Building A Binary Classifier That Helps Locating Displaced People

Tom Lever

08/16/2023

Introduction

In this project, we build a binary classifier that would help us locate people displaced by the earthquake in Haiti in 2010. More specifically, we build in a timely manner an accurate model that classifies pixels in geo-referenced aerial images of Haiti in 2010 as depicting blue tarps or depicting objects that are not blue tarps. People whose homes were destroyed by the earthquake often created temporary shelters using blue tarps. Blue tarps were good indicators of where displaced people lived.

Data

Our training and holdout data were collected likely by applying a Region Of Interest (ROI) Tool to high-resolution, orthorectified / geo-referenced images of Haiti in 2010. Our training image is presented below and is sourced from `HaitiOrthorectifiedImage.tif` at Pixel Values from Images over Haiti. One ROI tool is described at Region of Interest (ROI) Tool. Classes may be assigned to pixels by defining Regions Of Interest.

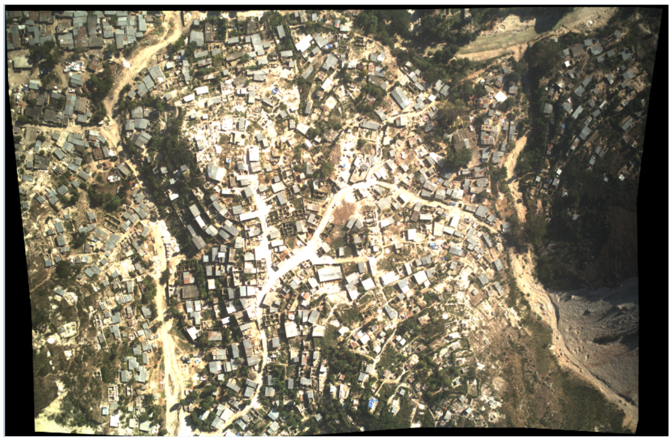


Figure 1: Orthorectified Image Of Haiti In 2010

Our training data is sourced from a CSV file like `HaitiPixels.csv` at Pixel Values from Images over Haiti. Our training data consists of $63,241$ observations. Each training observation consists of a class in the set $\{Vegetation,\ Soil,\ Rooftop,\ Various\ Non-Tarp,\ Blue\ Tarp\}$ and a pixel. A pixel is a colored dot. A pixel is represented by a tuple of intensities of color *Red*, *Green*, and *Blue* in the range 0 to 255.

We conduct exploratory data analysis by considering the distributions of intensities of color *Red*, *Green*, and *Blue* in our training data. The distribution of intensity of color *Red* has a hill, some foothills, and a relatively high proportion of high intensities. The distributions of intensity of colors *Green* and *Blue* each have two hills and are not normal.

Please explore TomLeversRPackage.

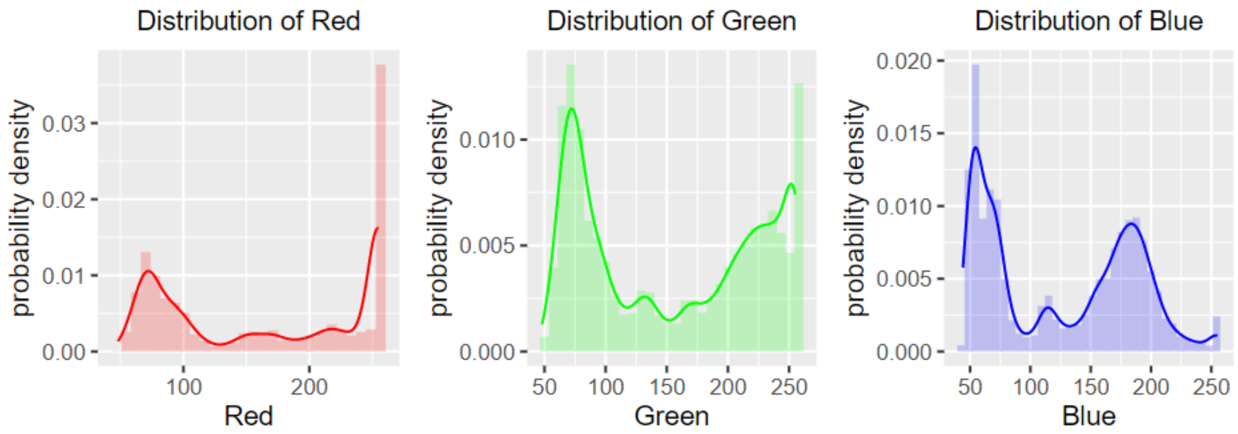
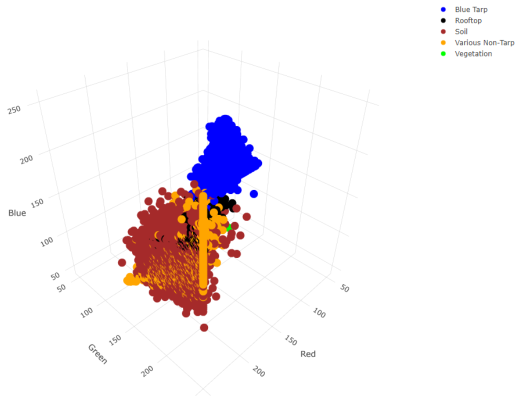
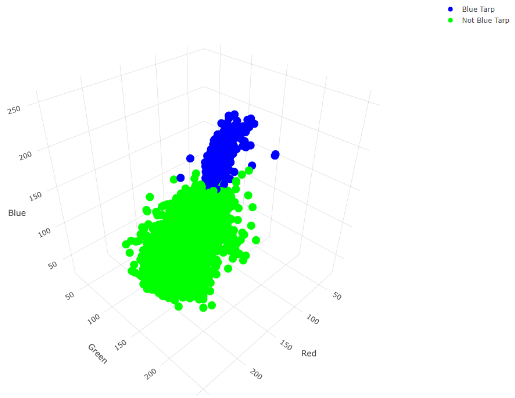


Figure 2: Distributions Of Intensities In Training Data

We examine in the left figure below the distribution of classes (such as *Blue Tarp*) in our training data in a space defined by intensities of color *Red*, *Green*, and *Blue*. The intensity space for pixels representing blue tarps is distinct from the intensity space for pixels representing objects that are not blue tarps.





Our holdout data is sourced from CSV files like

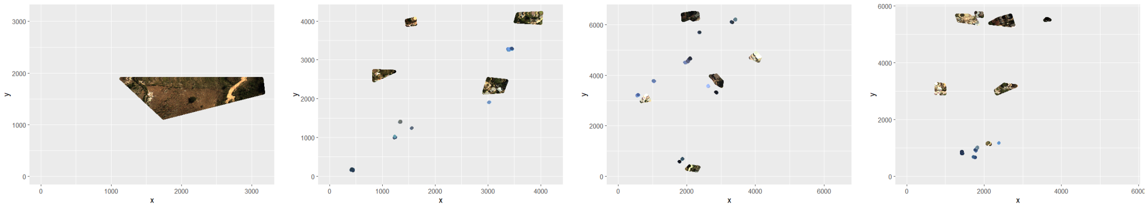
- `orthovnir057_ROI_NON_Blue_Tarps.txt`
- `orthovnir067_ROI_Blue_Tarps.txt`
- `orthovnir067_ROI_NOT_Blue_Tarps.txt`
- `orthovnir069_ROI_Blue_Tarps.txt`
- `orthovnir069_ROI_NOT_Blue_Tarps.txt`
- `orthovnir078_ROI_NON_Blue_Tarps.txt`

at Pixel Values from Images over Haiti and from Dr. Peter Gedeck

- `orthovnir078_ROI_Blue_Tarps.txt`

`orthovnir067_ROI_Blue_Tarps_data.txt` from Dr. Gedeck is a duplicate of `orthovnir067_ROI_Blue_Tarps.txt`. Each holdout observation consists of a class in the set $\{Not\ Blue\ Tarp,\ Blue\ Tarp\}$ and a pixel. The class of an observation is determined according to the name of the source of that observation.

We assume that columns $B1$, $B2$, and $B3$ in the above source files correspond to intensities of colors *Red*, *Green*, and *Blue*. Values in these columns vary from 255 down to 27, 28, and 25. Partial reconstructions of images of Regions 57, 67, 69, and 78 based on columns $B1$, $B2$, $B3$, X, and Y from our holdout source files look realistic.



We conduct exploratory data analysis by considering the distributions of intensities of color *Red*, *Green*, and *Blue* in our holdout data. The distributions of intensity of colors *Red*, *Green* and *Blue* are normal and broad relative to our training distributions. Our holdout distributions have long right tails and relatively high proportions of high intensities.

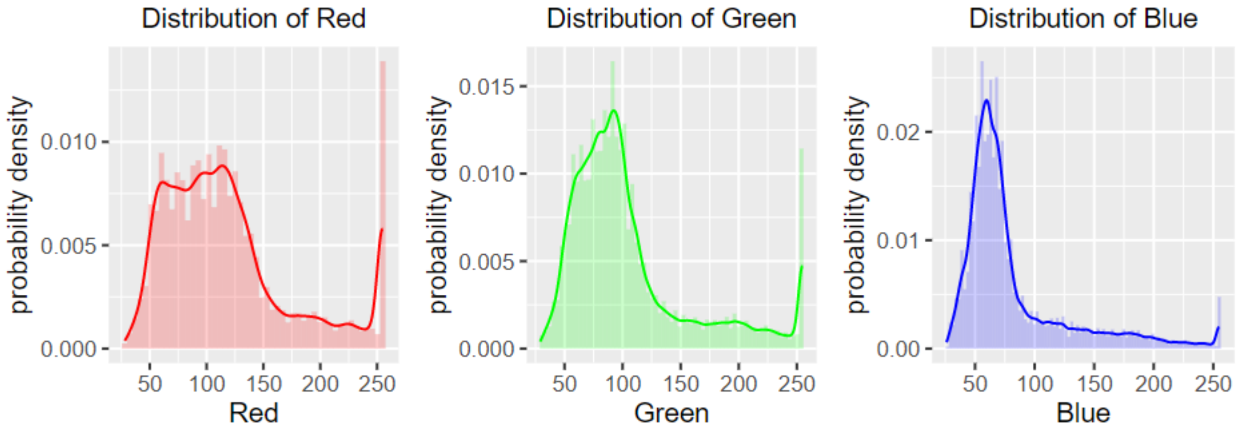


Figure 3: Distributions Of Intensities In Holdout Data

We examine in the right figure above the distribution of classes– i.e., *Not Blue Tarp* and *Blue Tarp* –in our holdout data in a space defined by intensities of color *Red*, *Green*, and *Blue*. The intensity space for pixels representing blue tarps is distinct from the intensity space for pixels representing objects that are not blue tarps.

The proportion of training observations that correspond to blue tarps (0.032) is about 4.4 times the proportion of holdout observations that correspond to blue tarps (0.007).

We expect that, due to differences in our training and holdout distributions, a binary classifier trained on our training data and holdout tested on our holdout data will have a maximum F1 measure that is less than that of a classifier cross validated on our training data. We recommend cross validating on our composite data or cross validating on 90 percent of our composite data and holdout testing on the other 10 percent. By training on our composite data our classifiers will become more realistic and robust and our holdout testing results will align with our cross-validation results.

Methods

Choosing A Binary Classifier

Since the intensity space for pixels representing blue tarps is distinct from the intensity space for pixels representing objects that are not blue tarps, we consider our optimal model to be a binary classifier that classifies pixels as depicting blue tarps or depicting objects that are not blue tarps. We may ignore non-binary classifiers that predict probabilities for all classes and may be used to locate pixels that more likely depict blue tarps than objects that are not blue tarps.

Choosing A Performance Metric

A threshold is a probability. A model classifies a pixel as representing a blue tarp if the probability that the pixel represents a blue tarp that the model predicts is greater than the threshold. Recall / True Positive Rate (TPR) is the ratio of true positives to actual positives. False Positive Rate (FPR) is the ratio of false positives to actual negatives. Precision / Positive Predictive Value (PPV) is the ratio of true positives to predicted positives. An F1 measure is the harmonic mean of PPV and TPR. A decimal of true positives is the ratio of number of true positives to total number of predictions. All of these quantities lie between 0 and 1.

According to Mr. Allwright, "Accuracy is a simple and widely understood error metric whose score ranges from 0% to 100%, where 100% is a perfect score and 0% is the worst. It is calculated as the number of correct predictions" divided by the number of total predictions. "This simplicity creates problems when the classes are imbalanced as a model which only predicts for the majority class will seem to be performing well according to its accuracy score… Accuracy does not perform well on imbalanced datasets which often leads to misleading results… Accuracy should be used when the dataset is balanced or when communicating the results to end users is important."

Receiver Operator Characteristic (ROC) graphs are graphs of TPR Vs. FPR for different thresholds. According to ROC and AUC, Clearly Explained!, the Area Under The Curve (AUC) of an ROC graph "makes it easy to compare one ROC curve to another" and to compare one binary classifier to another. If the AUC for one ROC curve is greater than the AUC for another ROC curve, the classifier with the former ROC curve is better than the classifier with the latter ROC curve. A given classifier's threshold may be tuned so that its FPR is close to 0 and its TPR is close to 1. We may consider maximizing ROC AUC among classifiers and tuning a classifier's threshold to minimize the distance between the corresponding point (FPR, TPR) and (0, 1). According to Data Scientist Stephen Allwright, ROC "[AUC] does not perform well on imbalanced datasets which often leads to misleading results… AUC shoud be used when you have a balanced dataset."

"People often replace the False Positive Rate with Precision… If there were lots of samples that were not [Blue Tarps] relative to the number of [Blue Tarp] samples, then Precision might be more useful than the False Positive Rate. This is because Precision does not include the number of True Negatives in its calculation, and is not effected by the imbalance. In practice, this sort of imbalance occurs when studying a rare [occurrence]. In this case, the study will contain many more [pixels not corresponding to blue tarps than corresponding to blue tarps]. We may consider maximizing PR AUC among classifiers and tuning a classifier's threshold to minimize the distance between the corresponding point (TPR, PR) and (1, 1).

According to Category graph: Precision-Recall vs. Threshold, "The ideal threshold setting is the highest possible recall and precision rate. This goal is not always achievable, because the higher the recall rate, the lower the precision rate, and vice versa. Setting the most appropriate threshold for a category is a trade-off between these two rates."

According to Optimal Thresholding of Classifiers to Maximize F1 Measure, "The [balanced] harmonic mean of precision and recall, the F1 measure is widely used to evaluate the success of a binary classifier when one class is rare." According to The truth of the F-measure, "the F-measure was first introduced to evaluation tasks of information extraction technology at the Fourth Message Understanding Conference (MUC-4) in 1992."

The average of two ratios (e.g., precision and recall) is the balanced harmonic mean of those two ratios. According to Harmonic mean, “For instance, if a vehicle travels a certain distance d outbound at a speed s_1 (e.g., 60 km/h) and returns the distance at a speed s_2 (e.g., 20 km/h), then its average speed is the harmonic mean of s_1 and s_2 (30 km/h), not the arithmetic mean (40 km/h). The total travel time is the same as if it had traveled the whole distance at that average speed. This can be proven as follows:

$$Average\ speed\ for\ the\ entire\ journey = \frac{Total\ distance\ traveled}{Sum\ of\ time\ for\ each\ segment} = \frac{D}{T} = \frac{D}{t_1 + t_2} = \frac{2d}{\frac{d}{s_1} + \frac{d}{s_2}} = \frac{2}{\frac{1}{s_1} + \frac{1}{s_2}}$$

“[I]f each sub-trip covers the same distance, then the average speed is the [balanced] harmonic mean of [each] sub-trip speed.” If one sub-trip covers a greater distance than the other sub-trip, “then a weighted harmonic mean is needed.” To return to the F1 measure, if each of precision and recall has the same weight, then the F1 measure is the balanced harmonic mean of each of precision and recall. This balanced harmonic mean is

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

If recall has more weight than precision, then a weighted harmonic mean is needed. This weighted harmonic mean is

$$F_{w_P,\ w_R} = \frac{w_P + w_R}{\frac{w_P}{P} + \frac{w_R}{R}}$$

When searching for an optimal binary classifier of a given type, we choose as our optimal binary classifier of that type the binary classifier with the highest F1 measure and $F_{w_P=1,\ w_R=1}$.

