Multinomial classification with tidymodels using the TidyTuesday volcano data

Julie Silge modified by John Lewis

7/30/2020

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(tidymodels)
library(vip)
theme_set(theme_light())
```

Load the data

```
volcano_raw <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/ma
ster/data/2020/2020-05-12/volcano.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   volcano_number = col_double(),
##   latitude = col_double(),
##   longitude = col_double(),
##   elevation = col_double(),
##   population_within_5_km = col_double(),
##   population_within_10_km = col_double(),
##   population_within_30_km = col_double(),
##   population_within_100_km = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

For a complete source of information of this dataset please see the following page

https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-05-12/readme.md (https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-05-12/readme.md)

```
dim(volcano_raw)
```

```
## [1] 958  26
```

```
glimpse(volcano_raw)
```

```
## Rows: 958
## Columns: 26
## $ volcano_number         <dbl> 283001, 355096, 342080, 213004, 321040, 28...
## $ volcano_name           <chr> "Abu", "Acamarachi", "Acatenango", "Acigol...
## $ primary_volcano_type   <chr> "Shield(s)", "Stratovolcano", "Stratovolca...
## $ last_eruption_year     <chr> "-6850", "Unknown", "1972", "-2080", "950"...
## $ country                <chr> "Japan", "Chile", "Guatemala", "Turkey", "...
## $ region                 <chr> "Japan, Taiwan, Marianas", "South America"...
## $ subregion              <chr> "Honshu", "Northern Chile, Bolivia and Arg...
## $ latitude               <dbl> 34.500, -23.292, 14.501, 38.537, 46.206, 3...
## $ longitude              <dbl> 131.600, -67.618, -90.876, 34.621, -121.49...
## $ elevation              <dbl> 641, 6023, 3976, 1683, 3742, 1728, 1733, 1...
## $ tectonic_settings      <chr> "Subduction zone / Continental crust (>25 ...
## $ evidence_category      <chr> "Eruption Dated", "Evidence Credible", "Er...
## $ major_rock_1           <chr> "Andesite / Basaltic Andesite", "Dacite", ...
## $ major_rock_2           <chr> "Basalt / Picro-Basalt", "Andesite / Basal...
## $ major_rock_3           <chr> "Dacite", " ", " ", "Basalt / Picro-Basalt...
## $ major_rock_4           <chr> " ", " ", " ", "Andesite / Basaltic Andesi...
## $ major_rock_5           <chr> " ", " ", " ", " ", " ", " ", " ", " ", " ...
## $ minor_rock_1           <chr> " ", " ", "Basalt / Picro-Basalt", " ", "D...
## $ minor_rock_2           <chr> " ", " ", " ", " ", " ", "Basalt / Picro-B...
## $ minor_rock_3           <chr> " ", " ", " ", " ", " ", " ", " ", "Andesi...
## $ minor_rock_4           <chr> " ", " ", " ", " ", " ", " ", " ", " ", " ...
## $ minor_rock_5           <chr> " ", " ", " ", " ", " ", " ", " ", " ", " ...
## $ population_within_5_km   <dbl> 3597, 0, 4329, 127863, 0, 428, 101, 51, 0,...
## $ population_within_10_km  <dbl> 9594, 7, 60730, 127863, 70, 3936, 485, 604...
## $ population_within_30_km  <dbl> 117805, 294, 1042836, 218469, 4019, 717078...
## $ population_within_100_km <dbl> 4071152, 9092, 7634778, 2253483, 393303, 5...
```

```
#Explore the data
#Our modeling goal is to predict the type of volcano from one of the #TidyTuesday
#dataset based on other volcano characteristics like latitude, longitude, tectonic
#setting, etc. There are more than just two types of volcanoes, so this is an example
#of multiclass or multinomial classification instead of binary classification.
#Let’s use a random forest model, because this type of model performs well with
#defaults.

volcano_raw %>%
  count(primary_volcano_type, sort = TRUE)
```

```
## # A tibble: 26 x 2
##    primary_volcano_type     n
##    <chr>                <int>
##  1 Stratovolcano          353
##  2 Stratovolcano(es)      107
##  3 Shield                  85
##  4 Volcanic field          71
##  5 Pyroclastic cone(s)     70
##  6 Caldera                 65
##  7 Complex                 46
##  8 Shield(s)               33
##  9 Submarine               27
## 10 Lava dome(s)            26
## # ... with 16 more rows
```

```
#probably too many types of volcanoes for us to build a model for, especially with
#just 958 examples. Let’s create a new volcano_type variable and build a model to
#distinguish between four volcano types:
#stratovolcano
#shield volcano
#caldera
#everything else (other)

#While we use transmute() to create this new variable, let’s also select the
#variables to use in modeling, like the info about the tectonics around the volcano
#and the most important rock type.

volcano_df <- volcano_raw %>%
  transmute(
    volcano_type = case_when(
      str_detect(primary_volcano_type, "Stratovolcano") ~ "Stratovolcano",
      str_detect(primary_volcano_type, "Shield") ~ "Shield",
      str_detect(primary_volcano_type, "Caldera") ~ "Caldera",
      TRUE ~ "Other"
    ),
    volcano_number, latitude, longitude, elevation,
    tectonic_settings, major_rock_1
  ) %>%
  mutate_if(is.character, factor)

volcano_df %>%
  count(volcano_type, sort = TRUE)
```

```
## # A tibble: 4 x 2
##   volcano_type      n
##   <fct>         <int>
## 1 Stratovolcano   461
## 2 Other           305
## 3 Shield          118
## 4 Caldera          74
```

