Building A Model That Helps Locating Displaced People

Tom Lever

06/08/2023

In this project, we build a model 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.

Our training data was collected likely by applying a Region Of Interest (ROI) Tool to a high-resolution, orthorectified / geo-referenced image of Haiti in 2010. One ROI tool is described at https://www.l3harrisgeospatial.com/docs/regionofinteresttool.html. Classes may be assigned to pixels by defining Regions Of Interest.

Our training data frame consists of $63,241$ observations. Each 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.

According to https://www.esri.com/about/newsroom/insider/what-is-orthorectified-imagery/, an orthorectified image is an accurately georeferenced image that has been processed so that all pixels are in an accurate (x, y) position on the ground. Orthorectified images have been processed to apply corrections for optical distortions from the sensor system, and apparent changes in the position of ground objects caused by the perspective of the sensor view angle and ground terrain.

We load a data frame of classes and pixels based on an orthorectified image of Haiti at https://www.kaggle.com/datasets/billbasener/pixel-values-from-images-over-haiti?datasetId=1899167.

```
#        Class Red Green Blue
# 1 Vegetation  64    67   50
# 2 Vegetation  64    67   50
```

```
#           Class Red Green Blue
# 63240 Blue Tarp 132   139  149
# 63241 Blue Tarp 133   141  153
```

We build binary classifiers that classify 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. We may ignore non-binary classifiers since the intensity space for pixels representing blue tarps is distinct from the intensity space for pixels representing objects that are not blue tarps. See "Figure 1: Distribution Of Classes In Intensity Space".

In order to build binary classifiers, we create a data frame with a column of indicators of whether of not a pixel depicts a blue tarp instead of a column of classes.

```
#       Indicator        Red       Green        Blue
# 47073         0  0.1143414 -0.02288695  0.01403632
# 46839         0 -0.1265385 -0.24314015 -0.18142497
#       Indicator        Red      Green       Blue
# 61978         1  0.4186107  0.9820183  1.9686492
# 8377          0 -0.6463320 -0.5459883 -0.5723476
```

We use 10-fold cross-validation to evaluate the performance of 5 classifiers. A classifier will classify a pixel as depicting a blue tarp or depicting an object that is not a blue tarp. We consider graphs each with an increasing purple curve representing Average Precision vs. Threshold, a decreasing red curve representing Average Recall vs. Threshold, and a forest-green curve representing Average F1 Measure vs. Threshold.

A threshold is a probability between 0 and 1. A model classifies a pixel as representing a blue tarp if the model the probability that the pixel represents a blue tarp that the model predicts is greater than the threshold. Precision is the ratio of true positives to predicted positives. Recall is the ratio of true positives to actual positives. An $F1$ measure is the harmonic mean of precision and recall.

When tuning the thresholds of our models we prioritize recall at least as much as precision. We prioritize identifying as many positives correctly as possible over having predicted positives be correct. We recommend models with thresholds less than or equal to the least threshold corresponding to the maximum $F1$ measure.

We use 10-fold cross-validation to evaluate the performance of logistic-regression models.

We use 10-fold cross-validation to evaluation the performance of a logistic-regression model with formula $Indicator \sim Red + Green + Blue$.

```
# $ROC_curve


#
# $data_frame_corresponding_to_maximum_average_F1_measure
# # A tibble: 1 x 4
#   threshold precision recall F1_measure
#       <dbl>     <dbl>  <dbl>      <dbl>
# 1      0.21     0.942  0.926      0.934

# $ROC_curve


#
# $data_frame_corresponding_to_maximum_average_F1_measure
# # A tibble: 1 x 4
#   threshold precision recall F1_measure
#       <dbl>     <dbl>  <dbl>      <dbl>
# 1      0.84     0.766  0.761      0.763

# $ROC_curve


#
# $data_frame_corresponding_to_maximum_average_F1_measure
# # A tibble: 1 x 4
#   threshold precision recall F1_measure
#       <dbl>     <dbl>  <dbl>      <dbl>
# 1      0.23     0.968  0.866      0.914

# $ROC_curve
```