Regarding the weight of PPV and the weight of TPR, we prioritize TPR at least as much as PPV. We prioritize identifying as many pixels corresponding to blue tarps correctly as possible at least as much as having predictions that pixels correspond to blue tarps be correct. As we move toward providing resources for refugees, we may wish to prioritize identifying as many pixels corresponding to blue tarps as possible more than having predictions that pixels correspond to blue tarps be correct.

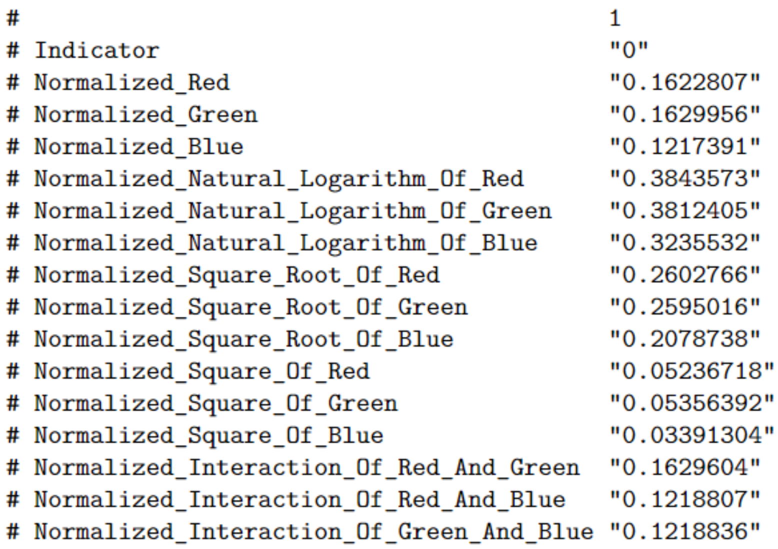
Data Frame For Modeling

In order to build binary classifiers, we create a composite data frame of our training data and our holdout data; substitute a column of classes for a column of indicators of whether of not a pixel depicts a blue tarp; add columns corresponding to normalized intensities and intensities normalized after transforming by the natural logarithm, the square root, the square, and interactions; and divide the composite data frame into training and holdout data frames again. We normalize as opposed to standardize intensities given that distributions of intensity are not normal. We shuffle our training and holdout data. See below the first observation in our training data frame of indicators and pixels.

Grid Search

Dimensions

First Dimension: Type Of Classifier We conduct a grid search for an optimal binary classifier. The first dimension of our grid is type of classifier. We compare Logistic Regression (LR) classifiers, Logistic Ridge Regression (LRR) classifiers, Linear Discriminant Analyses (LDA’s), Quadratic Discriminant Analyses (QDA’s), K Nearest Neighbors (KNN) classifiers, Random Forests (RF’s), Support-Vector Machines With Linear Kernel (SVMWLK’s), Support-Vector Machines With Polynomial Kernel (SVMWPK’s), and Support-Vector Machines With Radial Kernel (SVMWRK’s). Generally speaking, LR classifiers, LRR classifiers, LDA’s, and QDA’s are relatively inflexible with high bias and low variance; KNN classifiers, RF’s, and SVMWLK’s are relatively flexible with low bias and high variance. KNN classifiers may be flexible when K is approximately 1 or inflexible as K approaches the number of observations in our training data.



```
#                                          1
# Indicator                                "0"
# Normalized_Red                           "0.1622807"
# Normalized_Green                         "0.1629956"
# Normalized_Blue                          "0.1217391"
# Normalized_Natural_Logarithm_Of_Red      "0.3843573"
# Normalized_Natural_Logarithm_Of_Green    "0.3812405"
# Normalized_Natural_Logarithm_Of_Blue     "0.3235532"
# Normalized_Square_Root_Of_Red            "0.2602766"
# Normalized_Square_Root_Of_Green          "0.2595016"
# Normalized_Square_Root_Of_Blue           "0.2078738"
# Normalized_Square_Of_Red                 "0.05236718"
# Normalized_Square_Of_Green               "0.05356392"
# Normalized_Square_Of_Blue                "0.03391304"
# Normalized_Interaction_Of_Red_And_Green  "0.1629604"
# Normalized_Interaction_Of_Red_And_Blue   "0.1218807"
# Normalized_Interaction_Of_Green_And_Blue "0.1218836"
```

Figure 4: First Observation In Training Data Frame Of Indicators And Pixels

Second Dimension: Set Of Predictive Terms The second dimension of our grid is set of predictive terms. We consider classifiers with the terms in the above output other than *Indicator*. Per University Of Virginia courses Linear Models For Data Science and Statistical Learning, these predictive terms involve common transformations of predictors.

Third Dimension: Value Of Primary Hyperparameter The third dimension of our grid is value of primary hyperparameter. For LRR classifiers, the primary hyperparameter is λ. LRR classifiers are penalized for inclusion of predictive terms proportionally to λ; setting λ to be greater than 0 may decrease the variance and increase the performance of LRR classifiers.

For KNN classifiers, the primary hyperparameter is K. The class of a test observation depends on determining the classes of the K nearest neighboring observations in the training data. A KNN classifier with $K = 3$ is relatively flexible with low bias and high variance; such a model may overfit the training data and identify patterns with performance that is less than ideal. A KNN classifier with $K = n$, the number of training observations, predicts that all observations have one class and has high bias and low variance. A KNN classifier with $1 < K < n$ may perform best.

For RF's, the primary hyperparameter is *mtry*, the number of variables randomly sampled as candidates at each split / fraction of features available at each split. According to Dr. Bill Basener, "by selecting only some of the features for each split, we make our trees to be different from each other. We're creating more variety among our trees and are decorrelating the trees... [mtry] is probably the most important parameter... for a random forest. And the way this works: A lower fraction creates more decorrelation among the trees. So the trees are more different. But it also creates lower accuracy in the individual trees because they don't have as many parameters to select from for their classification. And so there's a bit of a tradeoff there."

For SVM's, the primary hyperparameter is cost C. According to *An Introduction to Statistical Learning* (Second Edition) (James et al. 2023), "Cost C is a nonnegative tuning parameter... C determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate. We can think of C as a budget for the amount that the margin can be violated by the n observations. If $C = 0$ then there is no budget for violations to the margin... For $C > 0$ no more than C observations can be on the wrong side of the hyperplane... As the budget C increases, we become more tolerant of violations to the margin, and so the margin will widen. Conversely, as C decreases, we become less tolerant of violations to the margin and so the margin narrows...

C controls the bias-variance trade-off of a support-vector [machine]. When C is small, we seek narrow margins that are rarely violated; this amounts to a classifier that is highly fit to the data, which may have low bias but high variance. On the other hand, when C is larger, the margin is wide and we allow more violations to it; this amounts to fitting the data less hard and obtaining a classifier that is potentially more biased but may have lower variance…

When the tuning parameter C is large, then the margin is wide, many observations violate the margin, and so there are many support vectors. In this case, many observations are involved in determining the hyperplane. This classifier has low variance (since many observations are support vectors) but potentially high bias. In contrast, if C is small, then there will be fewer support vectors and hence the resulting classifier will have low bias but high variance…

When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows."

Fourth Dimension: Value Of Secondary Hyperparameter The fourth dimension of our grid is value of secondary hyperparameter. For RF's, the secondary hyperparameter is *ntree*, the number of trees that we want to create. According to Dr. Basener, "And what is the effect of this parameter? So the more trees [you] have the more accuracy you get, but up to a point. Typically, you'll get a lot of increase when you go from 1 to 2 to 10 or 20 trees, and then you'll increase more up to 100. But at some point, increasing doesn't give you any more accuracy. So that may be at 500 that you don't want to create any more trees, or 2000 you have enough trees. It depends a lot of how complicated of a decision you need to make, how complicated your space is, how many features you have, how much data you have." According to Dr. Gedeck, "a good rule of thumb is that usually from 100 trees onward, not much happens."

The secondary hyperparameter for SVMWPK's is degree d; the secondary hyperparameter for SVMWRK's is γ. According to *An Introduction to Statistical Learning*, "It can be shown that… The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \left[\alpha_i \langle x, x_i \rangle\right]$$

where there are n parameters α_i, one per training observation."

According to Dr. Basener, "we can take this framework and instead of using the inner product there, we can replace that by what's called a kernel. And so it's a way of measuring distance between two points: between x, a new observation, and each x_i in our training set. And so we can use what's called the linear kernel

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} \left[x_{ij} x_{i'j}\right]$$

, which is [the inner product in] the linear support vector classifier. We can use a polynomial kernel of degree [d]

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} \left[x_{ij} x_{i'j}\right]\right)^d$$

So all we're doing is replacing th[e inner product] with one of these functions here. So [there's] a polynomial kernel of degree d or what's called a radial kernel

$$K(x_i, x_{i'}) = \exp\left\{-\gamma \sum_{j=1}^{p} \left[(x_{ij} - x_{i'j})^2\right]\right\}$$

And this is a common kernel to use. It's also called a radial basis function… So this is an exponential. Let's think about this function. There's an exponential of [a scaled] distance between the two points squared. So this is going to give some sort of bell-shaped curve and that γ tells us how quickly that bell-shaped curve drops off."

Fifth Dimension: Value Of Threshold The fifth dimension of our grid is value of threshold. Varying the threshold according to which a classifier classifies an observations as corresponding to a blue tarp or not will result in different numbers of false negatives, false positives, true negatives, and true positives and different F1 measures. We seek to maximize F1 measure.

Cross Validation

According to *An Introduction to Statistical Learning*, for models with categorical responses such as binary classifiers, "k-fold cross validation involves randomly dividing a set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds." A performance metric like F1 measure is "computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the" performance metric, $F_1, F_2, ..., F_k$. "The k-fold cross validation estimate is computed by averaging these values.

[W]e are interested only in the location of the [maximum] point in the estimated test [F1 measure] curve. This is because we might be performing cross validation on a number of statistical learning methods, or on a single method using different [hyperparameters], in order to identify the method that results in the [highest F1 measure. D]espite the fact that they sometimes [misestimate] the true test [F1 measure, most] of the CV curves [of F1 measure vs. threshold] come close to identifying the correct [threshold] corresponding to the [highest F1 measure]."

"[T]here is some variability in the CV estimates as a result of the variability in how the observations are divided into ten folds. But this variability is typically much lower than the variability in the test [F1 measures] that result from the validation set approach."

As described below, we apply cross validation to averaging performance metrics and choosing λ for LRR classifiers; K for KNN classifiers; $mtry$ and $ntree$ for RF's; and C, d, and γ for SVM's. We apply cross validation to averaging performance metrics using `rsample::vfold_cv`. We use `vfold_cv` by passing `vfold_cv` our training data and specifications of numbers of partitions of the data and times to repeat v-fold partitioning. We apply cross validation to choosing λ, K, $mtry$, $ntree$, C, d, and γ using `caret::train` by passing `train` a `caret::trainControl` constructed with `method = "cv"`.

Bidirectional Selection

To search for an optimal LR classifier, we perform pure bidirectional selection. For each formula, we perform 10-fold cross validation. For each fold, we record 100 thresholds and corresponding performance metrics including F1 measure. For each formula, we record 100 thresholds and corresponding average performance metrics. For each formula, we find the maximum average F1 measure. For each formula, we choose the threshold that yields the highest maximum average F1 measure. We choose the LR classifier with the formula and threshold with the highest maximum average F1 measure.

To describe pure bidirectional selection in detail, we determine the maximum average F1 measure for LR classifiers with formula $Indicator \sim normalize(Red)$. Similarly, we calculate the maximum average F1 measure for LR classifiers with formula $Indicator \sim normalize(Green), Indicator \sim normalize(Blue), ..., Indicator \sim normalize\,(Green : Blue)$. The classifiers with formula $Indicator \sim normalize(Blue^2)$ have the highest maximum average F1 measure of 0.446. We consider the maximum average F1 measures for LR classifiers with formulas $Indicator \sim normalize(Blue^2) + normalize(Red), ..., Indicator \sim normalize(Blue^2) + normalize(Green : Blue)$. The classifiers with formula $Indicator \sim normalize(Blue^2) + normalize(Red^2)$ have the highest maximum average F1 measure of 0.912. Since we have already considered the maximum average F1 measures for classifiers with formulas $Indicator \sim normalize(Red^2)$ and $Indicator \sim normalize(Blue^2)$, we do not perform backward selection. We consider the maximum average F1 measures for classifiers with formulas $Indicator \sim normalize(Blue^2) + normalize(Red^2) + normalize(Red), ..., Indicator \sim normalize(Blue^2) + normalize(Red^2) + normalize(Green : Blue)$. The classifiers with formula

$Indicator \sim normalize(Blue^2) + normalize(Red^2) + normalize\left(\sqrt{Blue}\right)$ have the highest maximum average F1 measure of 0.936. Per backward selection, we drop predictor $normalize(Blue^2)$ and consider the maximum average F1 measure for classifiers with formula $Indicator \sim normalize(Red^2) + normalize\left(\sqrt{Blue}\right)$. These classifiers have a maximum average F1 measure of 0.903. We choose a LR classifier with formula $Indicator \sim normalize(Blue^2) + normalize(Red^2) + normalize\left(\sqrt{Blue}\right)$.

Our optimal LR classifier has formula $Indicator \sim normalize(Blue^2)+normalize(Red^2)+normalize\left(\sqrt{Blue}\right)$.

To search for an optimal LRR classifier, we perform pure bidirectional selection. Unfortunately, pure bidirectional selection results ultimately in classifiers with formula $Indicator \sim normalize(Blue^2) + normalize(Green : Blue) + normalize[\ln(Blue)]$ and maximum average F1 measure 0.447. We conduct an exhaustive search for optimal classifiers with a formula with two predictors; the optimal classifiers have formula $Indicator \sim normalize[\ln(Blue)] + normalize(\sqrt{Red})$. We perform bidirectional selection using this formula as a baseline. Our optimal LRR classifier has formula $Indicator \sim normalize[\ln(Blue)] + normalize(\sqrt{Red}) + normalize\,[\ln(Green)] + normalize[\ln(Red)]$.

For the remaining types of classifiers, we perform pure bidirectional selection.

Our optimal LDA has formula $Indicator \sim normalize(Blue) + normalize(Red^2) + normalize\left(Green^2\right)$.

Our optimal QDA has formula $Indicator \sim normalize\,(Blue)+normalize\left(Red^2\right)+normalize\,(Red : Blue)$.

Our optimal KNN classifier has formula $Indicator \sim normalize(Red : Blue) + normalize(Red) + normalize[\ln(Green)]$.

Our optimal RF has formula $Indicator \sim normalize(Red^2)+normalize(Red : Green)+normalize\,(Green : Blue)+ normalize(Green)$.

Our optimal SVMWLK has formula $Indicator \sim normalize(Red)+normalize\left(Green^2\right)+normalize(Blue)+ normalize\left(Blue^2\right) + normalize\left(\sqrt{Red}\right) + normalize\left(\sqrt{Blue}\right)$.

Our optimal SVMWPK has formula $Indicator \sim normalize\left(Blue^2\right)+normalize\left(Red^2\right)+normalize(Green : Blue)$.

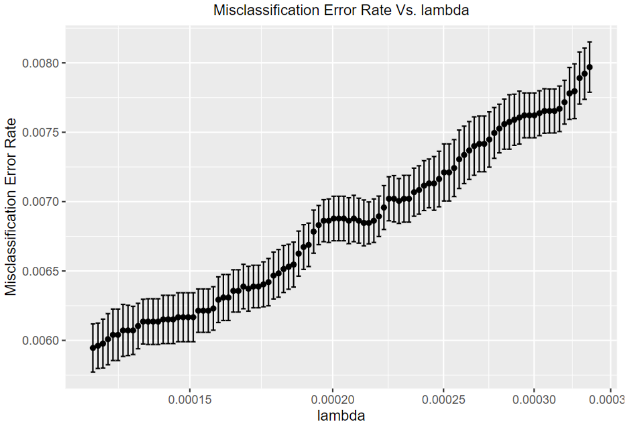
Our optimal SVMWRK has formula $Indicator \sim normalize(Blue)+normalize[\ln(Red)]+normalize\left(\sqrt{Green}\right)+ normalize[\ln(Blue)]$.