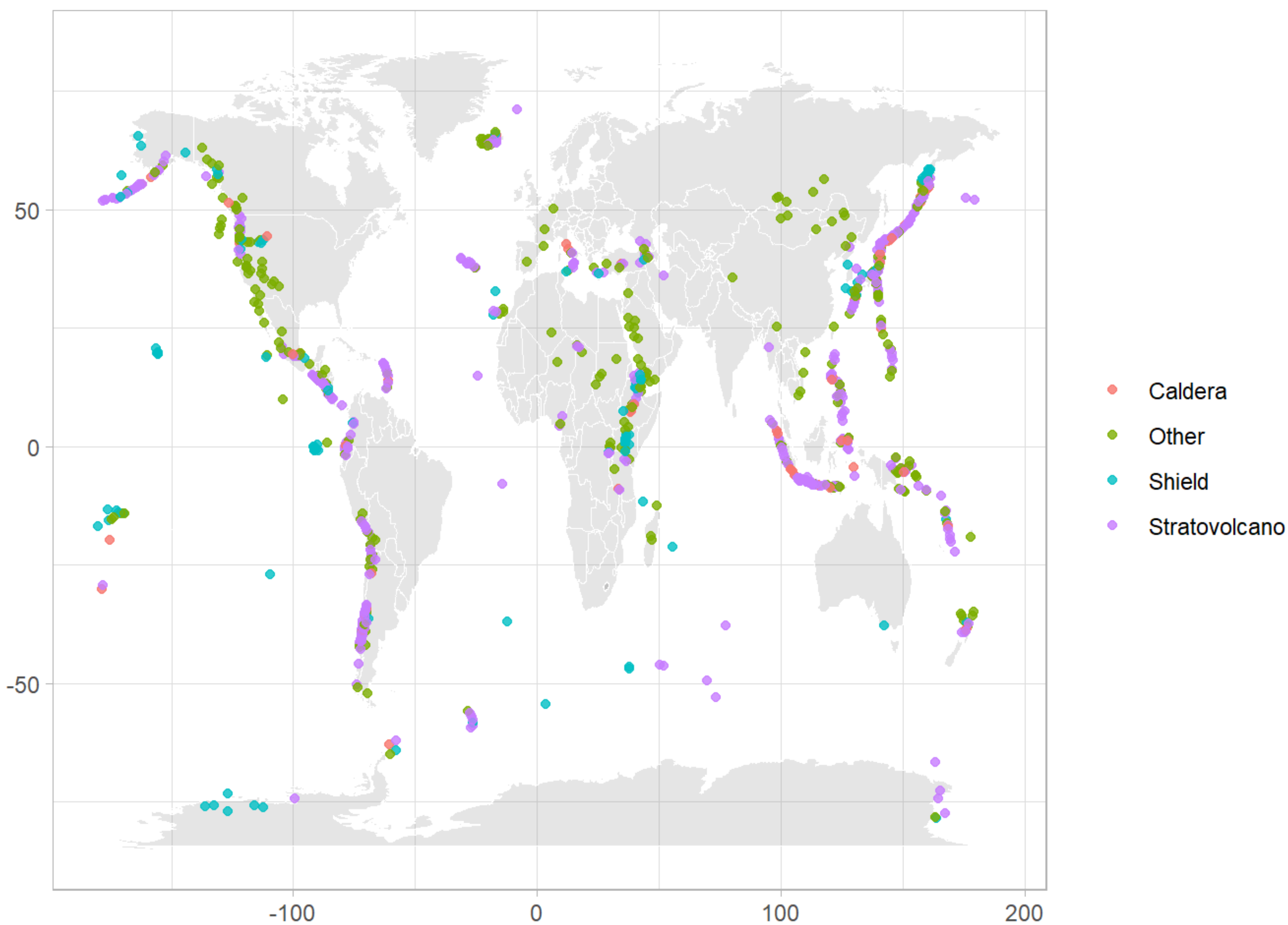
```
#not a lot of data to be building a random forest model but nice for mapping
```

