
0.65	0.91682	0.99491	0.87832	0.00124
0.70	0.91323	0.99473	0.86746	0.00106
0.75	0.90782	0.99445	0.85609	0.00098
0.80	0.90308	0.99421	0.84472	0.00085
0.85	0.89653	0.99388	0.83037	0.00072
0.90	0.88578	0.99334	0.80912	0.00057
0.95	0.85634	0.99190	0.75768	0.00036

Figure 5.1

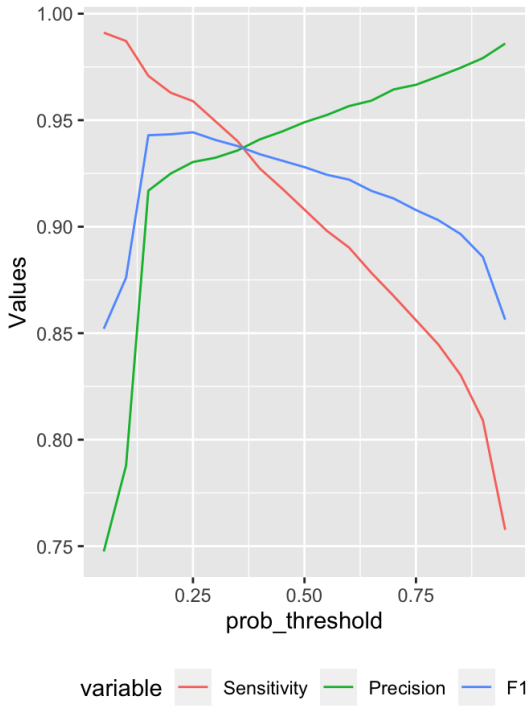
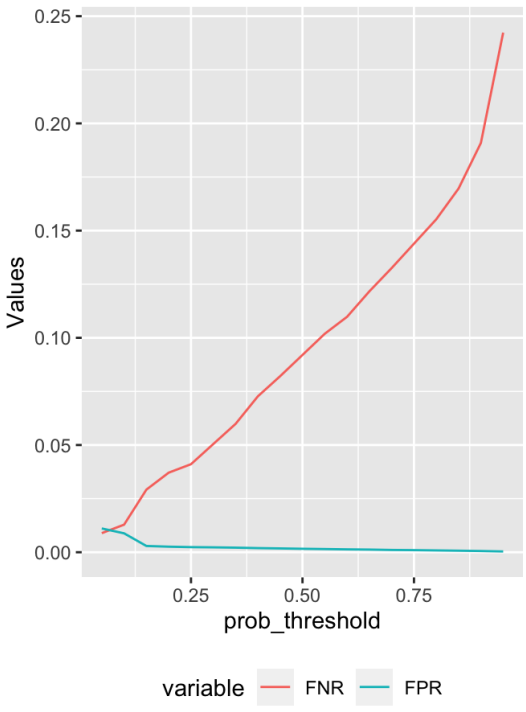


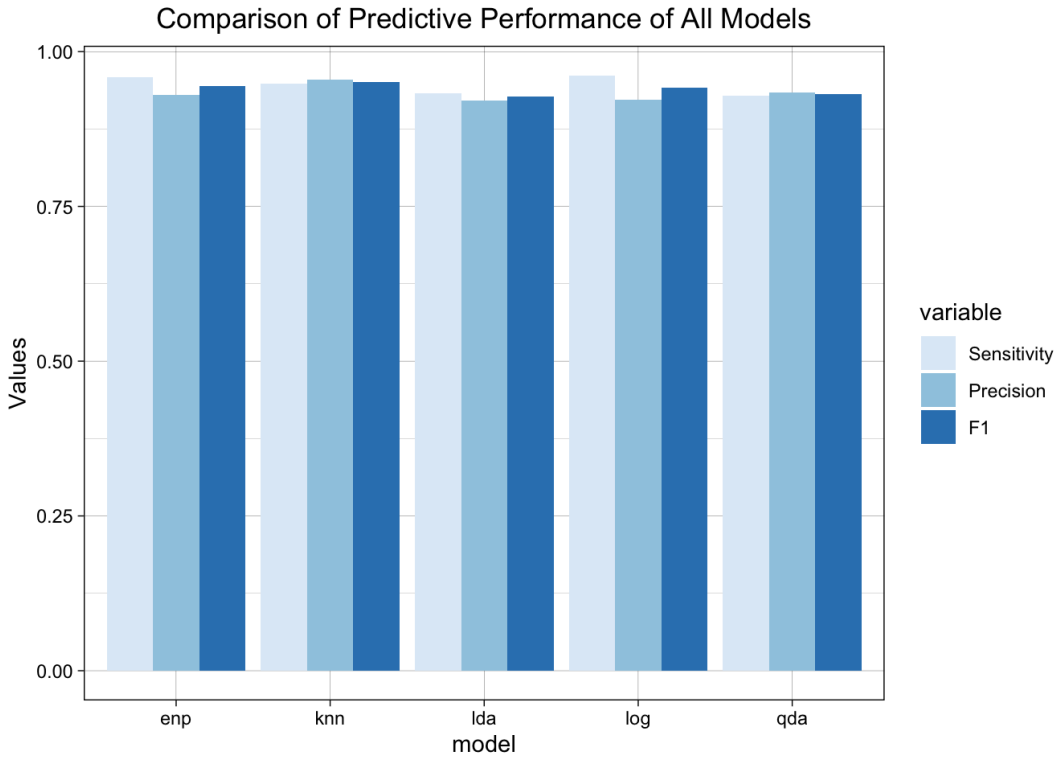
Figure 5.2



Performance Table

To summarize the top performing models across all five categories, I created a bar plot to compare their `F1`, `Sensitivity`, and `Precision` at the selected probability threshold. In theory, the `F1` metric should balance the issue of false positives and false negatives that `Sensitivity` and `Precision` quantify.