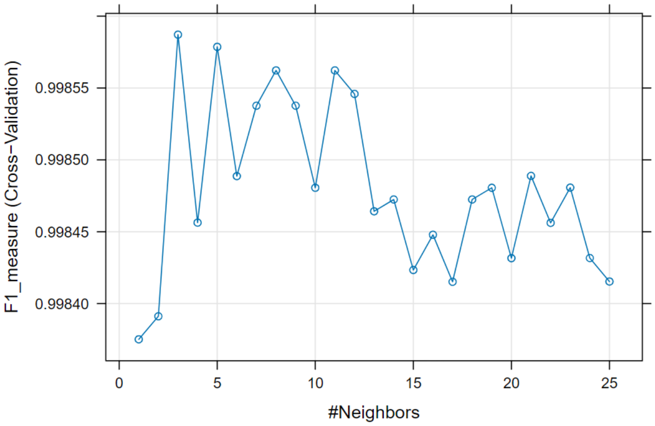
Tuning Hyperparameters

Logistic Ridge Regression Classifiers During bidirectional selection of LRR classifiers, for each formula, we search for one value of hyperparameter λ to be used in training a LRR classifier on 9 cross-validation folds. For our optimal formula, we use `caret::train` to choose $\lambda = 0.000123$. According to the documentation for `caret::train`, `caret::train` "sets up a grid of tuning parameters for a number of classification and regression routines, fits each model, and calculates a resampling based performance measure". Our inputs include our formula, our training data, the method "glmnet", the metric "F1_measure", a `trainControl` object, and a data frame containing one row for each combination of $\alpha = 0$ and value of λ to be evaluated. We construct the `trainControl` object with method "cv" denoting resampling method Cross Validation and with summary function `calculate_F1_measure` to compute F1 measures across resamples. $\alpha = 0$ indicates a combination of ridge regression and lasso regression that is effectively ridge regression.

λ is a constant that determines the importance of squares of coefficients to the quantity to be minimized when conducting LRR. We may use `glmnet::glmnet` to choose a sequence of values of λ to be evaluated. According to the documentation for `glmnet::glmnet`, `glmnet` fits "a generalized linear model via penalized maximum likelihood." Our inputs include a training matrix of predictor values, a vector of response values, an indication that the generalized linear model is a LR model, and $\alpha = 0$.

Unfortunately, `glmnet` chooses a sequence of values of λ with a minimum of 0.0035. For our optimal formula, `caret::train` considers the optimal value of λ to be this minimum. We assume that the true optimal value of λ is less than 0.0035. We use `glmnet::cv.glmnet` to graph Misclassification Error Vs. $\ln(\lambda)$ for $-9 < \ln(\lambda) < -8$. Misclassification error is an approximation of F1 measure. Our software fails to provide graphs of Performance Metrics Vs. Threshold and PR and ROC curves for some values of $\ln(\lambda)$ less than -9. We assume that the optimal value of λ is less than $\exp(-9) = 0.000123$.

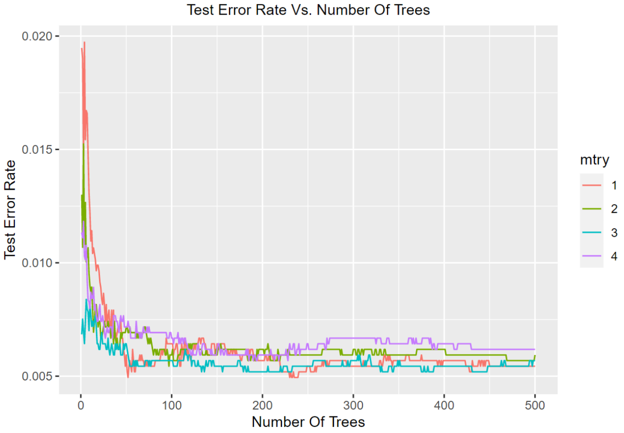


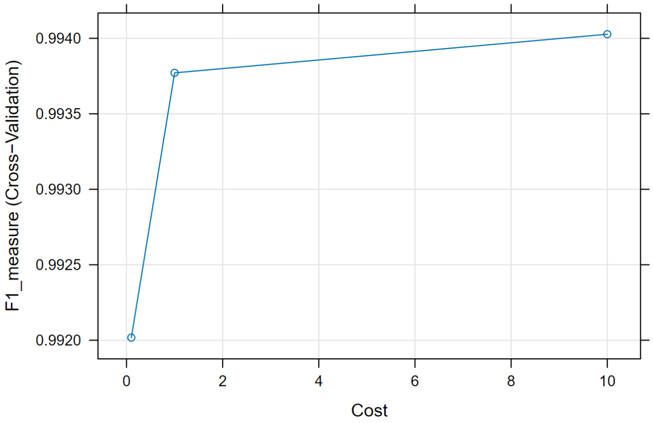


K Nearest Neighbors Classifiers During bidirectional selection of KNN classifiers, for each formula, we search for one value of hyperparameter K to be used in training a KNN classifier on 9 cross-validation folds. For our optimal formula, we use `caret::train` to choose $K = 3$ for all of our KNN classifiers. Our inputs include our formula, our training data, the method “knn”, the metric “F1_measure”, a `trainControl` object, and a data frame containing one row for each value of K to be evaluated. Dr. Gedeck recommended evaluating $K \in [1, 25]$.

See above graph, for our optimal formula, of average F1 measure vs. K for 10 cross-validated KNN classifiers. `caret::train` chose the value of K corresponding to the maximum average F1 measure. This value of K will be used in training a KNN classifier on 9 cross-validation folds.

Random Forest Classifiers During bidirectional selection of RF classifiers, for each formula, we search for one value of hyperparameter *mtry* and one value of hyperparameter *ntree* to be used in training an RF classifier on 9 cross-validation folds. We create plots of average test error rate vs. *ntree* for different values of *mtry*. Values of *mtry* vary from 1 to the number of predictors in the formula. Values of *ntree* vary from 1 to 500. We choose values $mtry = 1$ and $ntree = 52$ corresponding to the minimum test error rate.



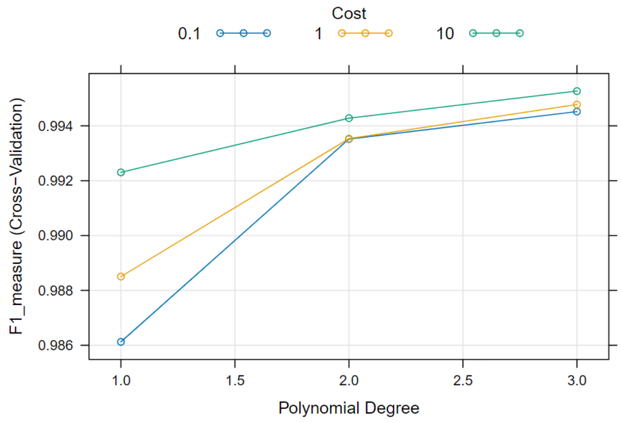


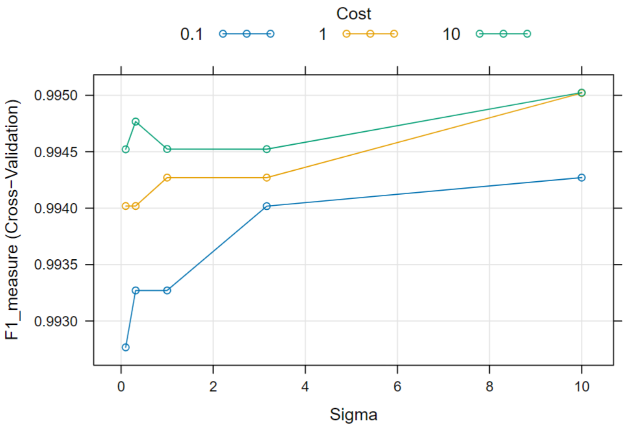
Support-Vector Machines With Linear Kernels During bidirectional selection of SVMWLK's, for each formula, we search for one value of hyperparameter C to be used in training a SVMWLK on 9 cross-validation folds. For our optimal formula, we use `caret::train` to choose $C = 10$ for all of our SVMWLK's. Our inputs include our formula, our training data, the method "svmLinear", the metric "F1_measure", a `trainControl` object, and a data frame containing one row for each value of C to be evaluated. Dr. Gedeck recommended evaluating $C \in \left\{10^{-3}, 10^{-2.75}, \cdots, 10^{2.75}, 10^{3}\right\}$. Given time limits, we choose $C \in \left\{10^{-1}, 10^{0}, 10^{1}\right\}$.

See above graph, for our optimal formula, of average F1 measure vs. C for 10 cross-validated SVMWLK's. `caret::train` chose the value of C corresponding to the maximum average F1 measure. This value of C will be used in training an SVMWLK on 9 cross-validation folds.

Support-Vector Machines With Polynomial Kernels During bidirectional selection of SVMWPK's, for each formula, we search for one value of hyperparameter C and one value of hyperparameter d to be used in training a SVMWPK on 9 cross-validation folds. For our optimal formula, we use `caret::train` to choose $C = 10$ and $d = 3$ for all of our SVMWPK's. Our inputs include our formula, our training data, the method "svmPoly", the metric "F1_measure", a `trainControl` object, and a data frame containing one row for each value of C and each value of d to be evaluated. Dr. Gedeck recommended evaluating $C \in \left\{10^{-3}, 10^{-2.75}, \cdots, 10^{2.75}, 10^{3}\right\}$ and $d \in \{1, 2, 3, 4, 5, 6\}$. Given time limits, we choose $C \in \left\{10^{-1}, 10^{0}, 10^{1}\right\}$ and $d \in \{1, 2, 3\}$.

See below graph, for our optimal formula, of average F1 measure vs. d for different values of C for 10 cross-validated SVMWPK's. `caret::train` chose the values of C and d corresponding to the maximum average F1 measure. These values of C and d will be used in training an SVMWLK on 9 cross-validation folds.





Support-Vector Machines With Radial Kernels During bidirectional selection of SVMWRK's, for each formula, we search for one value of hyperparameter C and one value of hyperparameter γ to be used in training a SVMWRK on 9 cross-validation folds. For our optimal formula, we use `caret::train` to choose $C = 10$ and $\gamma = 10$ for all of our SVMWPK's. Our inputs include our formula, our training data, the method "svmPoly", the metric "F1_measure", a `trainControl` object, and a data frame containing one row for each value of C and each value of d to be evaluated. Dr. Gedeck recommended evaluating $C \in \left\{10^{-3}, 10^{-2.75}, \cdots, 10^{2.75}, 10^{3}\right\}$ and $\gamma \in \{10^{-1}, 10^{-0.5}, 10^{0}, 10^{0.5}, 10^{1}\}$. Given time limits, we choose $C \in \left\{10^{-1}, 10^{0}, 10^{1}\right\}$ and $\gamma \in \{10^{-1}, 10^{0}, 10^{1}\}$.

See above graph, for our optimal formula, of average F1 measure vs. γ for different values of C for 10 cross-validated SVMWRK's. `caret::train` chose the values of C and γ corresponding to the maximum average F1 measure. These values of C and γ will be used in training an SVMWRK on 9 cross-validation folds.

Validation / Holdout Test

An optimal classifier of a given type is a classifier that yields a high F1 measure relative to other classifiers of that type. An optimal classifier has certain type, formula, primary hyperparameter, and secondary hyperparameter. We determine optimal LR and LRR classifiers, LDA's, QDA's, KNN classifiers, RF's, SVMWLK's, SVMWPK's, and SVMWRK's per cross validation each with good maximum average F1 measure greater than 0.89.

That being said, the number of observations in our training data frame, $63,241$, is 3.2 percent of the number of observations in our holdout data frame, $2,004,177$. Our holdout distributions of intensity of colors *Red*, *Green* and *Blue* are normal and broad relative to our training distributions. The proportion of training observations that correspond to blue tarps is 4.4 times the proportion of holdout observations that correspond to blue tarps. We expect that, due to differences in our training and holdout distributions, a binary classifier trained on our training data and holdout tested on our holdout data will have a maximum F1 measure that is less than that of a classifier cross validated on our training data. While we recommend cross validating on our composite data or cross validating on 90 percent of our composite data and holdout testing on the other 10 percent, we train on our training data and holdout test our optimal classifiers on our holdout data. In this case, holdout testing will provide us performance metrics for a data frame that is larger and perhaps more realistic than our training data frame. Holdout testing will provide us ideas of the robustness of our optimal classifiers across data frames. We may factor holdout testing results into our recommendation of an optimal binary classifier.

To holdout test our optimal classifiers, we train each optimal classifier on our training data and visualize performance metrics on our holdout data.

Evaluating And Comparing Classifiers

For each type of model and optimal formula, primary hyperparameter, and secondary hyperparameter per cross validation, we provide plot of Average Performance Metrics Vs. Threshold, Precision-Recall Curve, ROC curve, and data frame of optimal performance metrics per cross validation. We also provide plot of Performance Metrics vs. Threshold, Precision-Recall Curve, ROC curve, and data frame of optimal performance metrics per holdout testing.

Big picture:

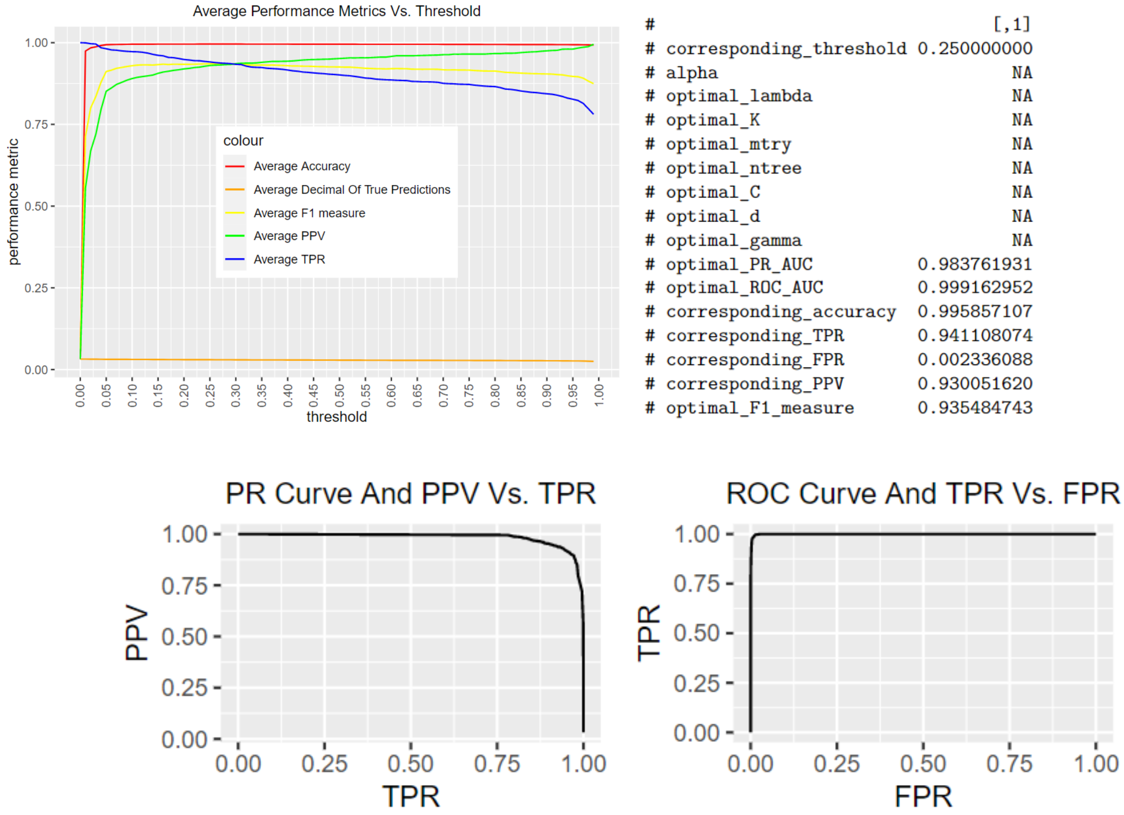
- Accuracy decreases across KNN CV, SVMWRK HT, SVMWPK CV, SVMWPK HT, SVMWPK CV, SVMWLK CV, RF CV, KNN HT, LR CV, QDA CV, RF HT, LR HT, SVMWLK HT, LRR CV, QDA HT, LRR HT, LDA CV, LDA HT.
- F1 measure decreases across KNN CV, SVMWRK CV, SVMWPK CV, SVMWLK CV, RF CV, LR CV, QDA CV, LRR CV, LDA CV, LRR HT, LR HT, SVMWLK HT, SVMWPK HT, KNN HT, RF HT, LDA HT, QDA HT, SVMWRK HT.
- PPV decreases across LRR CV, SVMWRK CV, QDA CV, RF CV, SVMWPK CV, SVMWLK CV, KNN CV, LR CV, LRR HT, RF HT, LR HT, LDA CV, LDA HT, QDA HT, SVMWLK HT, SVMWPK HT, KNN HT, SVMWRK HT.
- TPR decreases across SVMWLK HT, SVMWPK LT, LR HT, LRR HT, SVMWRK CV, SVMWLK CV, SVMWPK CV, RF CV, KNN CV, LR CV, QDA CV, LDA CV, LRR CV, KNN HT, LDA HT, RF HT, QDA HT, SVMWRK HT.
- PR AUC decreases across LR CV (0.984), SVMWLK CV (0.984), RF CV (0.984), SVMWRK CV (0.983), LRR CV (0.982), SVMWPK CV (0.980), KNN CV (0.979), LRR HT (0.963), QDA CV (0.959), LR HT (0.918), SVMWLK HT (0.917), LDA CV (0.905), SVMWPK HT (0.879), LDA HT (0.739), RF HT (0.710), KNN HT (0.669), QDA HT (0.504), SVWRK HT (0.294).
- ROC AUC decreases across LRR HT (0.9996), SVMWLK CV (0.9994), LRR CV (0.9992), LR CV (0.9992), SVMWLK HT (0.9991), LR HT (0.9991), SVMWLK CV (0.998), SVMWPK HT (0.998), KNN CV (0.994), RF CV (0.9939), SVMWRK CV (0.990), QDA CV (0.985), RF HT (0.979), LDA HT (0.953), LDA CV (0.950), KNN HT (0.933), SVMWRK HT (0.793), QDA HT (0.786).