Location of Volcanoes

```
world <- map_data("world")

ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "white", fill = "gray50", size = 0.05, alpha = 0.2
  ) +
  geom_point(
    data = volcano_df,
    aes(longitude, latitude, color = volcano_type),
    alpha = 0.8
  ) +
  labs(x = NULL, y = NULL, color = NULL)
```

```
## Warning: Ignoring unknown aesthetics: x, y
```



```
#Instead of splitting this small-ish dataset into training and testing data,
#let’s create a set of bootstrap resamples.

set.seed(456)
volcano_boot <- bootstraps(volcano_df)

volcano_boot
```

```
## # Bootstrap sampling
## # A tibble: 25 x 2
##    splits            id
##    <list>            <chr>
##  1 <split [958/347]> Bootstrap01
##  2 <split [958/361]> Bootstrap02
##  3 <split [958/361]> Bootstrap03
##  4 <split [958/334]> Bootstrap04
##  5 <split [958/364]> Bootstrap05
##  6 <split [958/353]> Bootstrap06
##  7 <split [958/339]> Bootstrap07
##  8 <split [958/348]> Bootstrap08
##  9 <split [958/353]> Bootstrap09
## 10 <split [958/347]> Bootstrap10
## # ... with 15 more rows
```

```
#Let’s train our multinomial classification model on these resamples, but keep in
#mind that the performance estimates can be somewhat biased.

#we could use SMOTE to upsampling (via the themis package) in order to balance the classes #but
 we are using a random forest so ,at least on the first run not do this

volcano_rec <- recipe(volcano_type ~ ., data = volcano_df) %>%
  update_role(volcano_number, new_role = "Id") %>%
  step_other(tectonic_settings) %>%
  step_other(major_rock_1) %>%
  step_dummy(tectonic_settings, major_rock_1) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())

# 1) we update the role for volcano number, since this is a variable we want to keep
# around for convenience as an identifier for rows but is not a predictor or outcome.
# 2) There are a lot of different tectonic setting and rocks in this dataset, so let’s
# collapse some of the less frequently occurring levels into an "Other" category,
# for each predictor.
# 3) we can create indicator variables and remove variables with zero variance.
# 4) Before oversampling, we center and scale (i.e. normalize) all the predictors.


volcano_prep <- prep(volcano_rec)
juice(volcano_prep) # just to look at our recipe
```

```
## # A tibble: 958 x 14
##    volcano_number latitude longitude elevation volcano_type tectonic_settin~
##             <dbl>    <dbl>     <dbl>     <dbl> <fct>                   <dbl>
##  1         283001   0.618      0.984   -0.875  Shield                 -0.289
##  2         355096  -1.21      -0.830    2.97   Stratovolca~           -0.289
##  3         342080  -0.0153    -1.04     1.50   Stratovolca~           -0.289
##  4         213004   0.746      0.101   -0.131  Caldera                -0.289
##  5         321040   0.988     -1.32     1.34   Stratovolca~           -0.289
##  6         283170   0.718      1.06    -0.0992 Stratovolca~           -0.289
##  7         221170  -0.156      0.158   -0.0956 Stratovolca~           -0.289
##  8         221110  -0.0601     0.158   -0.440  Stratovolca~           -0.289
##  9         284160   0.120      1.11    -0.644  Stratovolca~           -0.289
## 10         342100  -0.0165    -1.04     1.35   Stratovolca~           -0.289
## # ... with 948 more rows, and 8 more variables:
## #   tectonic_settings_Rift.zone...Oceanic.crust....15.km. <dbl>,
## #   tectonic_settings_Subduction.zone...Continental.crust...25.km. <dbl>,
## #   tectonic_settings_Subduction.zone...Oceanic.crust....15.km. <dbl>,
## #   tectonic_settings_other <dbl>, major_rock_1_Basalt...Picro.Basalt <dbl>,
## #   major_rock_1_Dacite <dbl>,
## #   major_rock_1_Trachybasalt...Tephrite.Basanite <dbl>,
## #   major_rock_1_other <dbl>
```

```
#Build a model
rf_spec <- rand_forest(trees = 1000) %>%
  set_mode("classification") %>%
  set_engine("ranger")

#workflow
volcano_wf <- workflow() %>%
  add_recipe(volcano_rec) %>%
  add_model(rf_spec)

volcano_wf
```

```
## == Workflow ===================================================================================
=
## Preprocessor: Recipe
## Model: rand_forest()
##
## -- Preprocessor ----------------------------------------------------------------------------
-
## 5 Recipe Steps
##
## * step_other()
## * step_other()
## * step_dummy()
## * step_zv()
## * step_normalize()
##
## -- Model -----------------------------------------------------------------------------------
-
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   trees = 1000
##
## Computational engine: ranger
```

```
#Now we can run our model without using prep & juice

#fit our workflow info to the resample data - using bootstrapping instead of cv

volcano_res <- fit_resamples(
  volcano_wf,
  resamples = volcano_boot,
  control = control_resamples(save_pred = TRUE)
)
```

Review of terminology of performance metrics (not exhaustive)

accuracy - the proportion of the data that are predicted correctly

ppv - a measurement system compared to a reference result (the “truth” or gold standard)

sensitivity - the true positive value or the proportion of actual positives that are correctly identified

specificity - the true negative value or the proportion of actual negatives that are correctly identified

roc_auc - a metric that computes the area under the ROC curve

Tidymodels syntax for classification metrics

prediction for label types

- `type = "class"`

`predict(volcano_fit, newdata=volcano_test, type = "class")`

prediction for probabilities

- `type = "prob"`

`predict(volcano_fit, newdata=volcano_test, type = "prob")`

in addition:

