Project Part 1

Virginia Brame, Clay Harris, Hai Liu

2025-02-27

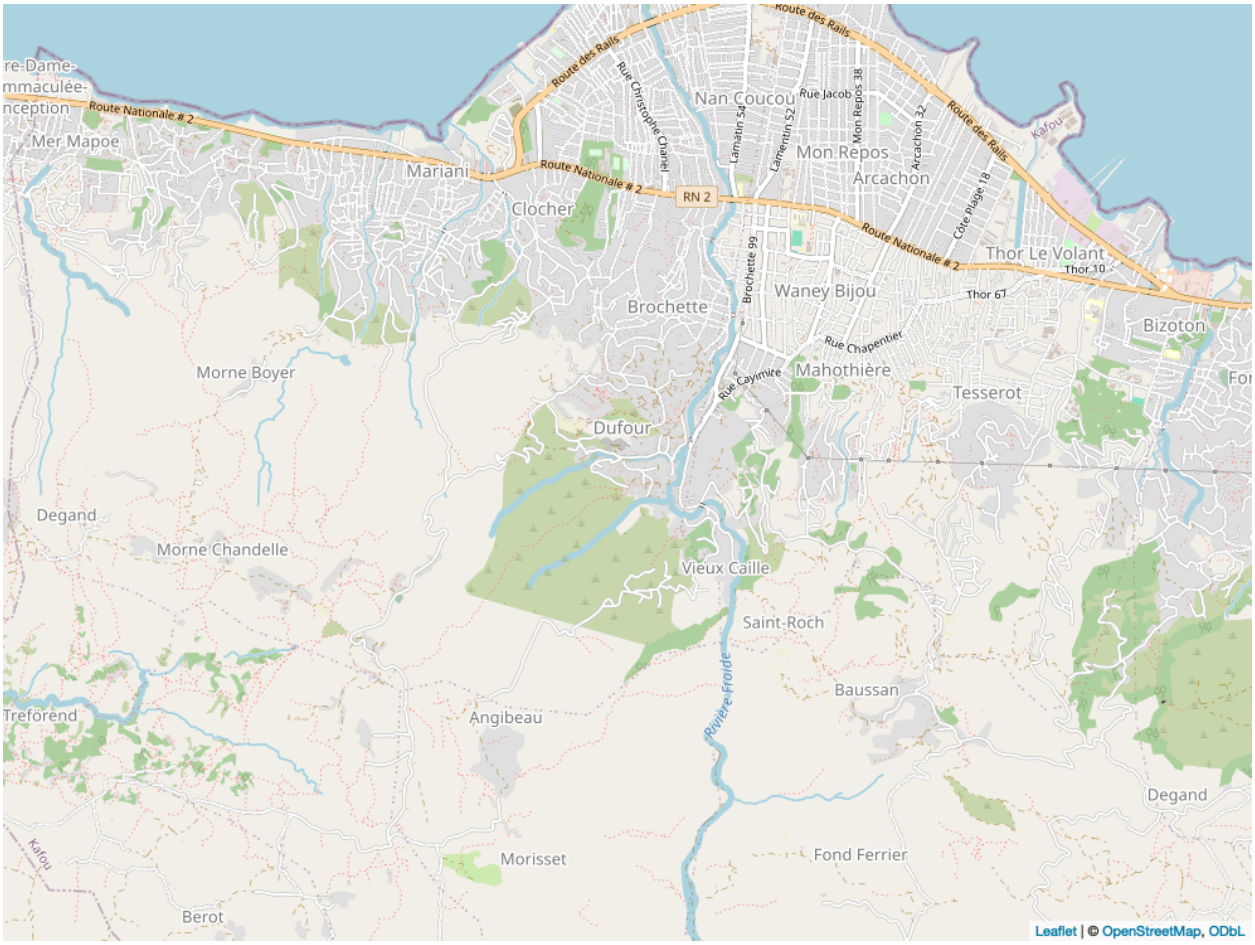
```
knitr::opts_chunk$set(
  echo = FALSE,
  cache = TRUE,
  autodep = TRUE,
  fig.align = "center",
  fig.pos = "H",
  out.width = "100%"
)
```

Data (loading, wrangling, EDA)

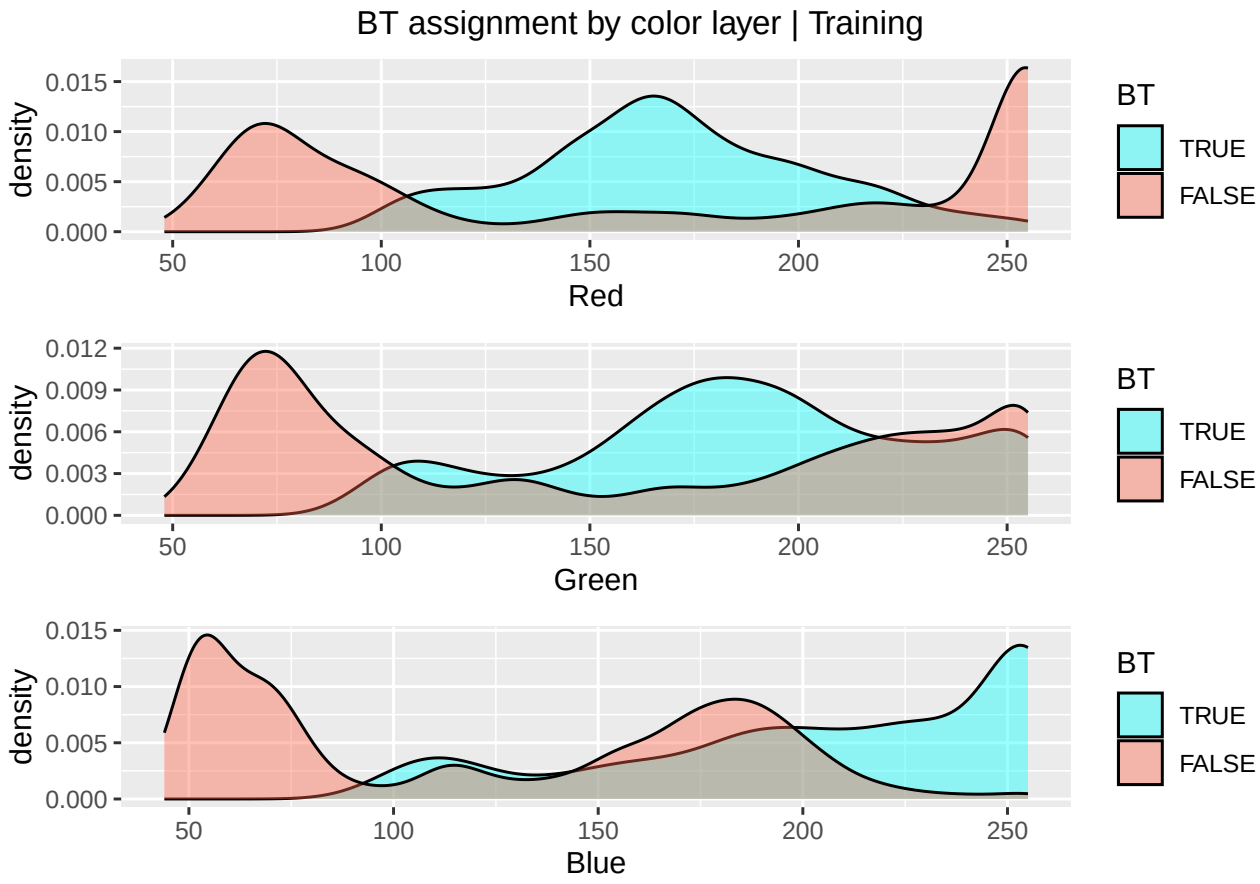
Data loading and wrangling

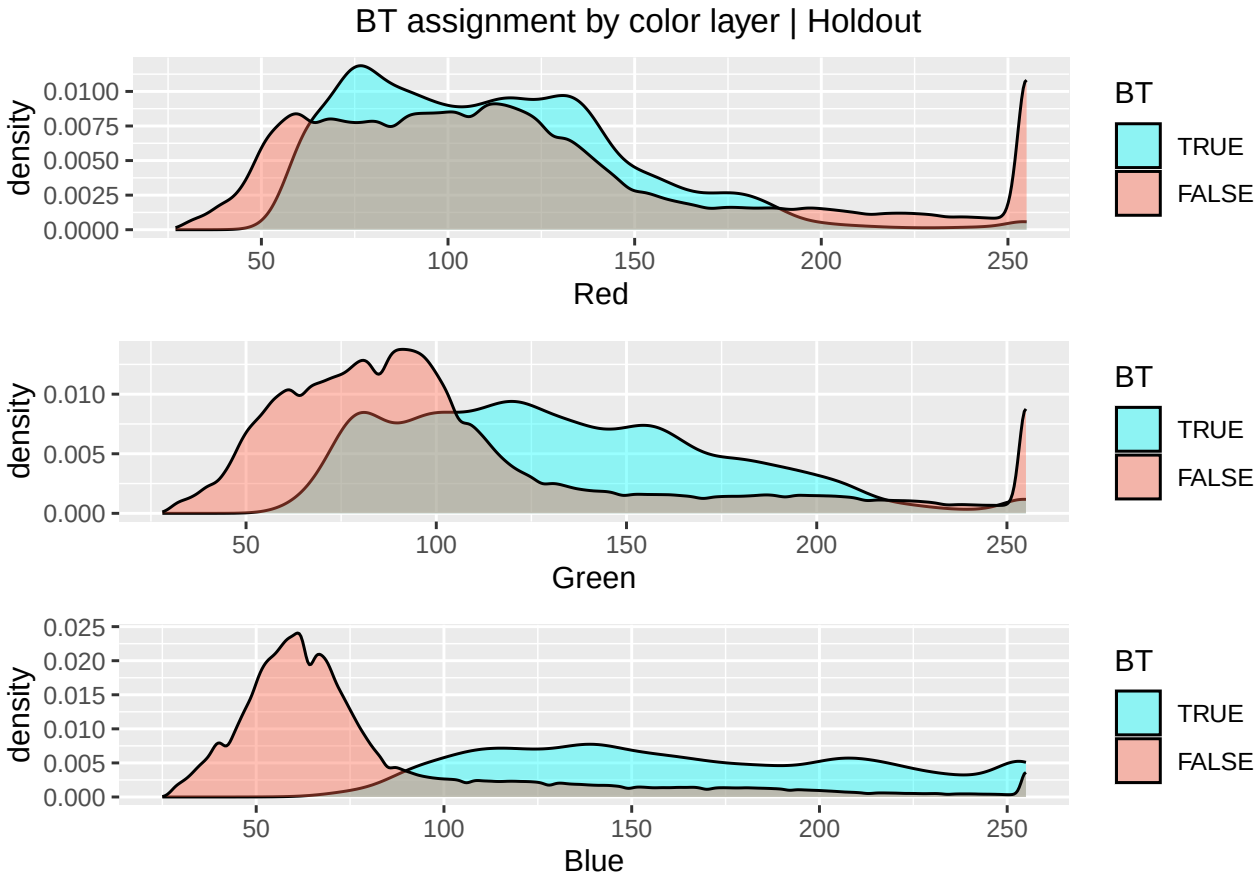
Since we are only interested in the level of "Blue Tarp", I create a new variable `BT` with only two classes, i.e., "TRUE" for "Blue Tarp" and "FALSE" for everything else.