Logistic Regression Classifiers Per LR cross validation:

- Our optimal LR classifier has formula $Indicator \sim normalize\left(Blue^2\right) + normalize\left(Red^2\right) + normalize\left(\sqrt{Blue}\right)$.
- The maximum average F1 measure is 0.936.
- The corresponding threshold is 0.25.
- Average accuracy is L-shaped and nearly 1 for about 95 percent of thresholds.
- Average F1 measure is U-shaped and nearly 0.936 for thresholds between about 0.1 and about 0.4.
- Average PPV increases logarithmically.
- Average TPR decreases linearly from a threshold of 0 to about 0.95.
- A Precision-Recall curve has an AUC of 0.984.
- An ROC curve has an AUC of 0.99920.

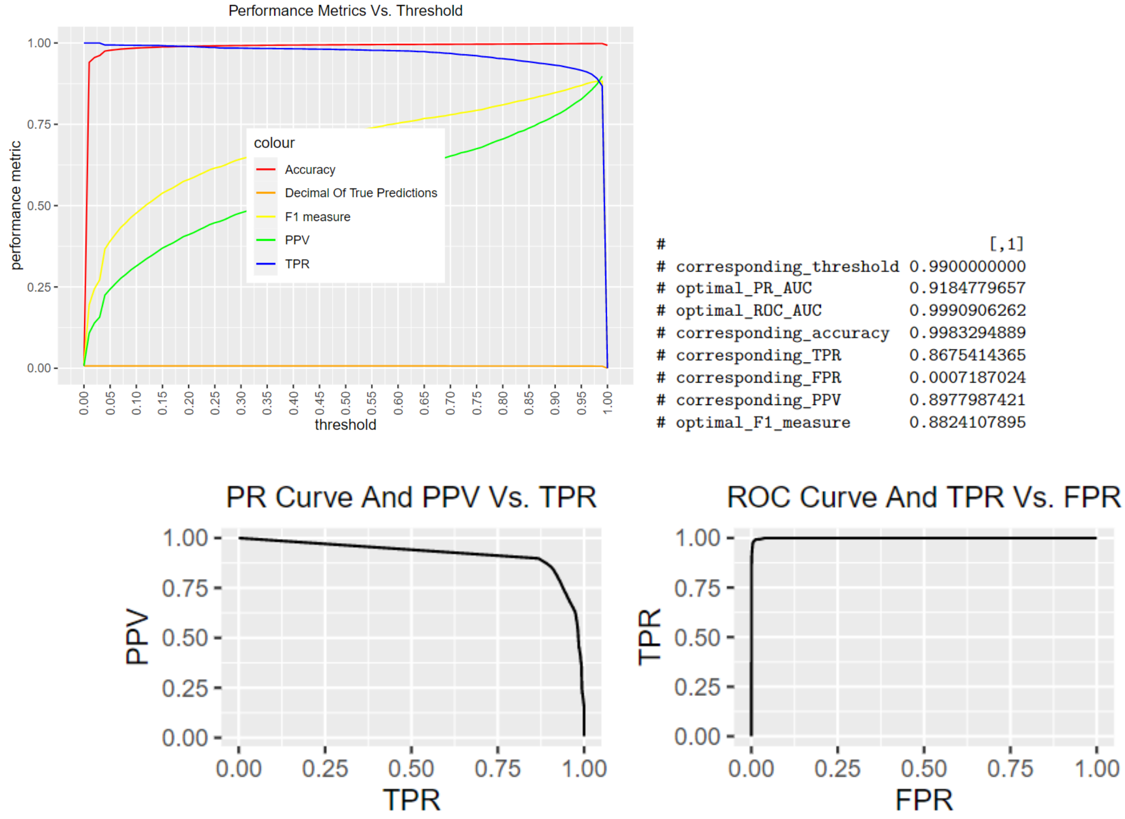
Per LR holdout testing:

- Our optimal LR classifier has a maximum F1 measure of 0.882.
- The corresponding threshold is 0.990.
- Accuracy is L-shaped and nearly 1 for about 85 percent of thresholds.
- F1 measure increases in the manner of $y = tan(x)$.
- PPV increases in the manner of $y = tan(x)$.
- TPR is L-shaped.
- A Precision-Recall curve has an AUC of 0.918.
- An ROC curve has an AUC of 0.99909.



```
#                                  [,1]
# corresponding_threshold 0.250000000
# alpha                            NA
# optimal_lambda                   NA
# optimal_K                        NA
# optimal_mtry                     NA
# optimal_ntree                    NA
# optimal_C                        NA
# optimal_d                        NA
# optimal_gamma                    NA
# optimal_PR_AUC          0.983761931
# optimal_ROC_AUC         0.999162952
# corresponding_accuracy  0.995857107
# corresponding_TPR       0.941108074
# corresponding_FPR       0.002336088
# corresponding_PPV       0.930051620
# optimal_F1_measure      0.935484743
```

Figure 5: PR And ROC Curves: Cross Validation: LR



```
#                                   [,1]
# corresponding_threshold 0.9900000000
# optimal_PR_AUC          0.9184779657
# optimal_ROC_AUC         0.9990906262
# corresponding_accuracy  0.9983294889
# corresponding_TPR       0.8675414365
# corresponding_FPR       0.0007187024
# corresponding_PPV       0.8977987421
# optimal_F1_measure      0.8824107895
```

Figure 6: PR And ROC Curves: Holdout Testing: LR

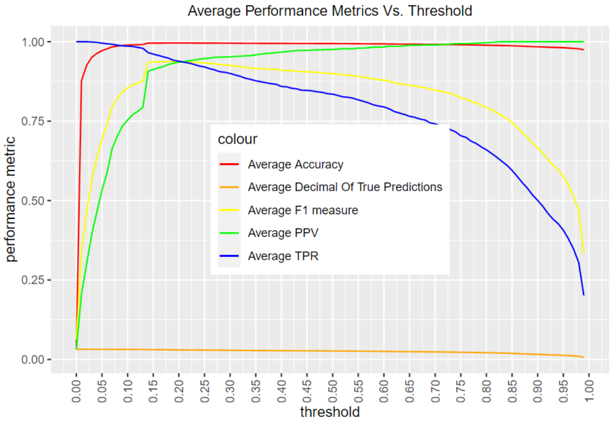
Logistic Ridge Regression Classifiers Per LRR cross validation:

- Our optimal LRR classifier has formula $Indicator \sim normalize[\ln(Blue)] + normalize\left(\sqrt{Red}\right) + normalize[\ln(Green)] + normalize[\ln(Red)]$.
- Our optimal value for λ is 0.000123.
- The maximum average F1 measure is 0.938.
- The corresponding threshold is 0.18.
- Average accuracy is L-shaped and nearly 1 for about 87.5 percent of thresholds.
- Average F1 measure is U-shaped and nearly 0.938 for maybe 11 percent of thresholds.
- Average PPV increases logarithmically.
- Average TPR decreases in the manner of $y = sqrt(x)$.
- A Precision-Recall curve has an AUC of 0.982.
- An ROC curve has an AUC of 0.99923.

Per LRR holdout testing:

- Our optimal LRR classifier has a maximum F1 measure of 0.917.
- The corresponding threshold is 0.18.
- Accuracy is L-shaped and nearly 1 for about 80 percent of thresholds.
- F1 measure is a mirror image of F1 measure per cross validation.
- PPV looks like a pointer finger.
- TPR is L-shaped.
- A Precision-Recall curve has an AUC of 0.963.

- An ROC curve has an AUC of 0.9996.



```
#                                  [,1]
# corresponding_threshold 0.1800000000
# alpha                   0.0000000000
# optimal_lambda          0.0001234098
# optimal_K                         NA
# optimal_mtry                      NA
# optimal_ntree                     NA
# optimal_C                         NA
# optimal_d                         NA
# optimal_gamma                     NA
# optimal_PR_AUC          0.9821724543
# optimal_ROC_AUC         0.9992398125
# corresponding_accuracy  0.9959994325
# corresponding_TPR       0.9480048364
# corresponding_FPR       0.0024178550
# corresponding_PPV       0.9286975209
# optimal_F1_measure      0.9380495281
```

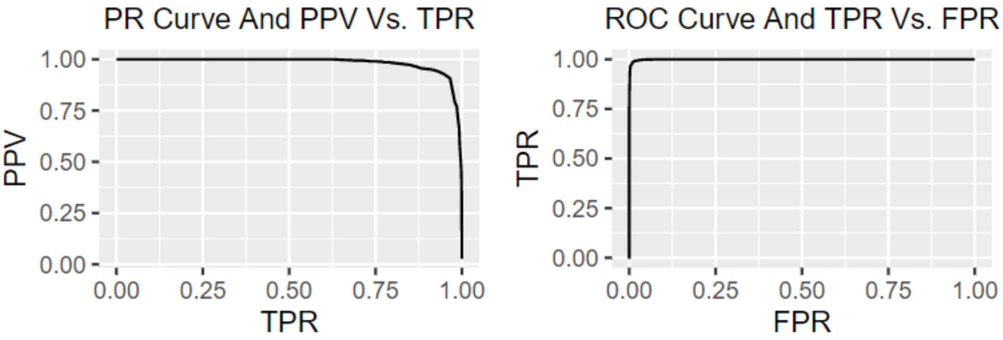
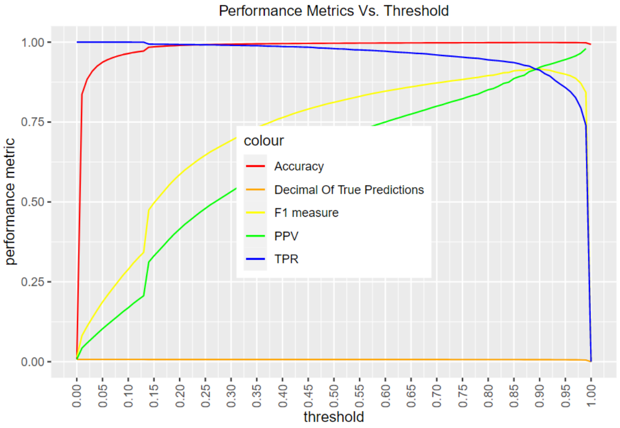


Figure 7: PR And ROC Curves: Cross Validation: LRR



```
#                                 [,1]
# corresponding_threshold 0.9000000000
# optimal_PR_AUC         0.9632743099
# optimal_ROC_AUC        0.9995993358
# corresponding_accuracy 0.9988049958
# corresponding_TPR      0.9127071823
# corresponding_FPR      0.0005684283
# corresponding_PPV      0.9211681885
# optimal_F1_measure     0.9169181670
```

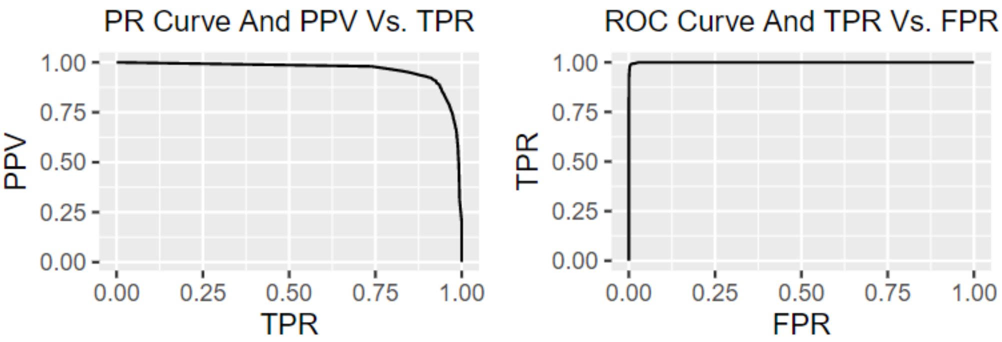


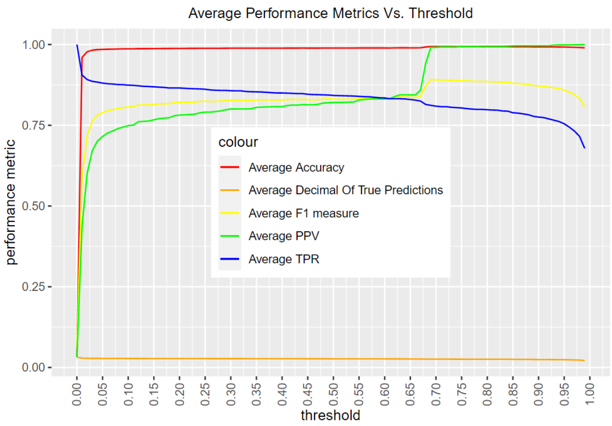
Figure 8: PR And ROC Curves: Holdout Testing: LRR

Linear Discriminant Analyses Per LDA cross validation:

- Our optimal LDA classifier has formula $Indicator \sim normalize(Blue) + normalize\left(Red^2\right) + normalize\left(Green^2\right)$.
- The maximum average F1 measure is 0.892.
- The corresponding threshold is 0.69.
- Average accuracy is L-shaped and nearly 1 for about 32.5 percent of thresholds.
- Average F1 measure is U-shaped and nearly 0.892 for maybe 6 percent of thresholds.
- Average PPV increases logarithmically and jumps to 1 for a threshold of 0.69.
- Average TPR decreases in the manner of $y = -tan(x)$.
- A Precision-Recall curve has an AUC of 0.905.
- An ROC curve has an AUC of 0.950.

