Water Pumps

Steven Gusenius, Zuber Saiyed, Margarita Linets

About this Project

Using data from Taarifa and the Tanzanian Ministry of Water, we set out to predict where water pumps were likely to be functional, in need of repair of not functional at a certainl locale.

A smart understanding of which water pumps will fail can improve maintenance operations and ensure that clean, potable water is available to communities across Tanzania.

More information about the challenge and the dataset can be found here - https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/

About the Dataset

The datasets for this project were downloaded from www.drivendata.org and consisted of a two files of comma separated format. This first file contains 40 characteristic data of each water pump, indexed by a pump ID, to be used as predictors. A list of these predictors is provided in *APPENDIX A*.

The second file contains the status_group for each water pump, also indexed by pump ID. The status_group is the response we are attempting to predict and indicates the condition of a water pump. Its value can be: Functional (F), FunctionalNeedsRepair (FNR), or NonFunctional(NF). The respective percentages of each are: 54.3%, 7.3%, 38.4%.

In total, there is data for 59,400 water pumps.

Data Cleaning

Data Modification Initially the datasets were cleaned to make them compatible with processing. Primarily this consisted of addressing missing data and special characters. Then the predictor data was merged with the response data into a single dataset.

Data Excluded Following the merge, the pump ID was eliminated as it is not a meaningful predictor. One predictor, **recorded_by** was excluded because it had minimal variation for all water pumps. Several other categorical predictors were eliminated for having an excessive number of (greater than 30) levels. A list of these factor variables, and their associated number of levels, is available in *APPENDIX B*. This step was needed when Lasso was used. This because Lasso requires the inputs to be of type *model.matrix*. A model matrix creates a separate column of data for each level of each factor variable. This has a detrimental impact on both memory requirements and processing speed. In this case, the retention of all such factor variables exceeded the capacity of the R software. Further, it is a reasonable assumption that if a large proportion of the data is spread across many nominal factor levels, that factor variable will have diminished predictive power.

Data Exploration

Prior to model fitting, some effort was invested in understanding the content of the data. Various hypotheses were made and then evaluated through a number of simple, ad hoc analyses.

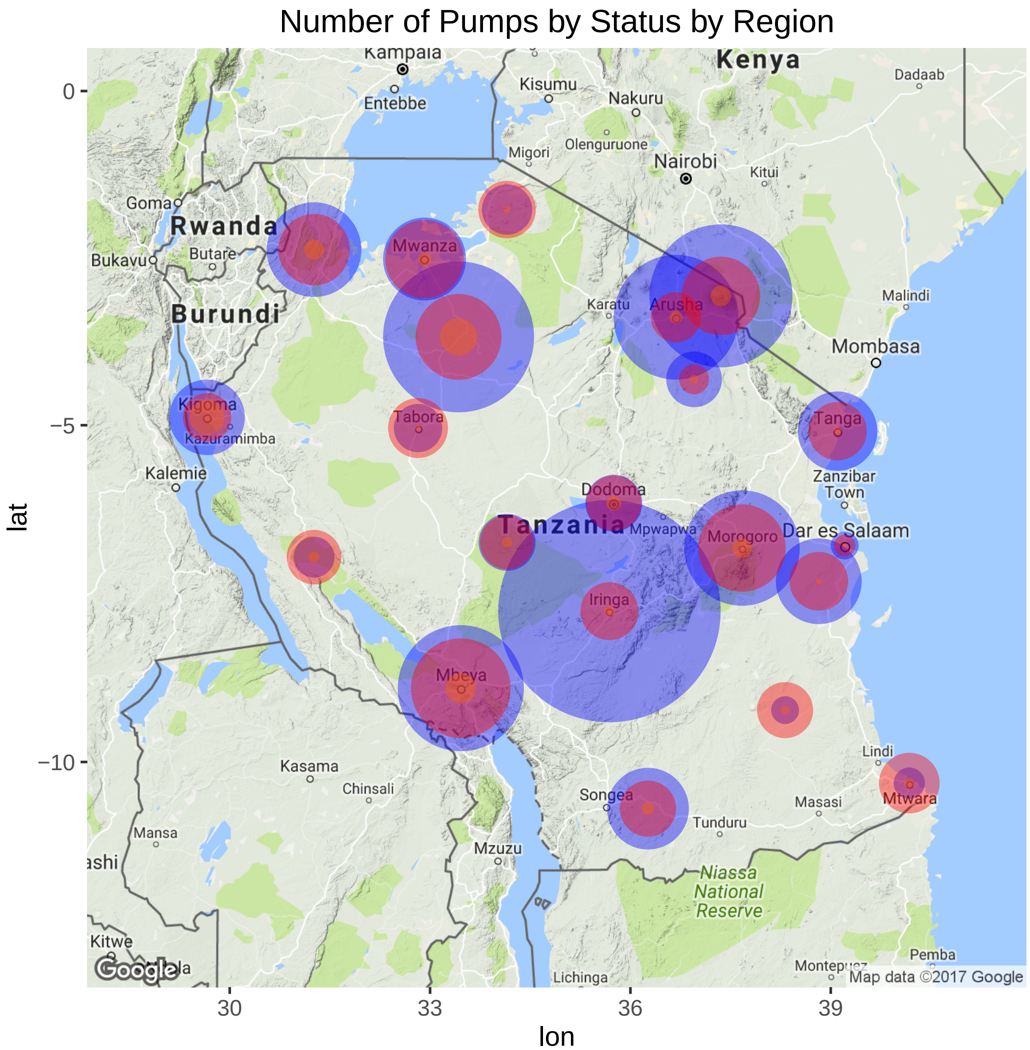
One such analysis was a data visualization where the frequency of the three **status_groups**, for each **region**, was plotted at the center of the respective region on a map of Tanzania. This map provide an understanding of how pump functionality was dispersed throughout the country, and serves as an indication of how many water pumps were contained in each region and whether each **region** had similar proportions of F, FNR, and NF water pumps.

