Disaster Relief Project Part 2

Group 13: Sarah Christen, Katherine Kelleher, Margaret Lindsay, and Brian Nolton

Introduction

Background and Motivation

- 2010 earthquake in Haiti caused widespread displacement
- Displaced individuals often used blue tarps as temporary shelters
- Blue tarps were easily visible in aerial imagery
- Rochester Institute of Technology gathered high-resolution images of affected areas

Challenge

- Large volume of imagery made manual identification of shelters difficult
- Urgent need to quickly locate and assist displaced individuals

Project Goal

- Develop statistical models to automatically identify blue tarps in aerial images

Exploratory Data Analysis

(1 of 6)

- Within the training data, the number of observations for each pixel class varied
- The training data predominately contained pixels classified as vegetation, soil, or rooftop
- The dataset contained the fewest observations in the class “Blue Tarp,” indicating an imbalance dataset between the blue tarp class and other classes

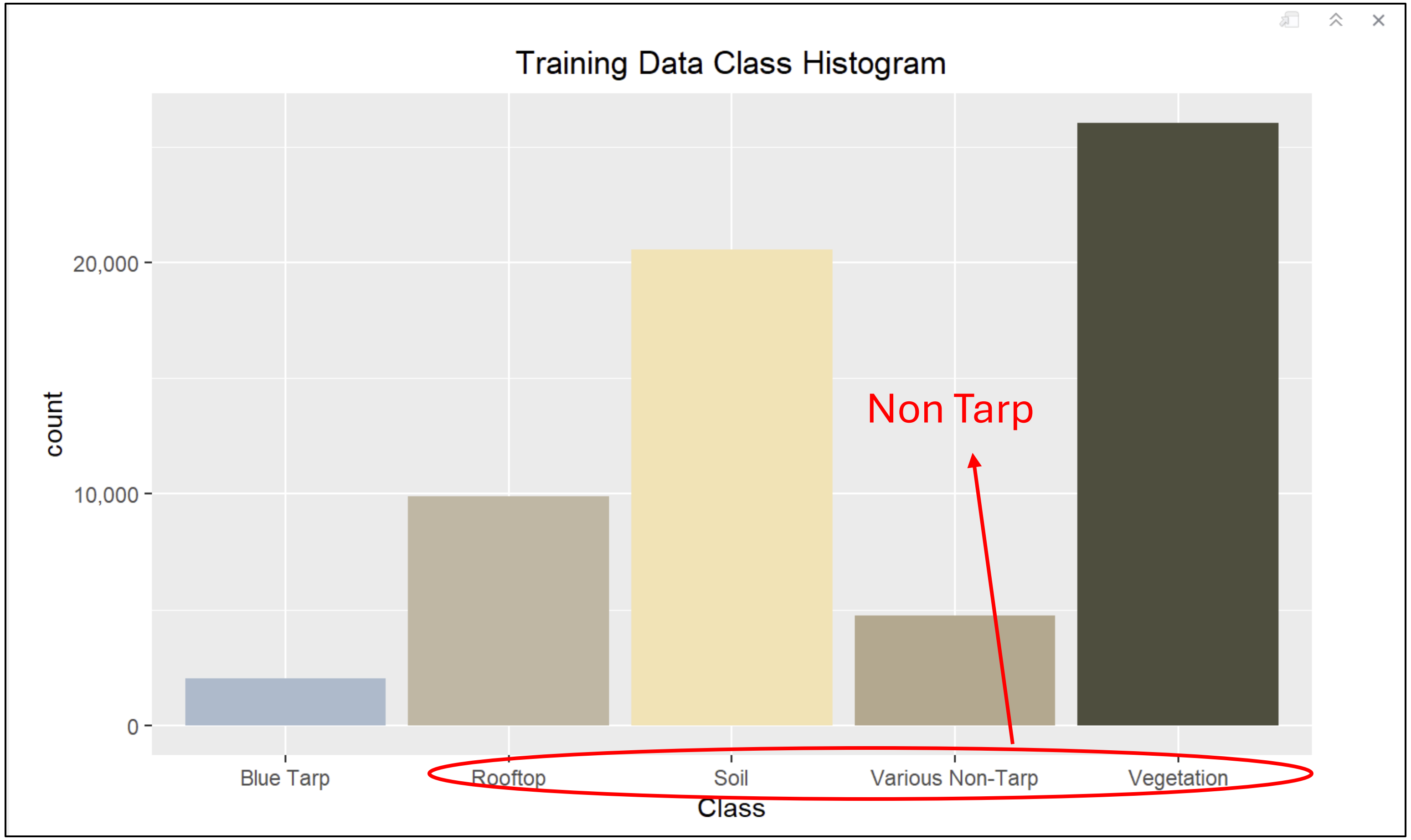


Figure 1. Count of Training Data by Pixel Class

Exploratory Data Analysis

(2 of 6)

- Both the training and holdout datasets are imbalanced
- The holdout dataset has a significantly larger number of observations than the training dataset
- There are significantly more Non_Tarp data points than Blue_Tarp data points in both datasets, which is consistent with the nature of data collected since blue tarps would only represent a small fraction of the aerial images captured

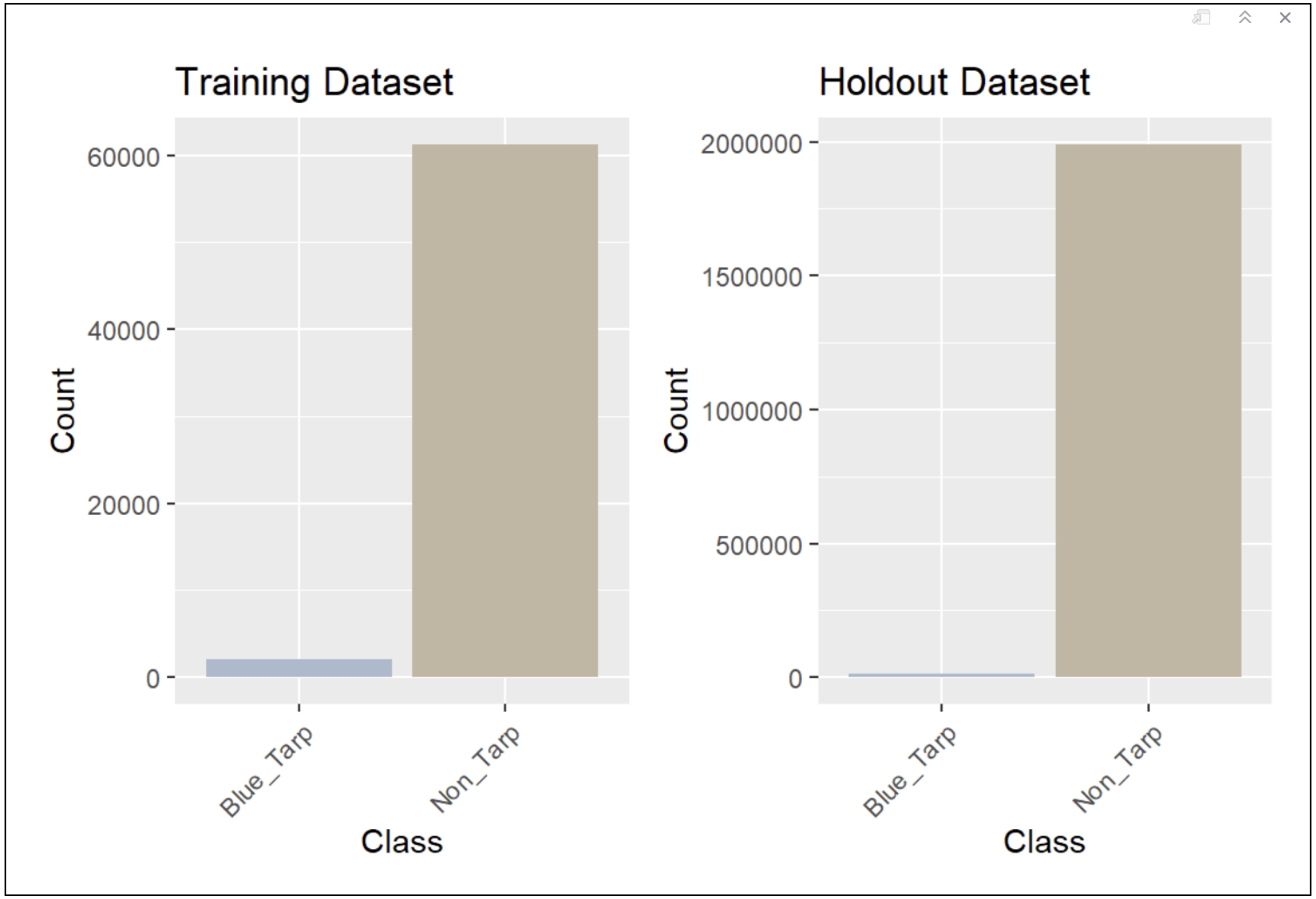


Figure 2. Count of Tarp and Non-tarp Pixel Class by Training Dataset and Holdout Dataset

Exploratory Data Analysis

(3 of 6)

- Training and test datasets had trends across values for colors for tarp and non-tarp data
  - Red had the lowest, green had middle, a blue had the highest median value for blue tarp data
  - Median values for non-tarp data had an opposite trend (decreasing values from red to green to blue)

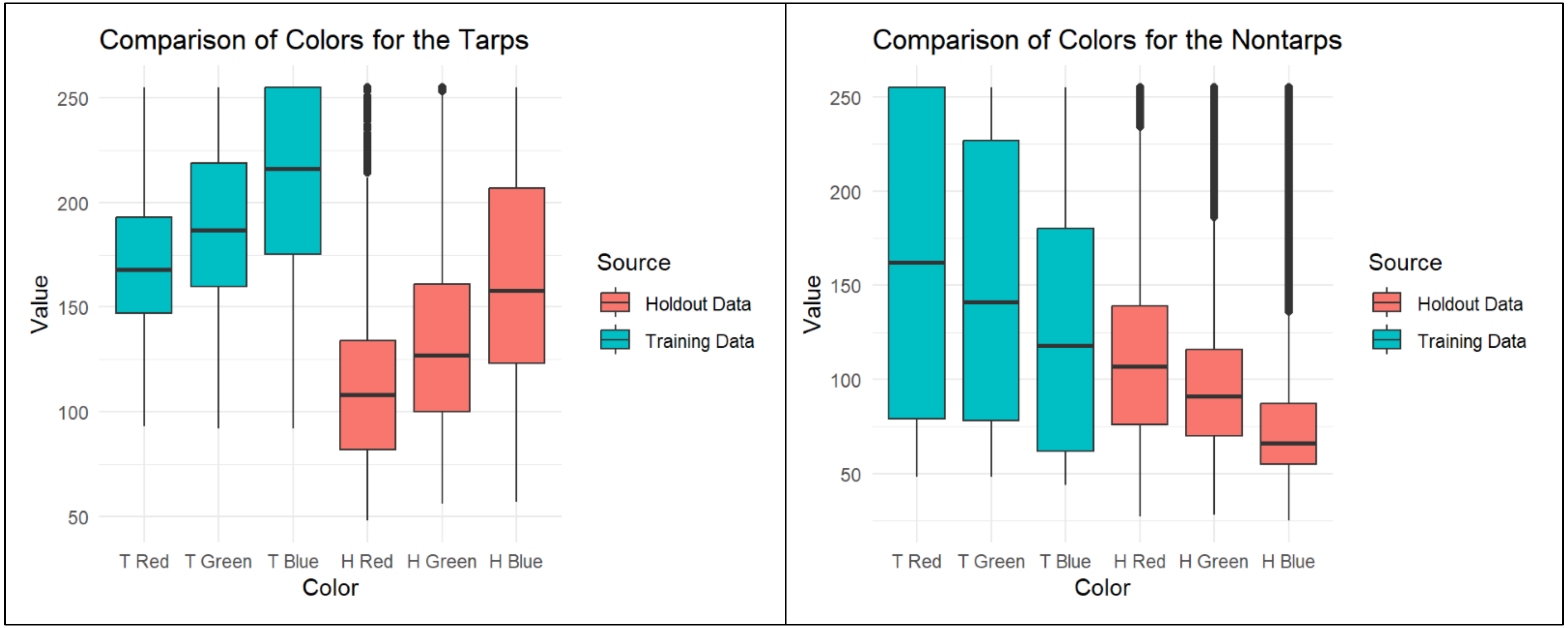


Figure 3. Boxplots of Red, Green, Blue Variables in Training and Holdout Datasets For Tarp and Non-Tarp Pixels, Picture

Exploratory Data Analysis

(4 of 6)

- Variance in median hue values present within a single class (e.g., color) across tarp and non-tarp datasets
  - Variance is observable across all colors in training and holdout data
  - Variance greater in non-tarp data, which contained a greater variety of pixels, compared to blue tarp
  - Variance is also greater between median values in training dataset, which is smaller
  - Differences more easily explained in the holdout data, due to the large sample size