Per LDA holdout testing:

- Our optimal LDA classifier has a maximum F1 measure of 0.742.
- The corresponding threshold is 0.68.
- Accuracy is L-shaped and nearly 1 for about 32.5 percent of thresholds.
- F1 measure increases stepwise.
- PPV increases stepwise.
- TPR decreases in the manner of $y = -tan(x)$.
- A Precision-Recall curve has an AUC of 0.739.
- An ROC curve has an AUC of 0.953.



```
#                                    [,1]
# corresponding_threshold 0.6900000000
# alpha                             NA
# optimal_lambda                    NA
# optimal_K                         NA
# optimal_mtry                      NA
# optimal_ntree                     NA
# optimal_C                         NA
# optimal_d                         NA
# optimal_gamma                     NA
# optimal_PR_AUC          0.9046700339
# optimal_ROC_AUC         0.9499424209
# corresponding_accuracy  0.9937224326
# corresponding_TPR       0.8116922240
# corresponding_FPR       0.0002449892
# corresponding_PPV       0.9910700503
# optimal_F1_measure      0.8919403753
```

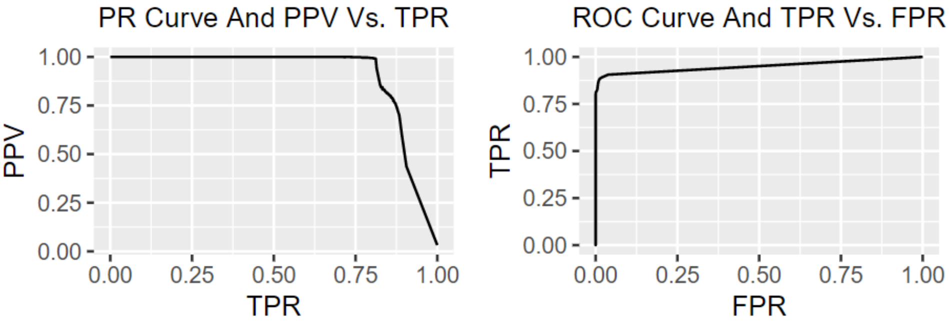
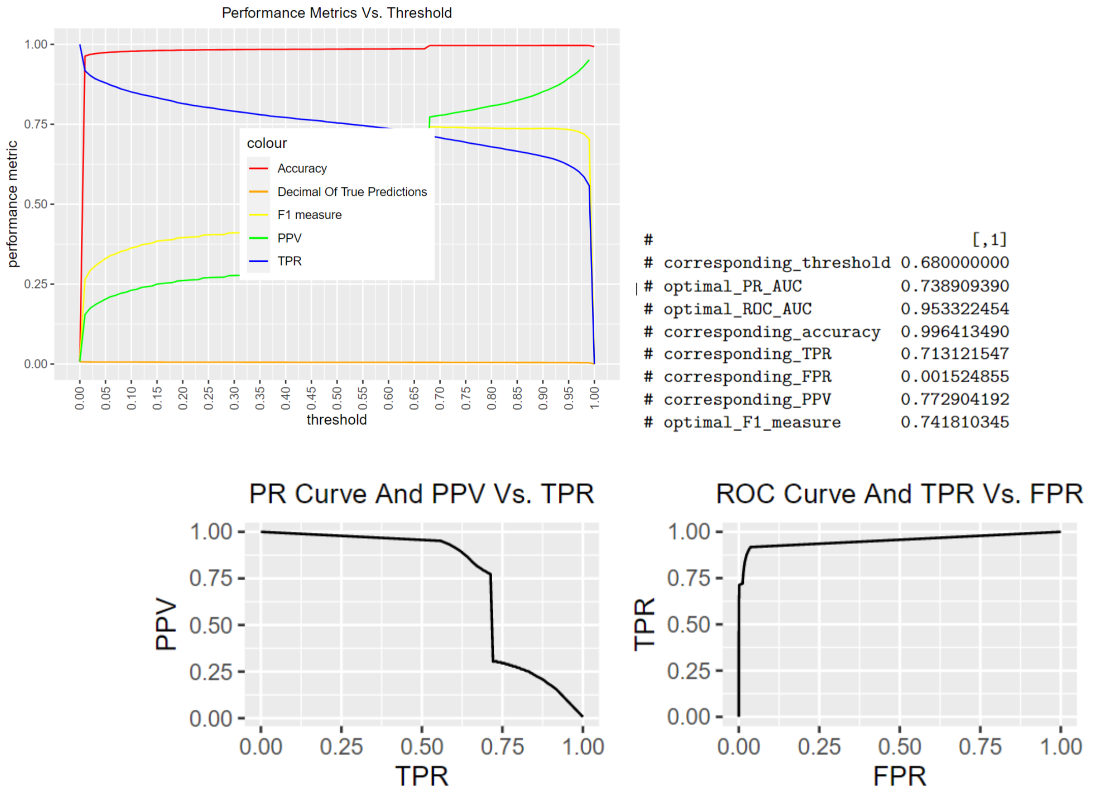


Figure 9: PR And ROC Curves: Cross Validation: LDA



```
#                             [,1]
# corresponding_threshold 0.680000000
# optimal_PR_AUC          0.738909390
# optimal_ROC_AUC         0.953322454
# corresponding_accuracy  0.996413490
# corresponding_TPR       0.713121547
# corresponding_FPR       0.001524855
# corresponding_PPV       0.772904192
# optimal_F1_measure      0.741810345
```

Figure 10: PR And ROC Curves: Holdout Testing: LDA

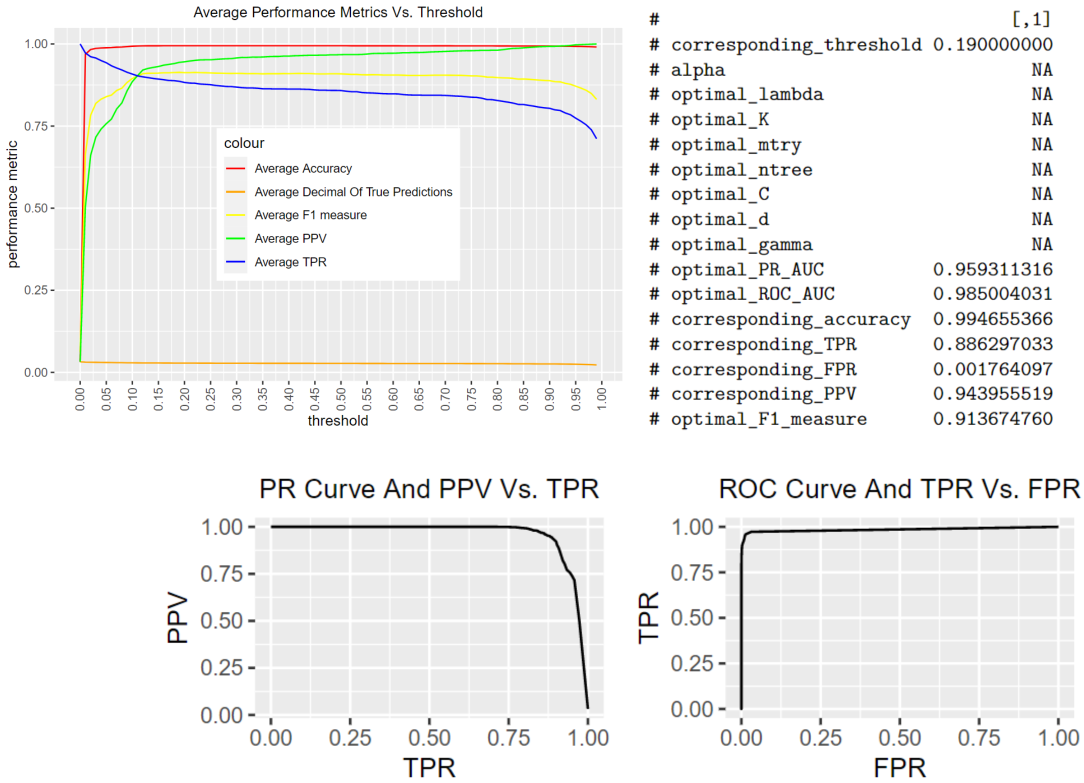
Quadratic Discriminant Analyses Per QDA cross validation:

- Our optimal QDA classifier has formula $Indicator \sim normalize(Blue) + normalize\left(Red^2\right) + normalize\left(Red : Blue\right)$.
- The maximum average F1 measure is 0.914.
- The corresponding threshold is 0.19.
- Average accuracy is L-shaped and nearly 1 for about 90 percent of thresholds.
- Average F1 measure is U-shaped and nearly 0.914 for maybe 58 percent of thresholds.
- Average PPV increases logarithmically.
- Average TPR decreases in the manner of $y = -tan(x)$.
- A Precision-Recall curve has an AUC of 0.959.

- An ROC curve has an AUC of 0.985.

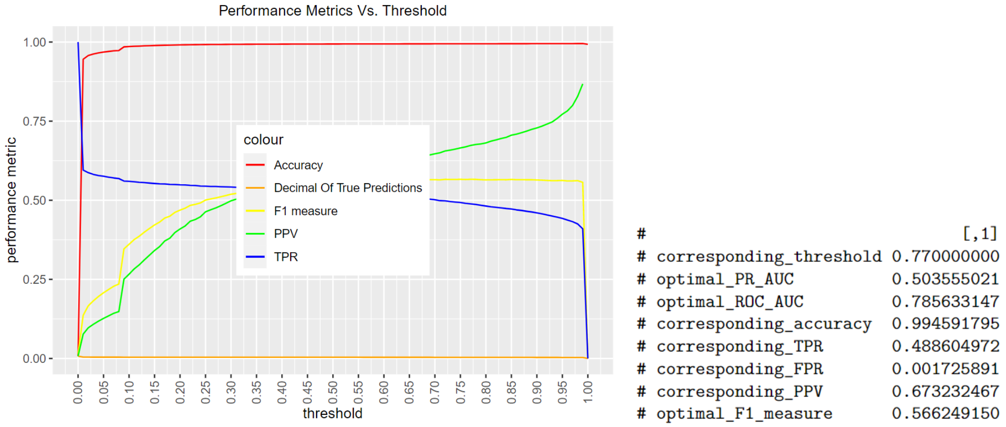
Per QDA holdout testing:

- Our optimal QDA classifier has a maximum F1 measure of 0.567.
- The corresponding threshold is 0.77.
- Accuracy is L-shaped and nearly 1 for about 80 percent of thresholds.
- F1 measure is U-shaped.
- PPV increases in the manner of $y = tan(x)$.
- TPR decreases in the manner of $y = -tan(x)$.
- A Precision-Recall curve has an AUC of 0.504.
- An ROC curve has an AUC of 0.786.



```
#                                   [,1]
# corresponding_threshold 0.190000000
# alpha                            NA
# optimal_lambda                   NA
# optimal_K                        NA
# optimal_mtry                     NA
# optimal_ntree                    NA
# optimal_C                        NA
# optimal_d                        NA
# optimal_gamma                    NA
# optimal_PR_AUC          0.959311316
# optimal_ROC_AUC         0.985004031
# corresponding_accuracy  0.994655366
# corresponding_TPR       0.886297033
# corresponding_FPR       0.001764097
# corresponding_PPV       0.943955519
# optimal_F1_measure      0.913674760
```

Figure 11: PR And ROC Curves: Cross Validation: QDA



```
#                                   [,1]
# corresponding_threshold 0.770000000
# optimal_PR_AUC          0.503555021
# optimal_ROC_AUC         0.785633147
# corresponding_accuracy  0.994591795
# corresponding_TPR       0.488604972
# corresponding_FPR       0.001725891
# corresponding_PPV       0.673232467
# optimal_F1_measure      0.566249150
```

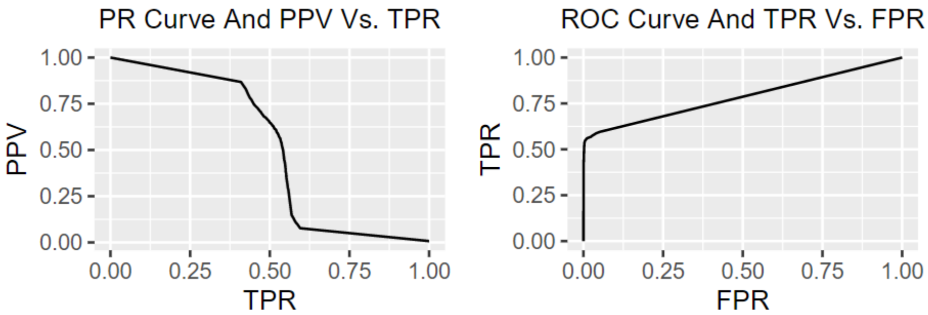


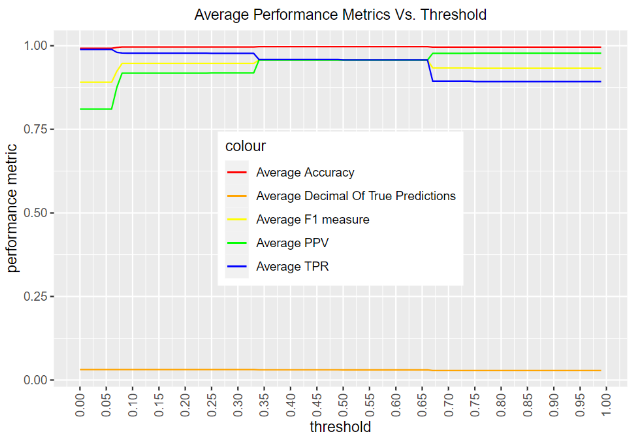
Figure 12: PR And ROC Curves: Holdout Testing: QDA

***K* Nearest Neighbors Classifiers** Per KNN cross validation:

- Our optimal KNN classifier has formula $Indicator \sim normalize(Red : Blue) + normalize(Red) + normalize[\ln(Green)]$.
- Our optimal value for K is 3.
- The maximum average F1 measure is 0.954.
- The corresponding threshold is 0.34.
- Average accuracy increases by 1 small step and is nearly 1 for 92 percent of thresholds.
- Average F1 measure increases by 1 step and is nearly 0.954 for 92 percent of thresholds.
- Average PPV increases by 3 steps.
- Average TPR decreases by 3 steps.
- A Precision-Recall curve has an AUC of 0.980.
- An ROC curve has an AUC of 0.996.

Per KNN holdout testing:

- Our optimal KNN classifier has a maximum F1 measure of 0.610.
- The corresponding threshold is 0.5.
- Accuracy increases by 1 small step and is nearly 1 for 92 percent of thresholds.
- F1 measure increases by 1 step and is nearly 0.610 for 92 percent of thresholds.
- PPV increases by 1 step and is nearly 0.491 for 92 percent of thresholds.
- TPR decreases by 1 small step and is nearly 0.804 for 92 percent of thresholds.
- A Precision-Recall curve has an AUC of 0.628.
- An ROC curve has an AUC of 0.901.



```
#                                 [,1]
# corresponding_threshold 0.340000000
# alpha                            NA
# optimal_lambda                   NA
# optimal_K               3.000000000
# optimal_mtry                     NA
# optimal_ntree                    NA
# optimal_C                        NA
# optimal_d                        NA
# optimal_gamma                    NA
# optimal_PR_AUC          0.978887719
# optimal_ROC_AUC         0.994095396
# corresponding_accuracy  0.997327693
# corresponding_TPR       0.958823576
# corresponding_FPR       0.001404108
# corresponding_PPV       0.957054909
# optimal_F1_measure      0.957848475
```