- `quantile`
- `numeric` -this category for regression metrics-

there are other prediction types-please see:

https://yardstick.tidymodels.org/reference/index.html (https://yardstick.tidymodels.org/reference/index.html)

Explore results

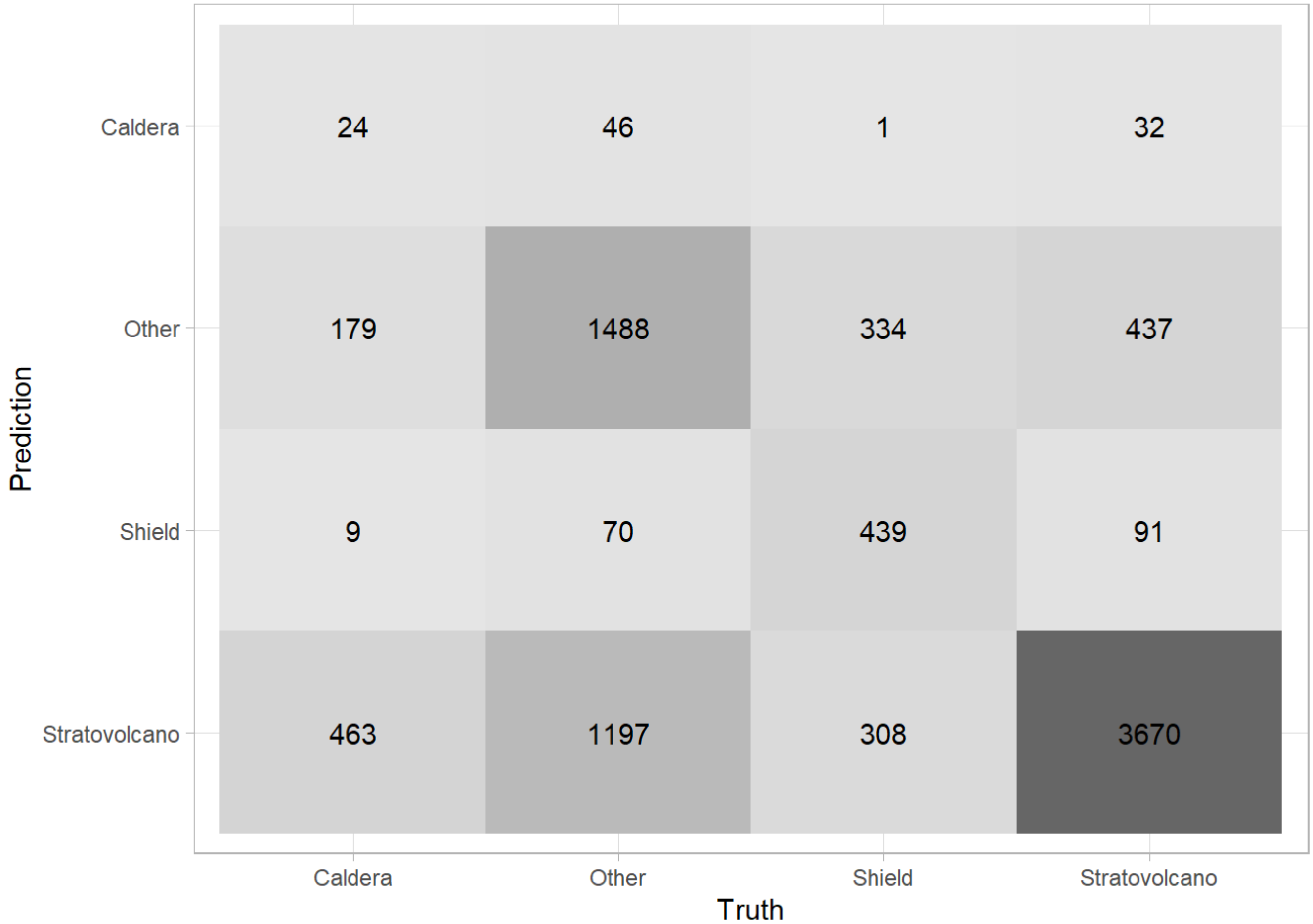
One of the biggest differences when working with multiclass problems is that your performance metrics are different from a two class problem

```
volcano_res %>%
  collect_metrics()
```

```
## # A tibble: 2 x 5
##   .metric  .estimator  mean     n std_err
##   <chr>    <chr>      <dbl> <int>   <dbl>
## 1 accuracy multiclass 0.640    25 0.00459
## 2 roc_auc  hand_till  0.769    25 0.00349
```

Confusion matrix - calculates a cross-tabulation of observed and predicted classes

```
volcano_con <-  volcano_res %>%
  collect_predictions() %>%
  conf_mat(volcano_type, .pred_class)
volcano_con %>%
  autoplot(type="heatmap")
```