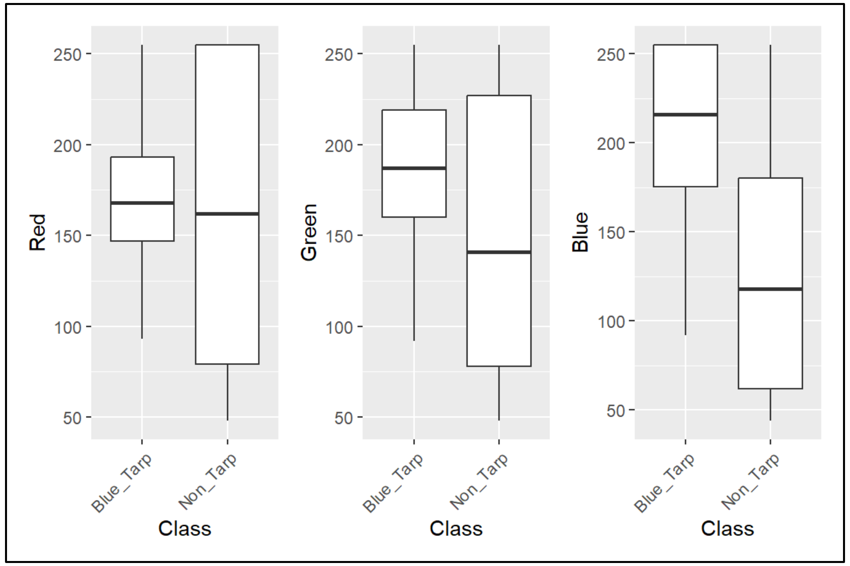


Figure 4. Boxplots of Categorical Variables for Training Data

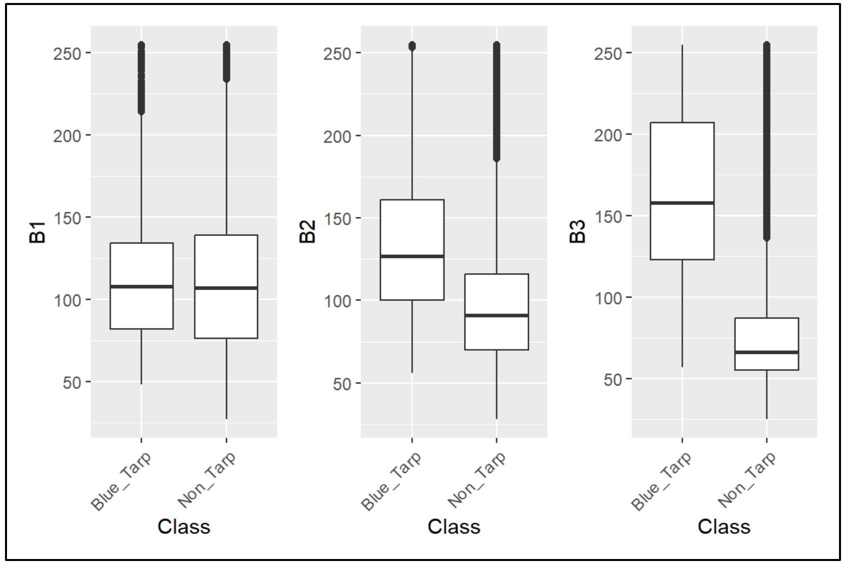


Figure 5. Boxplots of Categorical Variables for Holdout Data

Exploratory Data Analysis

(5 of 6)

- The color hues for vegetation in the training data tended to be lower in value and have less variation than the hues captured for rooftop, soil, and blue tarp observations
- Blue tarp pixels with higher color hue values tended to have more variation in their associated red or green color hue values

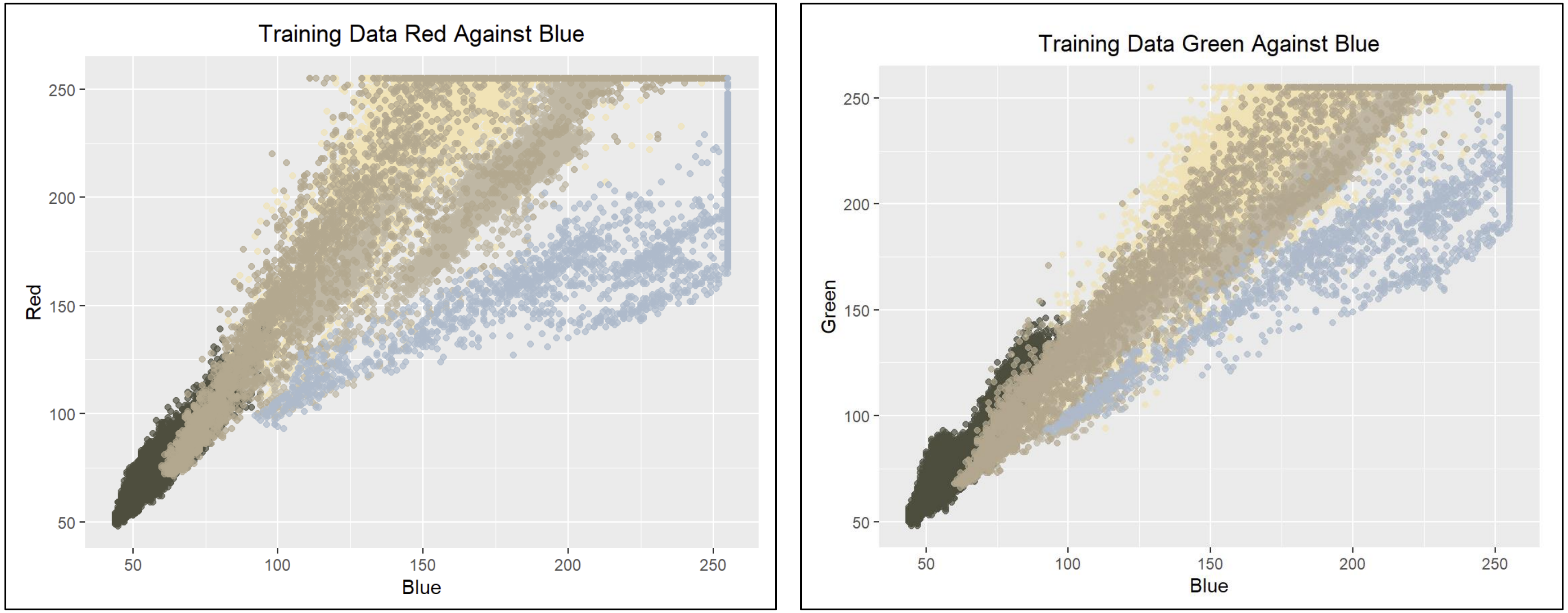
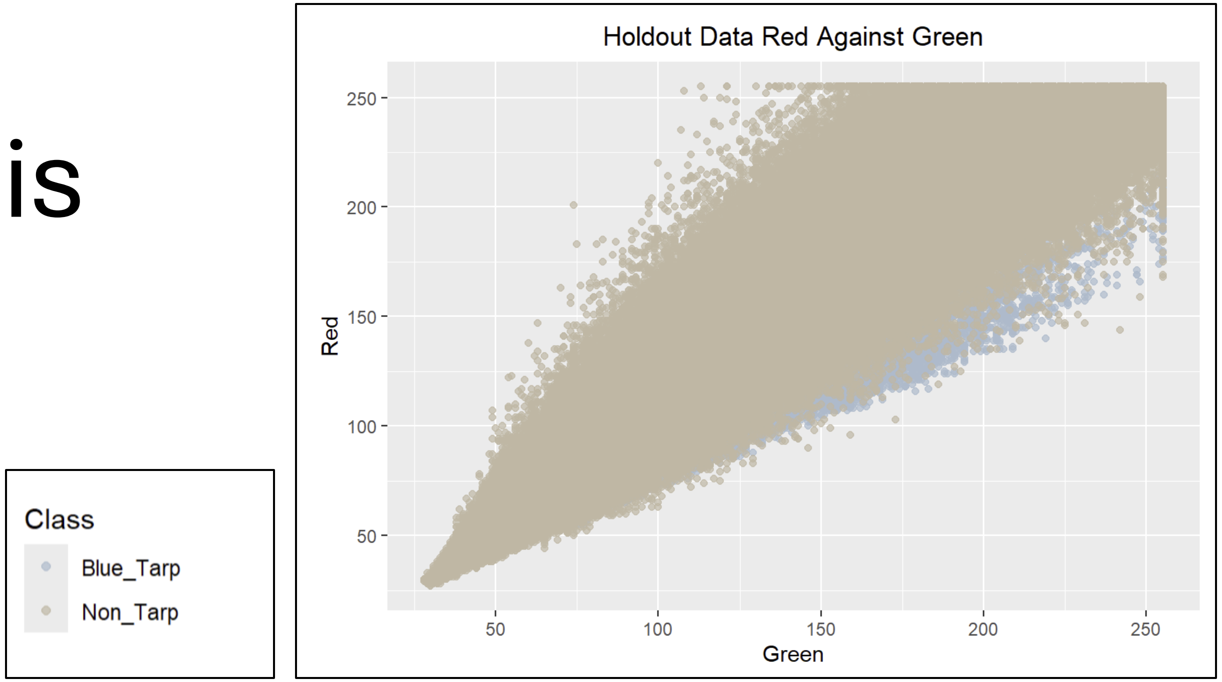


Figure 6. Scatterplot Distribution of Pixel Class in Training Data by RGB Color Values

Exploratory Data Analysis

(6 of 6)

- For non-tarp pixels in the holdout data, the variation between the value of color hues gets larger as the hue value gets larger
- For the holdout data, more variation in hue value occurs for blue_tarp data for all color classes (red, green, and blue) compared to the training data, which makes sense due to the larger sample size
- In the holdout data, blue-tarp observations tended to have higher hue values for the color class blue than red or green



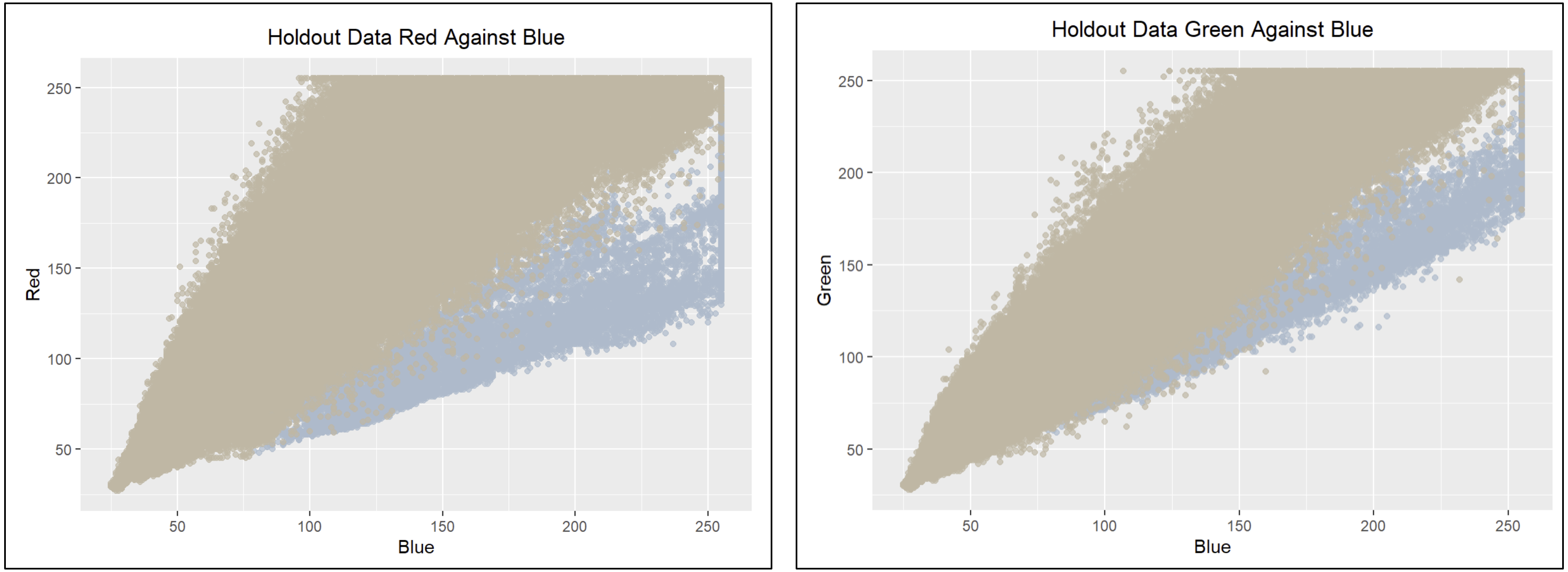


Figure 7. Scatterplot Distribution of Pixel Class in Holdout Data by RGB Color Values