EDA




```
## [1] 0.01344958
```

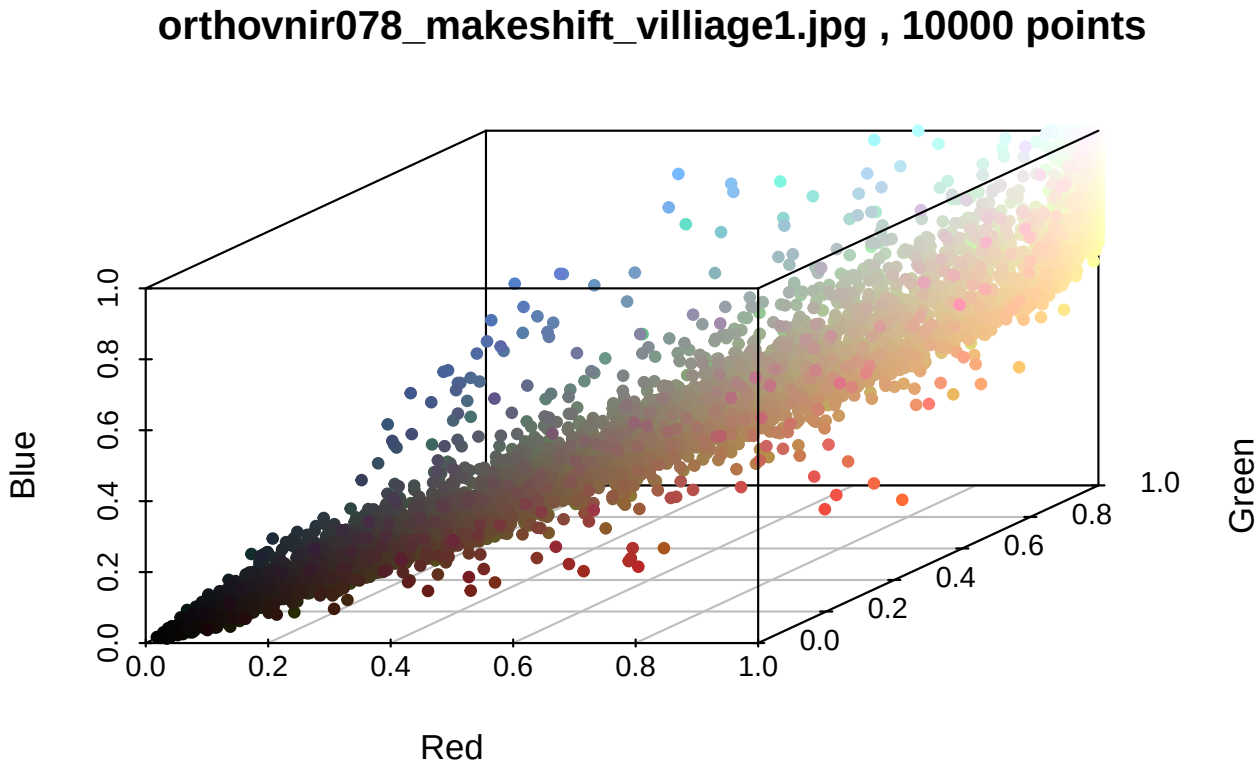

BT assignment by color layer | Training
density
0.015
0.010
0.005
0.000
50
100
150
200
250
Red
BT
TRUE
FALSE
density
0.012
0.009
0.006
0.003
0.000
50
100
150
200
250
Green
BT
TRUE
FALSE
density
0.015
0.010
0.005
0.000
50
100
150
200
250
Blue
BT
TRUE
FALSE



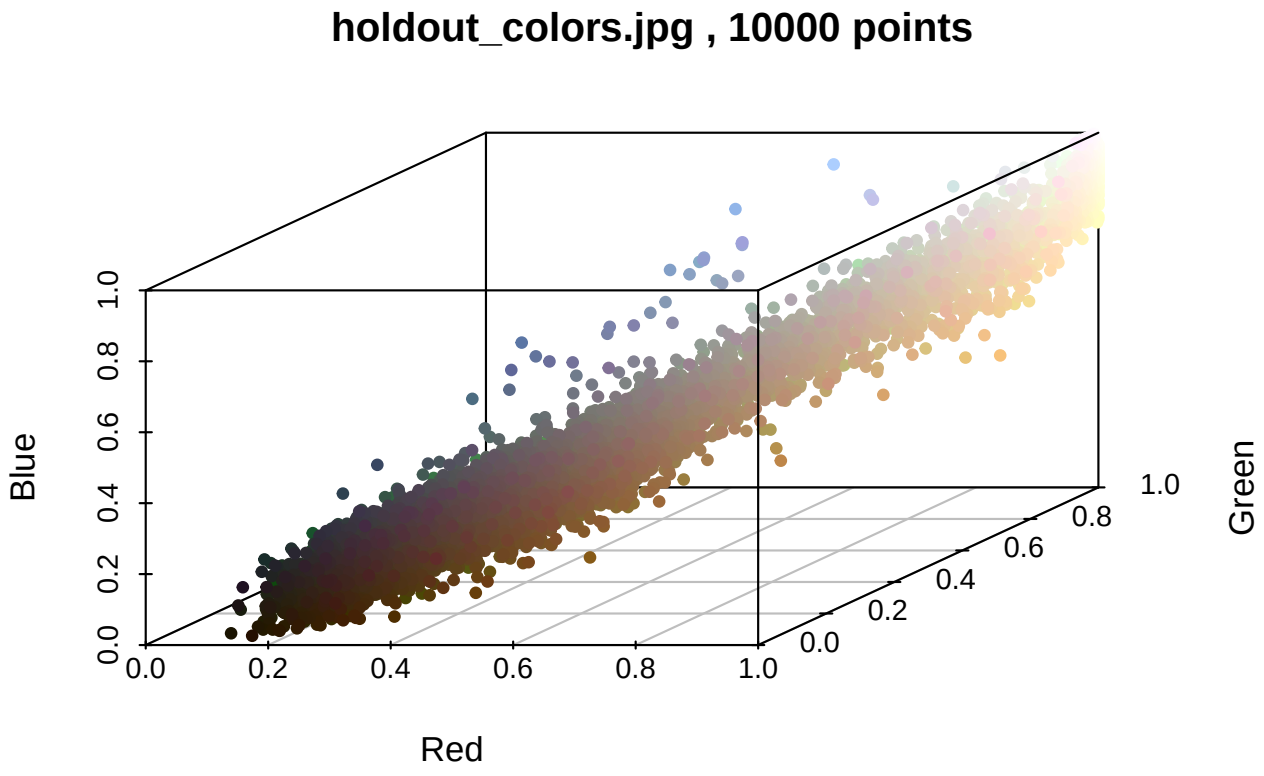
Understanding that: - Blue is 0,0,255 - Green is 0,255,0 - Red is 255,0,0

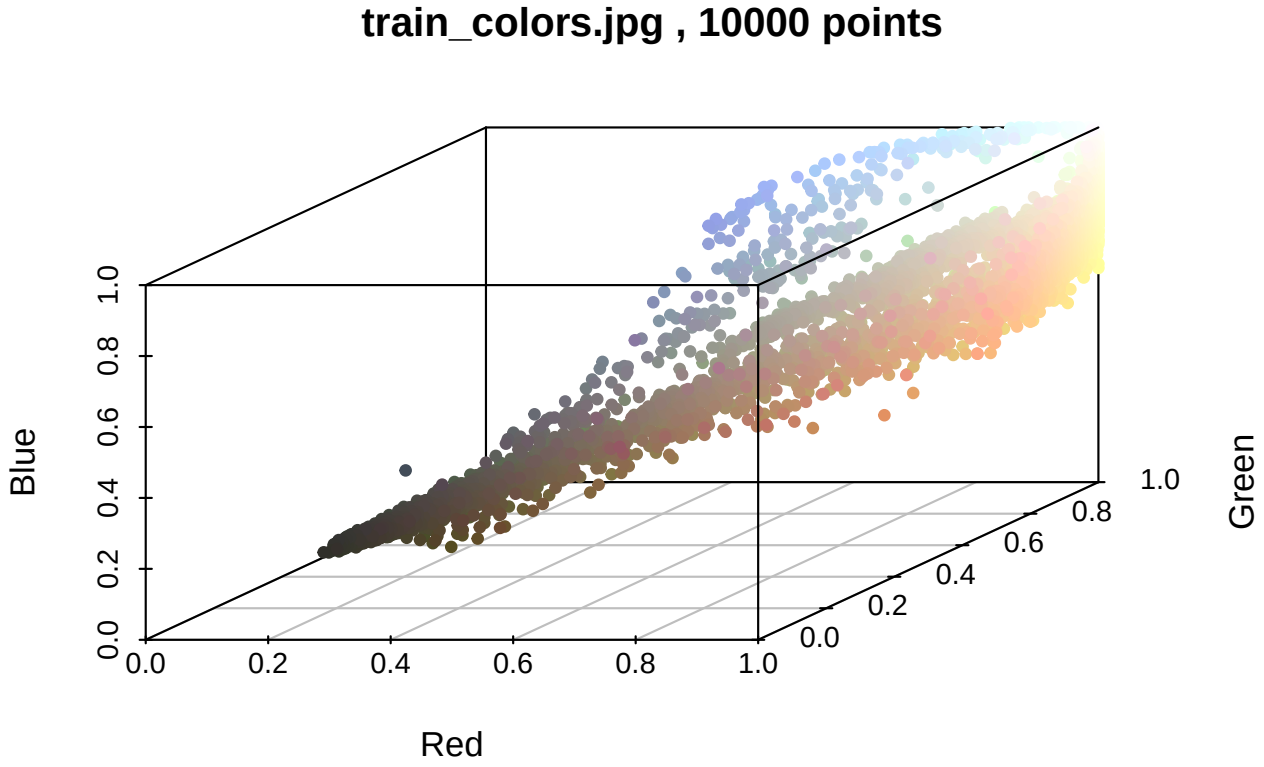
```
## No plotly plot.
```

Here we can clearly see the separation of the two levels of BlueTarp in the training data.

