

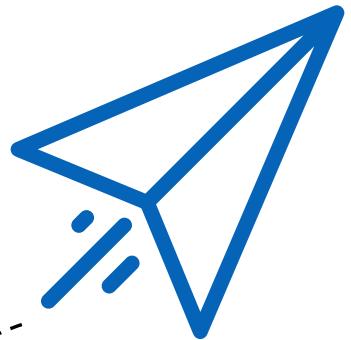
A photograph of a large elephant standing in a grassy savanna. The elephant is facing away from the camera, its large ears and textured skin visible. In the background, the majestic Mount Kilimanjaro rises, its peak covered in snow and surrounded by clouds. The sky is a clear, pale blue.

# Tanzania.

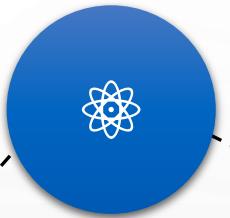
Predicting the Operational Status of Water Pumps by Thomas Skowronek

# Motivation

Why I choose this project.



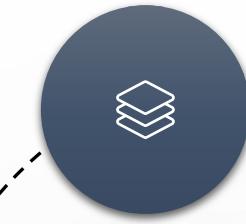
**Time Inc.:**  
Cape Town South Africa is  
90 days away from running  
out of water



**Home State of Michigan:**  
5 great lakes and 11K  
inland lakes > 5 acres



**DrivenData:**  
Pump it Up: Data  
Mining the Water  
Table



**Project Goal:**  
Apply classification  
algorithm using R



# The Data

59,400 observations, 39 predictor variables & 1 response variable

`amount_tsh` - Total static head (amount water available to waterpoint)

`date_recorded` - The date the row was entered

`funder` - Who funded the well

`gps_height` - Altitude of the well

`installer` - Organization that installed the well

`longitude` - GPS coordinate

`latitude` - GPS coordinate

`wpt_name` - Name of the waterpoint if there is one

`num_private` -

`basin` - Geographic water basin

`subvillage` - Geographic location

`region` - Geographic location

`region_code` - Geographic location (coded)

`district_code` - Geographic location (coded)

`lga` - Geographic location

`ward` - Geographic location

`population` - Population around the well

`public_meeting` - True/False

`recorded_by` - Group entering this row of data

`scheme_management` - Who operates the waterpoint

`scheme_name` - Who operates the waterpoint

`permit` - If the waterpoint is permitted

`construction_year` - Year the waterpoint was constructed

`extraction_type` - The kind of extraction the waterpoint uses

`extraction_type_group` - The kind of extraction the waterpoint uses

`extraction_type_class` - The kind of extraction the waterpoint uses

`management` - How the waterpoint is managed

`management_group` - How the waterpoint is managed

`payment` - What the water costs

`payment_type` - What the water costs

`water_quality` - The quality of the water

`quality_group` - The quality of the water

`quantity` - The quantity of water

`quantity_group` - The quantity of water

`source` - The source of the water

`source_type` - The source of the water

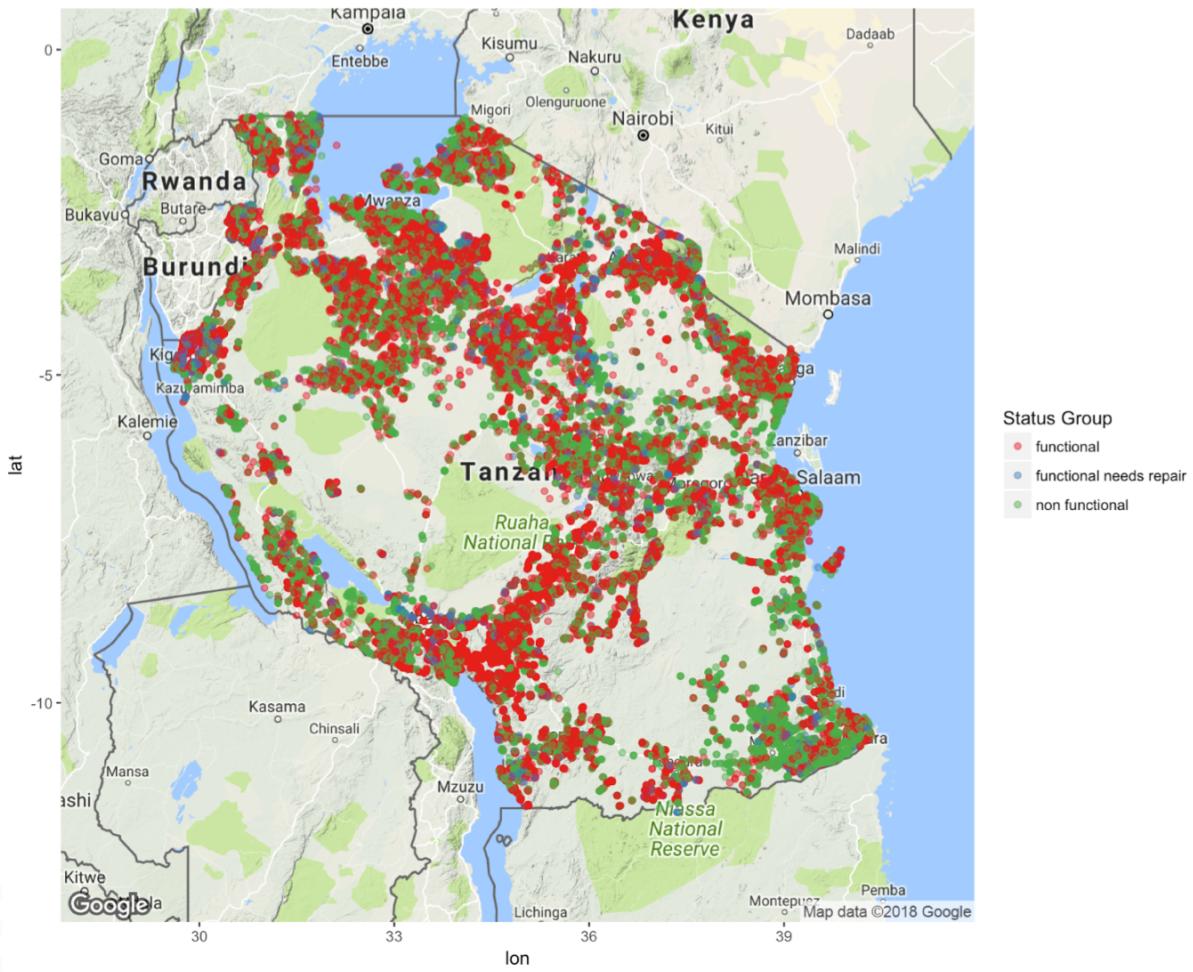
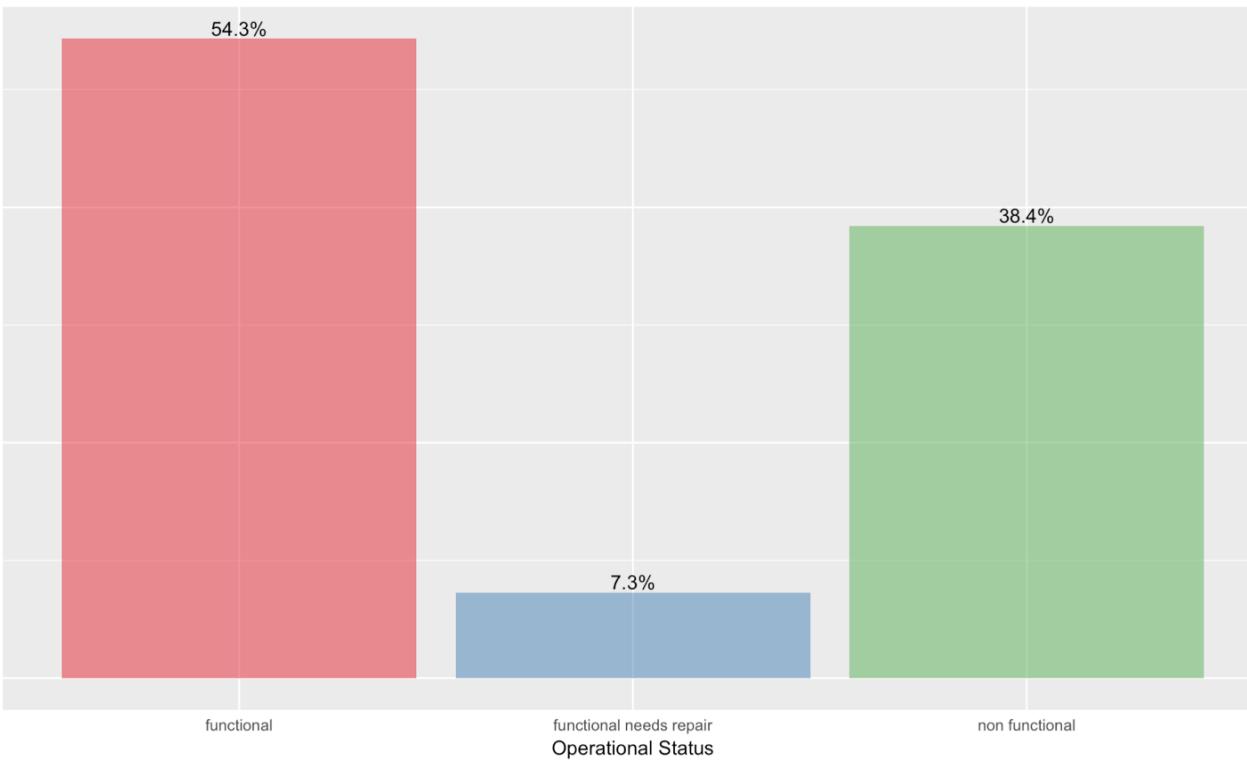
`source_class` - The source of the water