Description of Methodology

- Utilized R, tidymodels, and tidyverse
- Addressed classification problem by fitting, tuning, and comparing model metrics across 10 different models
- Built and trained each model on a "training" dataset and tested model fit by measuring ROC curves, confusion matrices, and performance metrics on a "holdout" dataset
  - Assessed accuracy, TPR/sensitivity, false positive rate (FPR), precision ROC AUC, and F-measure
- Emphasized F-measure and sensitivity due to the imbalanced datasets and performed 10-fold cross validation to choose thresholds and tuning parameters

Models without tuning parameters

Logistic Regression

LDA (Linear Discriminant Analysis)

QDA (Quadratic Discriminant Analysis)

Thresholds

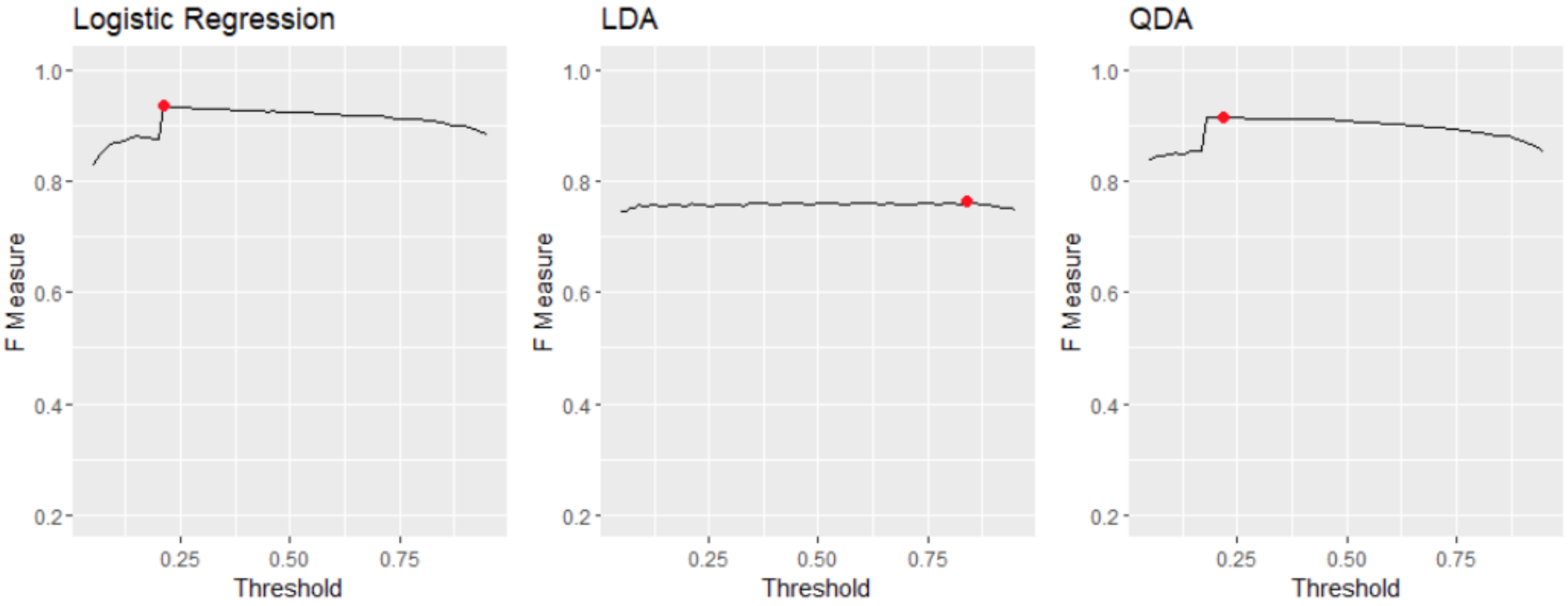


Figure 8. Threshold Scan of Each Classification Model using the F-Measure

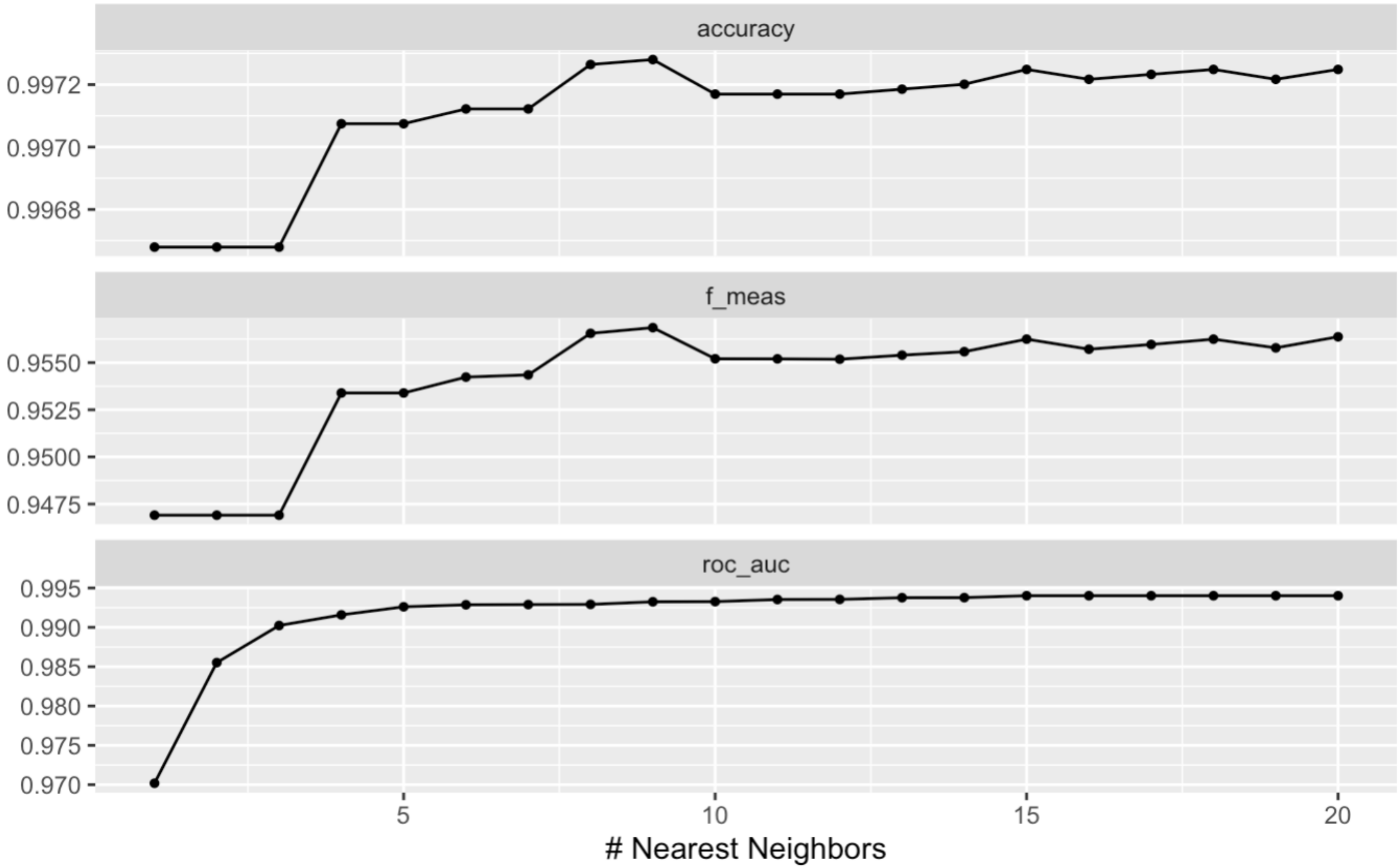
Models with tuning parameters

- K-nearest neighbor
- Penalized Logistic Regression
- Random Forest
- XGBoost
- Support Vector Machines
  - linear
  - polynomial
  - radial basis function

K-nearest Neighbor

Tuning parameters:

