Classification Metrics

Setup

Setup

Modeling steps:

1. Clean the data
    - What do we do about NA's?
    - Convert categorical variables to factors
2. Establish a model
    - Or many models to try?
    - Do we need to tune any hyperparameters?
3. Establish a recipe
    - Or many recipes to try?
    - How will we transform our variables?
    - Categorical to dummy variables? (but not the response!)
    - (data = full dataset)
4. Make workflows

1. Send the workflow to cross validation for model selection
    - For comparing different *models*
    - For *tuning*
    - For comparing different *recipes*
2. Send your final model to cross-validation for final metrics
    - why do we cross-validate for final metrics?
3. Fit the final model on the full dataset - this is your finished product!

Setup

```
ins <- read_csv("https://www.dropbox.com/s/bocjjyo1ehr5auz/insurance.csv?dl=1")

ins <- ins %>%
  mutate(
    smoker = factor(smoker)
  ) %>%
  drop_na()

knn_mod <- nearest_neighbor(neighbors = 5) %>%
  set_engine("kknn") %>%
  set_mode("classification")

knn_recipe <- recipe(smoker ~ age + bmi + charges,
                     data = ins)

knn_wflow <- workflow() %>%
  add_recipe(knn_recipe) %>%
  add_model(knn_mod)

cvs <- vfold_cv(ins, v = 5)

knn_fit <- knn_wflow %>%
  fit_resamples(cvs)
```

Metric 1: Accuracy

Accuracy

What percent of our guesses were correct?

```
knn_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 5
##   .metric  .estimator  mean     n std_err
##   <chr>    <chr>      <dbl> <int>   <dbl>
## 1 accuracy binary     0.965     5 0.00359
## 2 roc_auc  binary     0.992     5 0.00417
```

Accuracy

The problem: Consider this data.

```
##   [1] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
##  [19] "B" "B" "B" "B" "B" "A" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
##  [37] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
##  [55] "A" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
##  [73] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
##  [91] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
```

If I guess "B" every time, I'll have 98% accuracy!

Metric 2: ROC

ROC

ROC = "reciever operating charateristic** (ew)

FALSE Positive Rate = (how many A's did we say were B)/(how many did we say were "B" total)

How many did we misclassify as B?

True Positive Rate = (how many B's did we say were B)/(how many B's are there total)

How many true B's did we miss?

ROC

ROC = plots TP and FP across many decision boundaries

First, find the probability that the model assigns each observation for the first category of your categorical variable. (Generally, this is alphabetical:)

```
knn_final_fit <- knn_wflow %>%
  fit(ins)

ins <- ins %>%
  mutate(
    prob_nonsmoker = predict(knn_final_fit, ins, type = "prob")$.pred_no
  )
```

ROC

```
ins %>%
  roc_curve(truth = smoker, prob_nonsmoker) %>%
  autoplot()
```

ROC

GOOD: The ROC curve is way above the line (we can achieve a really good TP rate without sacrificing FP rate)

MEDIUM: The ROC curve is on the line (FP/TP is a trade-off)

BAD: The ROC curve is way below the line (we can't have good TP without bad FP)

ROC-AUC is the area under the curve - large values are good!

Try it!

Open Activity-Classification-2.Rmd again

Go to

Scroll down to the list of metrics

As a group, research one of the metrics that we haven't discussed in class, and compute it for some of your models.