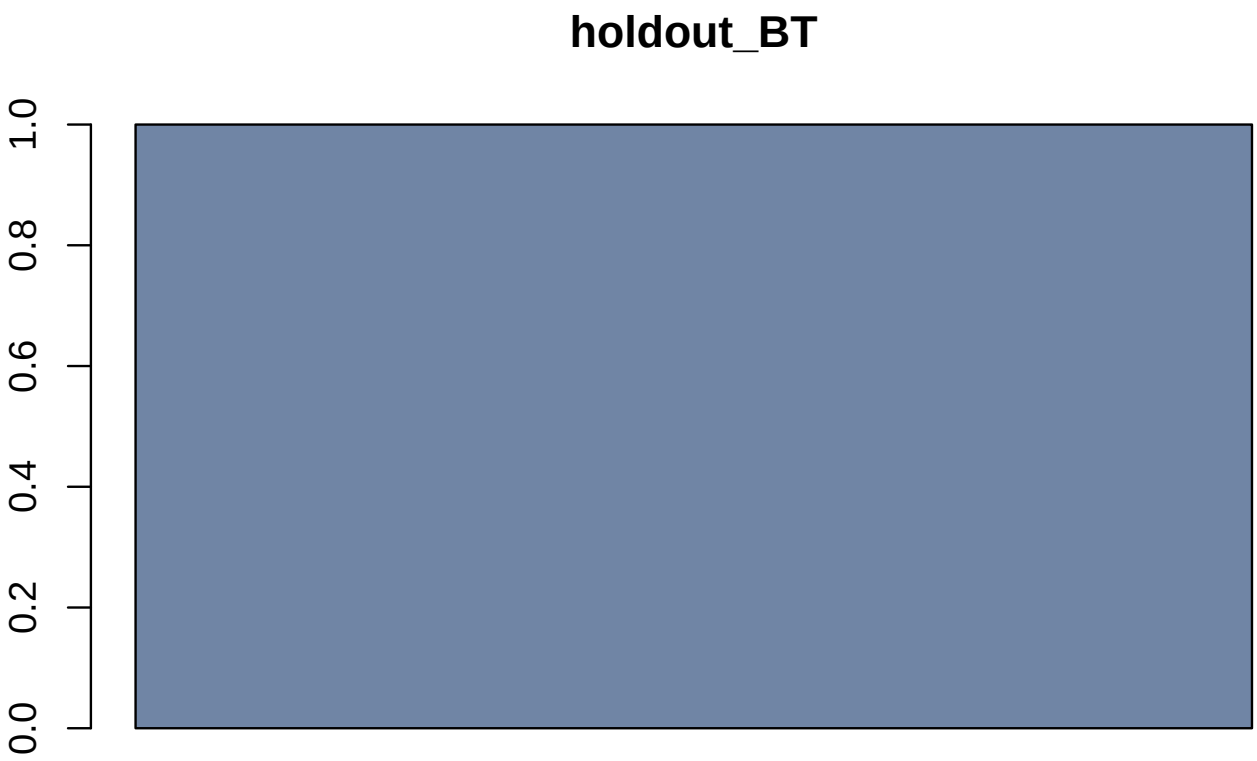

orthovnir078_makeshift_villiage1.jpg , 10000 points
Blue
Green
Red
0.0
0.2
0.4
0.6
0.8
1.0


orthovnir078_makeshift_villiage1
0.0
0.2
0.4
0.6






```
## RGB and HSV are device-dependent, perceptually non-uniform color spaces. See 'Color spaces' vignette
##
## Using 1*1*1 = 1 bins
```




```
## RGB and HSV are device-dependent, perceptually non-uniform color spaces. See 'Color spaces' vignette
##
## Using 1*1*1 = 1 bins
```

Have a look at the distributioin of the two classes for the outcome named “BT” (for BlueTarp).

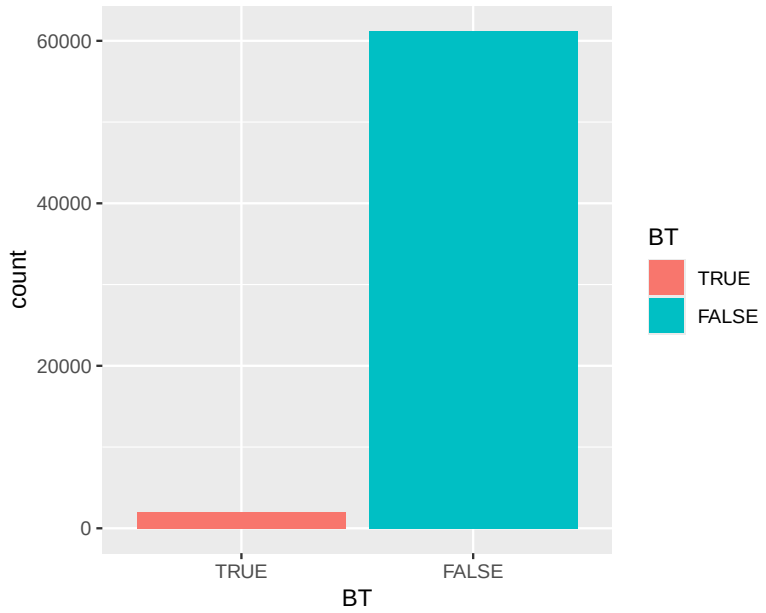


Figure 1: Distribution of Blue Tarp among all the observations.

I can see that the two outcome classes are extremely unbalanced. I will keep this in mind and deal with it later.

Methods

Model Preprocessing and Feature Selection Three classification models—logistic regression, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA)—were trained across three distinct color spaces: RGB (`Red`, `Green`, `Blue`), CIELab (`Luminance`, `a`, `b`), and HSV (`Hue`, `Saturation`, `Value`).

For each model, a `recipe()` function was defined to construct the feature set. No normalization was applied, as all three variables within each color space share the same ranges and units. This decision was made to preserve the original scale of the data and allow for potential coefficient interpretability in logistic regression.

Three distinct feature sets were defined: - **RGB:** Predictor variables include `Red`, `Green`, and `Blue`. - **CIELab:** Predictor variables include `Luminance`, `a`, and `b`. - **HSV:** Predictor variables include `Hue`, `Saturation`, and `Value`.

The response variable, `BT`, was a binary indicator denoting whether a given pixel belonged to a blue tarp (`BT = TRUE`) or not (`BT = FALSE`). The same response variable was used across all three color spaces to ensure direct comparability between models.

Each `recipe()` was applied to its corresponding dataset before training the models.

Model Specification and Workflow Definition For each color space, a logistic regression, LDA, and QDA model were defined and combined into a `workflow()` object. This structure ensures consistency in model training and evaluation.

The following table summarizes the nine model configurations:

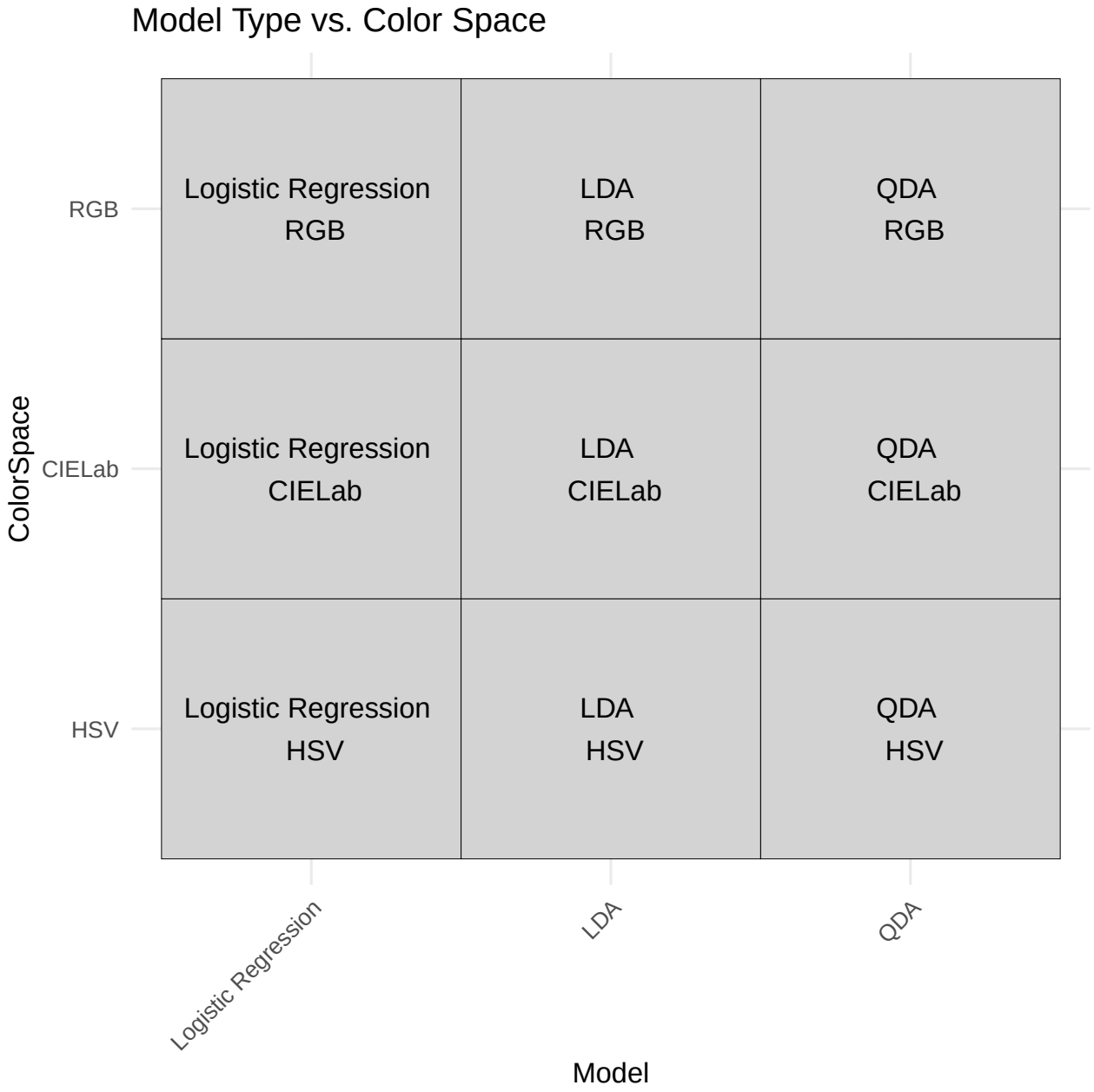


Figure 2: Model Type vs. Color Space

Combine preprocessing steps and model specification in workflow.

Cross-Validation A 10-fold cross-validation procedure was implemented using stratified sampling to ensure that each fold maintained the same proportion of positive (`BT = TRUE`) and negative (`BT = FALSE`) cases as the full dataset.