Below is a list of metrics from which to chose

```
summary(volcano_con)
```

```
## # A tibble: 13 x 3
##    .metric              .estimator .estimate
##    <chr>                <chr>          <dbl>
##  1 accuracy             multiclass     0.640
##  2 kap                  multiclass     0.393
##  3 sens                 macro          0.460
##  4 spec                 macro          0.844
##  5 ppv                  macro          0.554
##  6 npv                  macro          0.865
##  7 mcc                  multiclass     0.407
##  8 j_index              macro          0.304
##  9 bal_accuracy         macro          0.652
## 10 detection_prevalence macro          0.25
## 11 precision            macro          0.554
## 12 recall               macro          0.460
## 13 f_meas               macro          0.473
```

We computed accuracy and AUC during fit_resamples(), but we can always go back and compute other metrics we are interested in if we saved the predictions. We can even group_by() resample, if we like.

```
#ppv - positive predictive value
volcano_res %>%
  collect_predictions() %>%
  group_by(id) %>%
  ppv(volcano_type, .pred_class)
```

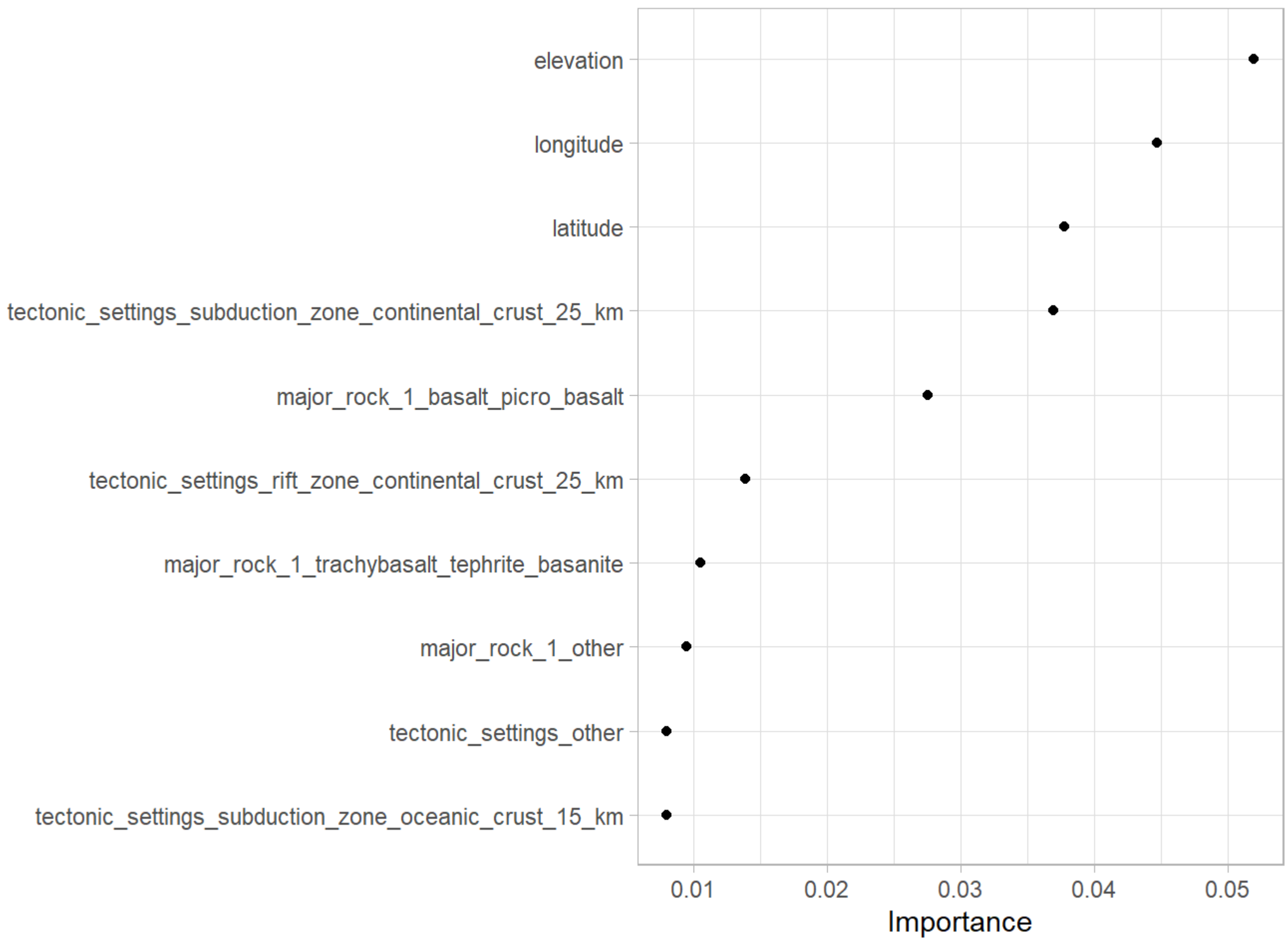
```
## # A tibble: 25 x 4
##    id          .metric .estimator .estimate
##    <chr>       <chr>   <chr>          <dbl>
##  1 Bootstrap01 ppv     macro          0.570
##  2 Bootstrap02 ppv     macro         NA
##  3 Bootstrap03 ppv     macro         NA
##  4 Bootstrap04 ppv     macro          0.575
##  5 Bootstrap05 ppv     macro         NA
##  6 Bootstrap06 ppv     macro          0.621
##  7 Bootstrap07 ppv     macro          0.584
##  8 Bootstrap08 ppv     macro          0.630
##  9 Bootstrap09 ppv     macro          0.552
## 10 Bootstrap10 ppv     macro          0.560
## # ... with 15 more rows
```

```
#roc results of bootstrap resampled rf model
volcano_res %>%
  collect_predictions() %>%
  group_by(id) %>%
  roc_auc(volcano_type,.pred_Caldera:.pred_Stratovolcano)
```

```
## # A tibble: 25 x 4
##    id          .metric .estimator .estimate
##    <chr>       <chr>   <chr>          <dbl>
##  1 Bootstrap01 roc_auc hand_till      0.773
##  2 Bootstrap02 roc_auc hand_till      0.788
##  3 Bootstrap03 roc_auc hand_till      0.737
##  4 Bootstrap04 roc_auc hand_till      0.787
##  5 Bootstrap05 roc_auc hand_till      0.780
##  6 Bootstrap06 roc_auc hand_till      0.785
##  7 Bootstrap07 roc_auc hand_till      0.774
##  8 Bootstrap08 roc_auc hand_till      0.789
##  9 Bootstrap09 roc_auc hand_till      0.791
## 10 Bootstrap10 roc_auc hand_till      0.760
## # ... with 15 more rows
```