`waterpoint_type` - The kind of waterpoint

`waterpoint_type_group` - The kind of waterpoint

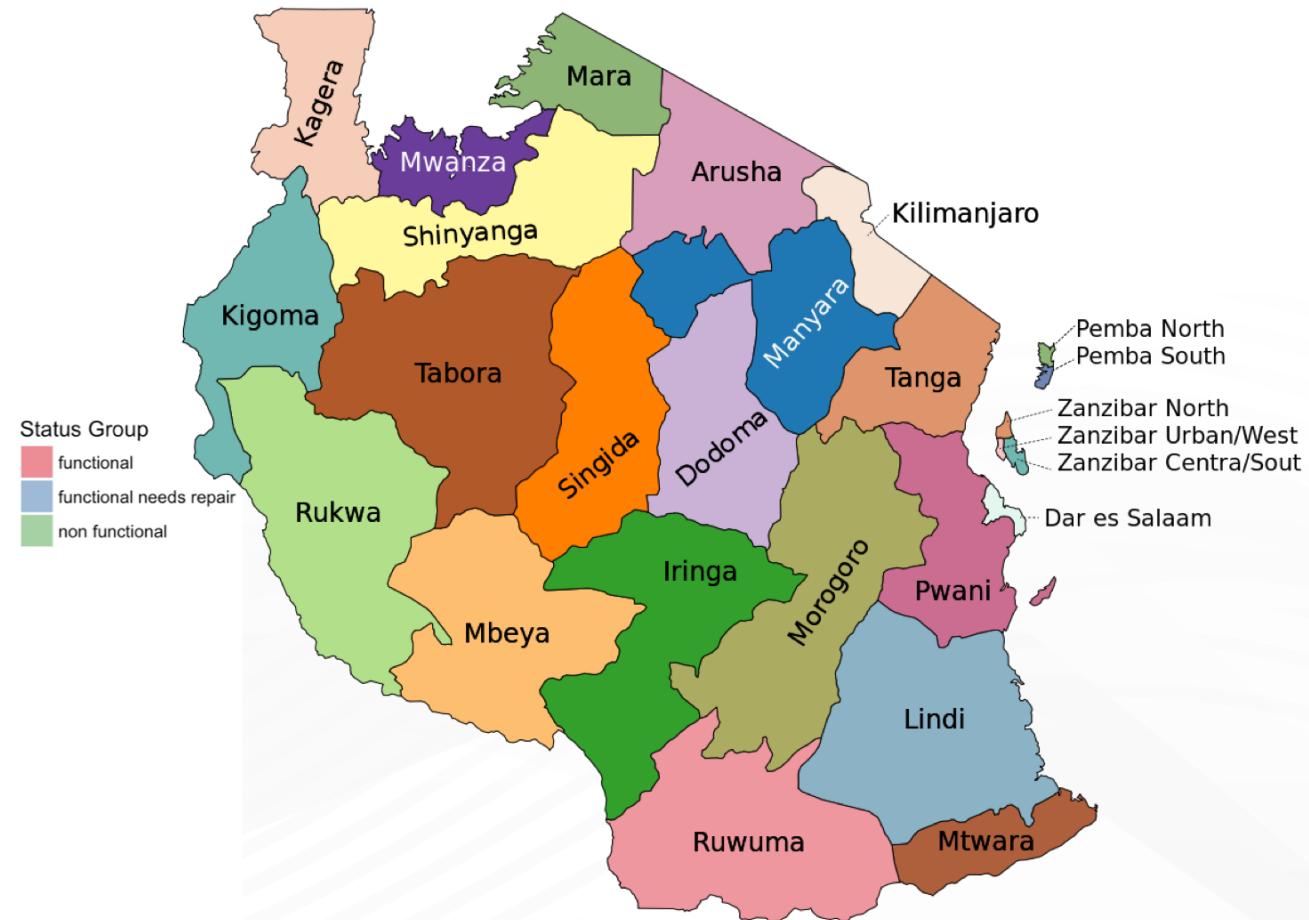
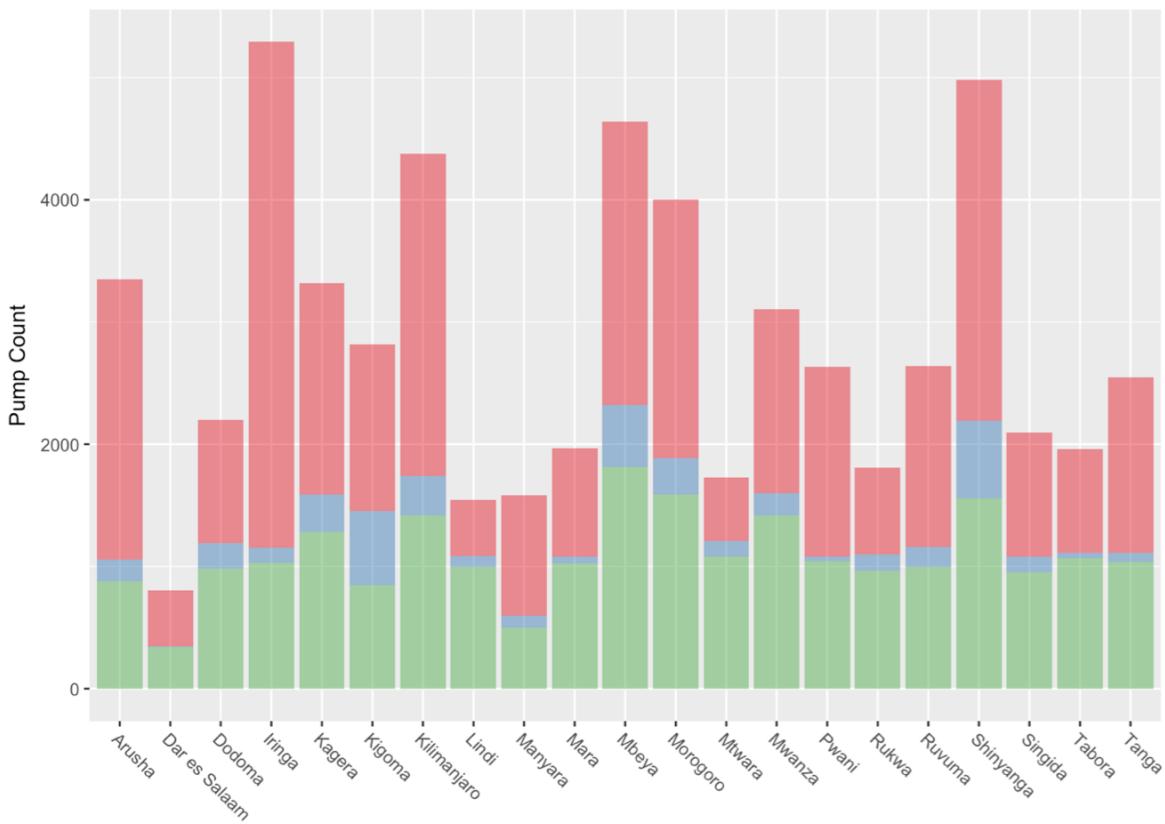
# Pump Status Group