- Neighbors



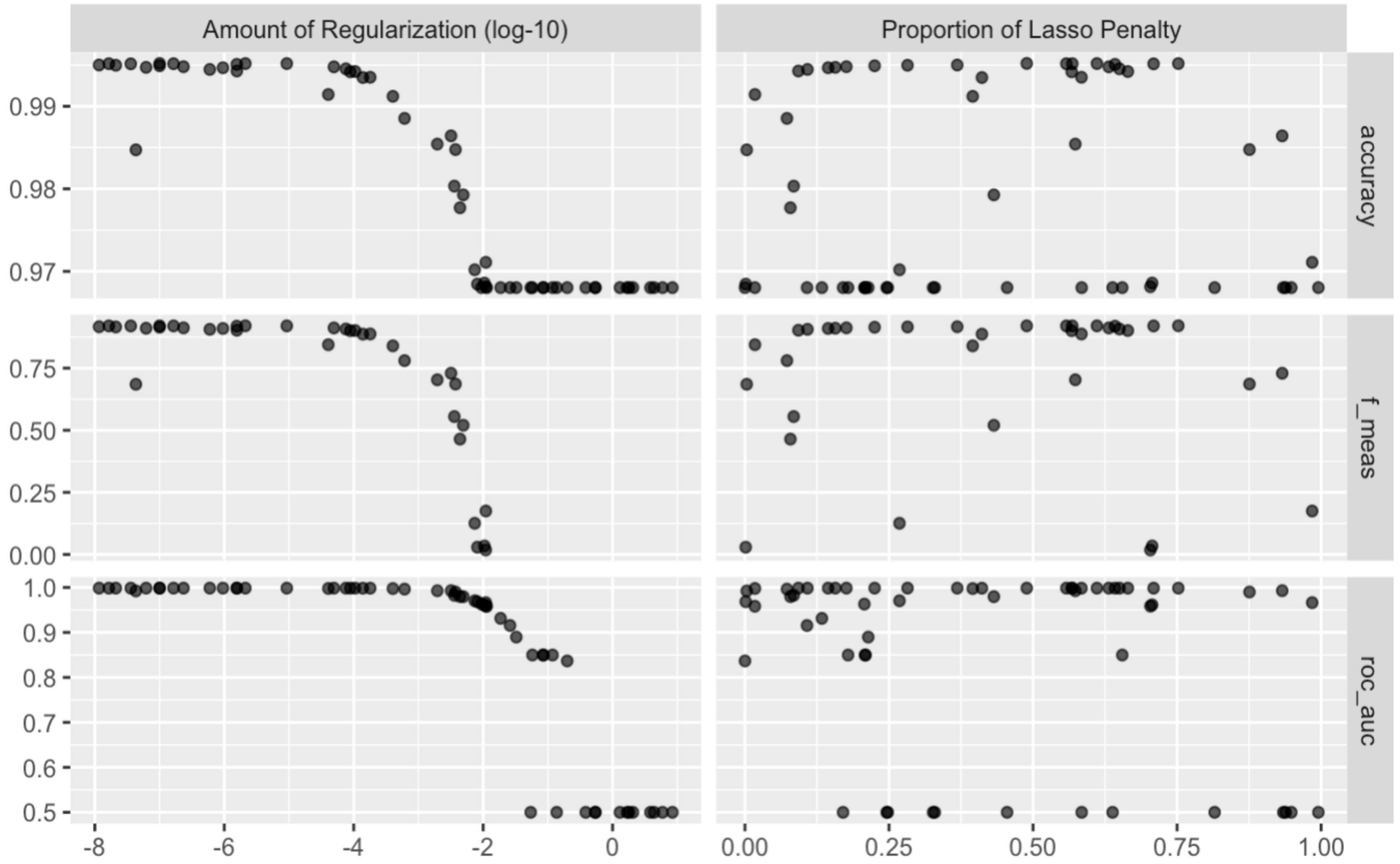
Neighbors: 9

Figure 9. Plot of Neighbor Tuning Parameter

Penalized Logistic Regression

Tuning parameters

- Penalty
- Mixture



Penalty = 0.00000212

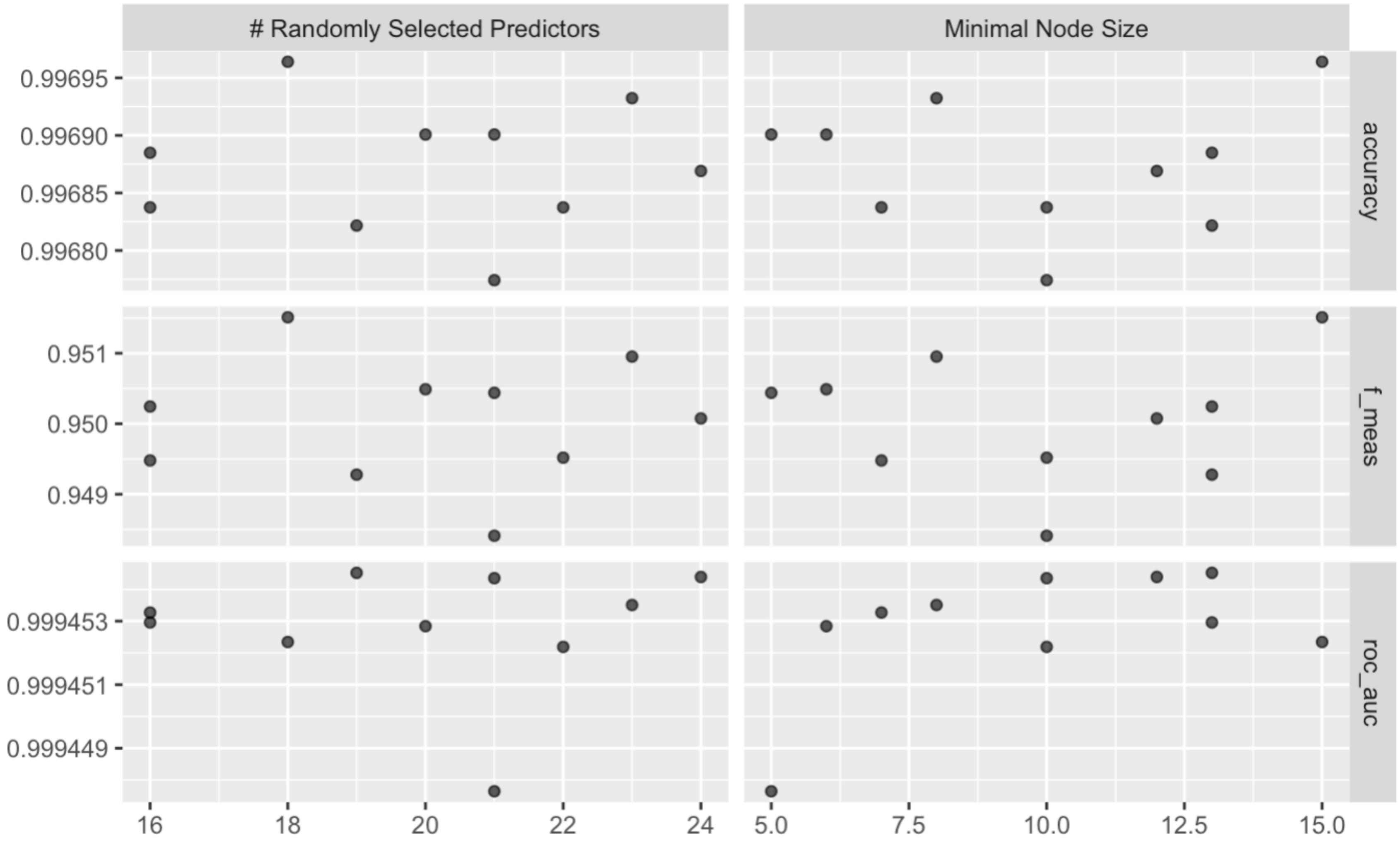
Mixture = 0.155556

Figure 10. Plot of Penalty and Mixture Tuning Parameters

Random Forest

Tuning Parameters

- Mtry
- Min_n



mtry = 18

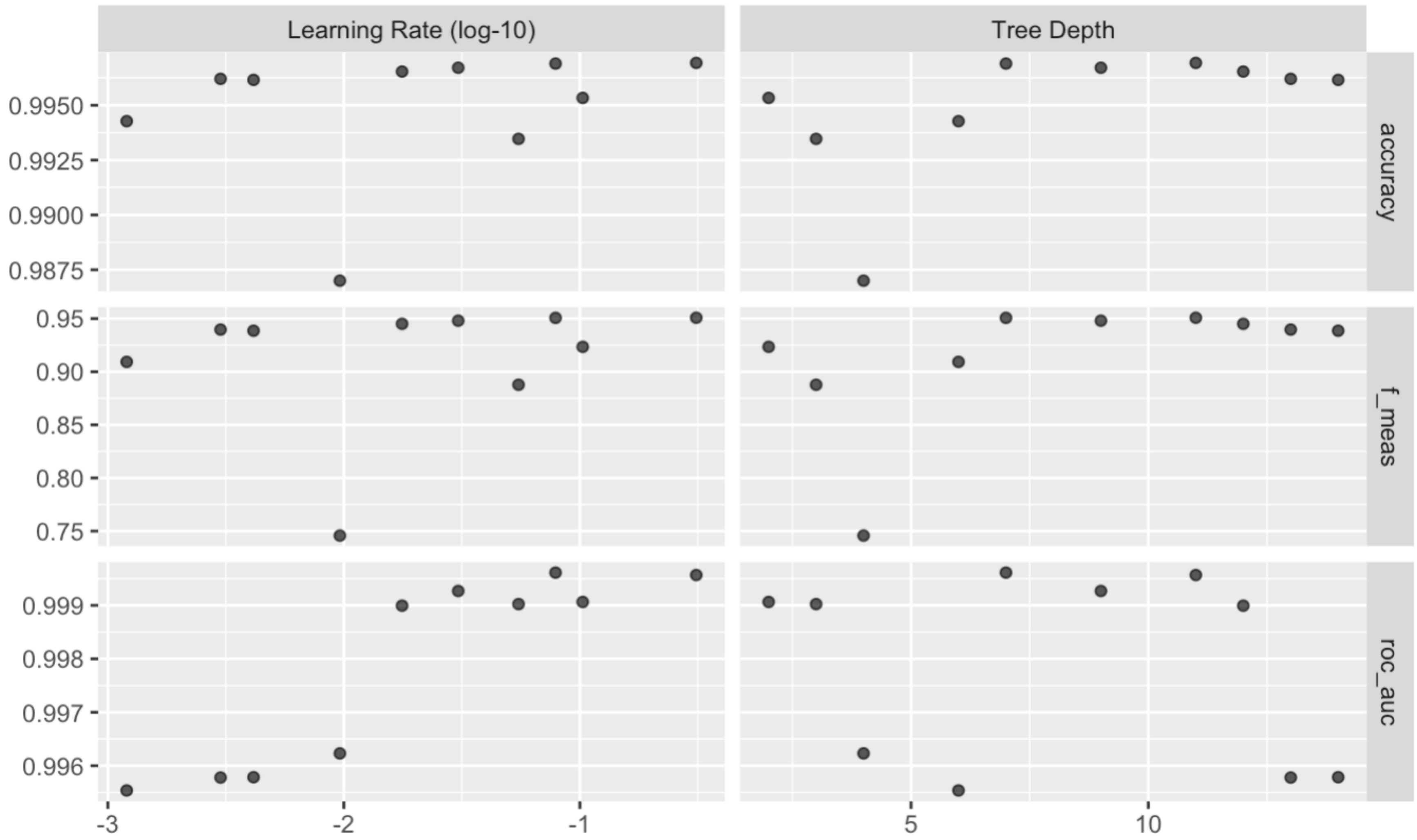
min_n = 15

Figure 11. Plot of Number of Predictor Variables Sampled ("mtry") and Minimal Node Size ("min_n") Tuning Parameters for Random Forest Model

Boosting

Tuning Parameters

- Mtry
- Min_n



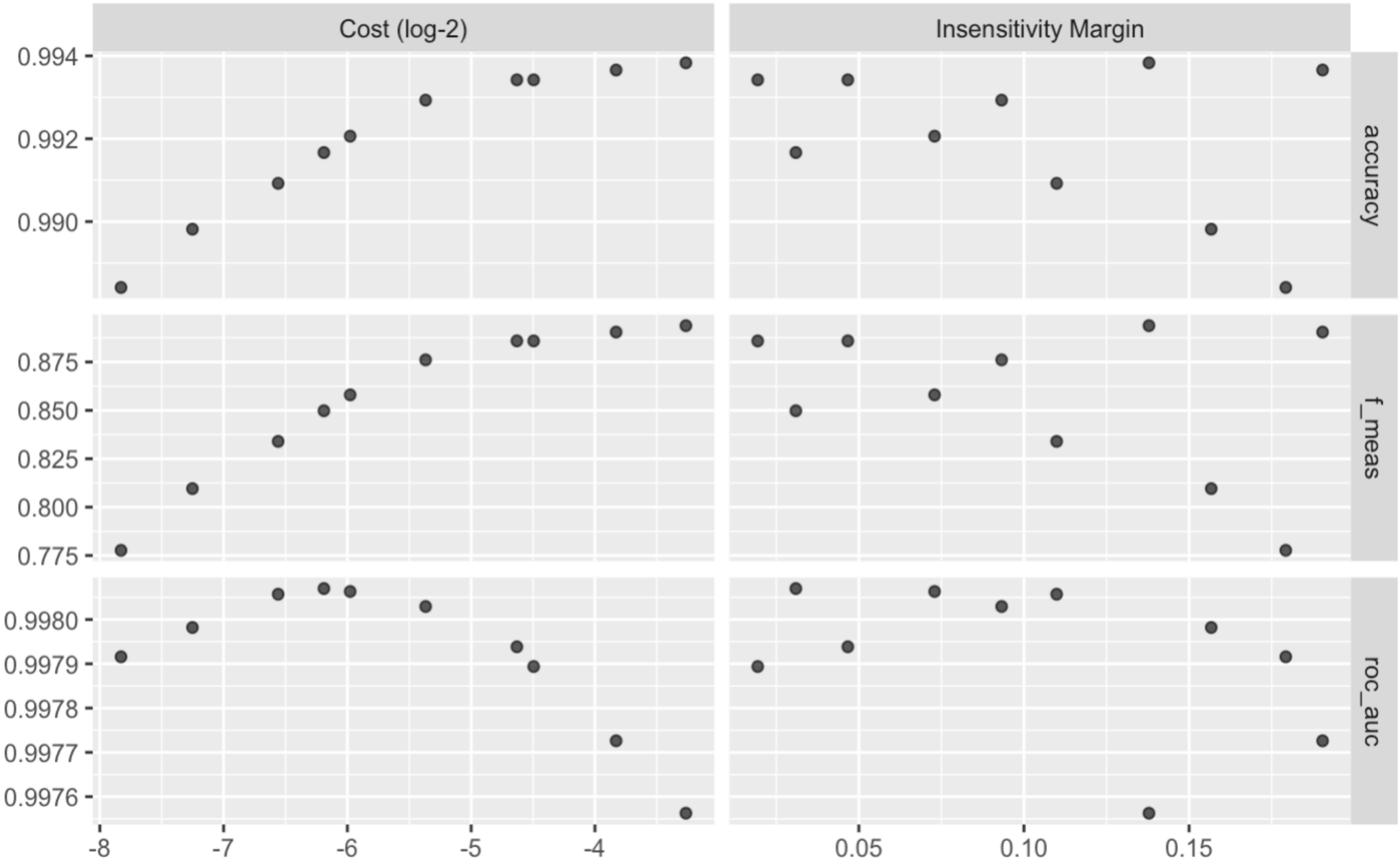
Learn_rate = 0.311 Tree_depth = 11

Figure 12. Plot of Number of Predictor Variables Sampled (“mtry”) and Minimal Node Size (“min_n”) Tuning Parameters for Boosting Model