Looking for the important variables driving the model results

```
rf_spec %>%
  set_engine("ranger", importance = "permutation") %>%
  fit(
    volcano_type ~ .,
    data = juice(volcano_prep) %>%
      select(-volcano_number) %>%
      janitor::clean_names()
  ) %>%
  vip(geom = "point")
```




```
#Let's join the predictions back to the original data.

volcano_pred <- volcano_res %>%
  collect_predictions() %>%
  mutate(correct = volcano_type == .pred_class) %>%
  left_join(volcano_df %>%
    mutate(.row = row_number()))
```

```
## Joining, by = c(".row", "volcano_type")
```

```
volcano_tab <- volcano_pred %>%
  select(volcano_type,.pred_class,.pred_Caldera:.pred_Stratovolcano)
# Predicted vs Observed (with probabilities)
knitr::kable(head(volcano_tab,n=15))
```

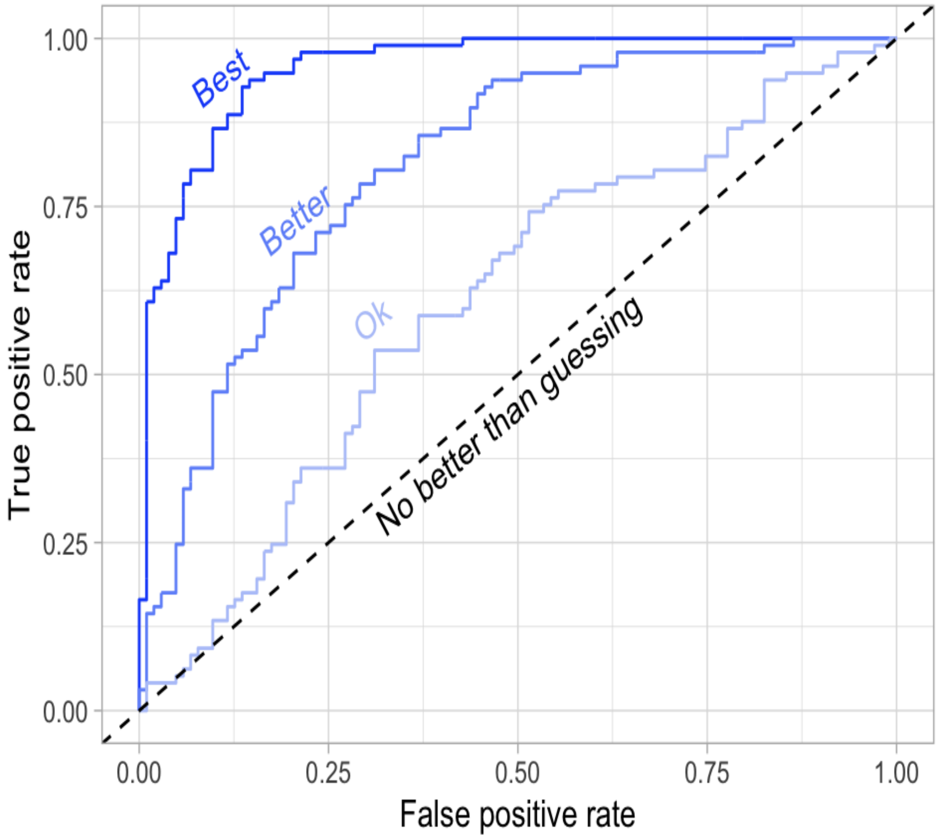
volcano_type	.pred_class	.pred_Caldera	.pred_Other	.pred_Shield	.pred_Stratovolcano
Stratovolcano	Stratovolcano	0.0492405	0.4105554	0.0128832	0.5273209
Stratovolcano	Stratovolcano	0.0302477	0.1119554	0.0192366	0.8385603
Caldera	Stratovolcano	0.1506736	0.1799692	0.0986523	0.5707049
Stratovolcano	Stratovolcano	0.0227586	0.2515078	0.0813246	0.6444090
Stratovolcano	Stratovolcano	0.1025157	0.1627114	0.0720256	0.6627474
Caldera	Stratovolcano	0.1025754	0.2283050	0.0710851	0.5980345
Stratovolcano	Stratovolcano	0.0433612	0.3173439	0.1076967	0.5315982
Stratovolcano	Stratovolcano	0.1421933	0.1956676	0.0245373	0.6376017
Stratovolcano	Stratovolcano	0.0072190	0.2201121	0.3101891	0.4624799
Shield	Stratovolcano	0.0081318	0.2994466	0.2495726	0.4428490
Stratovolcano	Stratovolcano	0.0392190	0.1353619	0.0171054	0.8083138
Other	Stratovolcano	0.0508098	0.1247130	0.0204936	0.8039836
Other	Stratovolcano	0.0396255	0.1505500	0.2714264	0.5383982
Shield	Other	0.0256512	0.4154344	0.3055788	0.2533357
Caldera	Stratovolcano	0.0199015	0.2069077	0.0793793	0.6938115