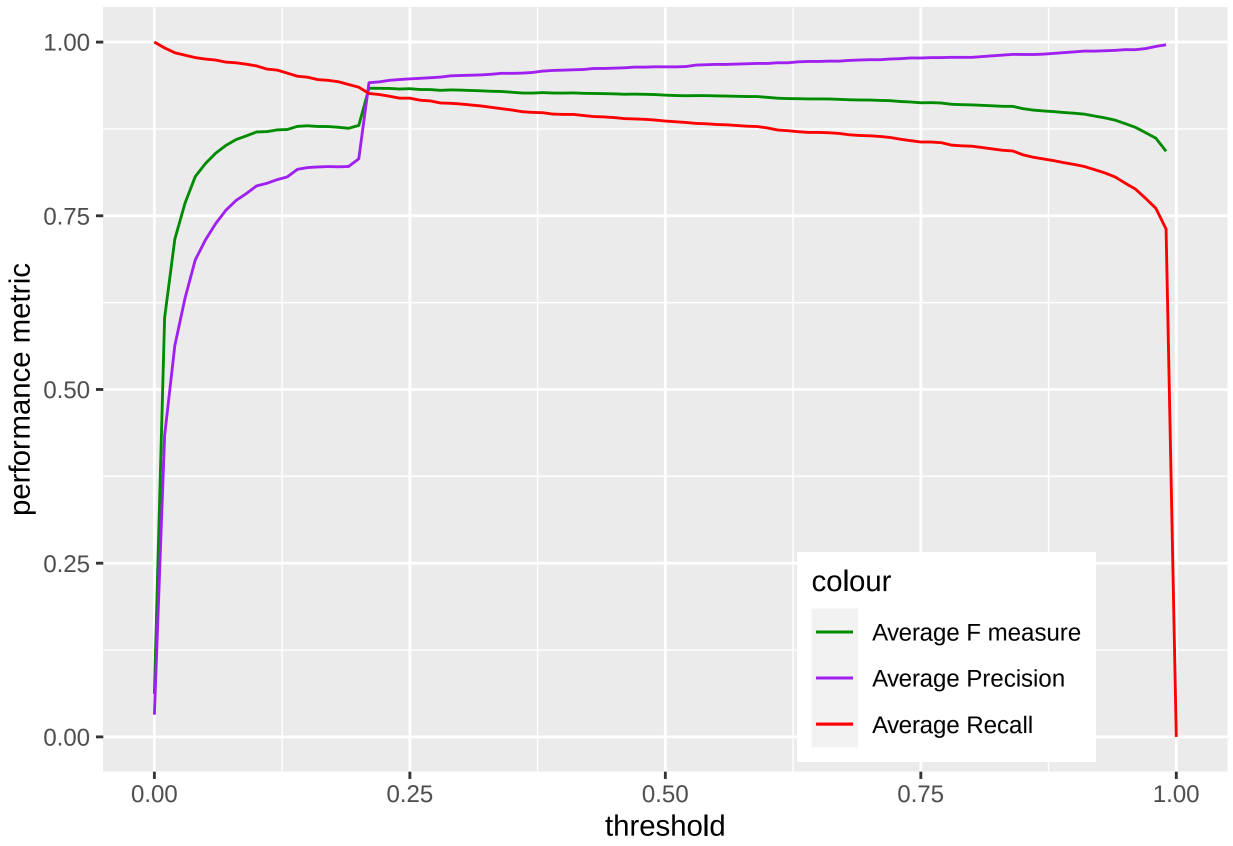

Figure 1: Precision / Recall Curve For Logistic Regression Model With Formula Indicator vs. Red + Green + Blue