Distribution and Location



# Regions

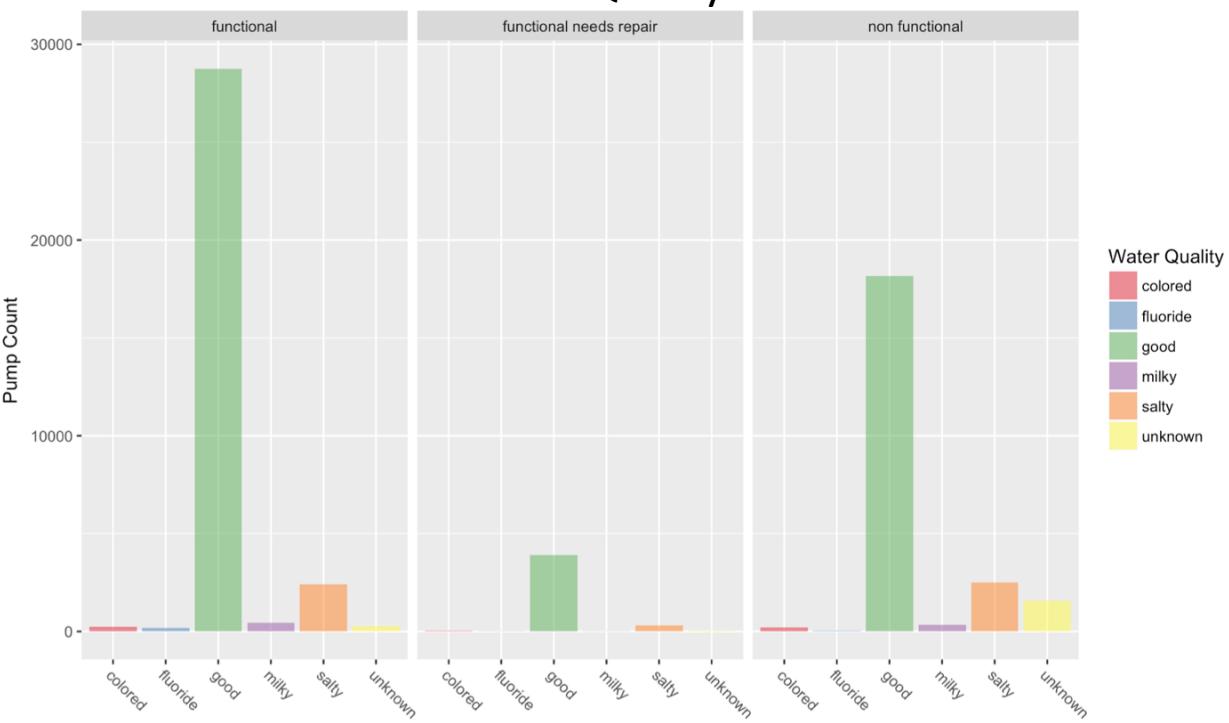
Geographic location



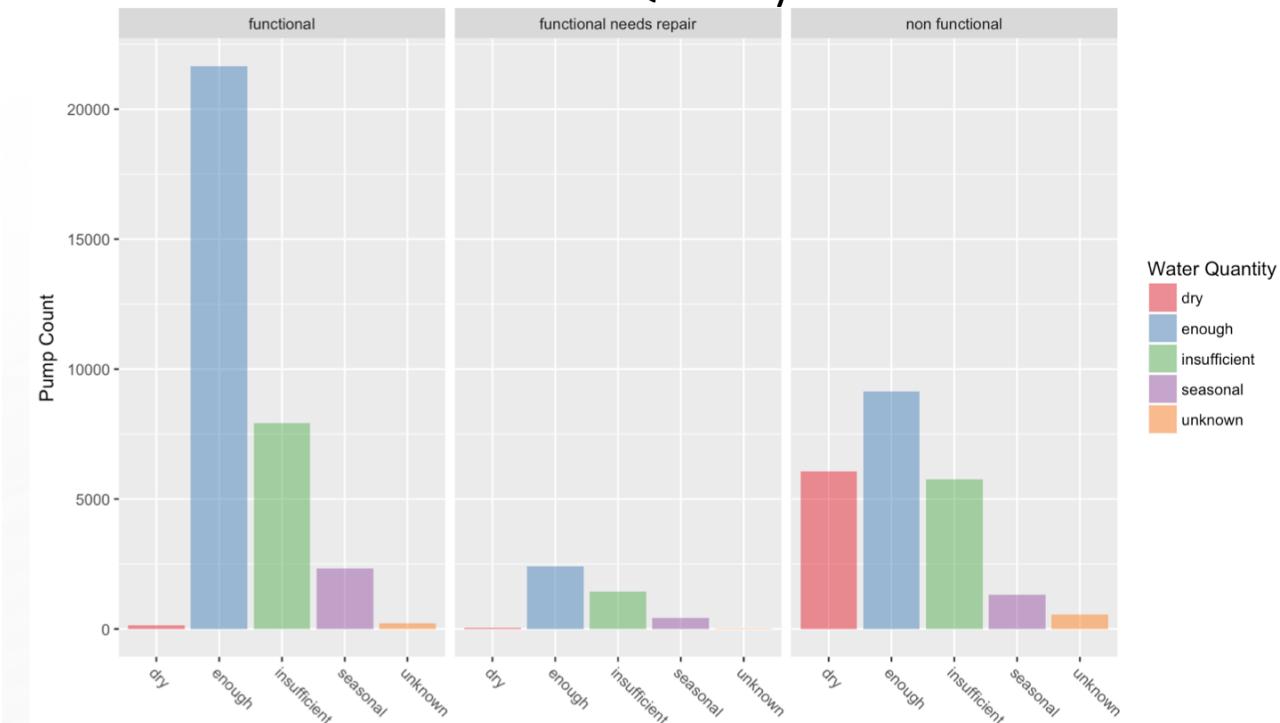
# Water Quality & Quantity

The quality of the water and the quantity of water

## Water Quality



## Water Quantity



Water Quality

- colored
- fluoride
- good
- milky
- salty
- unknown

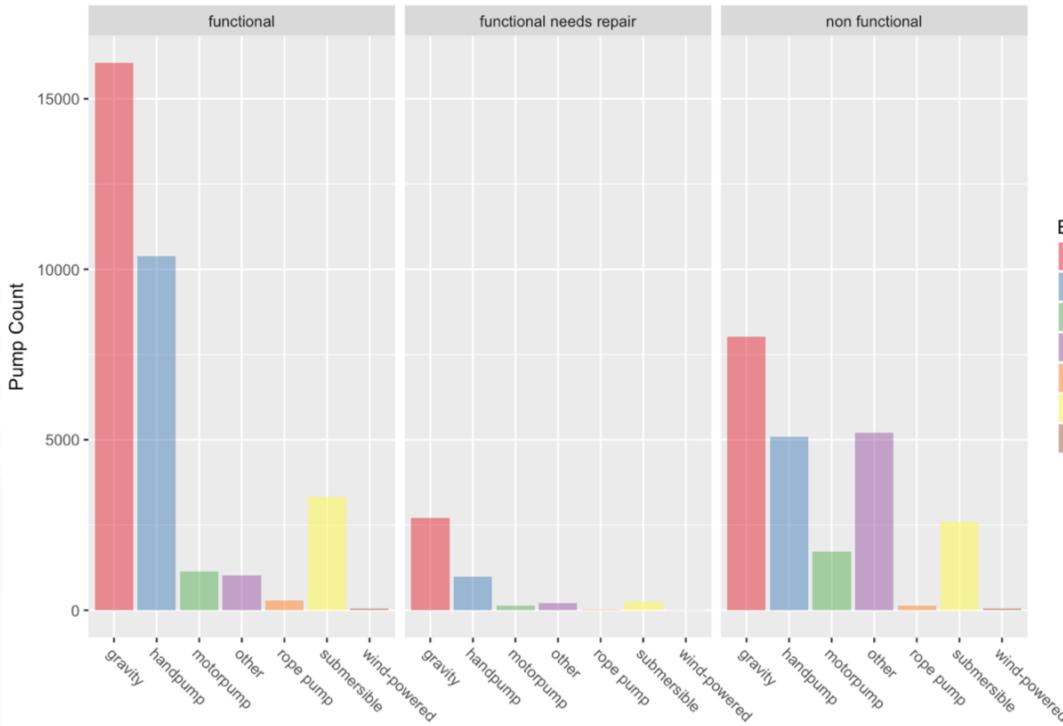
Water Quantity

- dry
- enough
- insufficient
- seasonal
- unknown