Number of correct vs non-correct

```
volcano_pred %>% count(correct==TRUE)
```

```
## # A tibble: 2 x 2
##   `correct == TRUE`     n
##   <lgl>             <int>
## 1 FALSE              3167
## 2 TRUE               5621
```

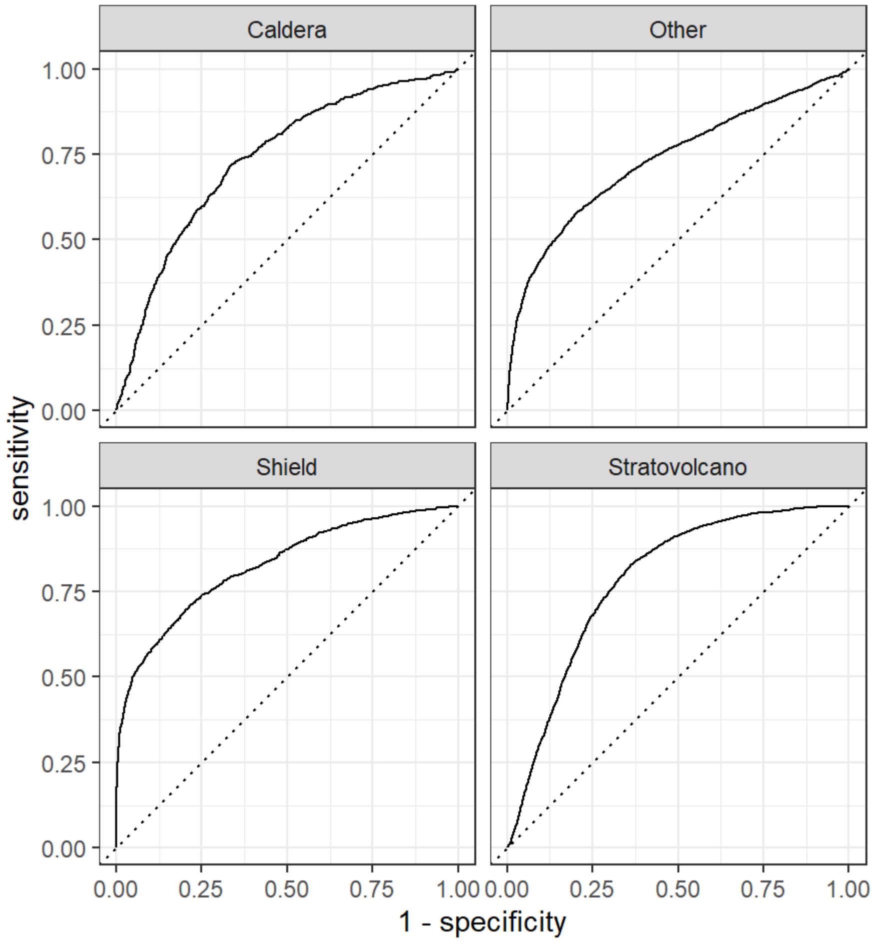
Example of a ROC curve



Source:Boehmke, B. and Greenwell, B., 2020: Hands-On Machine Learning with R, CRC Press, NY.

ROC for the 4 Volcano Types

```
volcano_pred %>%
   roc_curve(volcano_type, .pred_Caldera:.pred_Stratovolcano) %>%
    autoplot()
```