Performance was evaluated using ROC-AUC as the primary metric, with accuracy, precision, and F-measure also recorded for additional comparison. Predictions from each resampling iteration were saved to enable the construction of ROC curves based on cross-validation results, allowing for a detailed examination of model performance across different thresholds.

Each model—logistic regression, LDA, and QDA—was cross-validated separately within its respective color

space (RGB, CIELab, and HSV), resulting in a total of nine cross-validation experiments.

Cross-validation

Cross-Validation Performance by Color Space The table and plot below show the cross-validation performance of logistic regression, LDA, and QDA across three color spaces: RGB, CIELab, and HSV. The metrics include accuracy, F-measure, precision, and ROC-AUC. At this point, no specific threshold has been set; the models are evaluated using the default 0.5 decision boundary. Later sections will discuss threshold tuning in detail. ROC-AUC is emphasized here because it measures the model's ability to distinguish between positive (blue tarp) and negative (not blue tarp) classes independently of any specific threshold.

Logistic regression and QDA achieve similar accuracy and ROC-AUC, both approaching 0.999 in certain color spaces. LDA shows lower values for F-measure and precision, which indicates a higher rate of misclassifications relative to the other two models. However, its accuracy and ROC-AUC remain close to 0.98 or higher, suggesting that it still separates the classes reasonably well across thresholds.

The RGB and CIELab color spaces yield nearly identical results for logistic regression and QDA, both reaching an accuracy of about 0.995 and an ROC-AUC near 0.998–0.999. CIELab provides a slight improvement for LDA, with an F-measure of 0.776 compared to 0.761 in RGB. HSV's accuracy is marginally lower (about 0.993–0.994), yet its F-measure is more balanced across all three models.

Because ROC-AUC is computed independent of setting a decision threshold, it remains the primary indicator of each model's overall discriminative ability. Both logistic regression and QDA maintain high ROC-AUC scores in all color spaces, indicating strong separation between positive and negative classes without relying on a fixed threshold. LDA remains slightly less effective under the default threshold, but it still achieves a reasonable separation, with ROC-AUC values generally above 0.95.

No specific threshold tuning has been performed yet. Future steps will explore how threshold adjustment affects metrics such as F-measure and precision for each model-color space combination.

Table 1: Cross-validation performance metrics by color space.

Color Space	Model	Accuracy	F-measure	Precision	ROC-AUC
RGB	Logistic Regression	0.995	0.923	0.964	0.998
	LDA	0.984	0.761	0.725	0.989
	QDA	0.995	0.908	0.989	0.998
CIELab	Logistic Regression	0.995	0.926	0.965	0.999
	LDA	0.985	0.776	0.737	0.992
	QDA	0.995	0.911	0.985	0.999
HSV	Logistic Regression	0.993	0.894	0.880	0.977
	LDA	0.994	0.904	0.874	0.953
	QDA	0.994	0.903	0.865	0.989

Visualization of the same data

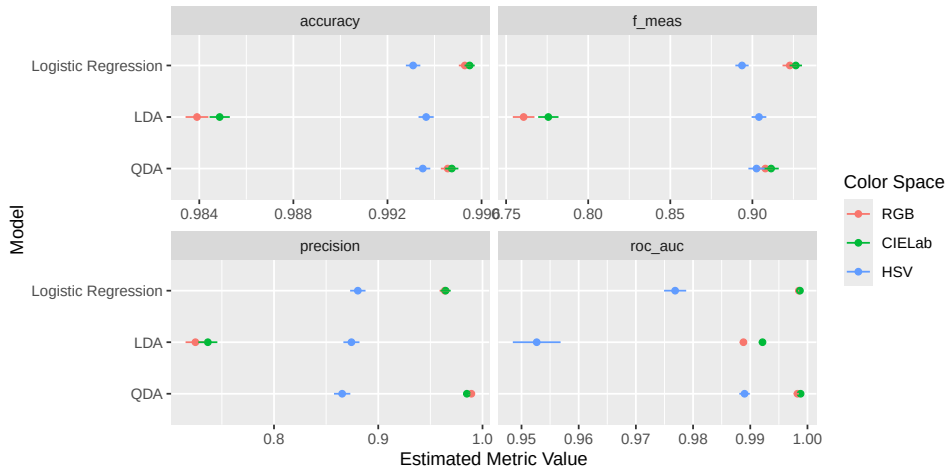


Figure 3: Cross-validation performance metrics by color space.

ROC Curves by Color Space In the figure below, each panel displays the ROC curve for logistic regression, LDA, and QDA within a given color space (RGB, CIELab, and HSV). The curves illustrate how each model's sensitivity evolves as the false-positive rate (1 – specificity) increases.

In the RGB and CIELab panels, both logistic regression and QDA reach near-perfect classification quickly, with very little area left above the curves. LDA remains consistently lower, indicating a reduced ability to separate the classes at most thresholds. The similar performance between RGB and CIELab is evident in how closely the curves for these two color spaces align.

The HSV panel shows that all three models require a slightly higher false-positive rate before attaining the same level of sensitivity observed in RGB and CIELab. LDA in HSV trails the other two models further, never fully approaching a sensitivity of 1. This indicates that in HSV, both false-positive rates and false-negative rates are somewhat higher compared to the other color spaces, particularly for the LDA model.

Across all color spaces, logistic regression and QDA yield high ROC-AUC values. LDA underperforms in comparison, with its curve consistently lying below those of the other two models. The marginal difference between RGB and CIELab curves suggests that both color spaces provide similar predictive power, whereas the HSV models lag slightly, requiring a higher trade-off (i.e., higher false-positive rate) for the same level of sensitivity.

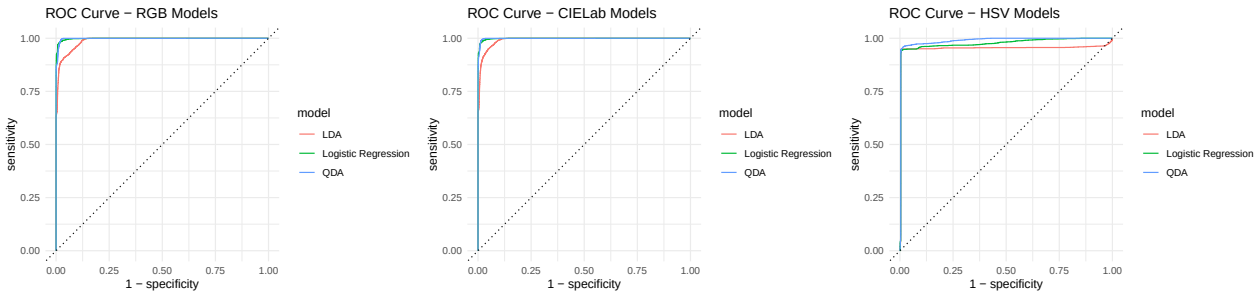


Figure 4: Cross-validation ROC curves for each color space

Threshold Selection and Optimization In many real-world classification tasks, the proportion of positive and negative cases is highly imbalanced. This dataset exhibits similar characteristics, prompting the need to adjust the decision threshold. A single default threshold (often 0.5) may overlook numerous minority-class instances or inflate false positives.

We use the `probably` package to systematically evaluate how varying the threshold affects metrics such as `accuracy`, `f_meas`, and `sens`. These three measures were chosen based on their direct relevance to the final application. A high sensitivity (`sens`) is desirable when missed positives have serious implications, while `f_meas` balances precision and recall to address the importance of correct positive classifications. Accuracy remains useful for an overall check, though it can be misleading in highly imbalanced settings.

Two functions are defined to explore threshold selection and illustrate the resulting confusion matrices: - `threshold_graph` scans multiple thresholds and collects metrics computed by `threshold_perf`. - `visualize_conf_mat` produces confusion matrices for specific thresholds selected from the metrics output.

Threshold Tuning by Color Space The figures below shows threshold tuning for logistic regression, LDA, and QDA across RGB, CIELab, and HSV. Each panel presents three metrics—accuracy, F-measure, and sensitivity—plotted against the decision threshold. Confusion matrices highlight the effect of selecting specific thresholds that maximize each metric.

In RGB and CIELab, the models yield similar patterns, reaching high metric values around comparable thresholds. HSV diverges somewhat, particularly in the LDA and QDA models where each metric has a similar value across all thresholds.

In this application, classifying blue tarps in satellite imagery for humanitarian aid, there is a strong need to balance precision (avoiding too many false alarms) with recall (finding as many actual tarps as possible). F-measure provides a middle ground, penalizing both missed targets and erroneous detections. Optimizing for sensitivity alone results in a large number of false positives, which demands additional field validation and can slow down response efforts. F-measure is therefore favored to maintain a reasonable rate of correct detections without overwhelming humanitarian workers and remote sensing specialists with false alarms.