Based on this map, it appears that districts with the fewest water pumps might have a larger number of pumps that are NF. For this reason a variable **regionalPumpCount** was added to the dataset.

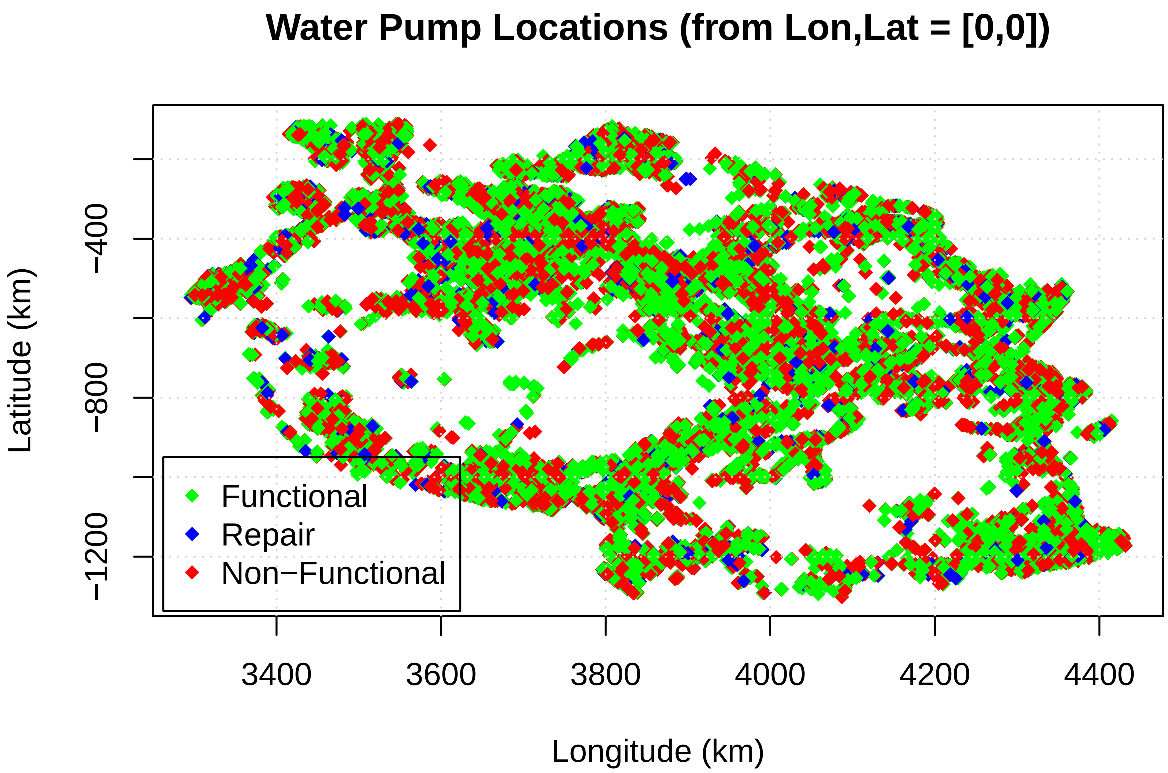
```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
##
## [[7]]
## [1] TRUE
##
## [[8]]
## [1] TRUE
```

```
# Clean Data: Filter out variables with more than 30 factors - model matrix becomes too large
KeepVars = setNames(data.frame(sapply(WaterPumps[,which(sapply(WaterPumps, is.character))], function(x)
                    c('factorlevels'))
KeepVars$vars = rownames(KeepVars)
WaterPumps = WaterPumps[,-which(names(WaterPumps) %in% KeepVars$vars[which(KeepVars$factorlevels==FALSE

# Remove Multicollinearity
```