Support Vector Machine – Linear Kernel

Tuning Parameters

- Cost
- Margin



Cost = 0.104

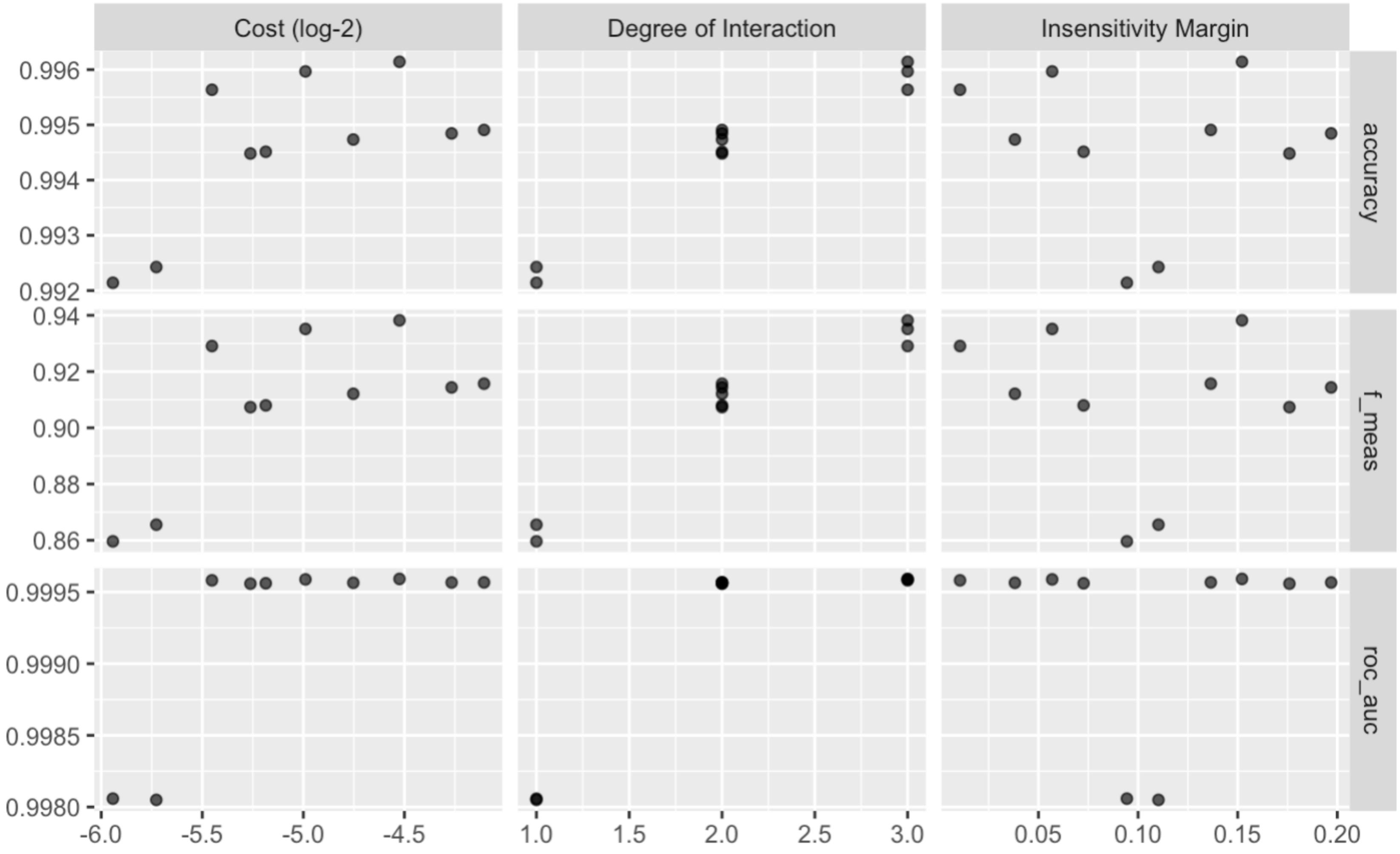
Margin = 0.138

Figure 13. Plot of Number of Cost and Insensitivity Margin Tuning Parameters for SVM Linear Model

Support Vector Machine – Polynomial Kernel

Tuning Parameters

- Cost
- Margin
- Degree



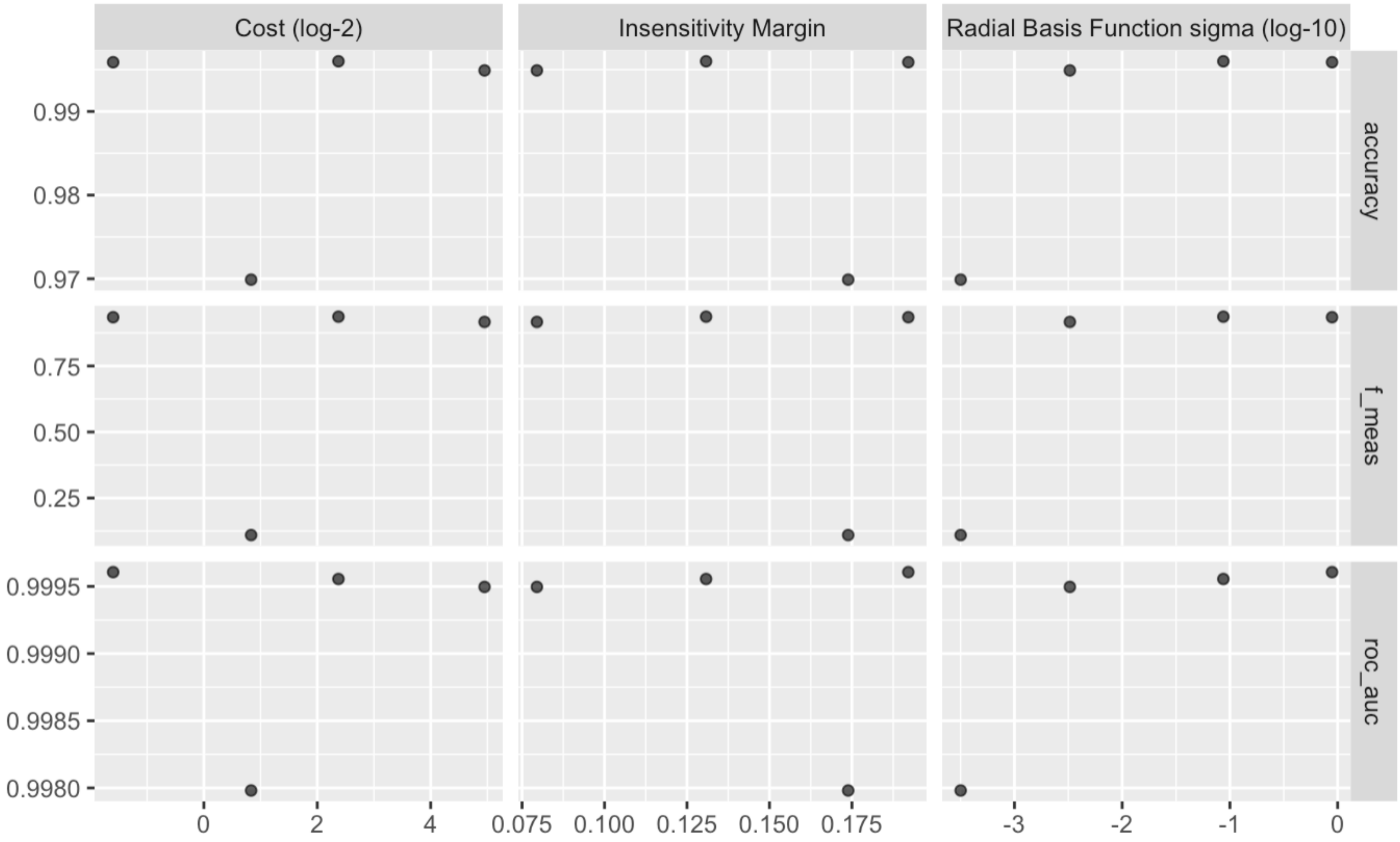
Cost = 0.0435 Degree = 3 Margin = 0.152

Figure 14. Plot of Number of Cost, Degree of Interaction, and Insensitivity Margin Tuning Parameters for SVM Polynomial Model

Support Vector Machine – Radial Basis Function Kernel

Tuning Parameters

- Cost
- Margin
- Rbf_sigma



Cost = 5.18225 Margin = 0.1307902 RBF sigma = 0.08695213

Figure 15. Plot of Number of Cost, Insensitivity Margin, and Inverse Kernel Width ("RBF_sigma") Tuning Parameters for SVM Polynomial Model

Cross-Validation

CV Metrics

model	f_meas	roc_auc	accuracy
Logistic Regression	0.933	0.998	0.996
LDA	0.761	0.989	0.985
QDA	0.914	0.998	0.995
Tuned Logistic Regression	0.952	0.999	0.997
Tuned KNN	0.957	0.993	0.997
Tuned Random Forest	0.952	0.999	0.997
Tuned Boost	0.954	0.999	0.997
Tuned Linear SVM	0.888	0.998	0.993
Tuned Polynomial SVM	0.936	1.000	0.996
Tuned RBF SVM	0.943	1.000	0.996

Table 1. Cross-validation Performance Metrics

Best performance for metric

Overall best model performance

CV ROC Graphs

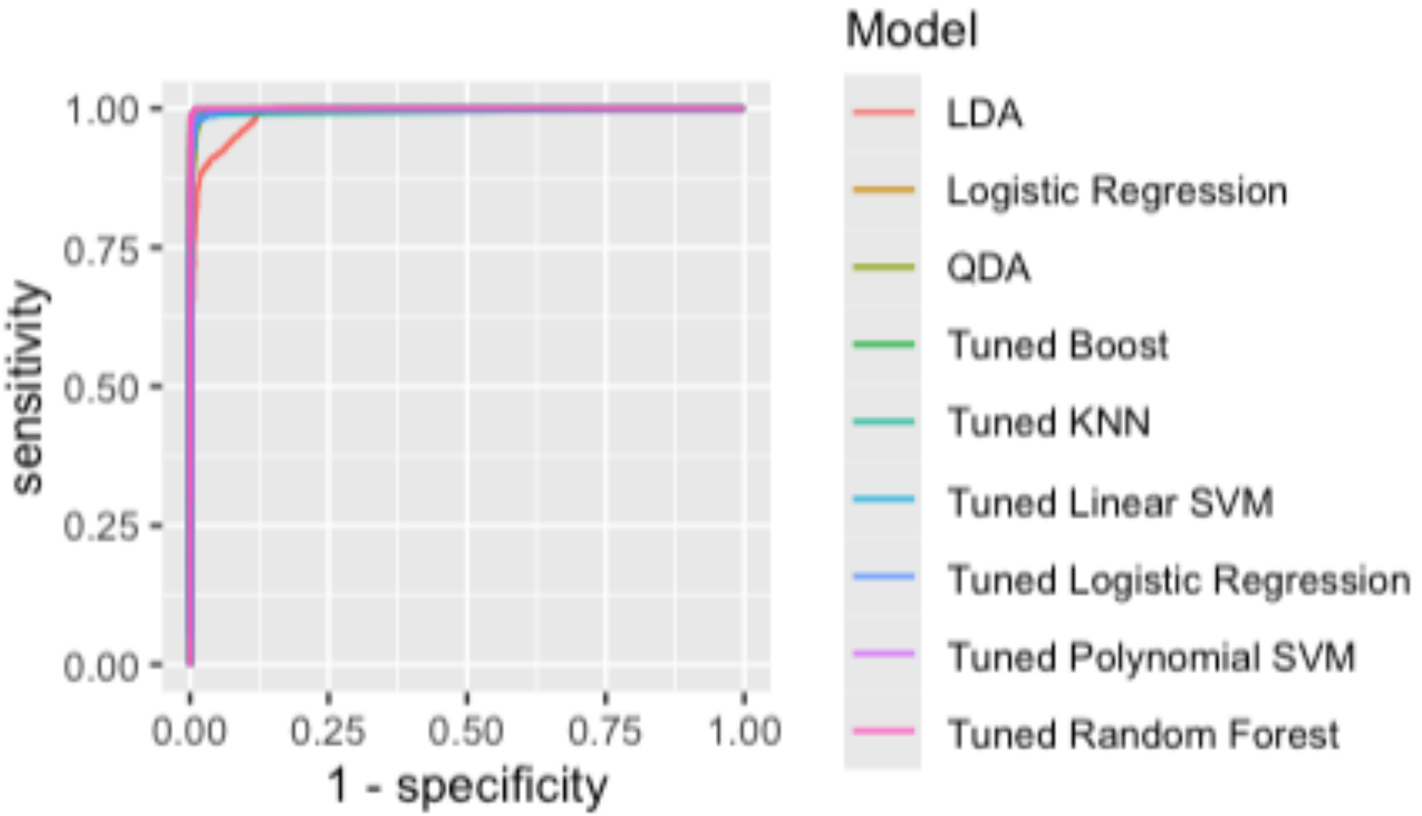


Figure 16. ROC Curve Comparison on Cross Validation at Threshold Selection and for Tuned Models

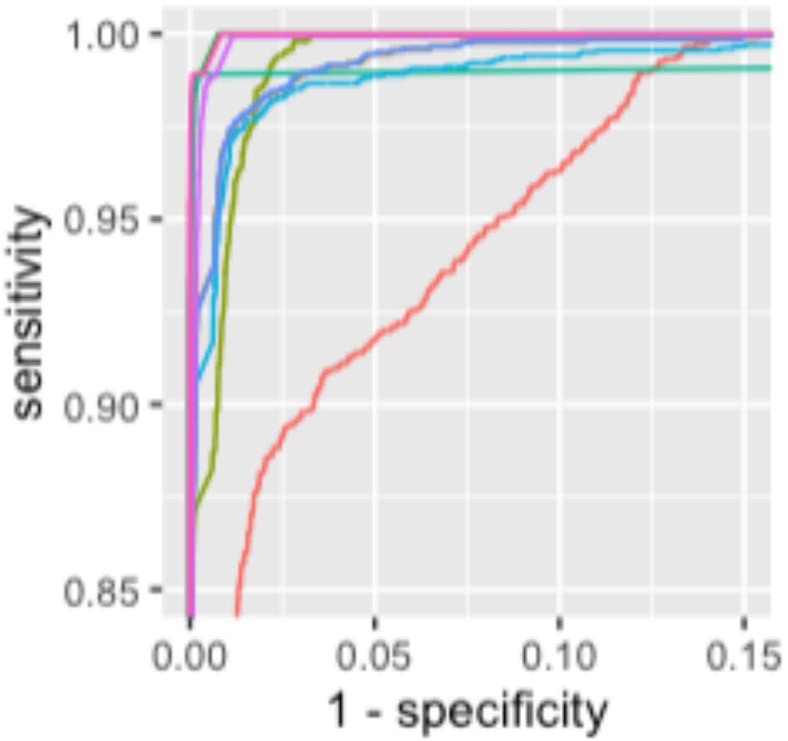


Figure 17. ROC Curve Comparison on Cross Validation at Threshold Selection and for Tuned Models (Zoomed In)