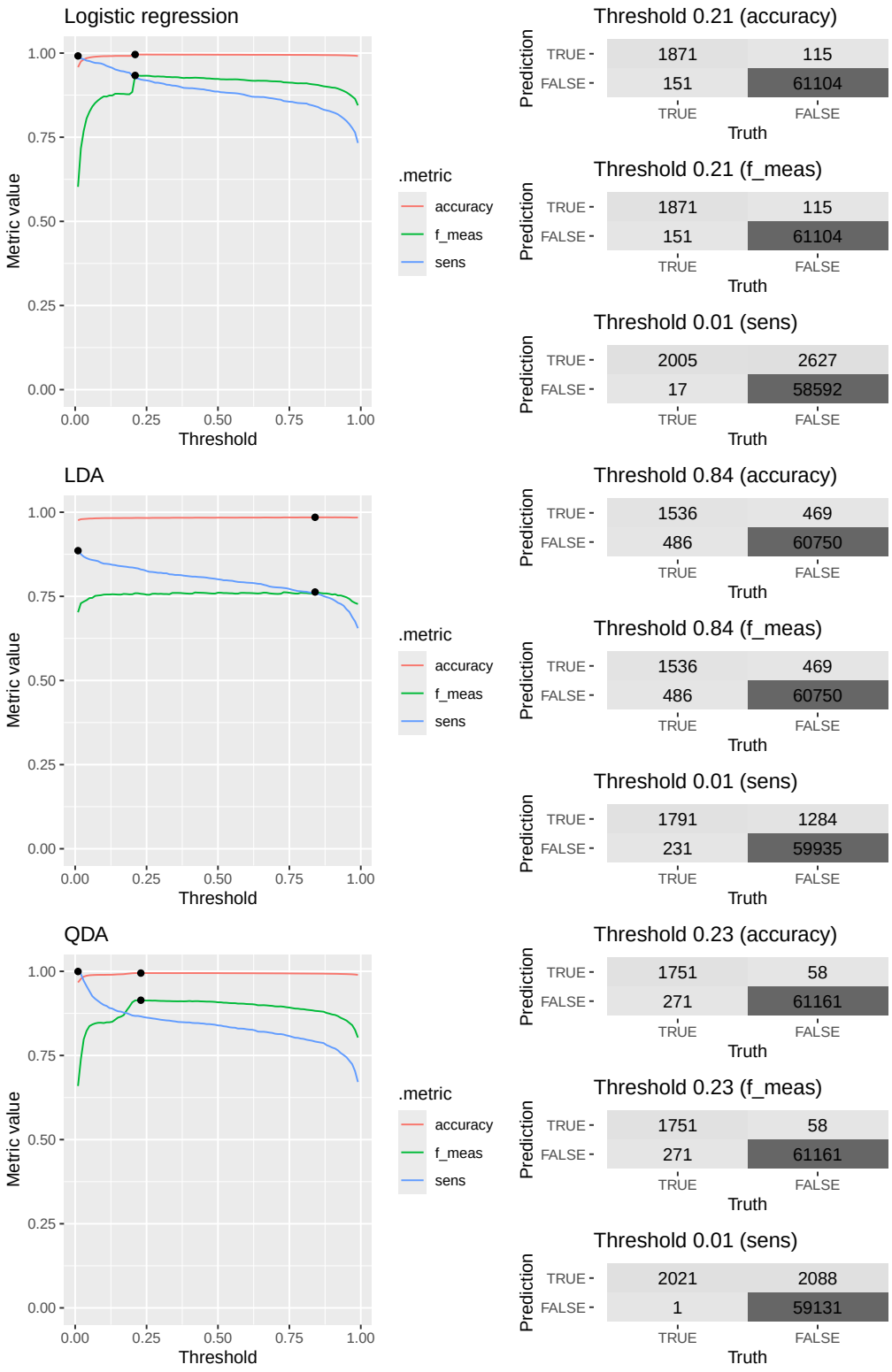


Figure 5: Metrics as a function of threshold optimization in the RGB color space.

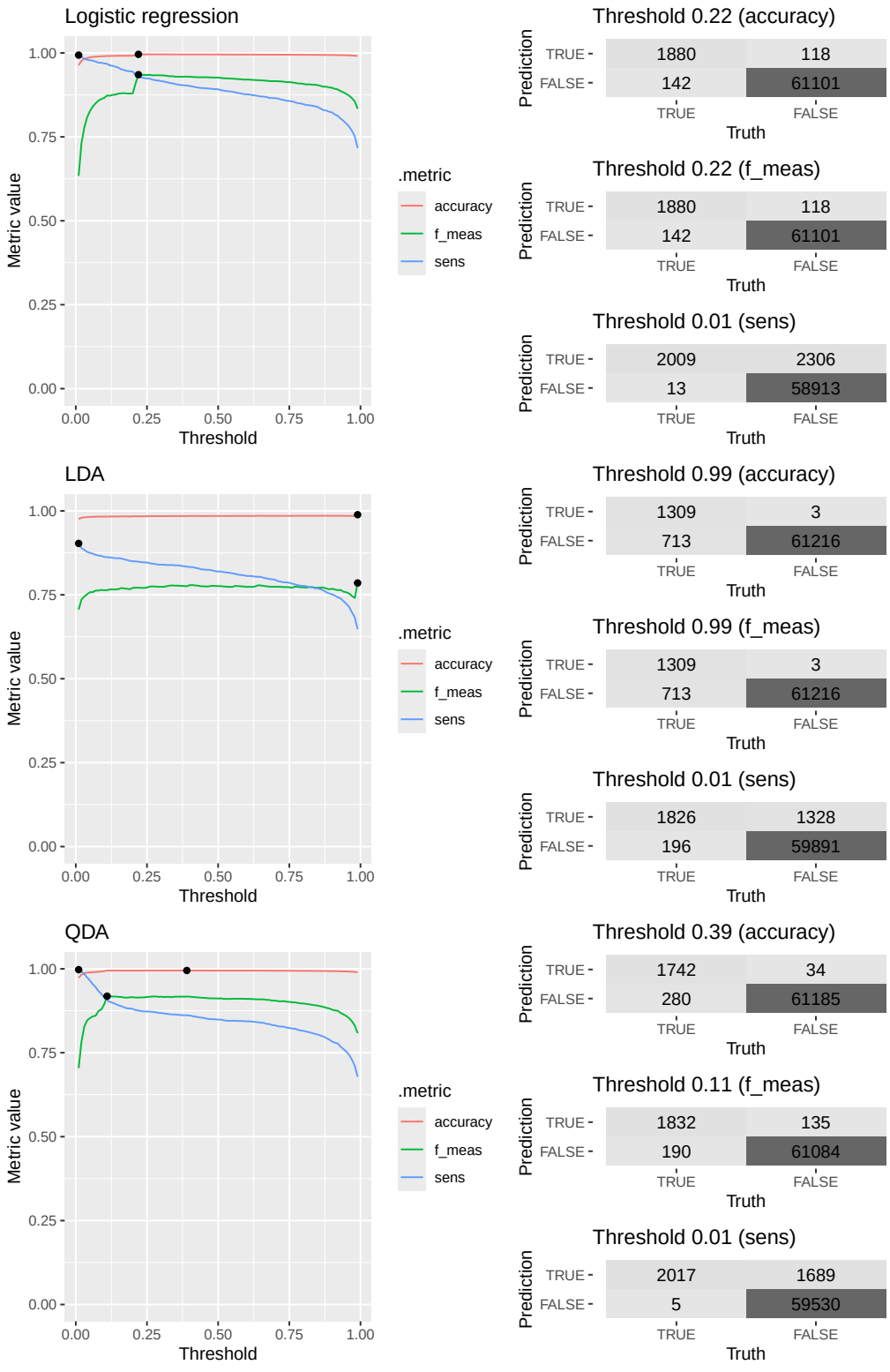


Figure 6: Metrics as a function of threshold optimization in the CEILab color space.

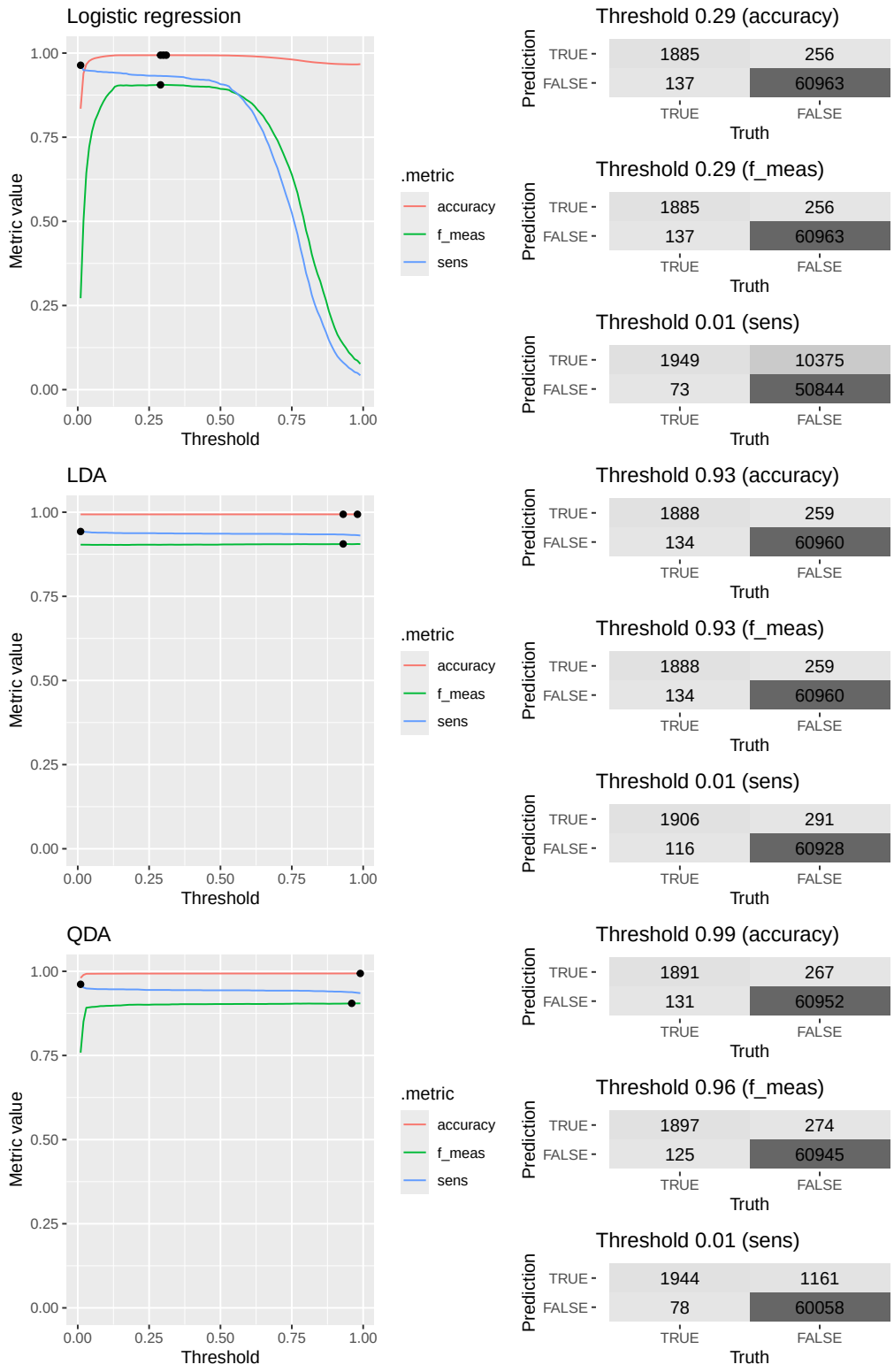


Figure 7: Metrics as a function of threshold optimization in the HSV color space.

Adjusting the F-Measure Weight While F-measure (`f_meas`) balances precision and recall, it places equal weight on both by default. In some domains, avoiding missed detections (false negatives) is critical,

prompting an increase in the weight of recall relative to precision. One approach is to adjust the F-measure parameter `beta`, where higher values of `beta` emphasize recall over precision.

In this research, the `yardstick` package's `metric_tweak` function was used to create custom F-measure metrics (`f_meas_adj2` with `beta = 2` and `f_meas_adj3` with `beta = 3`). These metrics were applied at a range of thresholds for logistic regression, LDA, and QDA, providing an in-depth view of how heavily prioritizing recall affects overall performance.

Although placing additional weight on recall reduces missed blue tarps, the number of false positives increased sharply. For instance, in the RGB color space, adjusting `beta` to 3 for logistic regression produced 1,960 true positives but 545 false positives, with only 62 missed detections. This outcome was deemed problematic, as every false-positive pixel would require manual validation, slowing humanitarian response efforts.