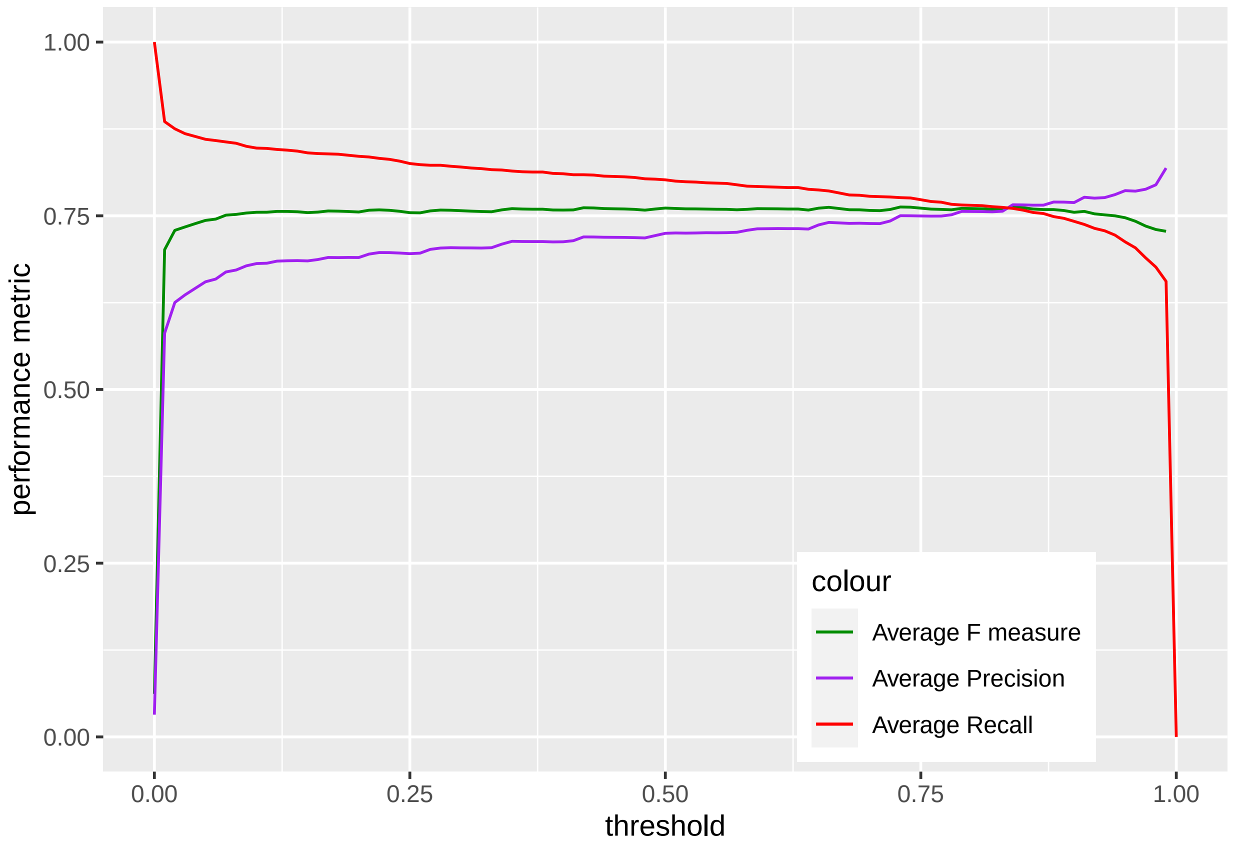

Figure 2: Precision / Recall Curve For Linear Discriminant Analysis Model With Formula Indicator vs. Red + Green + Blue