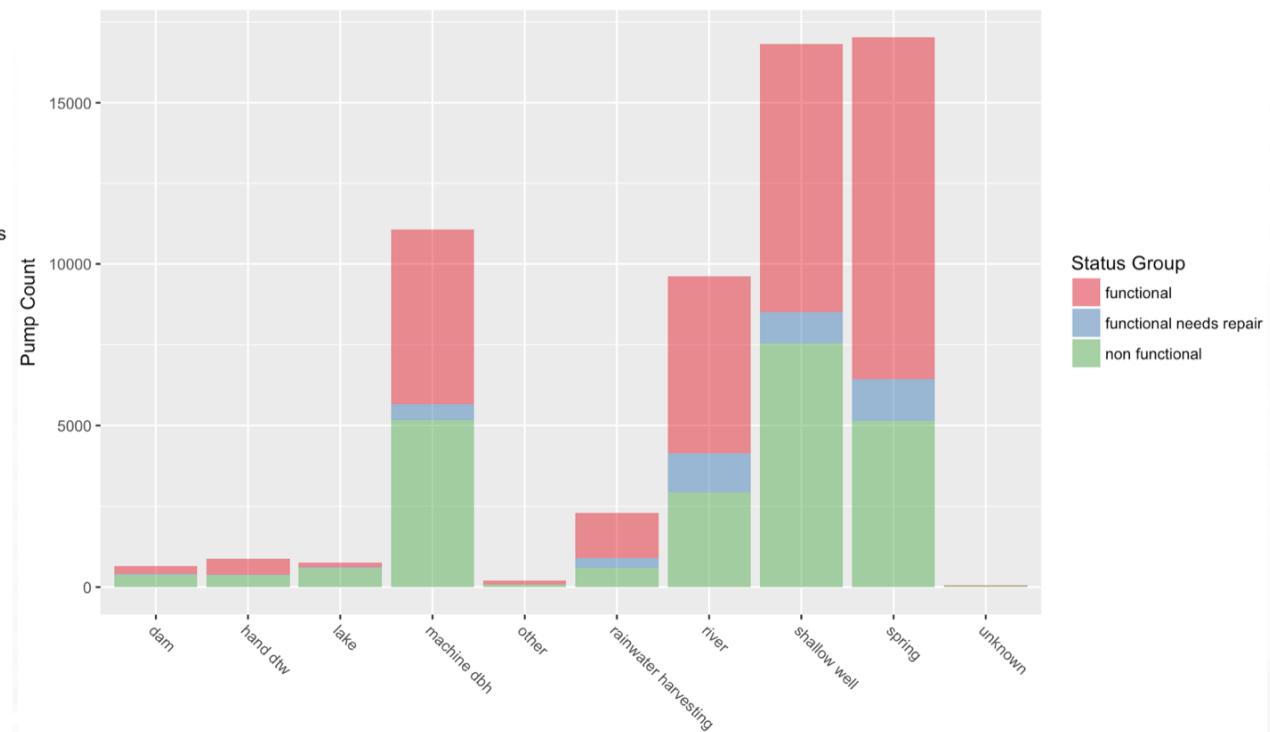
# Extraction Type & Water Source

The kind of extraction the water point uses and the source of the water

## Extraction Type

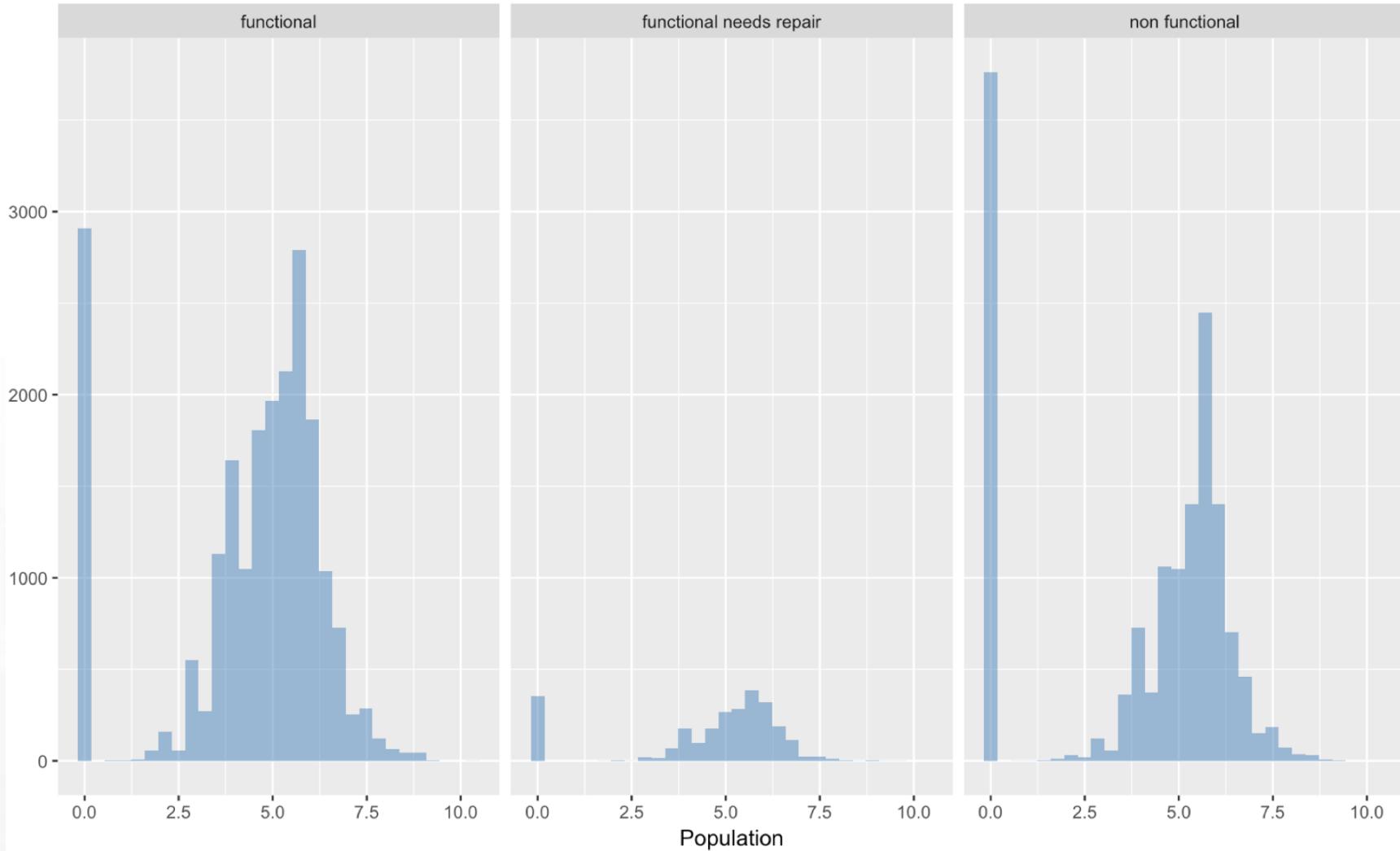


## Water Source



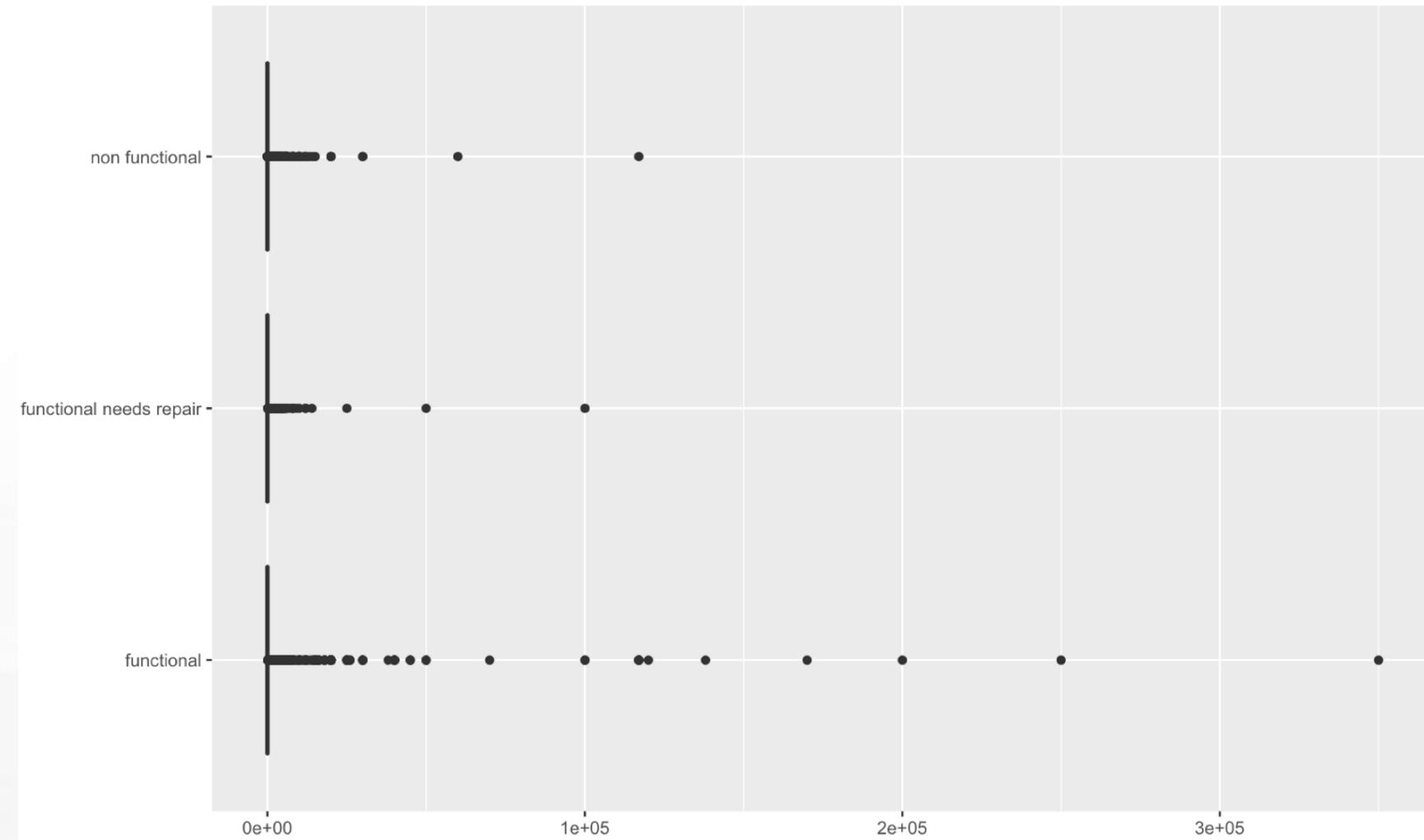
# Population

Population around the well



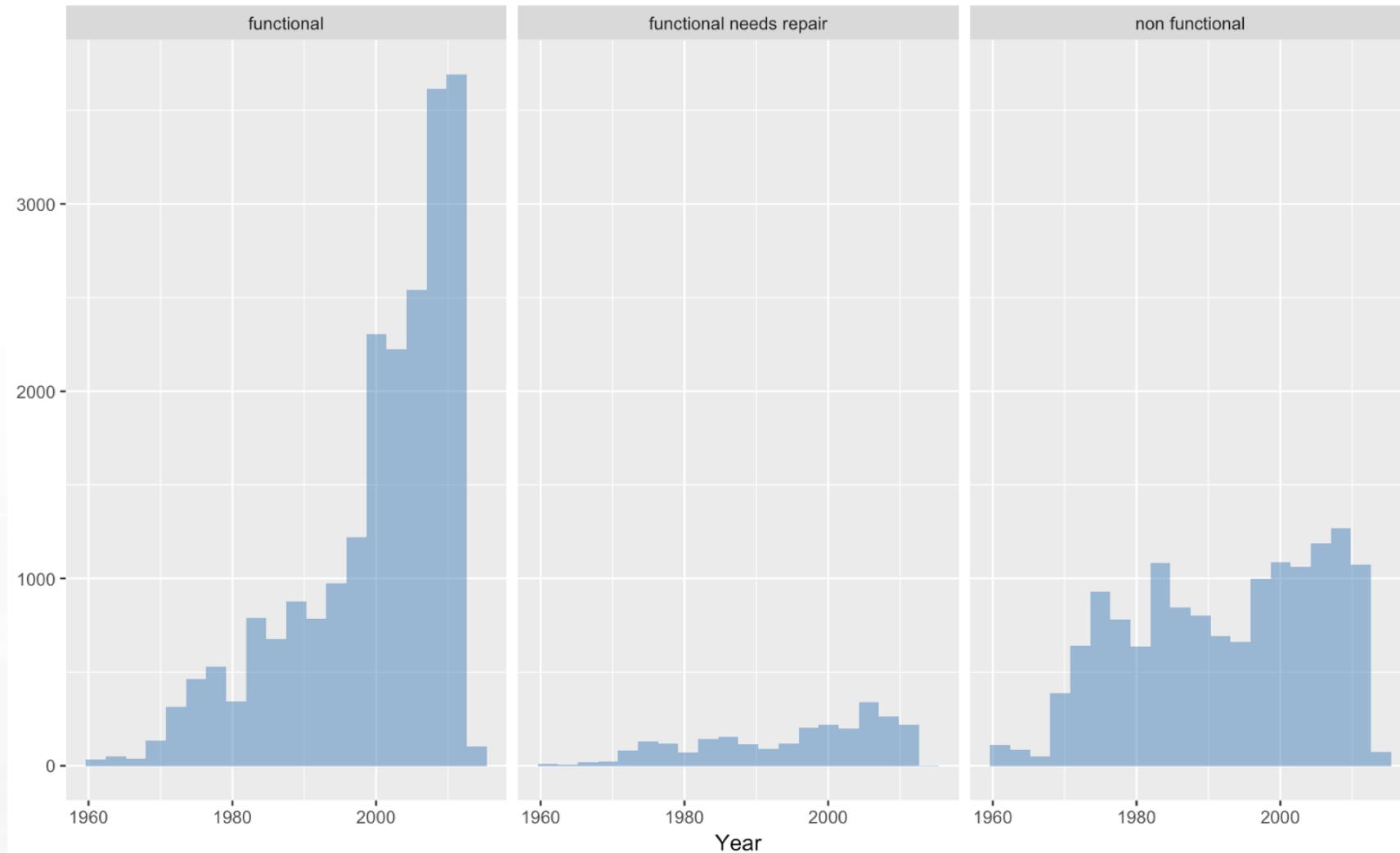
# Total Static Head Amount

Amount of water available to water point