Additionally visual analysis was performed by plotting the position of each water pumps, color coded by **status_group**. A simple spherical earth transformation allowed the water pump positions in longitude and latitude to be plotted in on a flat plane using linear units of kilometers. Given that linear units are preferable for fitting models, these transformed positions were added to the dataset as **East_km** and **North_km**. Because these would exhibit exceptionally high correlation with **longitude** and **latitude**, respectively, the latter variables were removed from the dataset.



The plot above was examined to see if there were signs of clustering among pumps of a specific **status_group**. While it did appear that there were some areas of the country with elevated proportions NF pumps, there was no recognizable pattern that could be leveraged for this evaluation. A visual comparison against a mean annual rainfall map of Tanzania (available on the internet), looked like it might exhibit correlation between areas with more rain and the location of all water pumps. A similar map of Tanzania average temperatures showed a potential positive correlate between hot temperatures and NF pumps. However, defining these relationships is beyond the scope of this effort.

Fit Approaches

For all models, cross validation was used. This consisted of separating the data into **training** and **validation** sets. The models were constructed using the **training** data, then their performances were evaluated using the **validation** data. The split between the two sets was approximately 70% **training** and 30% **validation**.

Given that relatively few of the variables contained numeric data, model approaches that utilize Euclidean distances between datapoints could not be used.

Because a small proportion of water pumps were of **status_group** FNR, some model types would ignore this state completely their predictions.

One approach for addressing this was the use of a Binary Outcome Lasso using a one-vs-one selection strategy. With this strategy, three **sub-models** were built. Each sub-model was assigned a level of the response variable. The remaining two levels were given the value of **other**. This forced the sub-model to focus on fitting only its assigned level. The outcome of each sub-model was an estimated probability that the each datapoint belonged to the assigned level. Each point was assessed against these three sets of predictions. The level with the highest probability was selected.

Given their general suitability for datasets of this nature, Random Forest and Random Forest with Boosting were also used.