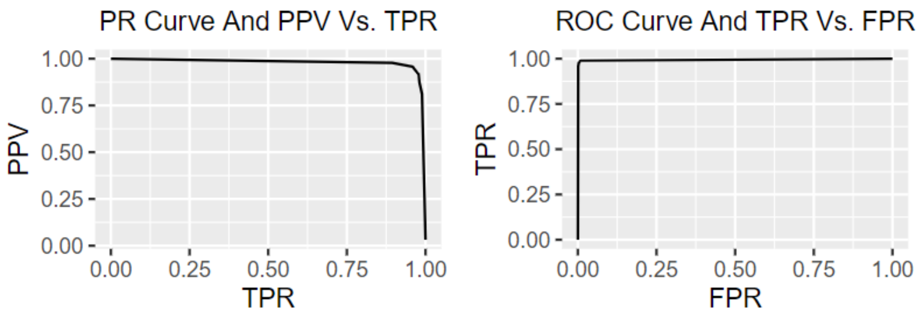
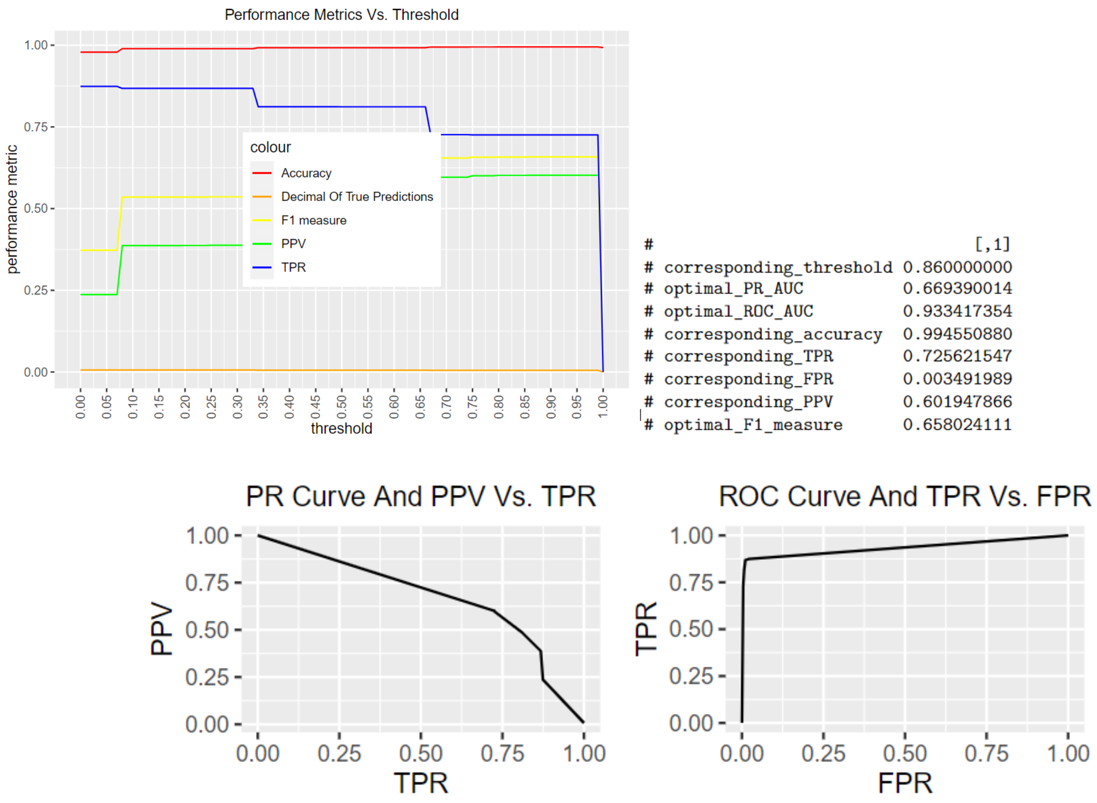


Figure 13: PR And ROC Curves: Cross Validation: KNN



```
#                              [,1]
# corresponding_threshold 0.860000000
# optimal_PR_AUC          0.669390014
# optimal_ROC_AUC         0.933417354
# corresponding_accuracy  0.994550880
# corresponding_TPR       0.725621547
# corresponding_FPR       0.003491989
# corresponding_PPV       0.601947866
# optimal_F1_measure      0.658024111
```

Figure 14: PR And ROC Curves: Holdout Testing: KNN

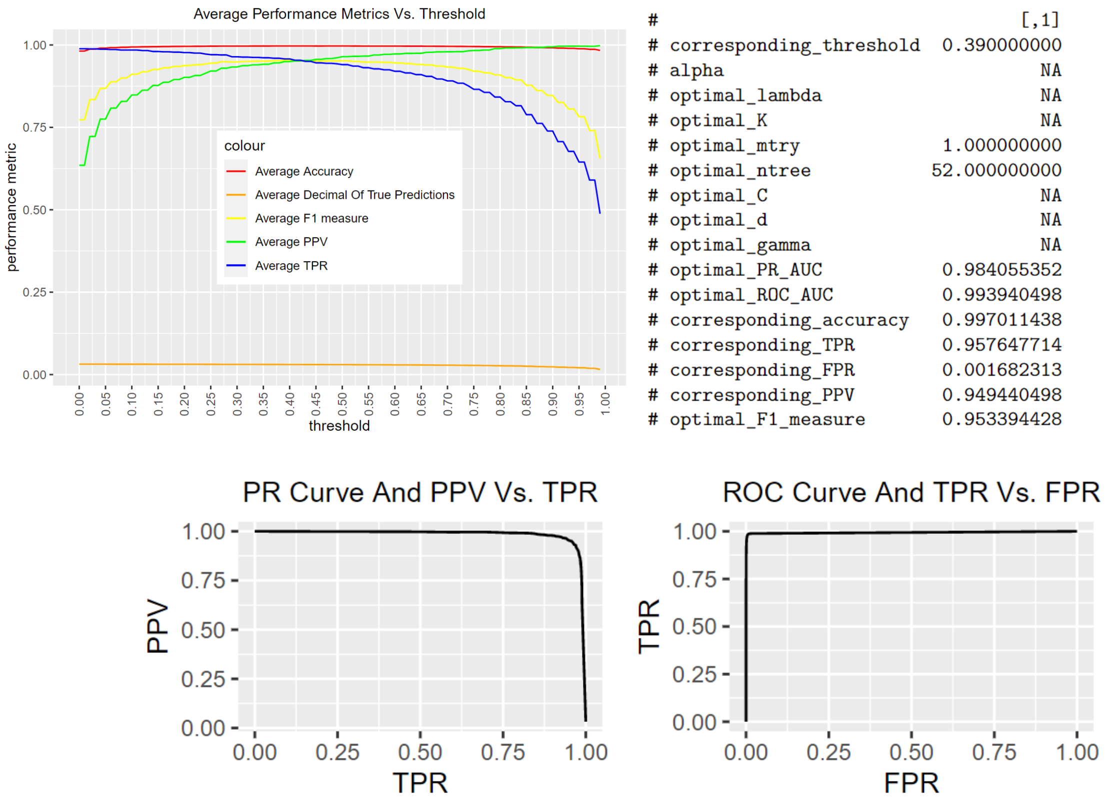
Random Forests Per RF cross validation:

- Our optimal RF has formula $Indicator \sim normalize\left(Red^2\right) + normalize(Red : Green) + normalize(Green : Blue) + normalize(Green)$.
- Our optimal value for $mtry$ is 1.
- Our optimal value for $ntree$ is 52.
- The maximum average F1 measure is 0.953.
- The corresponding threshold is 0.39.
- Average accuracy is U-shaped and is nearly 1 for 82 percent of thresholds.
- Average F1 measure is U-shaped and is nearly 0.953 for 35 percent of thresholds.
- Average PPV increases logarithmically.

- Average TPR decreases logarithmically.
- A Precision-Recall curve has an AUC of 0.984.
- An ROC curve has an AUC of 0.994.

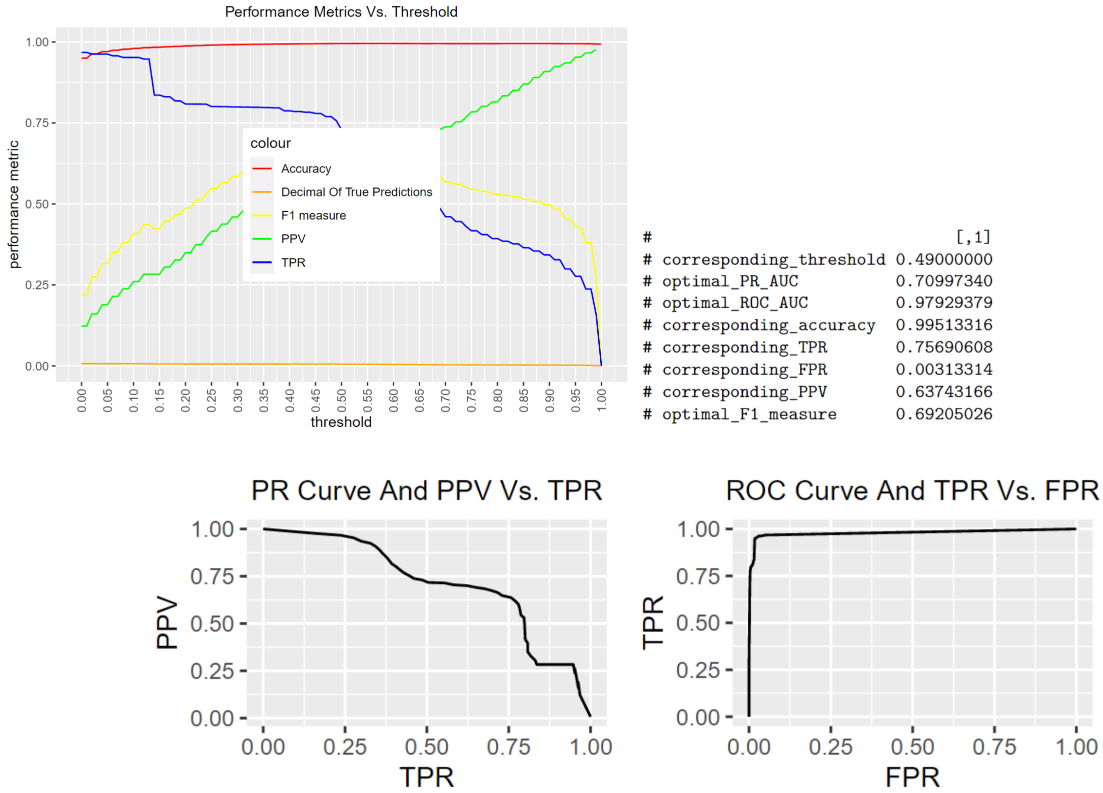
Per RF holdout testing:

- Our optimal RF has a maximum F1 measure of 0.692.
- The corresponding threshold is 0.49.
- Accuracy increases as a cantilever and is nearly 1 for 75 percent of thresholds.
- F1 measure is U-shaped and is nearly 0.692 for less than 35 percent of thresholds.
- PPV increases linearly with a few terraces.
- TPR decreases by 3 or 4 steps.
- A Precision-Recall curve has an AUC of 0.710.
- An ROC curve has an AUC of 0.979.



```
#                                    [,1]
# corresponding_threshold     0.390000000
# alpha                                NA
# optimal_lambda                       NA
# optimal_K                            NA
# optimal_mtry                1.000000000
# optimal_ntree              52.000000000
# optimal_C                            NA
# optimal_d                            NA
# optimal_gamma                        NA
# optimal_PR_AUC              0.984055352
# optimal_ROC_AUC             0.993940498
# corresponding_accuracy      0.997011438
# corresponding_TPR           0.957647714
# corresponding_FPR           0.001682313
# corresponding_PPV           0.949440498
# optimal_F1_measure          0.953394428
```

Figure 15: PR And ROC Curves: Cross Validation: RF



```
#                              [,1]
# corresponding_threshold 0.49000000
# optimal_PR_AUC          0.70997340
# optimal_ROC_AUC         0.97929379
# corresponding_accuracy  0.99513316
# corresponding_TPR       0.75690608
# corresponding_FPR       0.00313314
# corresponding_PPV       0.63743166
# optimal_F1_measure      0.69205026
```

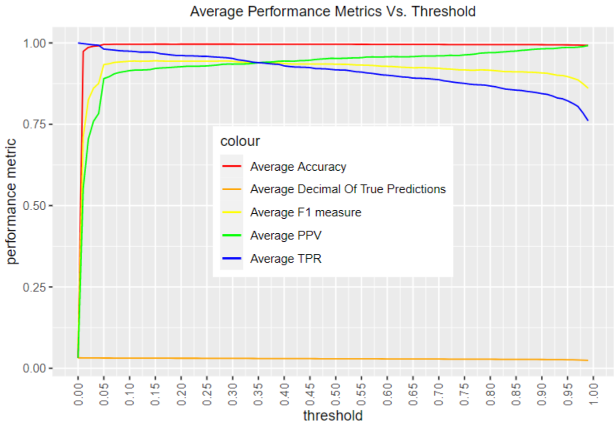
Figure 16: PR And ROC Curves: Holdout Testing: RF

Support-Vector Machines With Linear Kernels Per SVMWLK cross validation:

- Our optimal RF has formula $Indicator \sim normalize\left(Red^2\right) + normalize(Red : Green) + normalize(Green : Blue) + normalize(Green)$.
- Our optimal value for C is 10.
- The maximum average F1 measure is 0.953.
- The corresponding threshold is 0.15.
- Average accuracy is U-shaped and is nearly 1 for 82 percent of thresholds.
- Average F1 measure is U-shaped and is nearly 0.953 for 35 percent of thresholds.
- Average PPV increases logarithmically.
- Average TPR decreases logarithmically.
- A Precision-Recall curve has an AUC of 0.984.
- An ROC curve has an AUC of 0.994.

Per SVMWLK holdout testing:

- Our optimal RF has a maximum F1 measure of 0.692.
- The corresponding threshold is 0.49.
- Accuracy increases as a cantilever and is nearly 1 for 75 percent of thresholds.
- F1 measure is U-shaped and is nearly 0.692 for less than 35 percent of thresholds.
- PPV increases linearly with a few terraces.
- TPR decreases by 3 or 4 steps.
- A Precision-Recall curve has an AUC of 0.710.
- An ROC curve has an AUC of 0.979.



```
#                              [,1]
# corresponding_threshold  0.150000000
# alpha                             NA
# optimal_lambda                    NA
# optimal_K                         NA
# optimal_mtry                      NA
# optimal_ntree                     NA
# optimal_C               10.000000000
# optimal_d                         NA
# optimal_gamma                     NA
# optimal_PR_AUC           0.984226549
# optimal_ROC_AUC          0.998476597
# corresponding_accuracy   0.996410570
# corresponding_TPR        0.970282451
# corresponding_FPR        0.002727982
# corresponding_PPV        0.921533826
# optimal_F1_measure       0.945232620
```

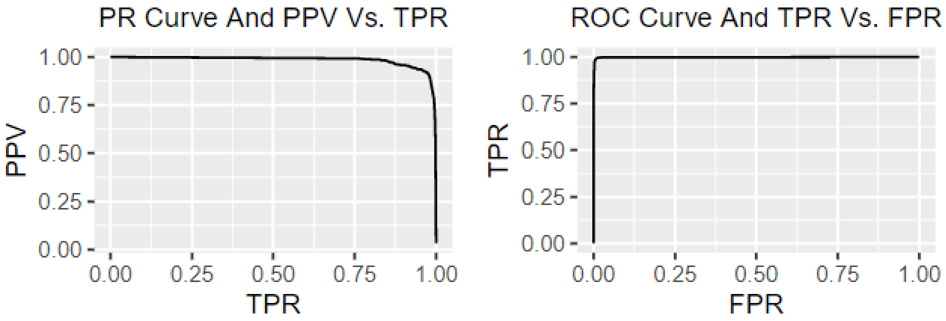
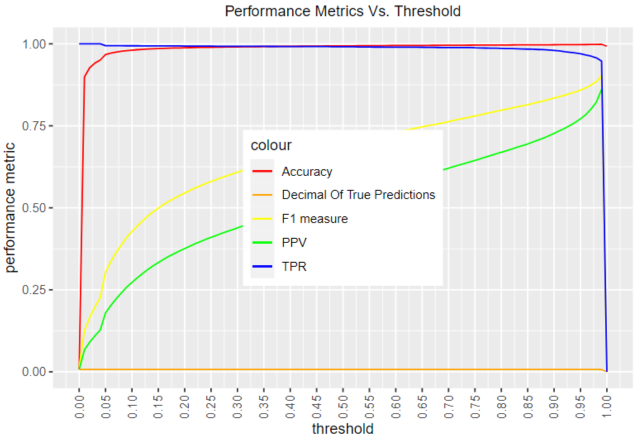


Figure 17: PR And ROC Curves: Cross Validation: SVMWLK



```
#                              [,1]
# corresponding_threshold 0.990000000
# optimal_PR_AUC          0.917028515
# optimal_ROC_AUC         0.999104180
# corresponding_accuracy  0.998516598
# corresponding_TPR       0.947306630
# corresponding_FPR       0.001110722
# corresponding_PPV       0.861241916
# optimal_F1_measure      0.902226461
```