After comparing these adjusted metrics to the unmodified F-measure, we concluded that the added recall did not justify the sharp rise in false positives. The default F-measure strikes a more balanced trade-off between precision and recall for our application, avoiding undue burden on field workers who must verify each potential target. Consequently, the unadjusted F-measure (`f_meas`) was retained as our preferred optimization metric.

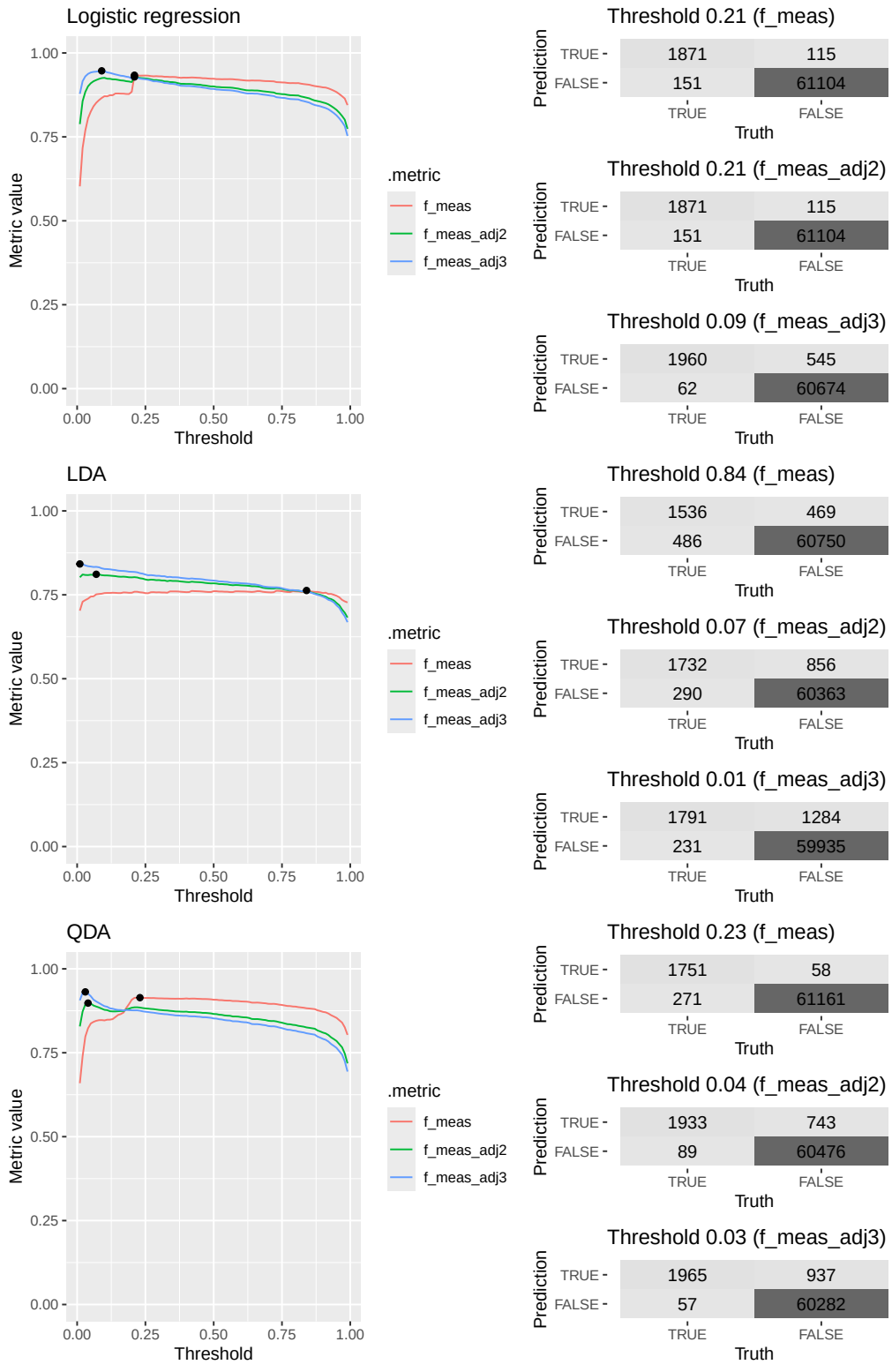


Figure 8: Metrics as a function of model performance

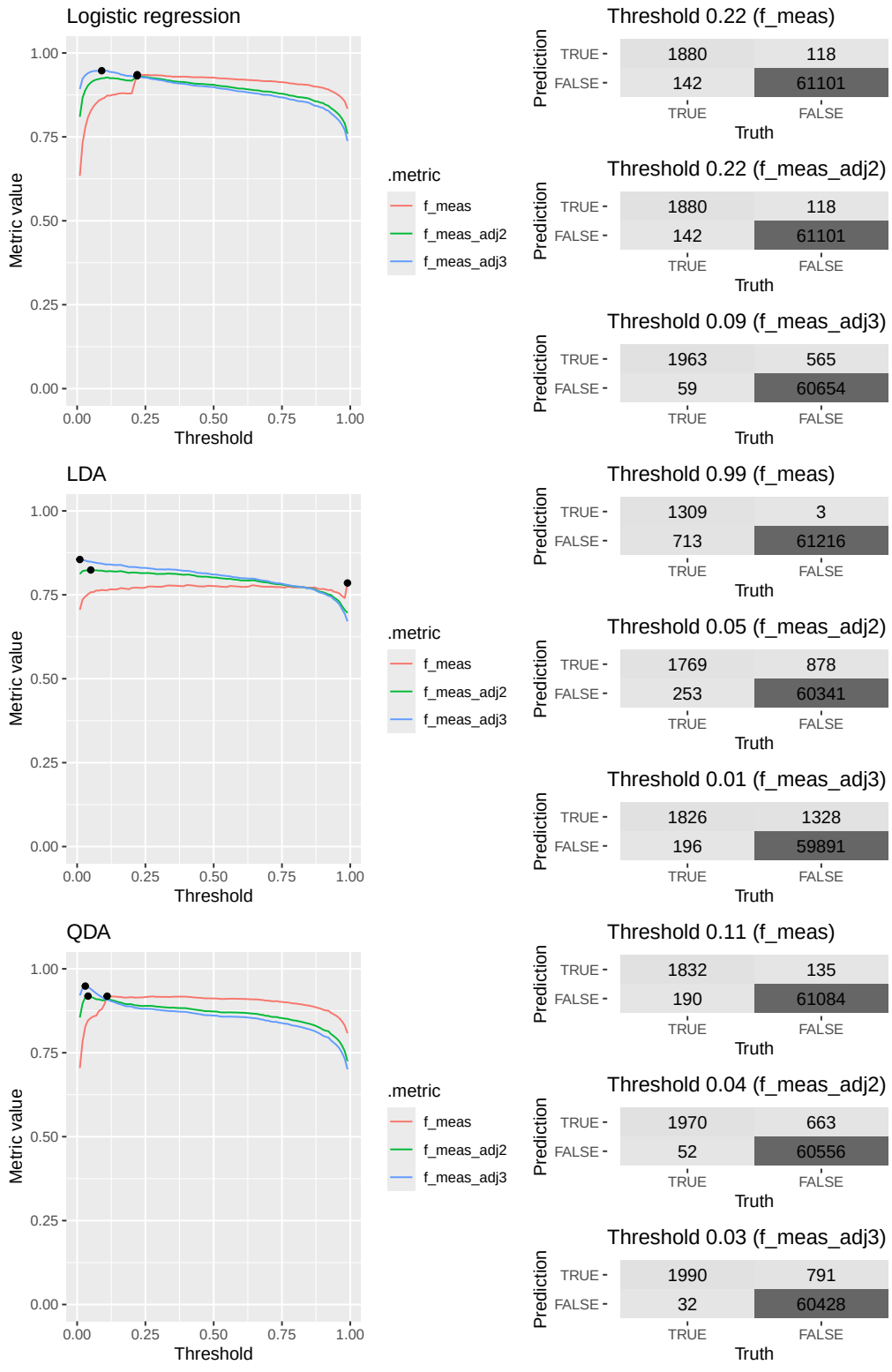


Figure 9: Metrics as a function of model performance

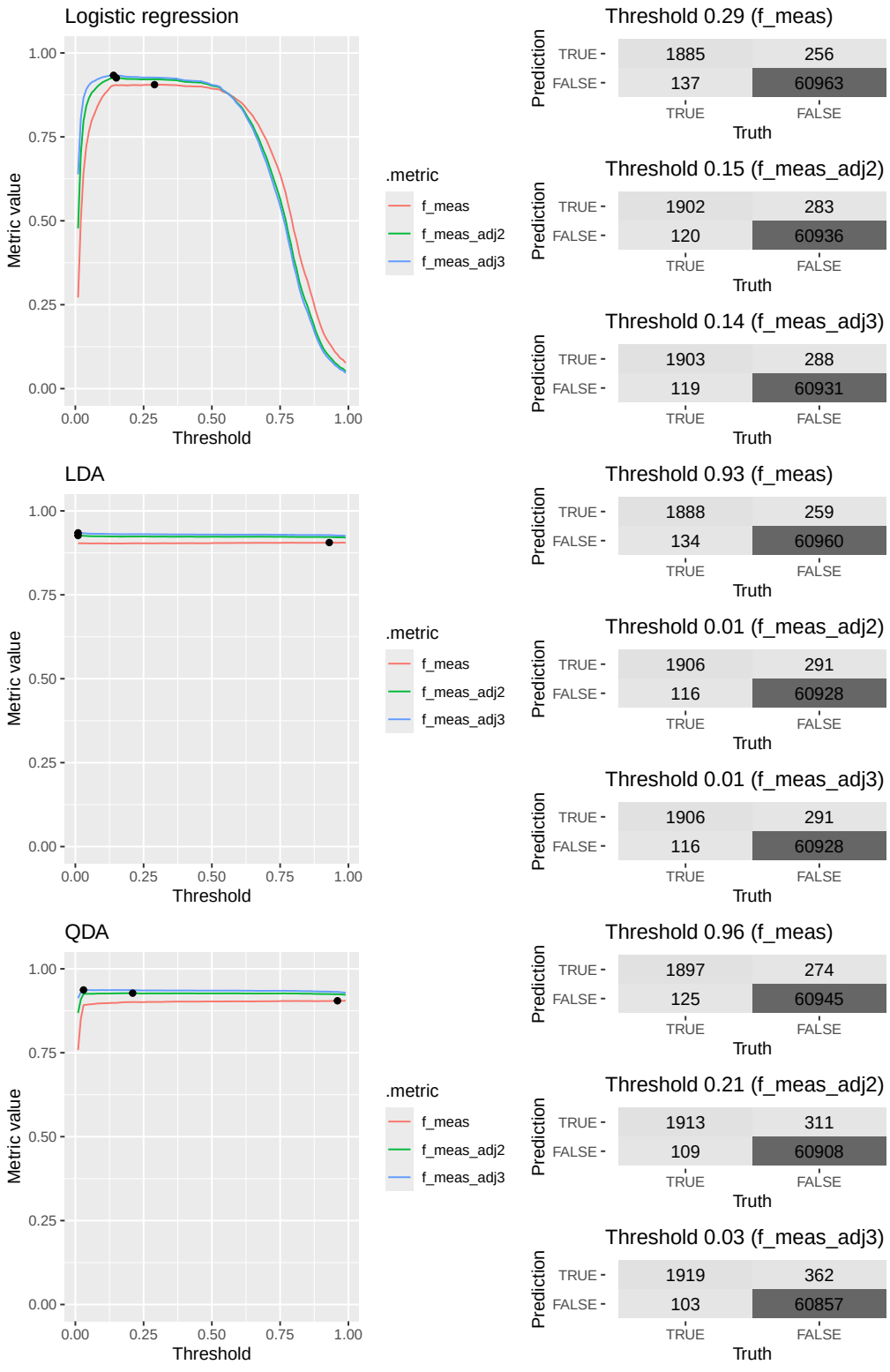


Figure 10: Metrics as a function of model performance

Final Model Fitting on Training Data

After evaluating performance in the cross-validation folds, each model was fit to the entire training dataset. The ROC-AUC curves for these full-training fits closely align with those obtained from the cross-validation predictions, suggesting that none of the nine models (logistic regression, LDA, or QDA in RGB, CIELab, or HSV) exhibits substantial overfitting at this stage. Each model's performance remains consistent when trained on the complete dataset.

These curves are nearly identical, reinforcing the conclusion that the additional training data did not inflate performance metrics in a way indicative of overfitting. On this basis, the final fitted models serve as reliable starting points for subsequent predictive analyses on the holdout set.

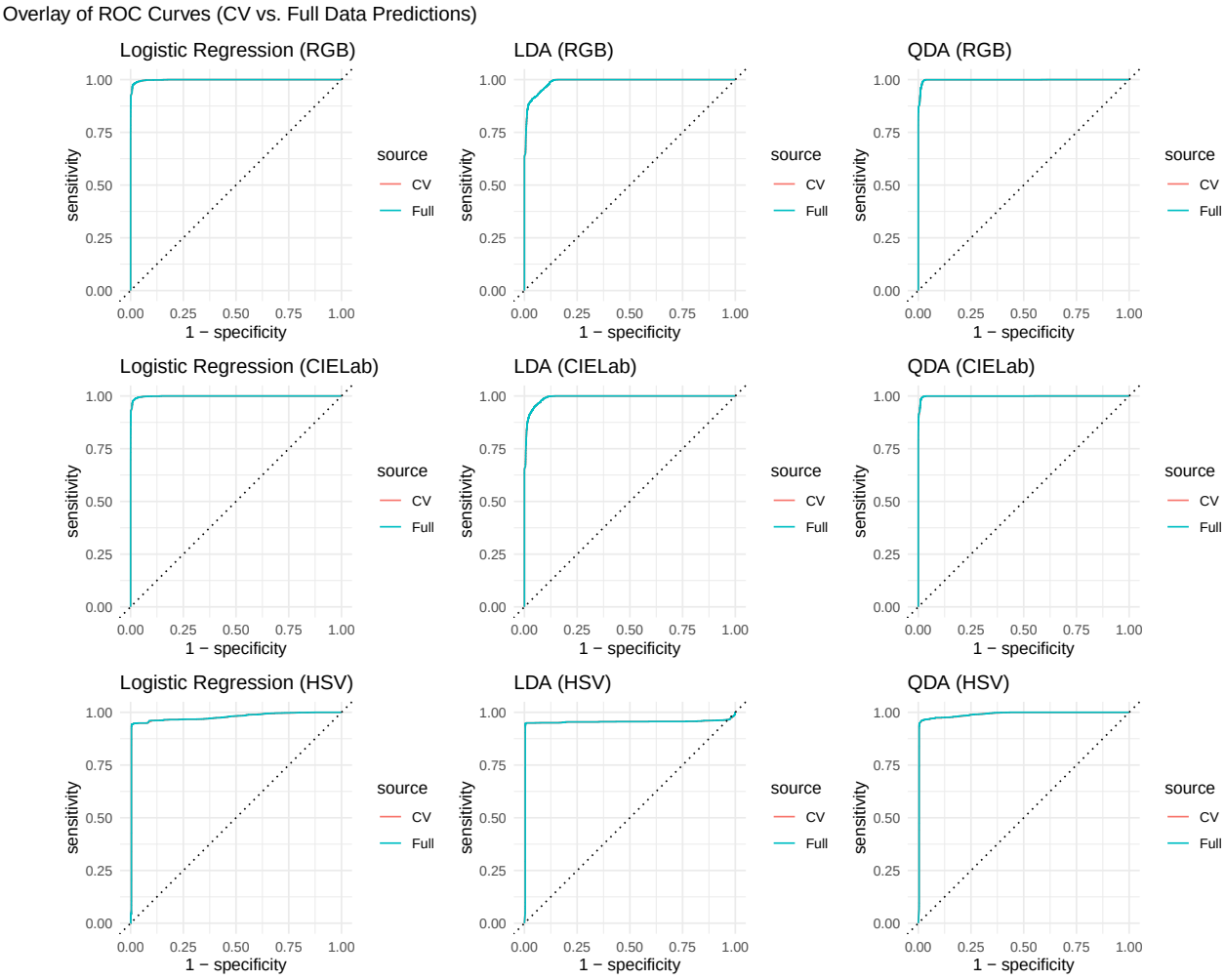


Figure 11: ROC curve comparison between cross-validation and full data set predictions

Alternate Presentation: Comparing ROC-AUC from Cross-Validation and Full Training While the previous figure overlays the ROC curves for cross-validation (CV) and full-training fits, an additional way to highlight potential overfitting is to compute the ROC-AUC under both conditions and compare the results directly in a table. This approach provides a numeric indication of whether the model's performance changes substantially when trained on the full dataset.

For each of the nine model–color space combinations: 1. Compute ROC-AUC from cross-validation predictions.
2. Fit the model on the full training dataset, then compute the ROC-AUC on the same dataset.