# Construction Year

Year the water point was constructed

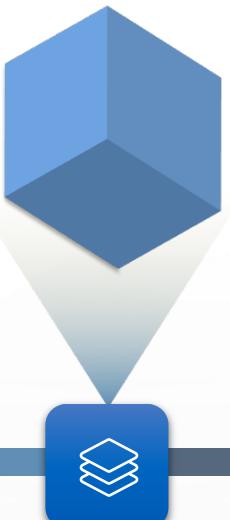


# Supervised Learning

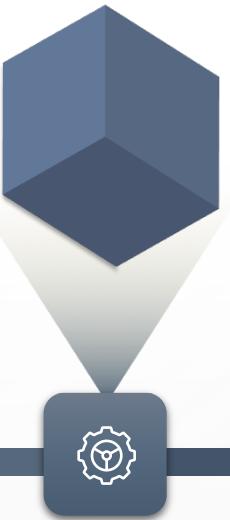
## Classification Algorithms



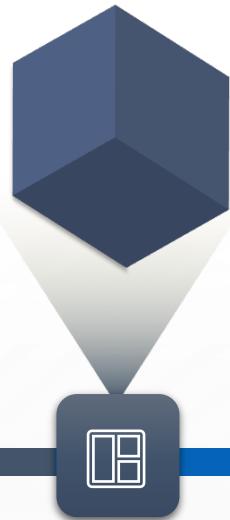
**Decision Tree**  
Package: C50



**Random Forest**  
Package: randomForest



**Extreme Gradient Boosting**  
Package: xgboost



**Support Vector Machine**  
Package: e1071



**Deep Neural Network**  
Package: h2o

# Iterative Approach

Start simple, then attempt to improve performance.

Create the initial models using a subset of the data.

**02**

Manually applied different parameters to tune the models.

**04**

**Step Number One**

**Step Number Two**

**Step Number Three**

**Step Number Four**

**01**

Increase the number of attributes used by the models.

**03**

Automate the tuning process to increase the quantity of model variation.