Results

Binary Outcome Lasso

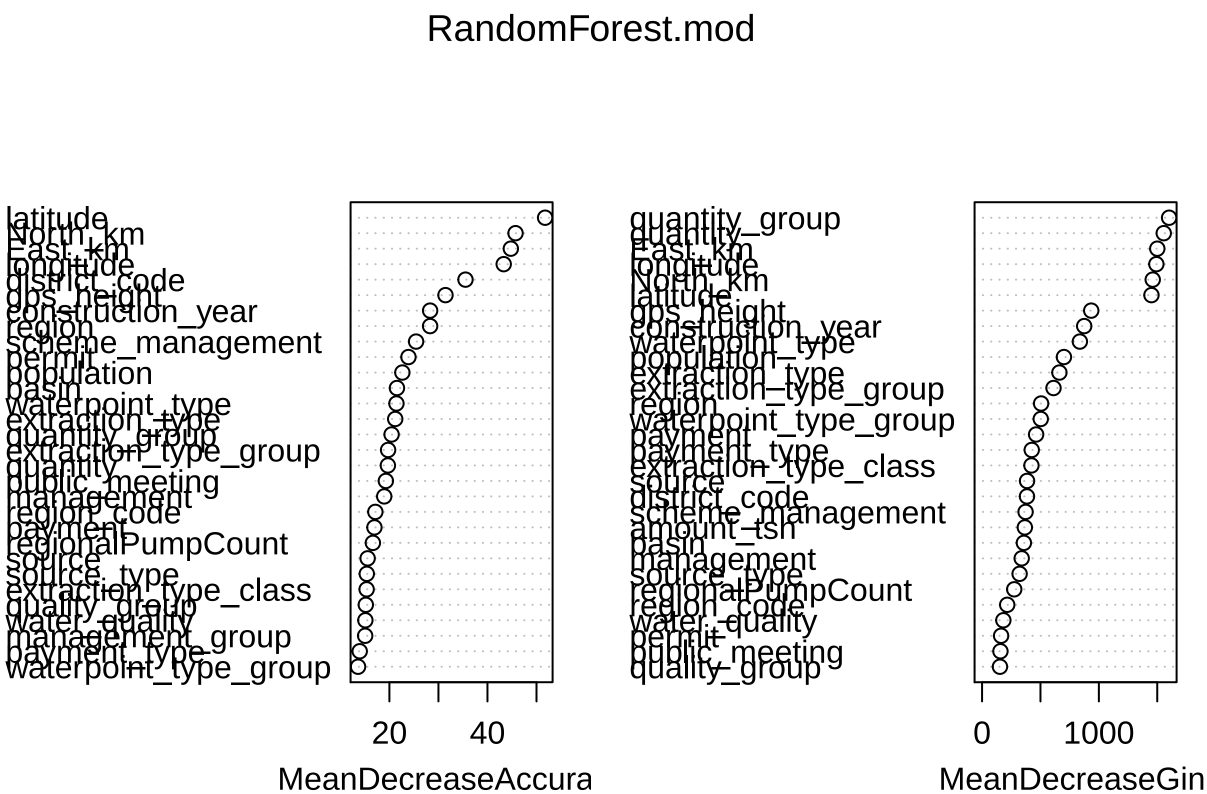
Table 1: Training Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.8979184	0.5739409	0.7136910	0.8979184	0.7359297
Class: functional needs repair	0.0517469	0.9966015	0.5451389	0.0517469	0.5241742
Class: non functional	0.6377146	0.8931351	0.7889533	0.6377146	0.7654249

Table 2: Validation Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.8949424	0.5652496	0.7122055	0.8949424	0.7300960
Class: functional needs repair	0.0584567	0.9961904	0.5434783	0.0584567	0.5273236
Class: non functional	0.6225584	0.8892925	0.7766581	0.6225584	0.7559255

Random Forest



From the random forest procedure, we obtained training set prediction accuracy of 95.4%. By contrast, in the validation set, the prediction accuracy was only 80.7%.

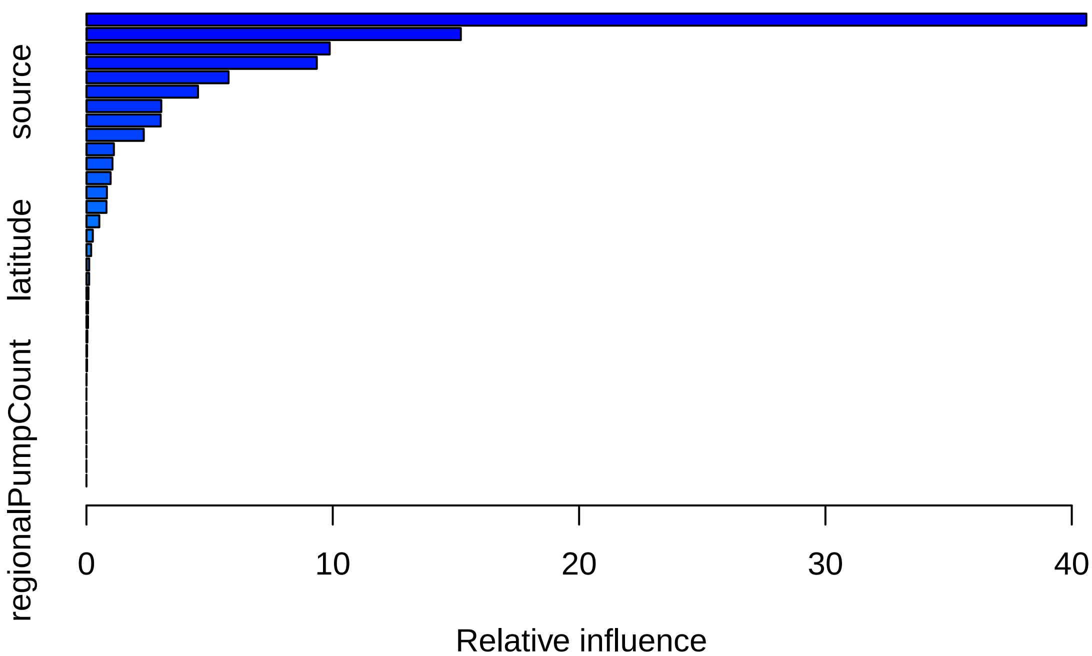
Table 3: Training Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.9871732	0.9243530	0.9391547	0.9871732	0.9557631
Class: functional needs repair	0.7511536	0.9957194	0.9324877	0.7511536	0.8734365
Class: non functional	0.9453637	0.9877567	0.9797450	0.9453637	0.9665602