3. Calculate the difference between these two values.

A small difference implies little or no overfitting, indicating that training on the entire dataset does not artificially inflate performance metrics. The table confirms that the difference in the performance metrics between the mean of the cross-validated folds and the metric of a fitted model are miniscule, confirming little or no overfitting.

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

Table 2: Comparison of ROC-AUC between cross-validation and full-training fits

Color Space	Model	ROC-AUC of CV Folds	ROC-AUC of Fitted Model	Difference
RGB	Logistic Regression	0.998497	0.998507	0.000010
	LDA	0.988809	0.988877	0.000068
	QDA	0.998249	0.998218	-0.000032
CIELab	Logistic Regression	0.998688	0.998709	0.000021
	LDA	0.992141	0.992189	0.000047
	QDA	0.998803	0.998762	-0.000041
HSV	Logistic Regression	0.976863	0.976829	-0.000034
	LDA	0.952659	0.952656	-0.000003
	QDA	0.988999	0.989180	0.000181

Evaluation on the Holdout Set

After confirming that our models do not exhibit excessive overfitting when trained on the full dataset, we apply them to the holdout data. Each model–color space pair undergoes a threshold scan using the same `f_meas` metric chosen earlier. This process identifies the threshold on the holdout set that maximizes the F-measure.

The figure below displays the F-measure across thresholds for logistic regression, LDA, and QDA in each color space, illustrating how model performance changes with different cutoffs. We see that the RGB and CEILab logistic regression models outperform all other models with achieving the highest f-measure. We also see that for those two models, the relationship is linear between threshold and f-measure performance: as the threshold increases and the model only classifies those pixels that can be most confidently classified as being a blue-tarp are indeed classified, the model performance by the f-measure increases.

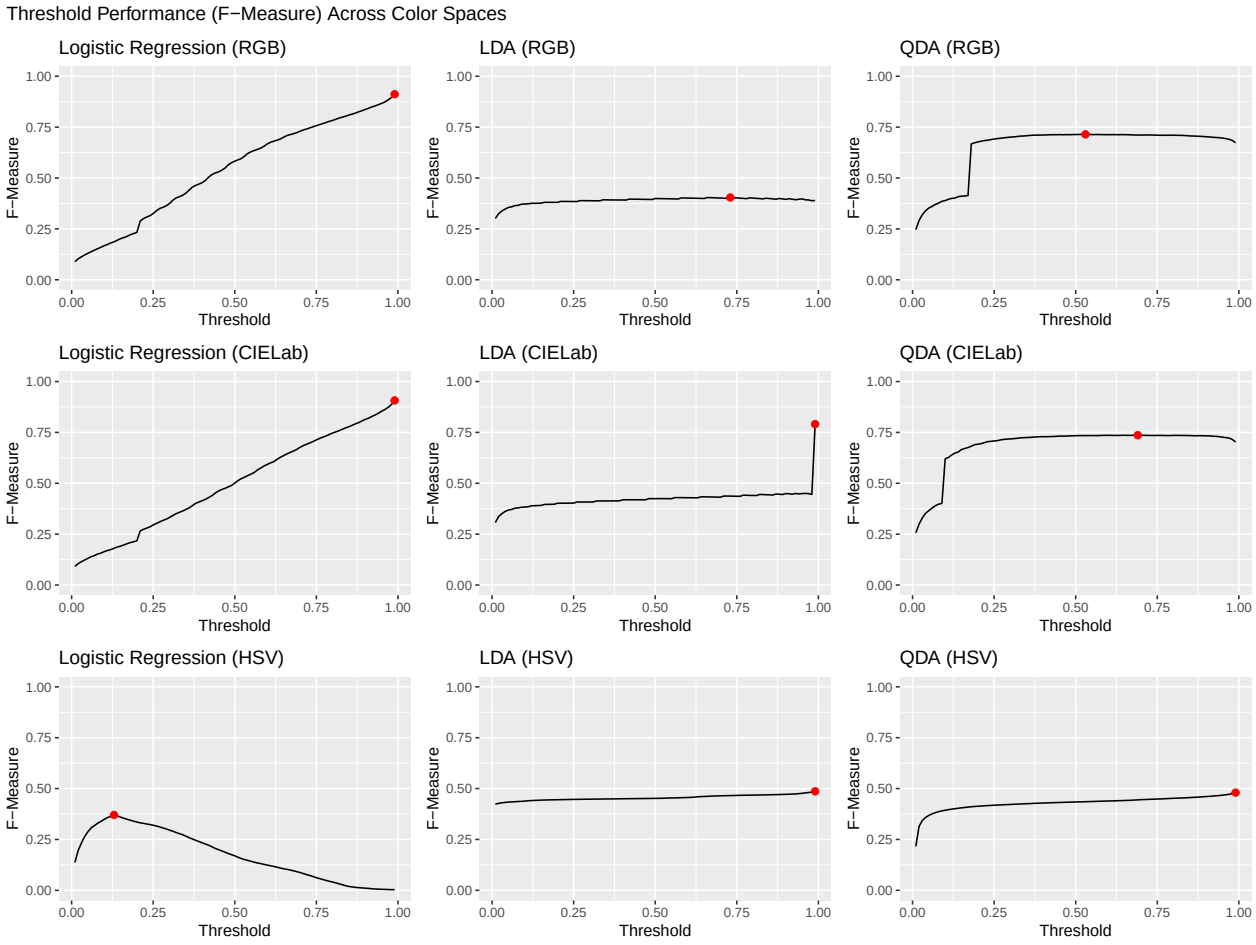


Figure 12: F-Measure by threshold for each model and color space

Once the optimal threshold is found on the holdout set, the final metrics are computed for both the training and holdout data using the optimal threshold determined in previous steps, allowing a direct comparison of how well each model transfers its learned classifications to unseen examples.