I also created a table that includes other threshold dependent (`FPR`) and threshold agnostic (`AUC` and `Accuracy`) metrics. All of the metrics were calculated using the re-sampling results from `caret` 's train object for each model. These metrics were calculated for each fold and the final reported metric is the average of the metrics calculated across all 10 folds.



Model	Optimal Tuning Parameters	AUC	Selected Threshold	F1	Accuracy	TPR (Sensitivity)	FPR
Logistic Regression	No. of Features: 5 - `Blue` , `Red` , `Green` , `Blue:Red` , `Blue:Green`	0.999474	0.25	0.94150	0.99617	0.96142	0.00268
Linear Discriminant Analysis	No. of Features: 5 - `Blue` , `Red` , `Green` , `Blue:Red` , `Blue:Green`	0.999027	0.10	0.92685	0.99529	0.93272	0.00265
Quadratic Discriminant Analysis	No. of Features: 3 - `Blue` , `Red` , `Green`	0.998685	0.25	0.93137	0.99562	0.92879	0.00217
K-Nearest Neighbors	No. of Features: 5 - `Blue` , `Red` , `Green` , `Blue:Red` , `Blue:Green` k = 25	0.999723	0.55	0.95134	0.99690	0.94807	0.00149
Penalized Logistic Regression	No. of Features: 4 - `Blue` , `Red` , `Green` , `Blue:Red` α = 1 λ = 1.6e-05	0.999397	0.25	0.94431	0.99638	0.95894	0.00238

Conclusions

1) Which algorithm works best and how confident is that conclusion?

Based on the predetermined metric of `F1` for model selection, the best performing algorithm is the K-Nearest Neighbors model with 5 features (`Blue` , `Red` , `Green` , `Blue:Red` , `Blue:Green`) and a $k = 25$. I chose this metric as the standard for model selection at the outset because it signaled the most optimal equilibrium of precision and sensitivity to blue tarps in the training data set. Because I am not aware of the types of resources that the rescue team has on the ground (both in terms of time and manpower), I decided to hedge the distribution of false positive and negatives in my final model.

The KNN model is the only non-parametric model this report considered. As such, our final model makes no assumptions about the data, has the lowest bias, but the most potential for variance. However, the large value for k cast a wide net for the neighborhood (reducing the flexibility of the model) and given that the training set had more than 63,000 observations, I think that our model is at a fairly low risk for overfitting the data. That said, the true test of confidence in this model will come when we fit and evaluate it's performance on a holdout set of the data.

2) Was there a clear winning algorithm?

While the non-parametric KNN model worked best given our `F1` metric, the parametric models still worked very well on this data set. The penalized logistic regression `F1` metric was less than one-tenth of a point lower than the `F1` value of the KNN model. Interestingly, this model reduced the set of predictor variables to four by eliminating the interaction term `Blue:Green` . This penalty effectively reduced the complexity of the original logistic model and could potentially lead to better generalized prediction performance when we validate on the holdout set of the data. Even the more biased models, the Quadratic and Linear discriminant analysis models had high `F1` values.

In our exploratory data analysis, we observed that the decision boundary between our two classes, `bluetarp` and `other` appeared approximately linear. There is a clear structure to the data, also evidenced by the large k value for our optimal algorithm. Therefore, there are multiple models that could perform well on this task. When considering future tasks of this kind, constraints like time/computation resources could even be a factor worth considering given that the nonparametric and parameter tuned models will take longer to train and evaluate.

3) How effective do you think your work here could actually be in terms of helping to save human life?

I think that this work could be very helpful in saving human life. I think that these approximate 95 point percentages for both precision - a predictive measure of how well we will do finding displaced people where we say they are, and sensitivity - a predictive measure of how well we will do finding all of the displaced people that are actually out there, are high enough that they justify the time and effort given to building and validation the models for use. I also think that there is value in having multiple models to consider on the validation set and in the field.

I've mentioned in this report that I don't know the extent of or lack of resources on the ground. However, if more information surfaces about the available resources, I can make necessary adjustment to the probability threshold to allow for more/less chance of false positives in the model. For exmaple, if there are additional resources, I view increasing the sensitivity and reducing the precision as the greatest elvel for potentially saving even more lives.

I am interested in validating these models on the holdout set and applying them to the geo-referenced imagery data as the true test of their value. However, I think that the resampling metrics that we considered from our 10-fold cross-validation of a relatively large dataset is a very promising foundation.