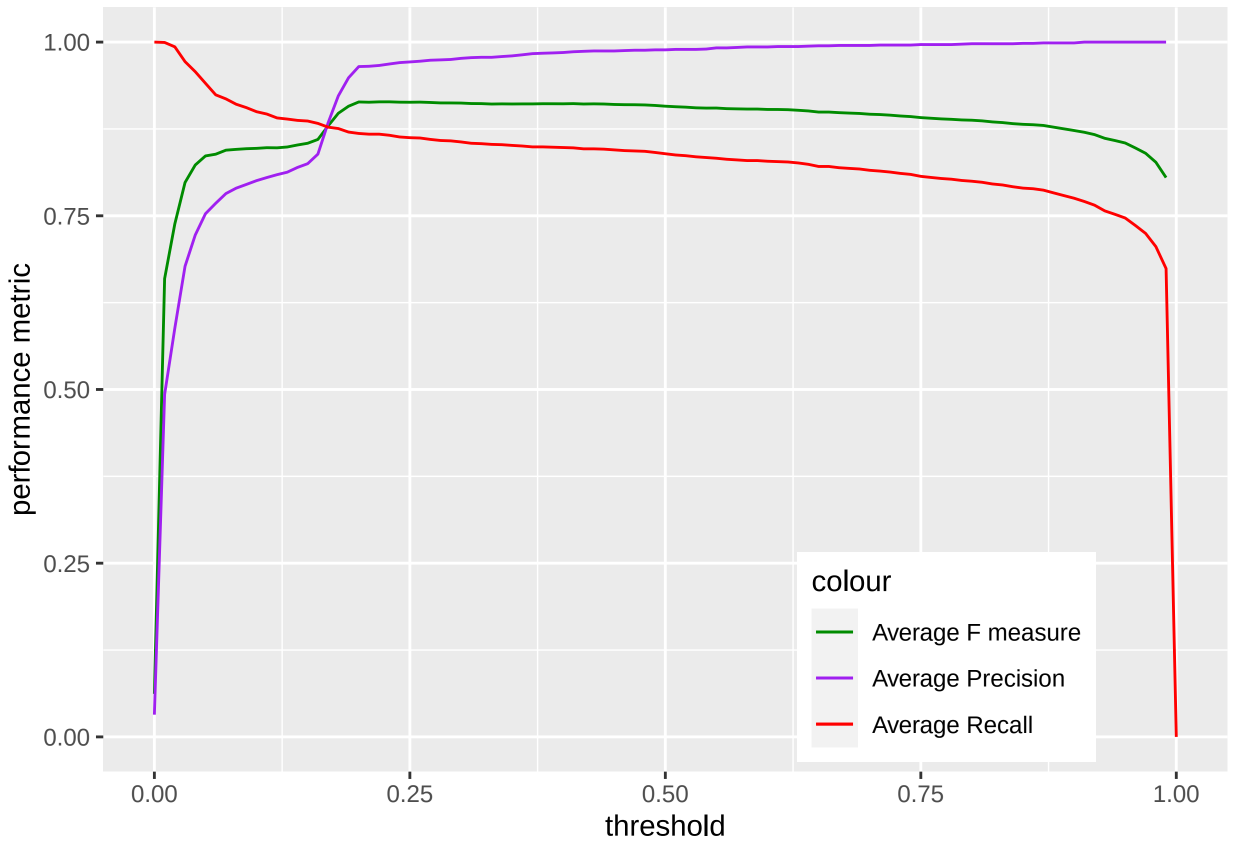

Figure 3: Precision / Recall Curve For Quadratic Discriminant Analysis Model With Formula Indicator vs. Red + Green + Blue