The table of performance metrics at the chosen thresholds to optimize f-measure demonstrates the final trade-offs each model achieves in terms of accuracy, F-measure, precision, and sensitivity for both the training and holdout sets. By examining these values, we can better understand the likely field performance of each model–color space combination.

Final Holdout F-Measure Results The table below summarizes the F-measure on the holdout set for each model–color space pair, along with each pair's chosen threshold optimized for F-measure.

- Logistic Regression (RGB) achieves the highest F-measure at 0.912 (threshold = 0.99).
- Logistic Regression (CIELab) follows closely at 0.907 (threshold = 0.99).
- QDA (RGB) and LDA (CIELab) have moderate F-measure values of 0.714 and 0.791, respectively.
- Most HSV models show notably lower F-measure, suggesting less effective balancing of precision and recall at their respective optimal thresholds.

Overall, logistic regression in RGB or CIELab provides the strongest combination of detecting actual blue tarps and minimizing false positives. This is crucial for rapid response and efficient resource deployment, as each false alarm requires manual verification and each missed blue tarp is potentially a missed person or family in need of aid.

Table 3: Final metrics for models at chosen thresholds.

dataset	color_space	model	threshold	accuracy	roc_auc	sens	f_meas
train	RGB	Logistic Regression	0.21	0.996	0.999	0.926	0.934
		LDA	0.84	0.985	0.989	0.759	0.763
		QDA	0.23	0.995	0.998	0.866	0.914
	CIELab	Logistic Regression	0.22	0.996	0.999	0.930	0.936
		LDA	0.41	0.985	0.992	0.832	0.779
		QDA	0.11	0.995	0.999	0.906	0.918
	HSV	Logistic Regression	0.29	0.994	0.977	0.932	0.906
		LDA	0.93	0.994	0.953	0.934	0.906
		QDA	0.81	0.994	0.989	0.942	0.904
holdout	RGB	Logistic Regression	0.99	0.999	0.999	0.953	0.912
		LDA	0.73	0.983	0.992	0.797	0.404
		QDA	0.53	0.996	0.992	0.689	0.714
	CIELab	Logistic Regression	0.99	0.999	0.999	0.956	0.907
		LDA	0.99	0.997	0.995	0.672	0.791
		QDA	0.69	0.996	0.993	0.691	0.736
	HSV	Logistic Regression	0.13	0.981	0.984	0.763	0.370
		LDA	0.99	0.985	0.985	0.991	0.486
		QDA	0.99	0.984	0.990	0.993	0.480

Confusion Matrices on the Holdout Set The figures below show the confusion matrices for each model–color space pair on the holdout set, evaluated at their respective F-measure–optimized thresholds. Each matrix indicates how many pixels were predicted as blue tarps and how many were actually blue tarps (`BT = TRUE`) versus not blue tarps (`BT = FALSE`).

A desirable matrix has high counts along the top-left to bottom-right diagonal, indicating correct classifications. The top-right entry in each matrix (false positives) corresponds to precision: fewer false positives means a higher precision. The bottom-left entry (false negatives) corresponds to recall: fewer missed blue tarps means higher recall.

In RGB and CIELab, logistic regression shows a balanced trade-off between precision and recall, resulting in relatively low false positives and few missed detections. This balance is essential for real-world deployments where each false positive demands field checks or additional remote-sensing analysis, while each missed tarp undermines the humanitarian response. The confusion matrices confirm that these two configurations (logistic regression in RGB or CIELab) provide strong overall performance at their ideal thresholds.

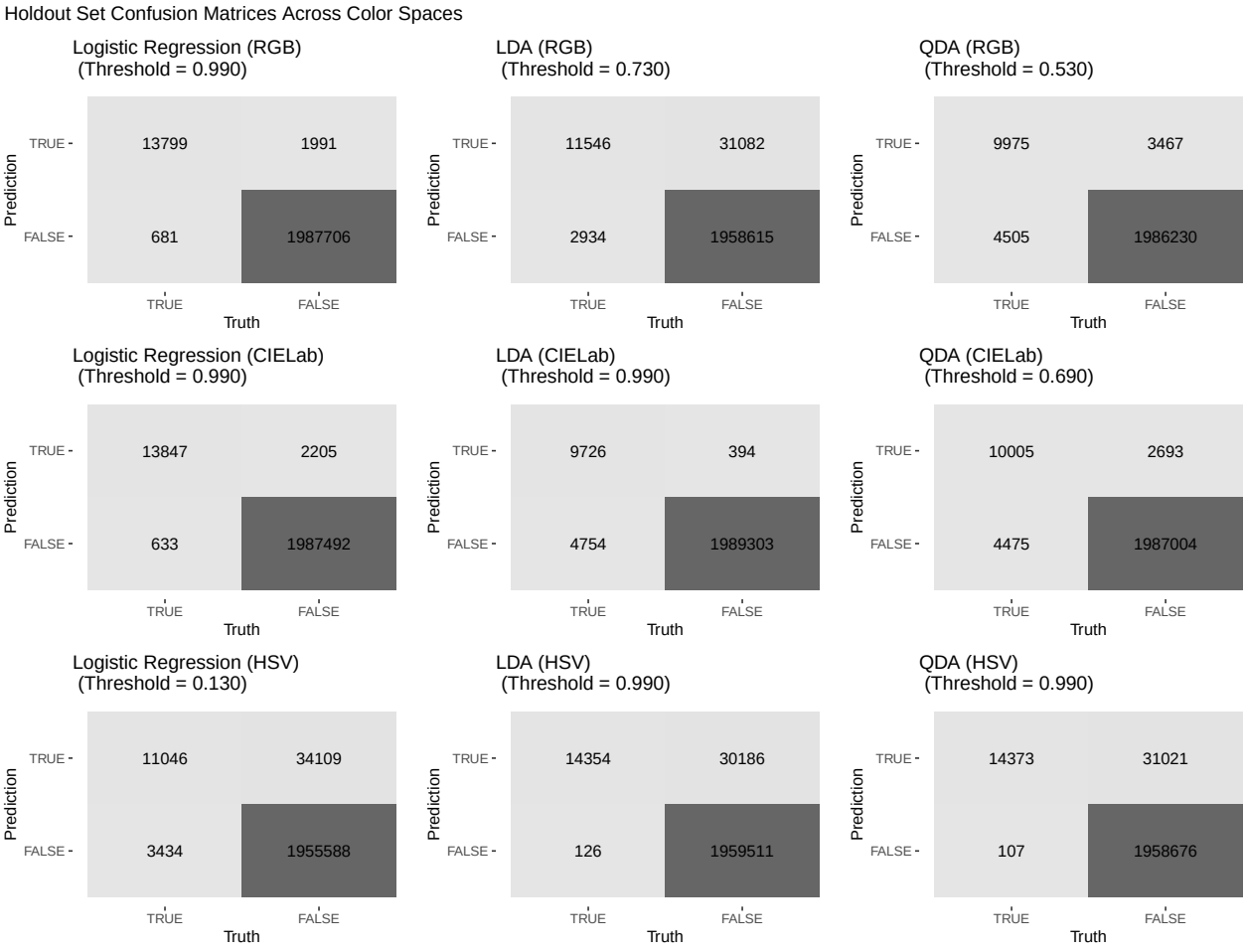


Figure 13: Holdout Set Confusion Matrices by Color Space

Threshold Fine-Tuning In our initial threshold search with increments of 0.01, the optimal thresholds for the two best models (logistic regression in RGB and CIELab) reached 0.99, the upper limit of the search range. To assess whether a more precise threshold could further improve performance, we conducted a refined search over the interval [0.99, 1] in increments of 0.001. This finer scan allows us to determine if a marginal increase in the threshold yields a significant improvement in the F-measure.

We implement this fine threshold search for the logistic regression models in RGB and CIELab. We then extract the optimal thresholds and compute performance metrics at these refined thresholds.

Analysis of Fine-Tuned Threshold Performance The results indicate that the optimal threshold on the holdout data is considerably higher than on the training data. This is intuitive because, on unseen data, the model must be much more certain that a pixel is a blue tarp before assigning it a positive classification; the training data primarily defines the color profiles, while the holdout data requires a more conservative decision rule.

Furthermore, the refined threshold scan—from 0.99 to 1 in 0.001 increments—shows that the logistic regression model using the CIELab color space achieves a slightly higher F-measure at its optimal threshold than the corresponding RGB model. Although the performance differences are marginal, this suggests that the CIELab representation may capture subtle variations in blue tarp appearance more effectively than RGB. Overall, however, the two models are very similar in performance.

Table 4: Performance metrics for models at ideal threshold (fine tuning).

dataset	color_space	model	threshold	accuracy	roc_auc	sens	f_meas
train	RGB	Logistic Regression	0.210	0.996	0.999	0.926	0.934
	CIELab	Logistic Regression	0.220	0.996	0.999	0.930	0.936
holdout	RGB	Logistic Regression	0.997	0.999	0.999	0.934	0.923
	CIELab	Logistic Regression	0.998	0.999	0.999	0.933	0.925

Final Confusion Matrix Analysis The final confusion matrices indicate that, to achieve a slightly improved F-measure, the logistic regression model in the CIELab color space produces 95 fewer false positives compared to its RGB counterpart. However, this gain is accompanied by an increase of 19 false negatives (i.e. 976 versus 957 missed blue tarps). Although these differences are numerically small, each misclassification corresponds to a pixel that may have significant real-world implications for blue tarp detection in humanitarian aid applications.