Table 4: Validation Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.8929893	0.7381364	0.8039052	0.8929893	0.8155629
Class: functional needs repair	0.3219018	0.9799238	0.5543624	0.3219018	0.6509128
Class: non functional	0.7750037	0.9099083	0.8417610	0.7750037	0.8424560

Random Forest with Boosting




```
##                                         var     rel.inf
## quantity                           quantity 40.59262238
## waterpoint_type             waterpoint_type 15.19947468
## region                               region  9.87623989
## extraction_type             extraction_type  9.35216887
## payment                             payment  5.77012737
## source                               source  4.52967018
## waterpoint_type_group waterpoint_type_group  3.03888441
## construction_year         construction_year  3.01411050
## management                       management  2.32777818
## basin                                 basin  1.11349481
## amount_tsh                       amount_tsh  1.05546585
## East_km                             East_km  0.98167070
## scheme_management         scheme_management  0.83037039
## longitude                         longitude  0.81260716
## public_meeting               public_meeting  0.52248686
## source_type                     source_type  0.25907818
## latitude                           latitude  0.19157544
## population                       population  0.11273933
## district_code                 district_code  0.11071461
## extraction_type_group extraction_type_group  0.08291090
## water_quality                 water_quality  0.06500981
## permit                               permit  0.06356290
## gps_height                       gps_height  0.03871681
```

```
## region_code                            region_code  0.02666798
## extraction_type_class extraction_type_class  0.02641021
## quality_group                        quality_group  0.00544160
## num_private                            num_private  0.00000000
## management_group              management_group  0.00000000
## payment_type                          payment_type  0.00000000
## quantity_group                      quantity_group  0.00000000
## source_class                          source_class  0.00000000
## North_km                                  North_km  0.00000000
## regionalPumpCount            regionalPumpCount  0.00000000
```

Table 5: Training Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.9233057	0.5744134	0.7195780	0.9233057	0.7488595
Class: functional needs repair	0.0906394	0.9958491	0.6321839	0.0906394	0.5432443
Class: non functional	0.6347175	0.9190299	0.8308132	0.6347175	0.7768737

Table 6: Validation Data Performance

	Sensitivity	Specificity	Precision	Recall	Balanced.Accuracy
Class: functional	0.9183799	0.5651261	0.7174175	0.9183799	0.7417530
Class: functional needs repair	0.1067810	0.9954647	0.6462264	0.1067810	0.5511228
Class: non functional	0.6184462	0.9142676	0.8168768	0.6184462	0.7663569

Conclusion

Based on the results obtained with the previous models, it is evident that the random forest peroforms best. Below is the comparison of prediction accuracy of all three models over the validation dataset.

Table 7: Model Comparison

Model	Accuracy
Binary Outcome Lasso	0.7306397
Random Forest	0.8067901
Boosted Random Forest	0.7453423

Appendices

Appendix A

Table 8: Metadata

Variable	Definition
amount_tsh	Total static head (amount water available to waterpoint)
date_recorded	The date the row was entered

Variable	Definition
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	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

Appendix B

Source code for the project can be found here: https://github.com/StatisticsGuru/WaterPump