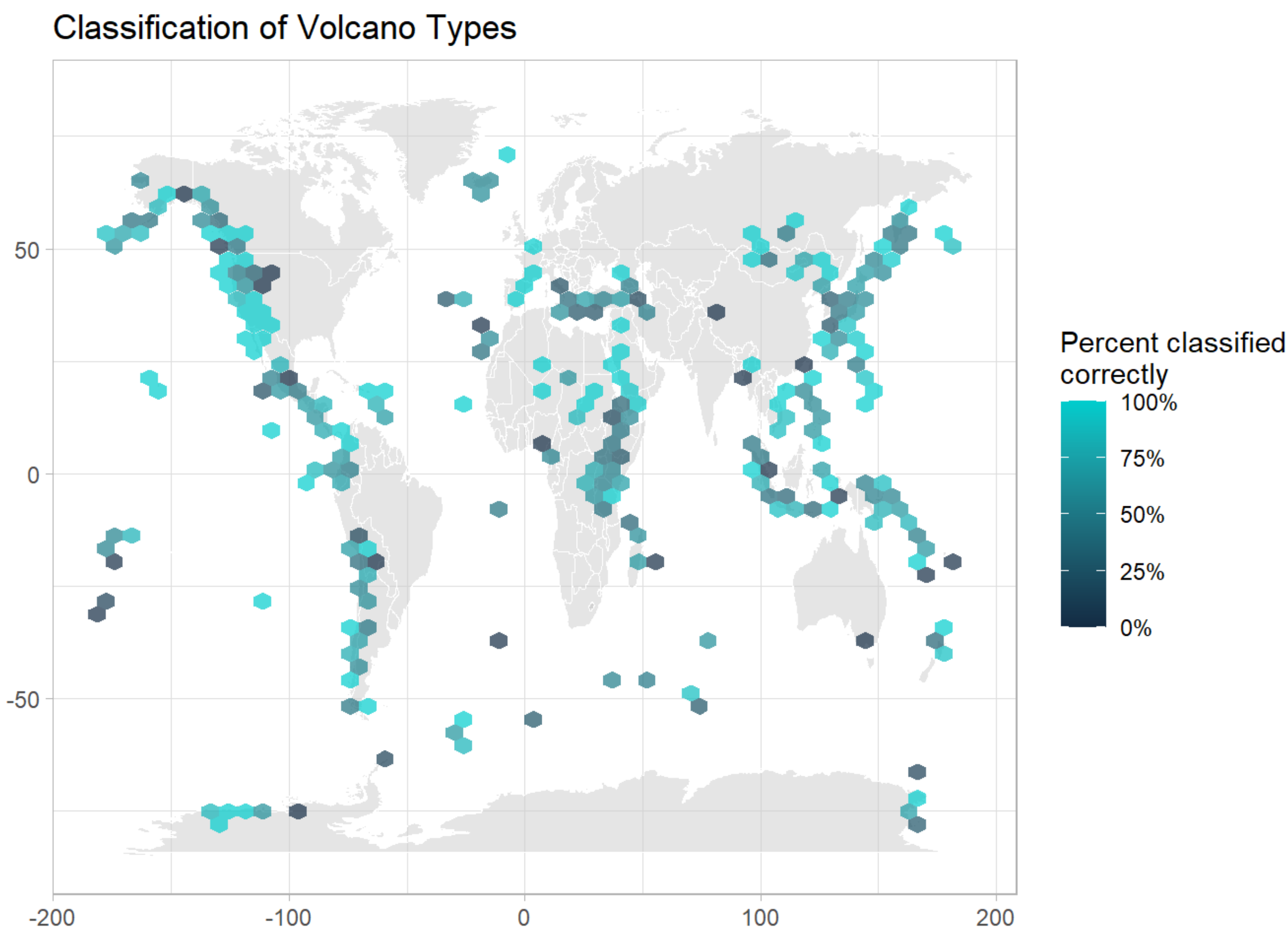
If you look through the performance results for the rf model, we certainly are not doing great!

The spatial information appears really important for the model, along with the presence of basalt and a subduction zone. Let's explore the spatial information a bit further, and make a map showing how right or wrong our modeling is across the world.

We'll make a map using stat_summary_hex(). Within each hexagon, let's take the mean of correct values to find what percentage of volcanoes were classified correctly, across all our bootstrap resamples.

```
ggplot() +
  geom_map(
    data = world, map = world,
    aes(long, lat, map_id = region),
    color = "white", fill = "gray80", size = 0.05, alpha = 0.5
  ) +
  stat_summary_hex(
    data = volcano_pred,
    aes(longitude, latitude, z = as.integer(correct)),
    fun = "mean",
    alpha = 0.7, bins = 50
  ) +
  scale_fill_gradient(high = "cyan3", labels = scales::percent) +
  theme_light() +
  labs(x = NULL, y = NULL, fill = "Percent classified\ncorrectly")+
  ggtitle("Classification of Volcano Types")
```

```
## Warning: Ignoring unknown aesthetics: x, y
```



The mapped results portray a much better picture. So the binning and spatial smoothing helped reduce some of the variance providing a much increased correct percentages in the spatial distribution of the 4 volcano types.

For further information on the analysis of the volcano dataset, please look at the following web sites:

https://rpubs.com/rhibarb6/volcano (https://rpubs.com/rhibarb6/volcano)

https://www.youtube.com/watch?v=vnxTGYL3C1M (https://www.youtube.com/watch?v=vnxTGYL3C1M) (tidyXep10)

https://juliasilge.com/blog/multinomial-volcano-eruptions/ (https://juliasilge.com/blog/multinomial-volcano-eruptions/) (Silge’s multinomial presentation)