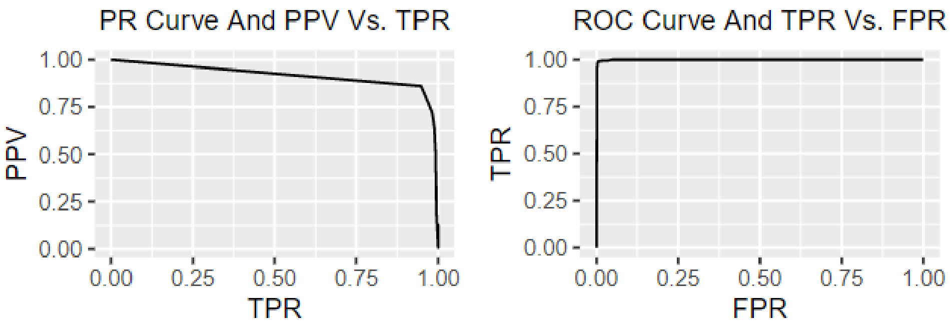


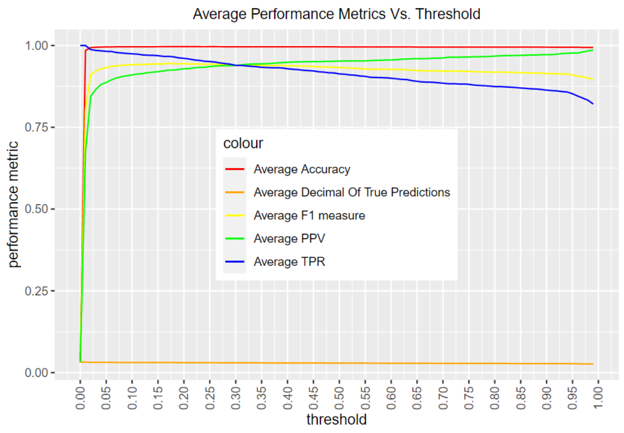
Figure 18: PR And ROC Curves: Holdout Testing: SVMWLK

Support-Vector Machines With Polynomial Kernels Per SVMWPK cross validation:

- Our optimal SVMPK has formula $Indicator \sim normalize\left(Blue^2\right) + normalize\left(Red^2\right) + normalize(Green : Blue)$.
- Our optimal value for C is 10.
- Our optimal degree and value for d is 3.
- The maximum average F1 measure is 0.945.
- The corresponding threshold is 0.17.
- Average accuracy is L-shaped and is nearly 1 for 97 percent of thresholds.
- Average F1 measure is U-shaped and is nearly 0.945 for 30 percent of thresholds.
- Average PPV increases logarithmically.
- Average TPR decreases linearly.
- A Precision-Recall curve has an AUC of 0.980.
- An ROC curve has an AUC of 0.999.

Per SVMWPK holdout testing:

- Our optimal SVMWPK has a maximum F1 measure of 0.858.
- The corresponding threshold is 0.99.
- Accuracy is L-shaped and is nearly 1 for 90 percent of thresholds.
- F1 measure increases in the manner of $y = tan(x)$.
- PPV increases in the manner of $y = tan(x)$.
- TPR is L-shaped and is nearly 0.99 for 80 percent of thresholds.
- A Precision-Recall curve has an AUC of 0.804.
- An ROC curve has an AUC of 0.998.



```
#                                   [,1]
# corresponding_threshold    0.170000000
# alpha                               NA
# optimal_lambda                      NA
# optimal_K                           NA
# optimal_mtry                        NA
# optimal_ntree                       NA
# optimal_C                 10.000000000
# optimal_d                  3.000000000
# optimal_gamma                       NA
# optimal_PR_AUC             0.980413987
# optimal_ROC_AUC            0.999435458
# corresponding_accuracy     0.996410537
# corresponding_TPR          0.966786340
# corresponding_FPR          0.002597001
# corresponding_PPV          0.924558337
# optimal_F1_measure         0.945075069
```

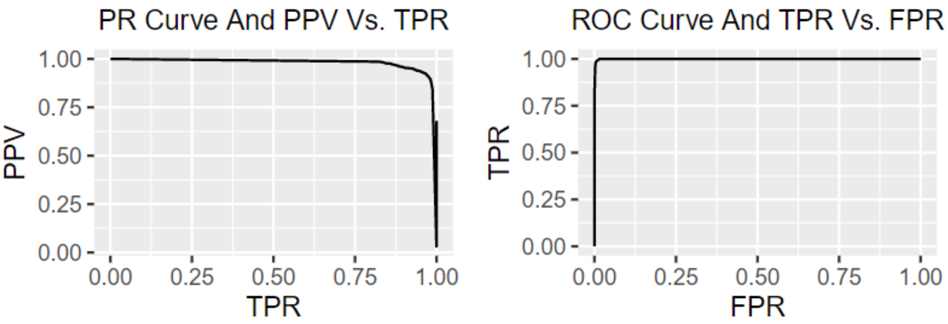
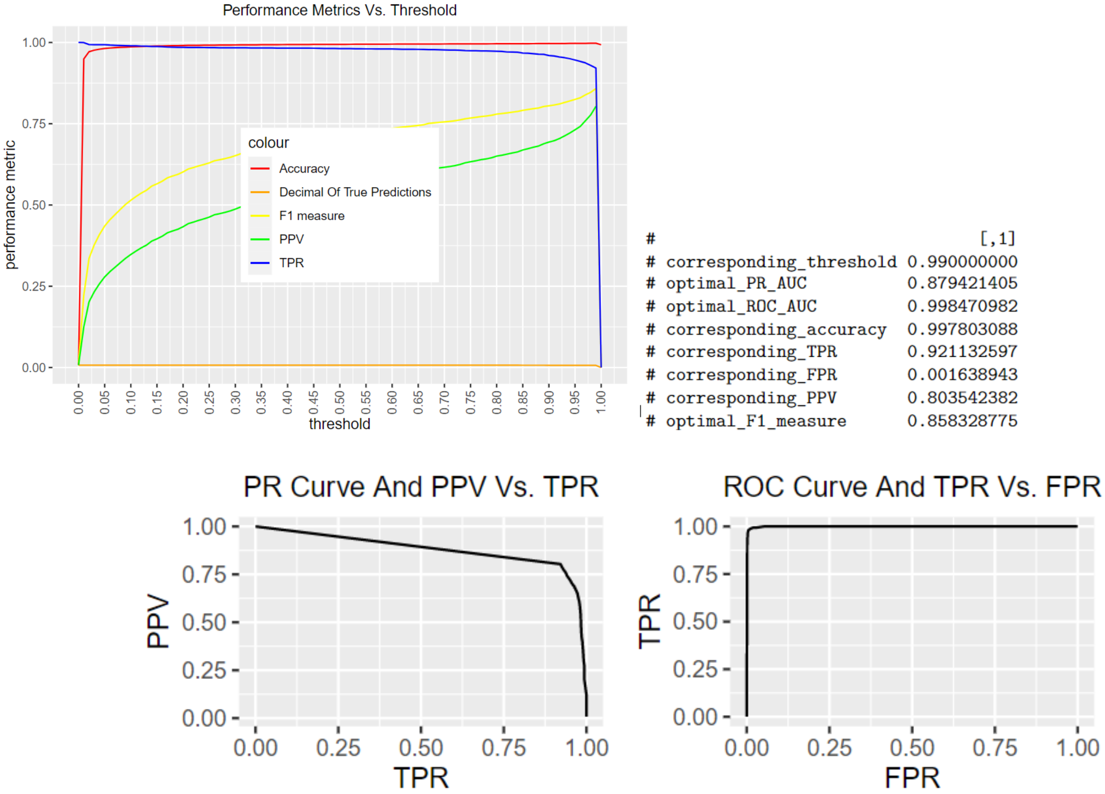


Figure 19: PR And ROC Curves: Cross Validation: SVMWPK



```
#                                [,1]
# corresponding_threshold 0.990000000
# optimal_PR_AUC          0.879421405
# optimal_ROC_AUC         0.998470982
# corresponding_accuracy  0.997803088
# corresponding_TPR       0.921132597
# corresponding_FPR       0.001638943
# corresponding_PPV       0.803542382
# optimal_F1_measure      0.858328775
```

Figure 20: PR And ROC Curves: Holdout Testing: SVMWPK

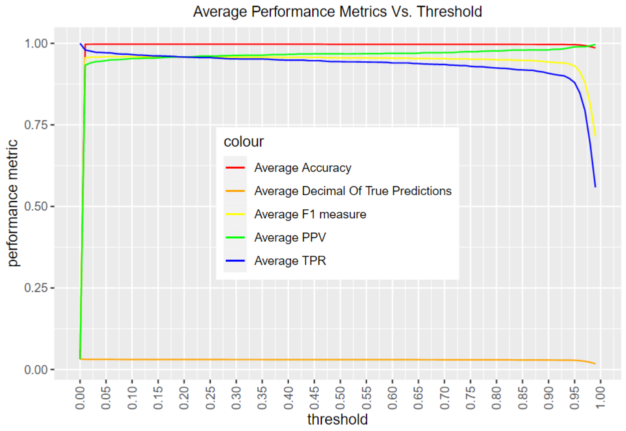
Support-Vector Machines With Radial Kernels Per SVMWRK cross validation:

- Our optimal SVMRK has formula $Indicator \sim normalize(Blue) + normalize[\ln(Red)] + normalize\left(\sqrt{Green}\right) + normalize[\ln(Blue)]$.
- Our optimal value for C is 10.
- Our optimal value of γ is 10.
- The maximum average F1 measure is 0.960.
- The corresponding threshold is 0.06.
- Average accuracy is L-shaped and is nearly 1 for 92 percent of thresholds.
- Average F1 measure is U-shaped and is nearly 0.960 for 85 percent of thresholds.
- Average PPV is L-shaped and is about 0.97 on average.

- Average TPR decreases in the manner of $y = tan(x)$.
- A Precision-Recall curve has an AUC of 0.983.
- An ROC curve has an AUC of 0.990.

Per SVMWRK holdout testing:

- Our optimal SVMWRK has a maximum F1 measure of 0.414.
- The corresponding threshold is 0.01.
- Accuracy is L-shaped and is nearly 98 for 98 percent of thresholds.
- F1 measure decreases in the manner of $y = tan(x)$.
- PPV is U-shaped and nearly 0.34 for 75 percent of thresholds.
- TPR decreases in the manner of $y = tan(x)$.
- A Precision-Recall curve has an AUC of 0.294.
- An ROC curve has an AUC of 0.793.



```
#                                   [,1]
# corresponding_threshold    0.060000000
# alpha                               NA
# optimal_lambda                      NA
# optimal_K                           NA
# optimal_mtry                        NA
# optimal_ntree                       NA
# optimal_C                 10.000000000
# optimal_d                           NA
# optimal_gamma             10.000000000
# optimal_PR_AUC             0.982888546
# optimal_ROC_AUC            0.989526955
# corresponding_accuracy     0.997406757
# corresponding_TPR          0.970481822
# corresponding_FPR          0.001698307
# corresponding_PPV          0.949344224
# optimal_F1_measure         0.959714228
```

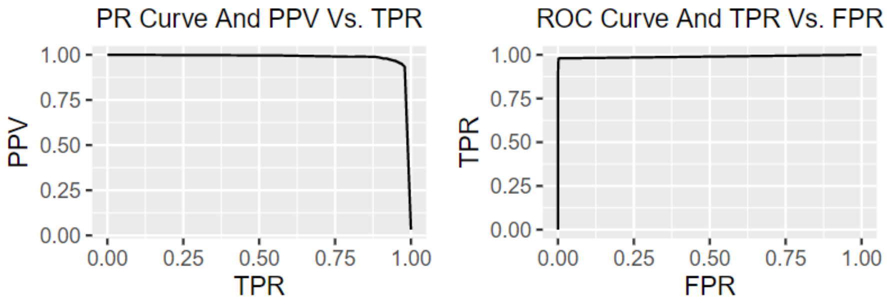
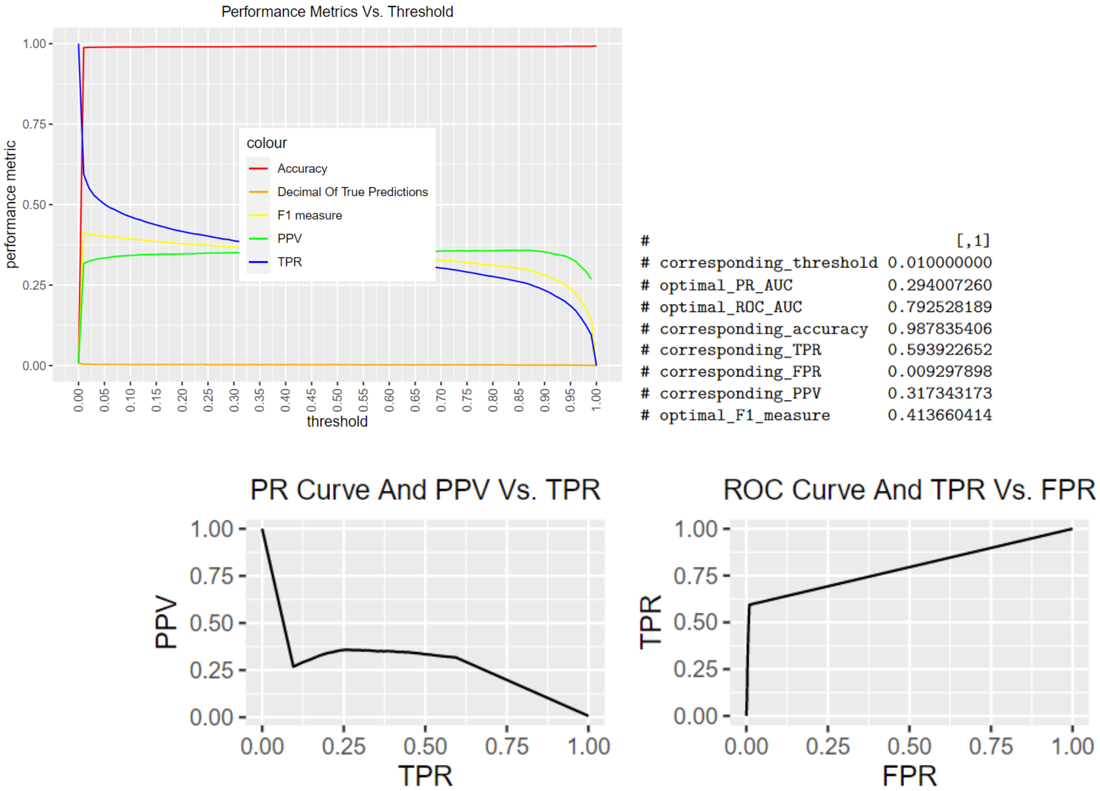


Figure 21: PR And ROC Curves: Cross Validation: SVMWRK



```
#                                  [,1]
# corresponding_threshold 0.010000000
# optimal_PR_AUC          0.294007260
# optimal_ROC_AUC         0.792528189
# corresponding_accuracy  0.987835406
# corresponding_TPR       0.593922652
# corresponding_FPR       0.009297898
# corresponding_PPV       0.317343173
# optimal_F1_measure      0.413660414
```

Figure 22: PR And ROC Curves: Holdout Testing: SVMWPK

Cross-Validation And Holdout Performance Tables

Please see Cross-Validation And Holdout Performance Tables.