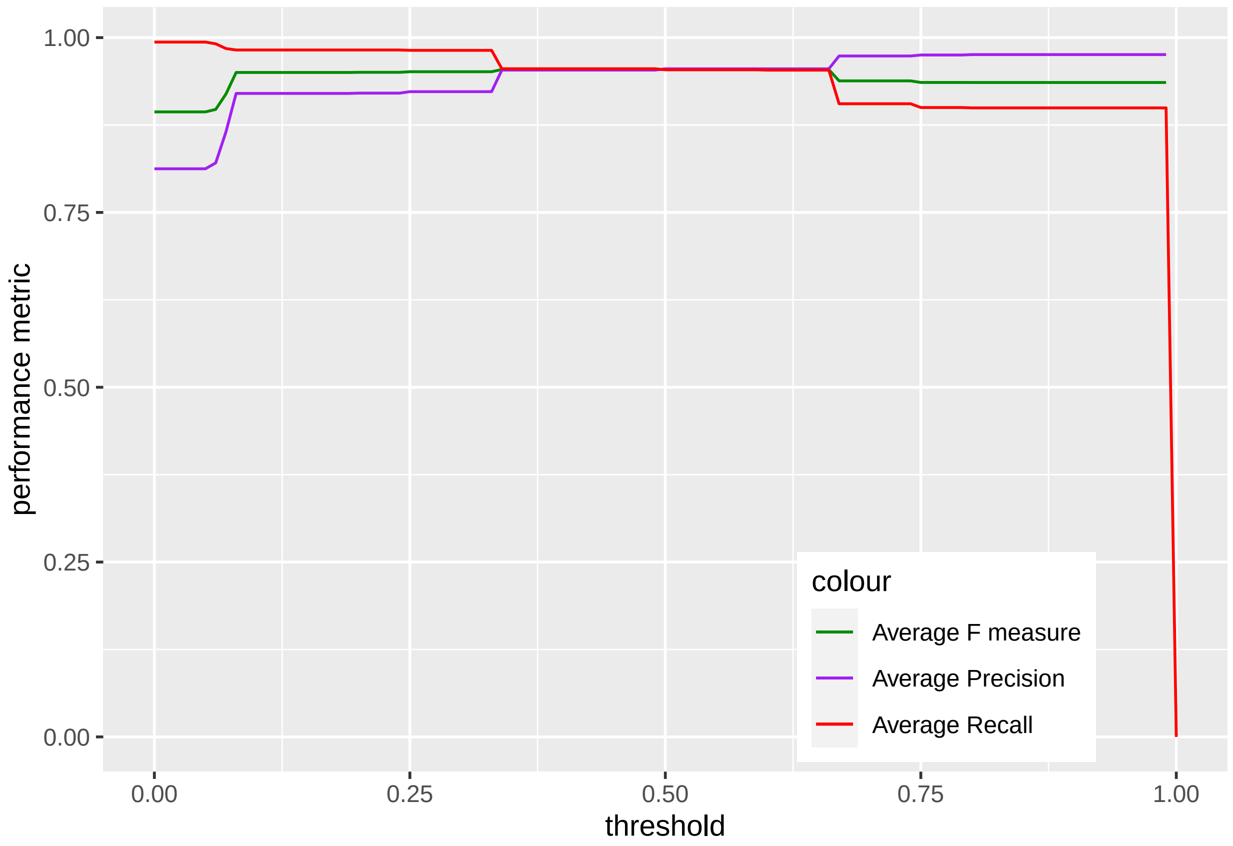

Figure 4: Precision / Recall Curve For K Nearest Neighbors Model With Formula Indicator vs. Red + Green + Blue And K = 3

```
#
# $data_frame_corresponding_to_maximum_average_F1_measure
# # A tibble: 10 x 4
#    threshold precision recall F1_measure
#        <dbl>     <dbl>  <dbl>      <dbl>
#  1      0.5      0.955  0.954      0.955
#  2      0.51     0.955  0.954      0.955
#  3      0.52     0.955  0.954      0.955
#  4      0.53     0.955  0.954      0.955
#  5      0.54     0.955  0.954      0.955
#  6      0.55     0.955  0.954      0.955
#  7      0.56     0.955  0.954      0.955
#  8      0.57     0.955  0.954      0.955
#  9      0.58     0.955  0.954      0.955
# 10      0.59     0.955  0.954      0.955
```

```r
library(TomLeversRPackage)
for (i in 1:99) {
 print(summarize_performance_of_cross_validated_models_using_dplyr(
     type_of_model = "KNN",
     formula = Indicator ~ Red + Green + Blue,
     data_frame = data_frame_of_indicators_and_pixels,
     K = i
 )$mean_AUC)
}
```