Training and Testing Models

Training Data Metrics

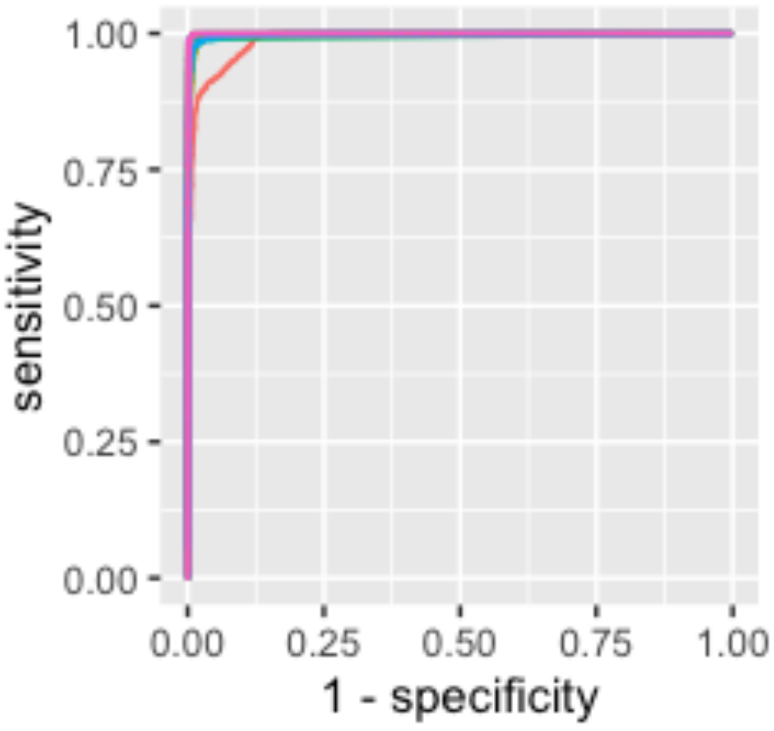
model	dataset	ROC	PR	Threshold	Accuracy	TPR	FPR	Precision	F_meas
Logistic Regression	Train	0.998	0.975	0.25	0.996	0.919	0.002	0.947	0.933
LDA	Train	0.989	0.859	0.85	0.985	0.757	0.008	0.765	0.761
QDA	Train	0.998	0.966	0.25	0.995	0.863	0.001	0.972	0.914
Tuned Logistic Regression	Train	0.999	0.975	NA	0.995	0.883	0.001	0.965	0.922
Tuned KNN	Train	0.994	0.993	NA	0.998	0.975	0.001	0.969	0.972
Tuned Random Forest	Train	1.000	0.998	NA	0.999	0.983	0.000	0.990	0.987
Tuned Boost	Train	1.000	0.998	NA	0.999	0.979	0.001	0.984	0.981
Tuned Linear SVM	Train	0.998	0.970	NA	0.994	0.811	0.000	0.990	0.891
Tuned Polynomial SVM	Train	1.000	0.990	NA	0.996	0.918	0.001	0.958	0.937
Tuned RBF SVM	Train	1.000	0.990	NA	0.996	0.944	0.002	0.946	0.945

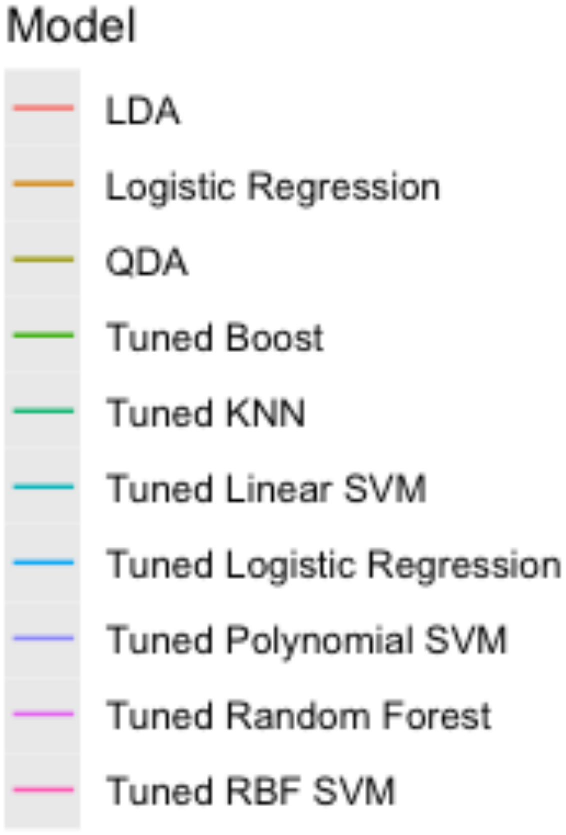
Best performance for metric

Overall best model performance

Table 2. Model Performance Metrics on Training Data

ROC on Training Data





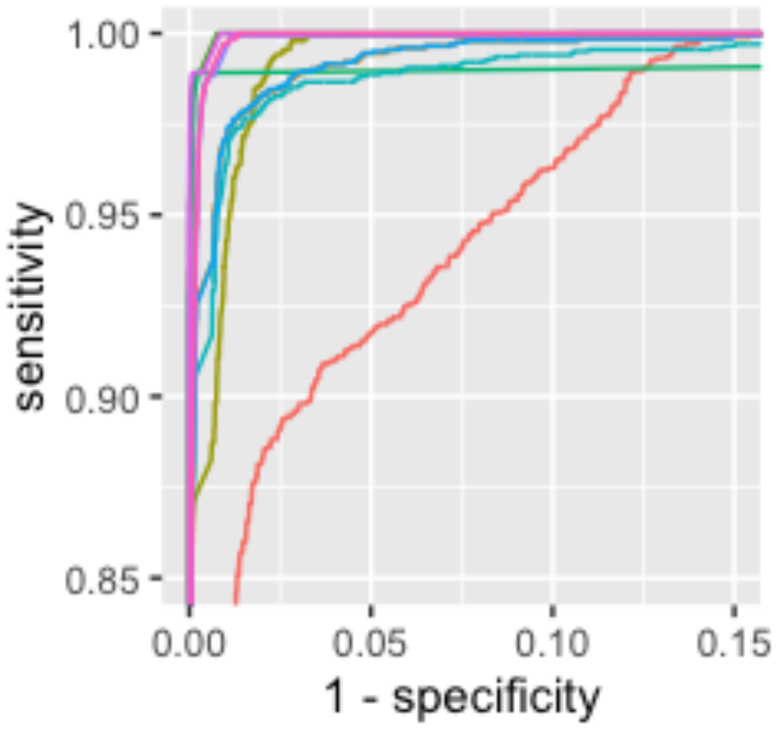


Figure 18. ROC Curve Comparison on Models Fit on Training Data

Figure 19. ROC Curve Comparison on Models Fit on Training Data (Zoomed In)

Training Data PR Curve

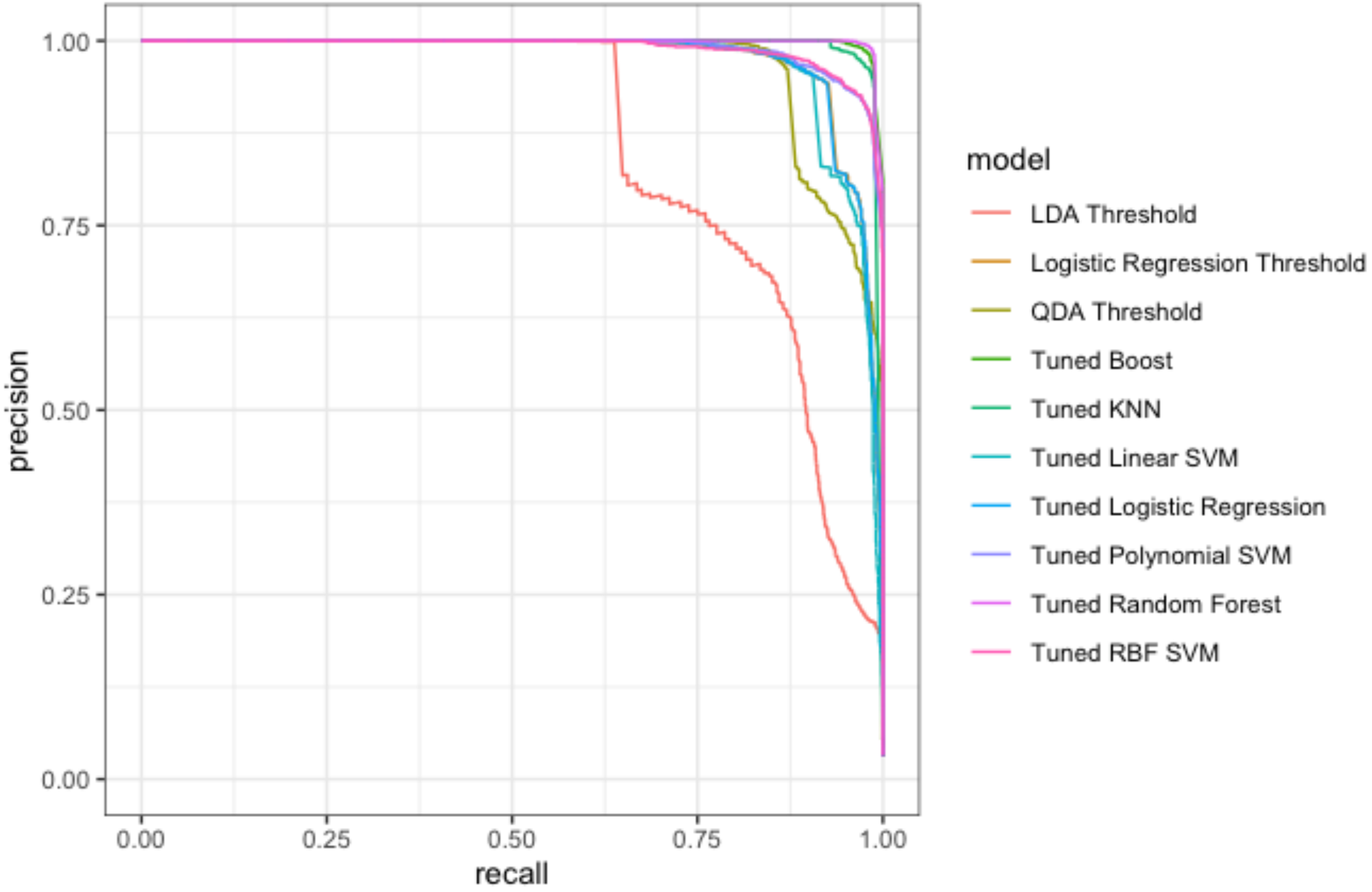


Figure 20. Precision-Recall Curve Comparison on Train Data

Testing Data Metrics

model	dataset	ROC	PR	Threshold	Accuracy	TPR	FPR	Precision	F_meas
Logistic Regression	Test	0.999	0.969	0.25	0.970	0.993	0.030	0.195	0.326
LDA	Test	0.992	0.683	0.85	0.984	0.750	0.015	0.272	0.399
QDA	Test	0.992	0.762	0.25	0.995	0.751	0.003	0.641	0.691
Tuned Logistic Regression	Test	0.999	0.970	NA	0.991	0.988	0.009	0.436	0.605
Tuned KNN	Test	0.961	0.703	NA	0.993	0.831	0.006	0.498	0.623
Tuned Random Forest	Test	0.975	0.650	NA	0.992	0.698	0.006	0.476	0.566
Tuned Boost	Test	0.962	0.716	NA	0.993	0.770	0.006	0.498	0.605
Tuned Linear SVM	Test	1.000	0.970	NA	0.998	0.969	0.002	0.772	0.859
Tuned Polynomial SVM	Test	1.000	0.990	NA	0.994	0.992	0.006	0.542	0.701
Tuned RBF SVM	Test	0.999	0.951	NA	0.993	0.976	0.007	0.507	0.667