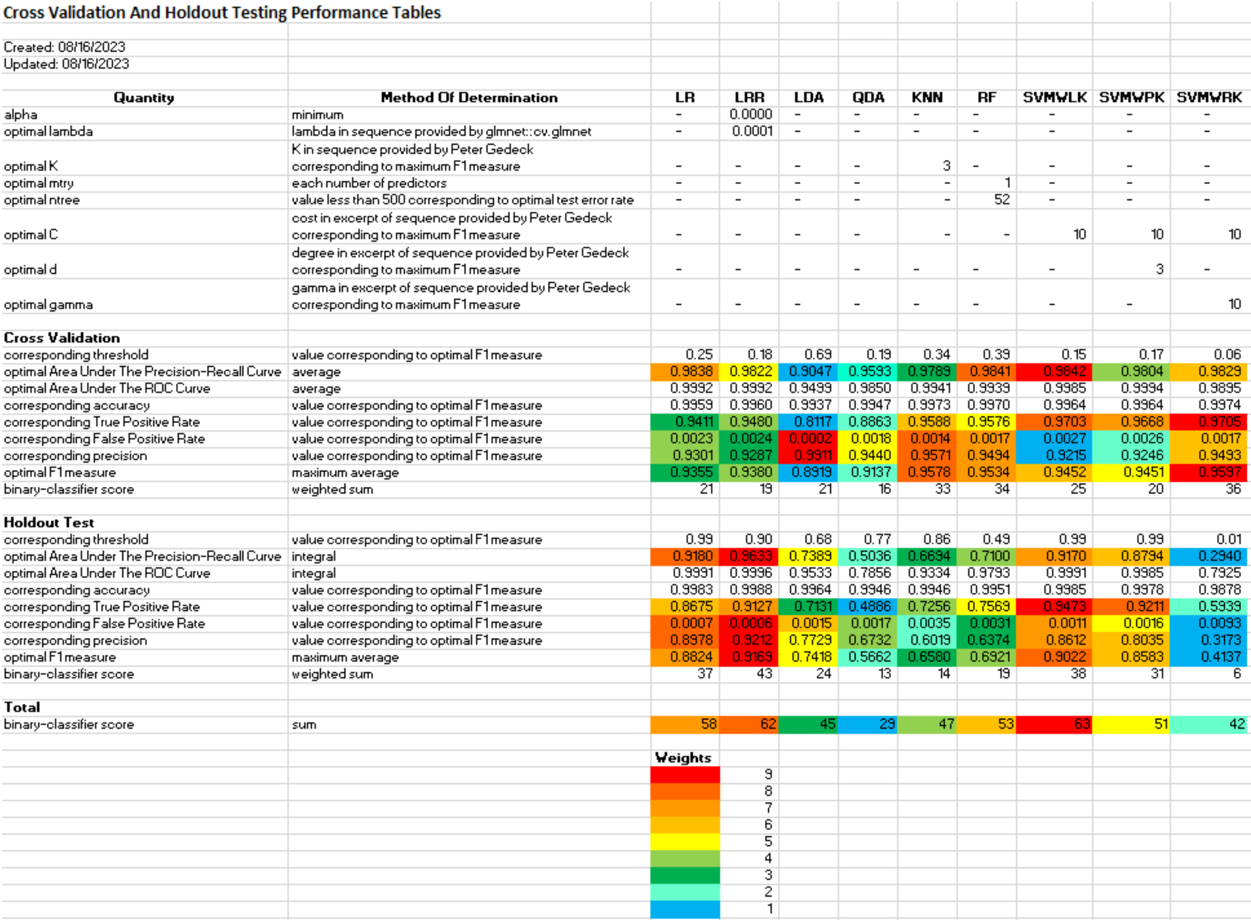
We calculate binary-classifier scores for each type of binary classifier for cross validation and holdout testing. Each binary-classifier score is a weighted sum of the 5 colored performance metrics above it in the appropriate table; namely, PR AUC, TPR, FPR, PPV, and F1 measure. We do not include ROC AUC or accuracy as they may be biased by our imbalanced data.

We calculate total binary-classifier scores for each type of binary classifier. Each total binary-classifier score is the sum of the cross-validation binary classifier score and the holdout binary classifier score for that type of classifier.

Our optimal SVMWLK with formula $Indicator \sim normalize(Red)+normalize\left(Green^2\right)+normalize(Blue)+normalize\left(Blue^2\right)+normalize\left(\sqrt{Red}\right)+normalize\left(\sqrt{Blue}\right)$, $C = 10$, and $t = 0.15$ has highest total binary-classifier score. Our optimal SVMWLK has for cross validation highest PR AUC, second-highest TPR, fourth highest F1 measure, and worst FPR and PPV. Our optimal SVMWLK has for holdout highest TPR, second highest F1 measure, and third-highest PR AUC, FPR, and PPV. Our SVMWLK has an average cross-validation binary-classifier score and the second-highest holdout binary-classifier score. If we were to recommend a classifier based on total binary-classifier score, we would recommend our optimal SVMWLK classifier.

Our optimal LRR classifier with formula $Indicator \sim normalize[\ln(Blue)] + normalize(\sqrt{Red}) + normalize\left[\ln(Green)\right] + normalize[\ln(Red)]$, $\lambda = 0.000123$, and $t = 0.18$ has highest holdout binary-classifier score. Our optimal SVMWLK has second-highest holdout binary-classifier score. Our optimal LRR classifier has for holdout highest PR AUC, FPR, PPV, and F1 measure and third-highest TPR. If we were to recommend a classifier based on holdout binary-classifier score, or holdout F1 measure, we would recommend our optimal LRR classifier.

Our SVMWRK has highest cross-validation binary classifier score by 2 but by far the worst holdout binary-classifier score, suggesting overfitting, and the second-worst holdout binary-classifier score. Our optimal RF with formula $Indicator \sim normalize(Red^2) + normalize(Red : Green) + normalize\,(Green : Blue) + normalize(Green)$, $mtry = 1$, $ntree = 52$, and $t = 0.39$ has the second-highest cross-validation binary-classifier score. If we were to recommend a classifier without knowledge of our holdout data based on cross-validation binary-classifier score, or cross-validation F1 measure, we would recommend our SVMWRK. If we were to recommend a classifier with knowledge of our holdout data based on cross-validation binary-classifier score, or cross-validation F1 measure, we would recommend our RF.



Cross Validation And Holdout Testing Performance Tables

Created: 08/16/2023
Updated: 08/16/2023

Quantity	Method Of Determination	LR	LRR	LDA	QDA	KNN	RF	SVMWLK	SVMWPK	SVMWRK
alpha	minimum	-	0.0000	-	-	-	-	-	-	-
optimal lambda	lambda in sequence provided by glmnet::cv.glmnet	-	0.0001	-	-	-	-	-	-	-
optimal K	K in sequence provided by Peter Gedeck corresponding to maximum F1 measure	-	-	-	-	3	-	-	-	-
optimal mtry	each number of predictors	-	-	-	-	-	1	-	-	-
optimal ntree	value less than 500 corresponding to optimal test error rate	-	-	-	-	-	52	-	-	-
optimal C	cost in excerpt of sequence provided by Peter Gedeck corresponding to maximum F1 measure	-	-	-	-	-	-	10	10	10
optimal d	degree in excerpt of sequence provided by Peter Gedeck corresponding to maximum F1 measure	-	-	-	-	-	-	-	3	-
optimal gamma	gamma in excerpt of sequence provided by Peter Gedeck corresponding to maximum F1 measure	-	-	-	-	-	-	-	-	10
Cross Validation										
corresponding threshold	value corresponding to optimal F1 measure	0.25	0.18	0.69	0.19	0.34	0.39	0.15	0.17	0.06
optimal Area Under The Precision-Recall Curve	average	0.9838	0.9822	0.9047	0.9593	0.9789	0.9841	0.9842	0.9804	0.9829
optimal Area Under The ROC Curve	average	0.9992	0.9992	0.9499	0.9850	0.9941	0.9939	0.9985	0.9994	0.9895
corresponding accuracy	value corresponding to optimal F1 measure	0.9959	0.9960	0.9937	0.9947	0.9973	0.9970	0.9964	0.9964	0.9974
corresponding True Positive Rate	value corresponding to optimal F1 measure	0.9411	0.9480	0.8117	0.8863	0.9588	0.9576	0.9703	0.9668	0.9705
corresponding False Positive Rate	value corresponding to optimal F1 measure	0.0023	0.0024	0.0002	0.0018	0.0014	0.0017	0.0027	0.0026	0.0017
corresponding precision	value corresponding to optimal F1 measure	0.9301	0.9287	0.9911	0.9440	0.9571	0.9494	0.9215	0.9246	0.9493
optimal F1 measure	maximum average	0.9355	0.9380	0.8919	0.9137	0.9578	0.9534	0.9452	0.9451	0.9597
binary-classifier score	weighted sum	21	19	21	16	33	34	25	20	36
Holdout Test										
corresponding threshold	value corresponding to optimal F1 measure	0.99	0.90	0.68	0.77	0.86	0.49	0.99	0.99	0.01
optimal Area Under The Precision-Recall Curve	integral	0.9180	0.9633	0.7389	0.5036	0.6694	0.7100	0.9170	0.8794	0.2940
optimal Area Under The ROC Curve	integral	0.9991	0.9996	0.9533	0.7856	0.9334	0.9793	0.9991	0.9985	0.7925
corresponding accuracy	value corresponding to optimal F1 measure	0.9983	0.9988	0.9964	0.9946	0.9946	0.9951	0.9985	0.9978	0.9878
corresponding True Positive Rate	value corresponding to optimal F1 measure	0.8675	0.9127	0.7131	0.4886	0.7256	0.7569	0.9473	0.9211	0.5939
corresponding False Positive Rate	value corresponding to optimal F1 measure	0.0007	0.0006	0.0015	0.0017	0.0035	0.0031	0.0011	0.0016	0.0093
corresponding precision	value corresponding to optimal F1 measure	0.8978	0.9212	0.7729	0.6732	0.6019	0.6374	0.8612	0.8035	0.3173
optimal F1 measure	maximum average	0.8824	0.9169	0.7418	0.5662	0.6580	0.6921	0.9022	0.8583	0.4137
binary-classifier score	weighted sum	37	43	24	13	14	19	38	31	6
Total										
binary-classifier score	sum	58	62	45	29	47	53	63	51	42
		Weights								
		🟥	9							
		🟧	8							
		🟧	7							
		🟧	6							
		🟨	5							
		🟩	4							
		🟩	3							
		🟦	2							
		🟦	1							

Figure 23: Cross-Validation And Holdout Performance Tables

Conclusions

As above, our best performing algorithm in the cross-validation data is our optimal SVMWRK, though it seems to overfit. Our best performing algorithm in the cross-validation data informed by holdout performance is our optimal RF. Our best performing algorithm in the holdout data is our optimal LRR classifier. Our best performing algorithm overall is our optimal SVMWLK.

Our findings are reconcilable given differences in our training and holdout distributions. A binary classifier trained on our training data and holdout tested on our holdout data will have a maximum F1 measure that is less than that of a classifier cross validated on our training data. Our optimal classifiers make assumptions about what constitutes a blue tarp based on our training data. We recommend cross validating on our composite data or cross validating on 90 percent of our composite data and holdout testing on the other 10 percent to determine one best binary classifier. By training on our composite data our classifiers will become more realistic and robust and our holdout testing results will align with our cross-validation results. In that case, our classifiers' assumptions will be based on both our training and holdout data and more based in the real world.

For now, we recommend in order our optimal SVMWLK with highest overall binary-classification score, our optimal RF with second-highest cross-validation binary-classifier score and far better performance on our holdout data than our SVMWRK, and our optimal LRR classifier with highest holdout binary-classification score.

TPR, the ratio of true positives to actual positives, is relevant to our application; we want TPR to be 1; we want the number of false negatives to be 0; we don't want to predict that a pixel corresponding to a blue tarp does not belong to a blue tarp; we want to identify all blue tarps. FPR, the ratio of false positives to actual negatives, is relevant to our application; we want FPR to be 0; we don't want to predict that a pixel corresponds to a blue tarp when the pixel does not correspond to a blue tarp; we want to visit locations as efficiently as possible. ROC AUC is not relevant to our application; according to Mr. Allwright, ROC "[AUC] does not perform well on imbalanced datasets." PPV, the ratio of true positives to predicted positives, is relevant to our application; we want PPV to be 1; we don't want there to be any false positives; we want all predicted positives to be true positives. PR AUC is relevant to our application; PR AUC is a good overall measure of the performance of our classifier and its PPV and TPR. Accuracy is not relevant to our application; according to Mr. Allwright, "Accuracy does not perform well on imbalanced datasets which often leads to misleading results." F1 measure is relevant to our application; according to Optimal Thresholding of Classifiers to Maximize F1 Measure, "the F1 measure is widely used to evaluate the success of a binary classifier when one class is rare."

According to Mr. Allwright, "A good F1 score is dependent on the data you are working with and the use case. For example, a model predicting the occurrence of a disease would have a very different expectation than a customer churn model. However, there is a general rule of thumb when it comes to F1 scores, which is as follows:" - An F1 score greater than 0.9 is very good. - An F1 score of 0.8 to 0.9 is good. - An F1 score of 0.5 to 0.8 is okay. - An F1 score less than 0.5 is not good.

Per cross-validation, all of our binary classifiers are adequately performing per cross-validation. Per holdout testing, all of our binary classifiers perform okay except for our SVMWRK.

Our optimal SVMWLK and LRR classifier and our SVMWLK and RF have statistically significant differences in average maximum F1 measures per cross validation. Our optimal LRR classifier and RF have a statistically insignificant difference. Following Text-Mining Evaluation: Statistical Tests, we cross validate our SVMWLK, LRR classifier, and RF. We use the same folds for each cross validation. We calculate maximum F1 measures for each held-out fold. We construct a data frame of fold ID's and maximum F1 measure for each binary classifier and fold. We calculate the average maximum F1 measures across held-out folds. We consider differences between the average maximum F1 measures to be test statistics. We consider an alternate hypothesis to be that our SVMWLK, for example, has a higher average maximum F1 measure than our LRR classifier, for example, for all possible test sets. We consider a null hypothesis to be that our SVMWLK has an equal average maximum F1 measure to that of our LRR classifier. We consider a p value to be the probability of observing that the difference in average maximum F1 measures for a random sample / test set is greater than the relevant test statistic. We consider our confidence level to be $CL = 0.95$ and our significance level to be $\alpha = 1 - CL = 0.05$. If the p value is greater than or equal to the significance level α, we fail to reject the null hypothesis that our SVMWLK has an average maximum F1 measure equal to that of our LRR classifier. If the p value is less than or equal to the significance level α, we reject the null hypothesis and have sufficient evidence to support the alternate hypothesis that our SVMWLK has an average maximum F1 measure greater than that of our LRR classifier.

To calculate our p value, we conduct Fisher's Randomization Test. We calculate our p value as the ratio of a count C and a number of iterations I. Our count C is the number of iterations $i \leq I$ for which a modified difference between an average maximum F1 measure for our SVMWLK and an average maximum

F1 for our LRR classifier is greater than our test statistic. For each iteration, an average maximum F1 measure for our SVMWLK is computed effectively by tossing a fair coin for each row in our data frame of fold ID's and maximum F1 measures, if heads swapping the maximum F1 measures for the present fold, and calculating the mean of the maximum F1 measures in the column corresponding to our SVMWLK. An average maximum F1 measure for our LRR classifier is computed similarly. A modified difference between the average maximum F1 measure for our SVMWLK and the average maximum F1 measure for our LRR classifier is compared to our test statistic. If the modified difference is greater than our test statistic, count C is incremented. Our p value is the ratio of count C and number of iterations I.

The p value for our SVMWLK and LRR classifier is 0.045. The probability of observing a random test statistic greater than our test statistic is 0.045. Because our p value is greater than our significance level, we cannot confidently say that our test statistic is not due to random chance. Our difference between the average maximum F1 measures of our SVMWLK and LRR classifier is statistically significant. The p value for our SVMWLK and LRR classifier is approximately 0. Our difference between the average maximum F1 measures of our SVMWLK and LRR classifier is statistically significant. The p value for our LRR classifier and RF is approximately 0.99. Our difference between the average maximum F1 measures of our LRR classifier and RF is statistically insignificant.

Given statistically significant differences between our optimal SVMWLK and either our optimal LRR classifier or our optimal RF, we select our optimal SVMWLK.

We are able to use predictive modeling tools as our data set of classes and pixels may be represented as a complete data frame of natural numbers. We are able to use classifiers as the data in column *Class* is nominal. We are able to use binary classifiers as pixels representing blue tarps exist in a distinct subspace defined by intensities of color *Red*, *Green*, and *Blue*. We are able to normalize intensities of color *Red*, *Green*, and *Blue*.

Our data set seems particularly well-suited to the class of prediction methods Binary Classifier as our data set is complete, observations may be represented as natural numbers, class of pixel is nominal, and pixels representing blue tarps exist in a distinct subspace defined by intensities of color *Red*, *Green*, and *Blue*.

Given that according to Mr. Allwright the maximum average F1 measures of our binary classifiers are good per cross validation, our research may be effective in terms of locating people displaced by the earthquake in Haiti in 2010 and helping to save human life.