# Extreme Gradient Boosting

Package xgboost

5th Place Submission – DrivenData Score: 0.3787

✓ 80/20 train/test

✓ sparse.model.matrix {Matrix}

✓ Select subset of attributes

✓ Round: 25, Max Depth: 15, ETA: 1

✓ Full set of predictors

✓ Round: 100, Max Depth: 15, ETA: 0.3

Classification Error: 19.01%

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0    1    2
##           0 5755  884  456
##           1  547 3601 143
##           2  149   79 264
##
## Overall Statistics
##
##                           Accuracy : 0.8099
##                           95% CI : (0.8027, 0.8169)
##   No Information Rate : 0.5431
##   P-Value [Acc > NIR] : < 2.2e-16
##
##                           Kappa : 0.6439
##   Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                         Class: 0 Class: 1 Class: 2
## Sensitivity          0.8921  0.7890  0.30591
## Specificity          0.7531  0.9057  0.97930
## Pos Pred Value       0.8111  0.8392  0.53659
## Neg Pred Value       0.8545  0.8731  0.94739
## Precision            0.8111  0.8392  0.53659
## Recall              0.8921  0.7890  0.30591
## F1                  0.8497  0.8133  0.38967
## Prevalence           0.5431  0.3842  0.07266
## Detection Rate       0.4845  0.3032  0.02223
## Detection Prevalence 0.5973  0.3613  0.04142
## Balanced Accuracy    0.8226  0.8473  0.64261
```

Class 0: functional, Class 1: non functional, Class 2: functional needs repair

# Support Vector Machine

Package e1071



80/20 train/test



Select subset of attributes



Full set of predictors



Kernels: radial, polynomial, sigmoid



Tuning via Caret with 10-fold cross validation

Classification Error: 32.12%

4th Place Submission – DrivenData Score: 0.6764

```
## Confusion Matrix and Statistics
##
##          functional functional needs repair non functional
##    functional                  5823                 733            2328
##    functional needs repair      18                   9            5
##    non functional              610                121            2231
##
## Overall Statistics
##
##               Accuracy : 0.6788
##                 95% CI : (0.6703, 0.6872)
##   No Information Rate : 0.5431
##   P-Value [Acc > NIR] : < 2.2e-16
##
##               Kappa : 0.3548
##   Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                                     Class: functional Class: functional needs repair Class: non functional
##   Sensitivity                      0.9027                  0.0104287           0.4888
##   Specificity                      0.4360                  0.9979119           0.9001
##   Pos Pred Value                   0.6554                  0.2812500           0.7532
##   Neg Pred Value                   0.7902                  0.9279082           0.7383
##   Precision                        0.6554                  0.2812500           0.7532
##   Recall                           0.9027                  0.0104287           0.4888
##   F1                               0.7594                  0.0201117           0.5929
##   Prevalence                       0.5431                  0.0726553           0.3842
##   Detection Rate                  0.4902                  0.0007577           0.1878
##   Detection Prevalence             0.7479                  0.0026941           0.2494
##   Balanced Accuracy                0.6693                  0.5041703           0.6944
```

# Deep Neural Network

Package h2o

3rd Place Submission – DrivenData Score: 0.7745

- 80/20 train/test
- Full set of predictors
- Three hidden layers with 200 nodes each
- Disabled early stopping
- Early stopping enabled with defined stopping criteria
- Grid search

Classification Error: 22.17%

```
## Confusion Matrix and Statistics
##
##                                     Reference
## Prediction                  functional functional needs repair non functional
##   functional                      5595                 493           1067
##   functional needs repair          72                   208            46
##   non functional                  764                   189           3434
##
## Overall Statistics
##
##                               Accuracy : 0.7783
##                               95% CI  : (0.7707, 0.7858)
##   No Information Rate : 0.5419
##   P-Value [Acc > NIR] : < 2.2e-16
##
##                               Kappa : 0.5814
##   Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                                     Class: functional Class: functional needs repair Class: non functional
## Sensitivity                      0.8700                  0.23371          0.7552
## Specificity                      0.7131                  0.98925          0.8698
## Pos Pred Value                   0.7820                  0.63804          0.7828
## Neg Pred Value                   0.8226                  0.94091          0.8512
## Precision                        0.7820                  0.63804          0.7828
## Recall                           0.8700                  0.23371          0.7552
## F1                                0.8236                  0.34211          0.7687
## Prevalence                        0.5419                  0.07499          0.3831
## Detection Rate                   0.4714                  0.01753          0.2893
## Detection Prevalence             0.6029                  0.02747          0.3696
## Balanced Accuracy                 0.7915                  0.61148          0.8125
```

# Decision Tree

Package C50

✓ 80/20 train/test

✓ Select subset of attributes

✓ Rules-based model

✓ Boosted ruleset

✓ Full set of predictors

✓ Winnowing for feature selection

✓ Tuning via Caret: tuneLength, trainControl[method, repeats]

Classification Error: 19.23%

2nd Place Submission – DrivenData Score: 0.8123

```
## Confusion Matrix and Statistics
##
##                                     Reference
## Prediction                  functional  functional needs repair non functional
##   functional                      5769                 448          928
##   functional needs repair        145                  271          82
##   non functional                   537                 144         3554
##
## Overall Statistics
##
##                           Accuracy : 0.8077
##                           95% CI  : (0.8005, 0.8148)
##   No Information Rate : 0.5431
##   P-Value [Acc > NIR]  : < 2.2e-16
##
##                           Kappa : 0.6394
##   Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                                     Class: functional Class: functional needs repair Class: non functional
##   Sensitivity                      0.8943                0.31402          0.7787
##   Specificity                       0.7465                0.97939          0.9069
##   Pos Pred Value                    0.8074                0.54418          0.8392
##   Neg Pred Value                    0.8559                0.94798          0.8679
##   Precision                          0.8074                0.54418          0.8392
##   Recall                            0.8943                0.31402          0.7787
##   F1                                0.8486                0.39824          0.8078
##   Prevalence                         0.5431                0.07266          0.3842
##   Detection Rate                     0.4857                0.02282          0.2992
##   Detection Prevalence              0.6015                0.04193          0.3565
##   Balanced Accuracy                  0.8204                0.64671          0.8428
```

# Random Forest

Package randomForest

✓ 80/20 train/test

✓ Select subset of attributes

✓ Full set of predictors

✓ 10-fold cross validation using Caret

✓ Tuning via Caret: tuneLength, trainControl[method, repeats]

1st Place Submission – DrivenData Score: 0.8170