Best performance for metric
Overall best model performance

Table 3. Model Performance Metrics on Holdout Data

ROC on Holdout Data

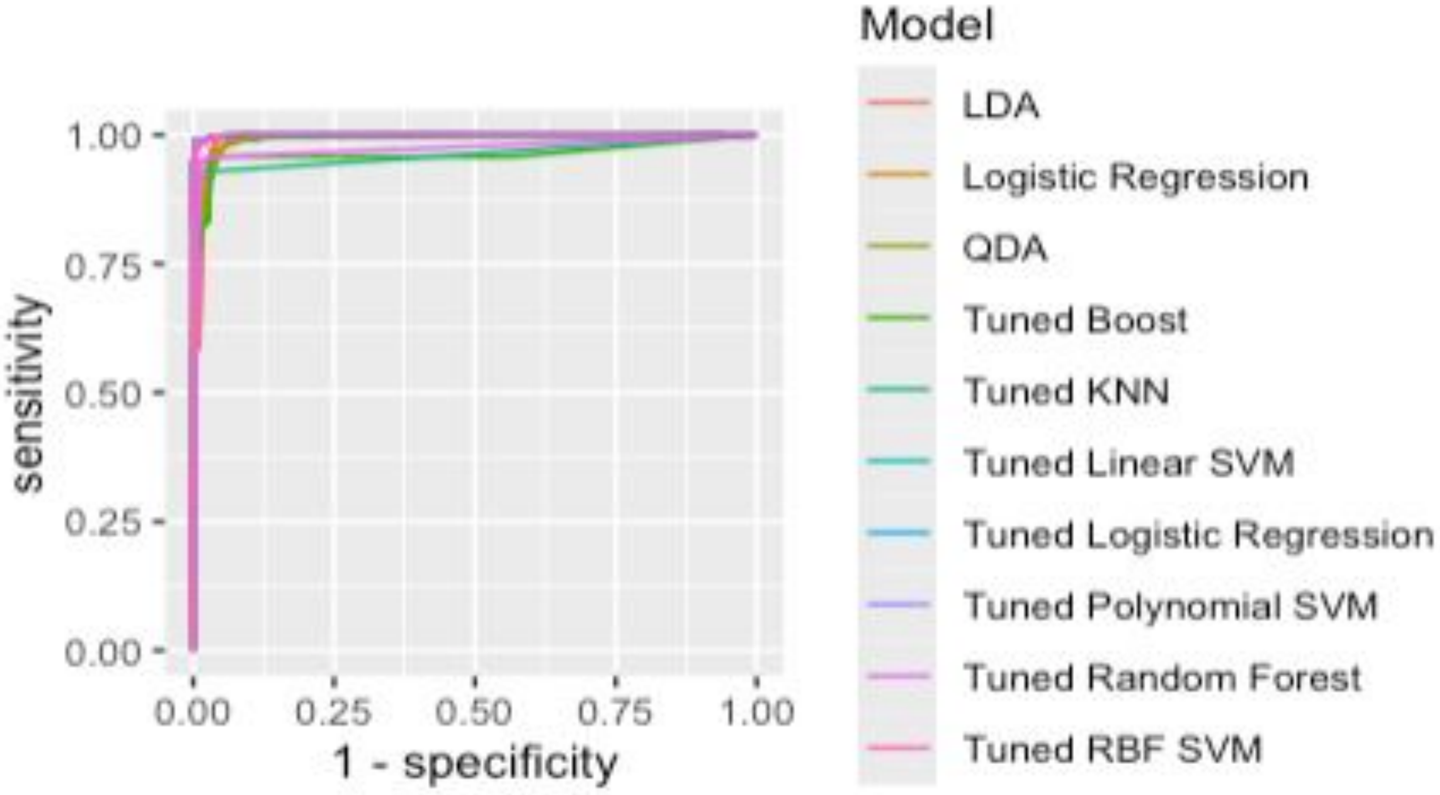


Figure 21. ROC Curve Comparison on Models Fit on Holdout Data

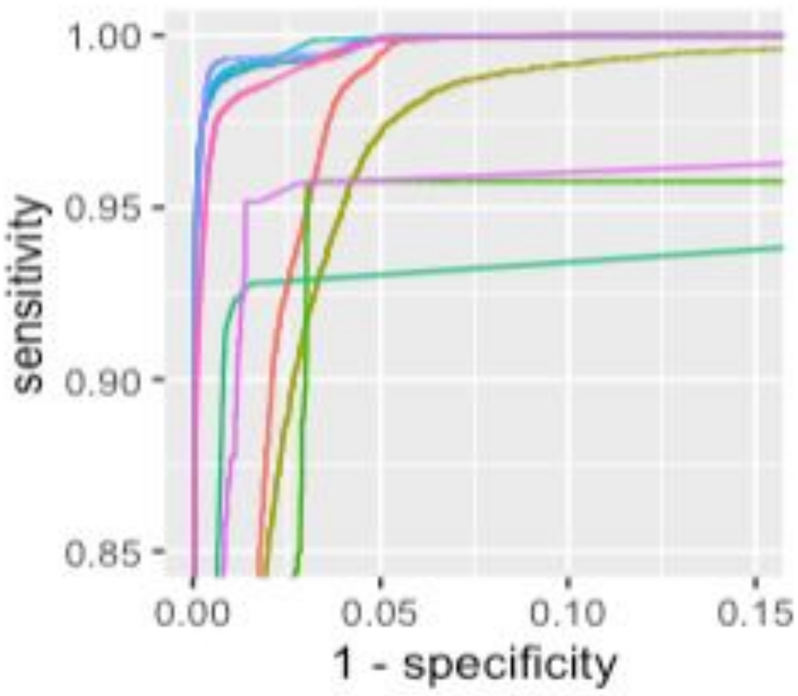


Figure 22. ROC Curve Comparison on Models Fit on Holdout Data (Zoomed In)

Testing Data PR Curve

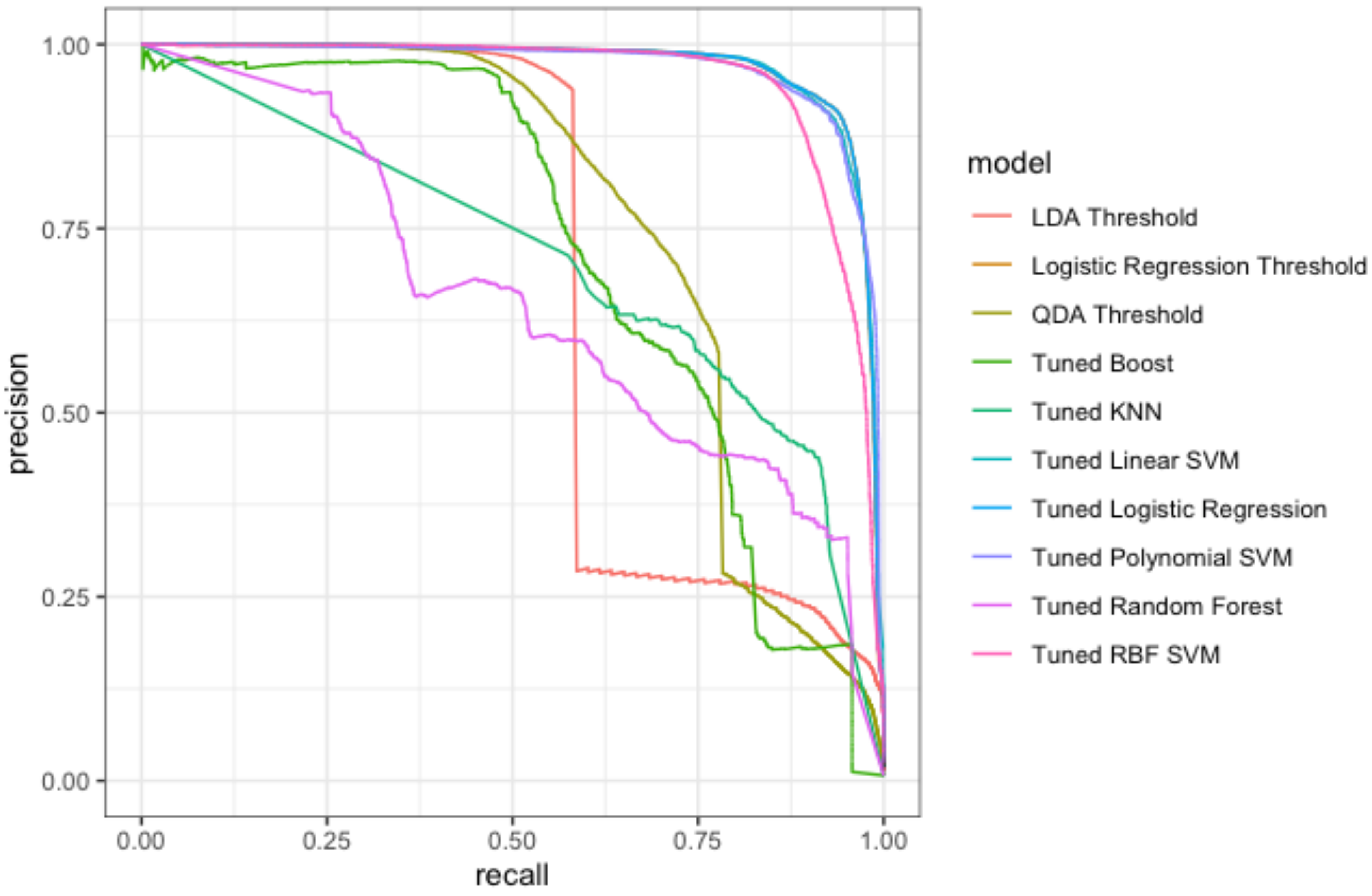


Figure 23. Precision-Recall Curve Comparison on Testing Data

Metric Comparison for Training and Holdout Data

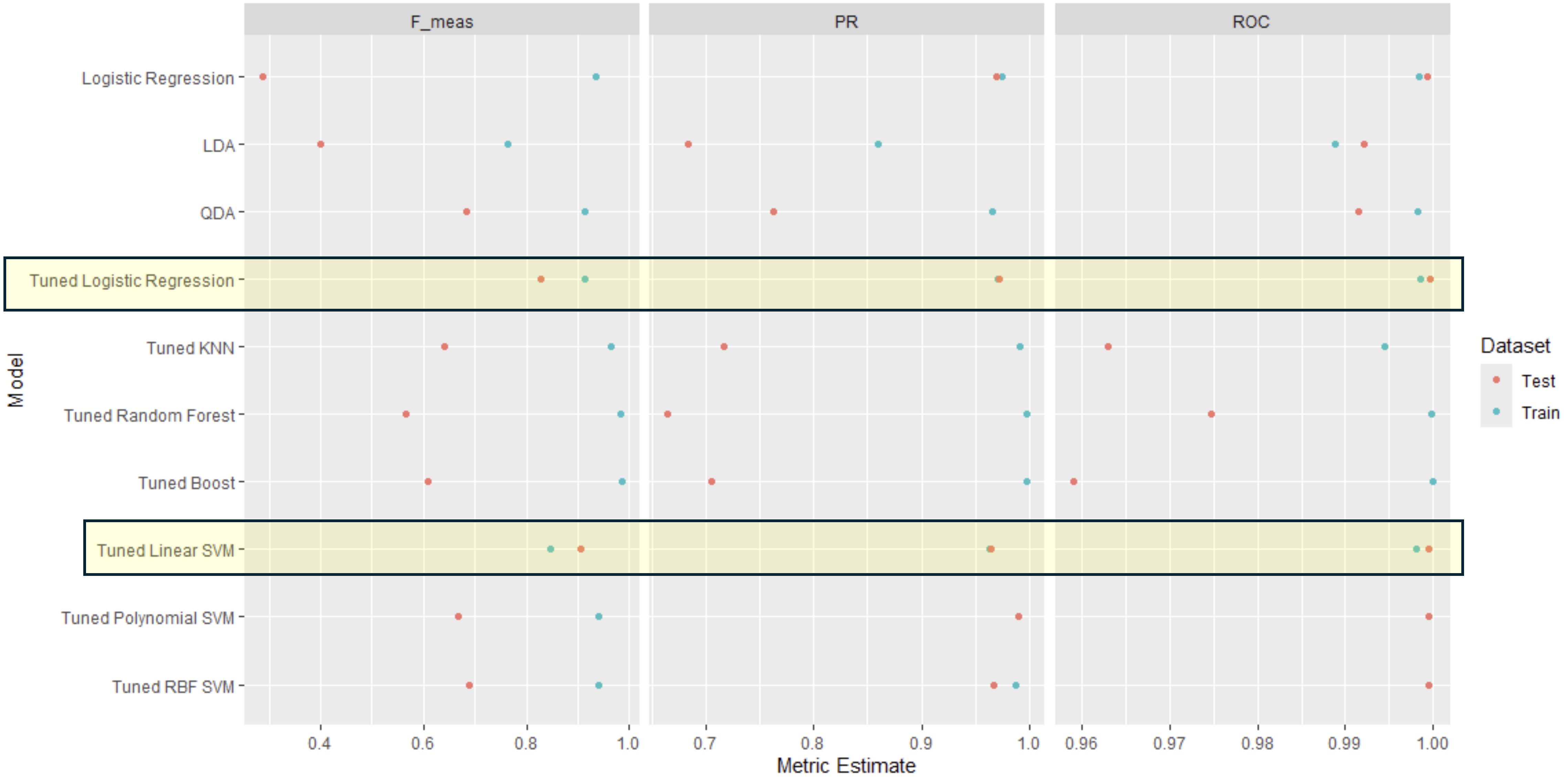


Figure 24. Metric Comparison for Models on Training and Holdout Data

Confusion Matrices

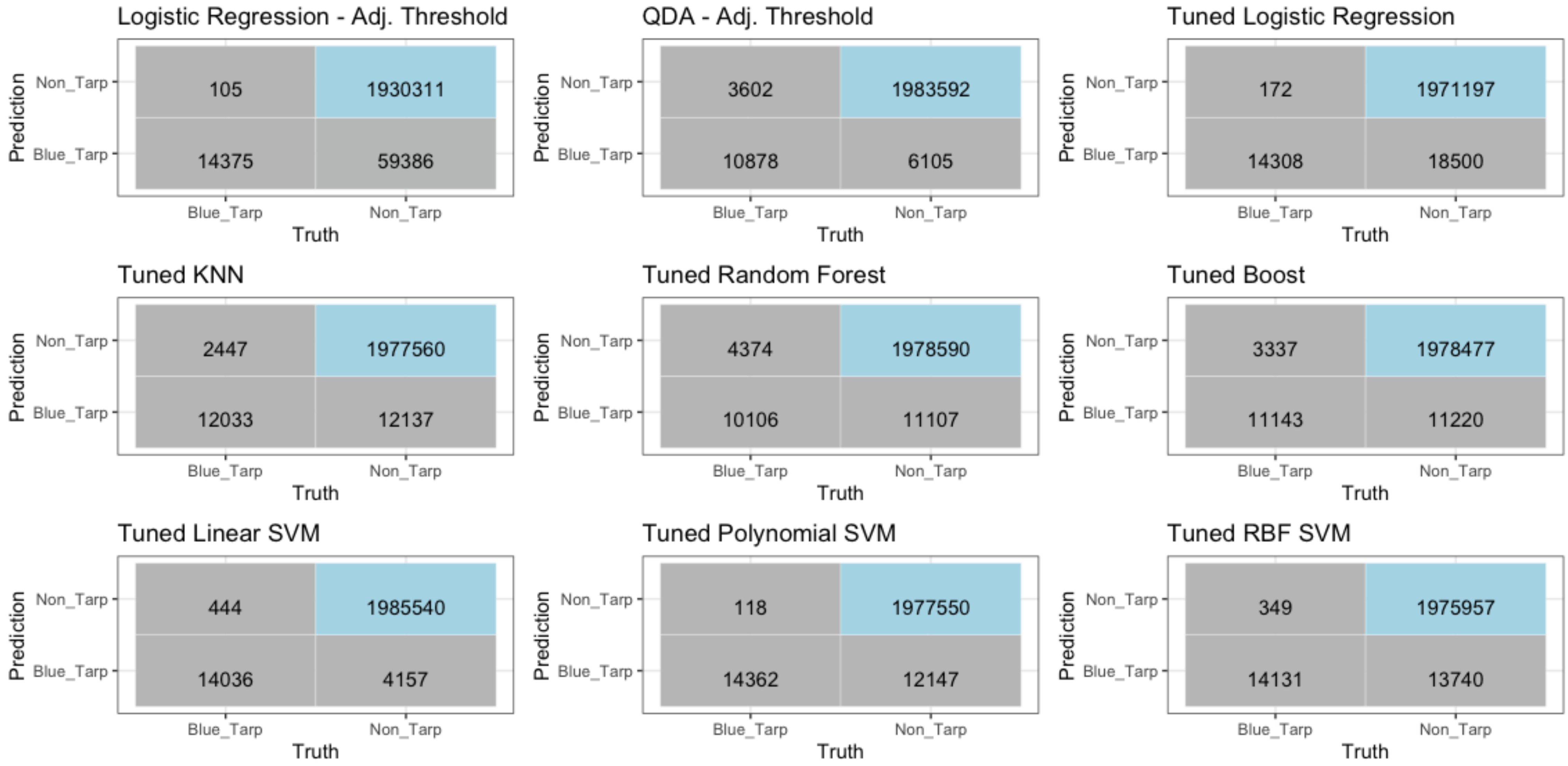


Figure 25. Confusion Matrix for Models on Holdout Data

Metrics from Linear SVM Confusion Matrix

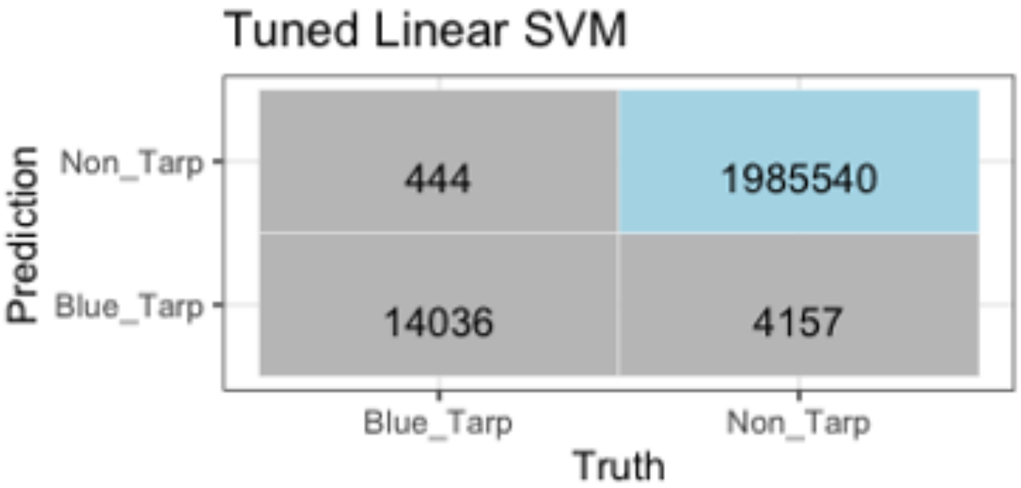


Figure 26. Metrics from Confusion Matrix for Tuned Linear SVM

False negative rate: $\frac{444}{444+14036} = 0.0307$

True positive rate (sensitivity): $\frac{14036}{444+14036} = 0.9693$

False positive rate: $\frac{4157}{1985540+4157} = 0.0021$

Accuracy: $\frac{14036+1985540}{2004177} = 0.9977$

Conclusions: Best Model on Holdout Data

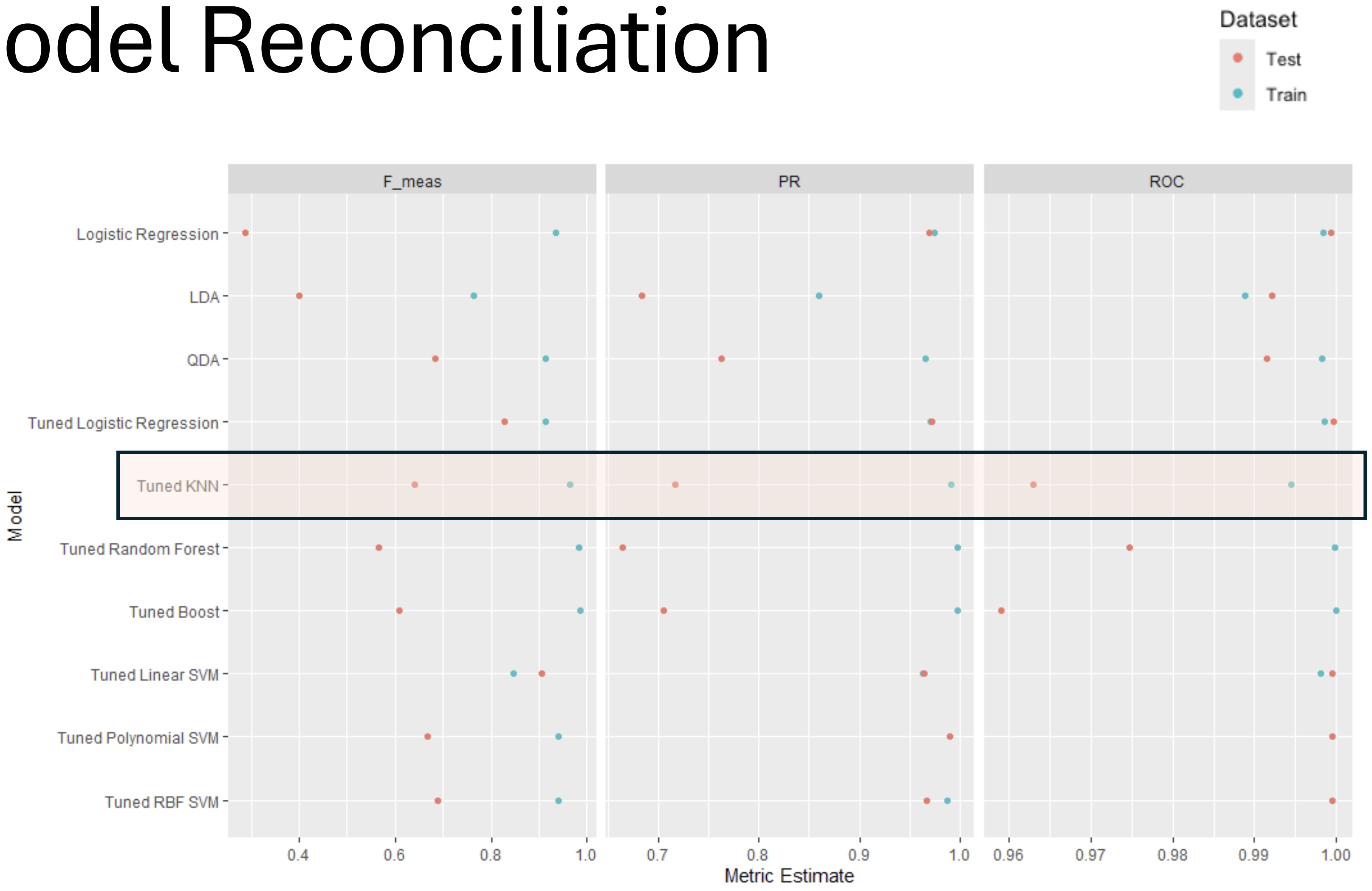
- Tuned linear SVM model performed the best overall on the testing metrics
    - High F-measure, high TPR, low FPR
- Tuned polynomial SVM model, QDA with a threshold of 0.22, and tuned RBF SVM model all performed fairly well

Conclusions: Best Model for CV

- On the cross-validation data, the tuned KNN model had the highest f-measure (0.957) and the highest accuracy (0.997)
  - Tied with tuned logistic regression, random forest, and boost on the accuracy value
- On the cross-validation data, the f-measure for the linear SVM model was lower relative to the other models but still fairly high (0.888)

Conclusions: Model Reconciliation

- Evidence of overfitting on the tuned KNN model
  - Large separation between test and train metrics
- SVMs allow some observations on the incorrect side of the margin and hyperplane
- More flexible methods often have lower training error rates but do not necessarily perform better on the test data



Conclusions: Recommended Model

- In order to efficiently identify displaced persons, the linear SVM model would work well
  - High TPR: blue tarps correctly labeled as blue tarps
  - Low FPR: non-blue tarps rarely identified as blue tarps
- Other models such as the QDA or tuned logistic regression models might be considered because they require less time to run and might be less “expensive” to run
- If resources to search for displaced persons are not limited, it might be worth choosing a model that maximizes the TPR and reduces the false negative rate at the expense of a higher FPR, such as the tuned polynomial SVM

Conclusions: Metric Selection

- The dataset is imbalanced so accuracy would be a misleading metric
  - There a significantly more non_tarp observations than blue_tarp observations. If a model classified all observations as non_tarp, it would still have a high accuracy but would not be useful in finding displaced people.
- Thresholds were selected and models were tuned based on the f-measure
  - F-measure relies on precision and recall
- Emphasis on decreasing the false negative rate to reduce the risk of failing to identify the location of displaced people

Conclusions: Worst vs Best Model Comparison

- LDA and untuned logistic regression had the lowest f-measure values on the hold-out data
  - LDA depends on the mean of all observations in the class
- Linear SVM had the highest f-measure value on the hold-out data
  - SVMs depend only on a subset of the training observations
    - SVMs are robust to observations far from the hyperplane, which allows for variation in things like lighting differences that could affect the pixel RGB values
  - The tuned value for cost is low, which means there are fewer support vectors
  - Low value for cost parameter suggests the model has low bias and high variance

Conclusions: Effects of Tuning

- When tuned, the logistic regression model f-measure for the holdout dataset increased from 0.326 to 0.605.
- The tuned penalty parameter was small, which parallels the relatively small cost tuning parameter in the linear SVM
- According to the authors of our textbook, *An Introduction to Statistical Learning*, logistic regression and SVM often yield similar results[1]
  - We see this confirmed in our models, as the tuned logistic regression model performs more similarly to the SVMs as compared to the untuned logistic regression model.
  - The SVMs perform better than the tuned logistic regression model, which is expected, as the blue tarp and non-tarp classes are fairly well separated.

[1]Gareth James et al., *An Introduction to Statistical*. *Learning with Applications in R*, 2nd ed. (New York: Springer, 2021), 357