```
## Confusion Matrix and Statistics
##
##                                     Reference
## Prediction                  functional functional needs repair non functional
##   functional                           6386                   141                  170
##   functional needs repair                25                   690                  15
##   non functional                         40                   32                  4379
##
## Overall Statistics
##
##                               Accuracy : 0.9644
##                               95% CI : (0.9609, 0.9676)
## No Information Rate : 0.5431
## P-Value [Acc > NIR] : < 2.2e-16
##
##                               Kappa : 0.9347
## Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                                     Class: functional Class: functional needs repair Class: non functional
## Sensitivity                          0.9899                  0.79954                 0.9595
## Specificity                           0.9427                  0.99637                 0.9902
## Pos Pred Value                      0.9536                  0.94521                 0.9838
## Neg Pred Value                      0.9875                  0.98448                 0.9751
## Precision                            0.9536                  0.94521                 0.9838
## Recall                               0.9899                  0.79954                 0.9595
## F1                                    0.9714                  0.86629                 0.9715
## Prevalence                           0.5431                  0.07266                 0.3842
## Detection Rate                       0.5376                  0.05809                 0.3687
## Detection Prevalence                 0.5638                  0.06146                 0.3747
## Balanced Accuracy                    0.9663                  0.89795                 0.9748
```

Classification Error: 3.56%

# DrivenData Confirmation

Best Submission

## Pump it Up: Data Mining the Water Table

HOSTED BY DRIVENDATA

### Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
0.8170	375	4409	1 / 3

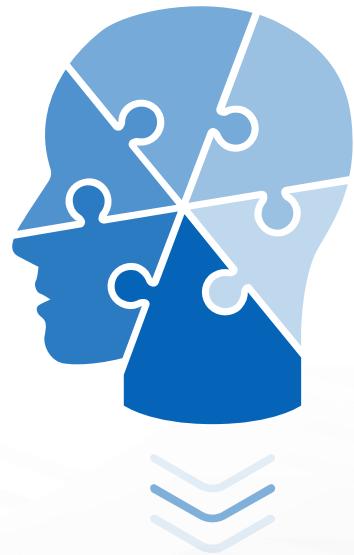
### EVALUATION METRIC

$$\text{Classification Rate} = \frac{1}{N} \sum_{i=0}^N I(y_i = \hat{y}_i)$$

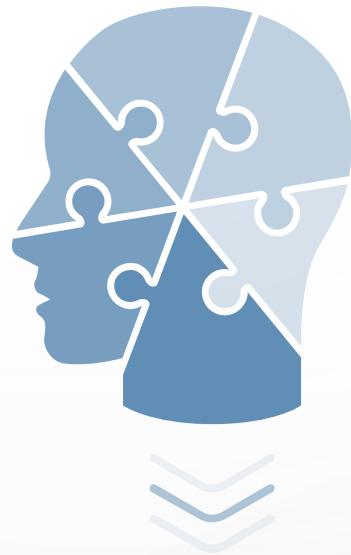
The metric used for this competition is the classification rate, which calculates the percentage of rows where the predicted class  $\hat{y}$  in the submission matches the actual class,  $y$  in the test set. The maximum is 1 and the minimum is 0. The goal is to maximize the classification rate.

# Conclusion

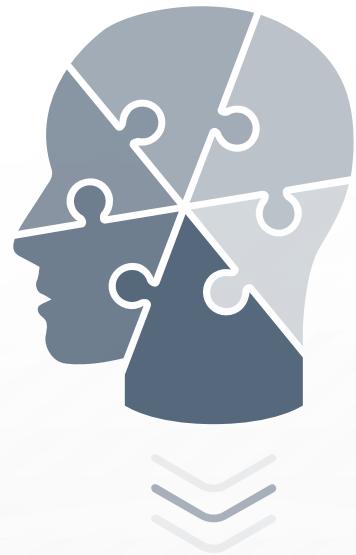
Project Takeaways.



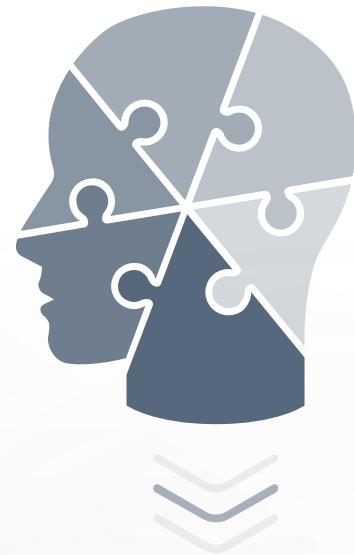
Data  
Cleansing



Feature  
Engineering



Model  
Tuning



Data  
Augmentation

# Project Repository

GitHub

The screenshot shows a GitHub repository page for 'tcskowronek/tanzanian-water-pumps'. The repository has 56 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made 15 seconds ago. The repository description is "Data mining the water table: Predicting the operating condition of a water points in Tanzania". The README.md file contains the title "Predicting the Operational Status of Tanzanian Water Pumps".

This repository

Pull requests Issues Marketplace Explore

tcskowronek / tanzanian-water-pumps

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Data mining the water table: Predicting the operating condition of a water points in Tanzania Edit

Add topics

56 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

tcskowronek Slide deck for presentation Latest commit 87eafb6 15 seconds ago

images Added region map of Tanzania a month ago

presentation Slide deck for presentation 15 seconds ago

src Final and best random forest model 6 hours ago

.gitignore Remove output 29 days ago

README.md Updated title a month ago

README.md

## Predicting the Operational Status of Tanzanian Water Pumps

Data mining the water table: Predicting the operating condition of a water points in Tanzania

GitHub Repository: <https://github.com/tcskowronek/tanzanian-water-pumps>

# References

- Baker, A. (2018). Cape Town is 90 days away from running out of water. Retrieved from <http://time.com/5103259/cape-town-water-crisis/>
- DrivenData. (2018a). How the process works. Retrieved from <https://www.drivendata.org/about/>
- DrivenData. (2018b). Pump it Up: Data mining the water table. Retrieved from <https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/>
- Kahle, D., & Wickham, H. (2013). ggmap: Spatial visualization with ggplot2. Retrieved from <https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf>
- Kuhn, M. (2013). Classification using C5.0. Retrieved from [https://static1.squarespace.com/static/51156277e4b0b8b2ffe11c00/t/51e67b45e4b0e6c130fb4d54/1374059333633/user\\_C5.0.pdf](https://static1.squarespace.com/static/51156277e4b0b8b2ffe11c00/t/51e67b45e4b0e6c130fb4d54/1374059333633/user_C5.0.pdf)
- LeDell, E. (2017). H2O Deep Learning. Retrieved from <https://github.com/ledell/sldm4-h2o/blob/master/sldm4-deeplearning-h2o.Rmd>
- MapFight. (n.d.). Tanzania Texas Comparison. Retrieved from <https://mapfight.appspot.com/tz-vs-texas/tanzania-texas-size-comparison>
- Michigan State University. (n.d.). Frequently asked questions regarding inland lakes management and use in Michigan. Retrieved from <http://michiganlakes.msu.edu/faq#WhatisthelargestinlandlakeinMichigan>
- Sarawat, M. (n.d.). Beginners tutorial on XGBoost and parameter tuning in R. Retrieved from <https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/>
- The World Bank. (2017). The World Bank in Tanzania. Retrieved from <http://www.worldbank.org/en/country/tanzania/overview>
- U.S. Census. (2017). Annual estimates of the resident population for the United States, regions, states, and Puerto Rico: April 1, 2010 to July 1, 2017. Retrieved from <http://www.worldbank.org/en/country/tanzania/overview>
- Water.org. (n.d.). Tanzania's water and sanitation crisis. Retrieved from <https://water.org/our-impact/tanzania/>
- World Health Organization. (n.d.). United Republic of Tanzania. Retrieved from <http://www.who.int/countries/tza/en/>



# Thank You For Watching.

Have a nice day ☺