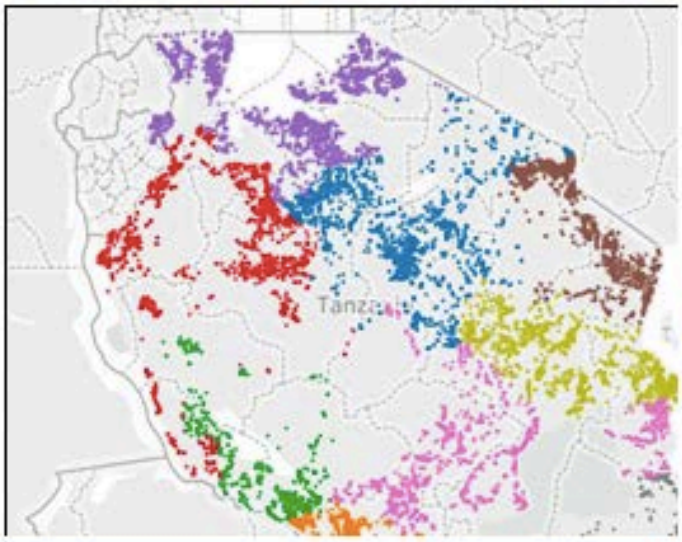
Pump
It
Up



Pump It Up:
Predicting the Operating Condition of Tanzanian Wells

Final Report

Tanzanian Water Ministry

Engagement Letter

November 15, 2016

Dr. Donald Wedding, CEO

Tanzanian Ministry of Water

Dear Dr. Wedding,

We are pleased to submit the final deliverable for the Pump It Up project. As laid out in the project goals stated at the beginning of the project, this deliverable includes the final recommended model for predicting the functionality of the wells. We also offer a recommendation on which tools should be used to solve future modeling problems and provide data visualizations that aid in analyzing the data as well as provide maps of the location and functionality of the wells.

This deliverable begins by outlining the problem statement and providing a description of the data and the data quality. We then describe and analyze the different transformations performed on the data before providing an overview of the data visualization and exploratory data analysis completed for this report. In the next section we performed and evaluated different models using three different tools, R, Azure and ANGOSS. The conclusions and recommendations portion of the report includes the final results of our analysis. It lays out the final recommended model for predicting the functionality of the wells, the tool that is recommended for future modeling problems and an overview of the data visualization tool and dashboards used in the report. As requested, all research and technical material have been included within a separate reference document to support any organizations needing to validate statements made within the body of this work.

We look forward to having an opportunity to present this final deliverable to you. In the meantime, should you need any additional material for your Board of Directors, please do not hesitate to contact any team member.

Sincerely,

Team 2

Carlos Fuentes, Robert Herold, Chris Pelkey, and Susan Poole

Table of Contents

Problem Statement

Tanzania's wells are the lynchpin of a healthy and stable society. Without a reliable source of water, local food supplies collapse leading to civil unrest. The Ministry of Water is tasked with ensuring that the 59,400 wells spread across a landmass of 365,756sq miles reliably serve a population of 51.82 million people. With 88% of the water sector being dependent on foreign donors, there is pressure to limit well down time by optimizing well maintenance teams and deploying them to proactively maintain wells before they fail.

To see the link between water and food security in Africa, one need look no further than the 2011 drought in the Horn of Africa. – International Committee of the Red Cross.

Figure 1. Current State of Wells

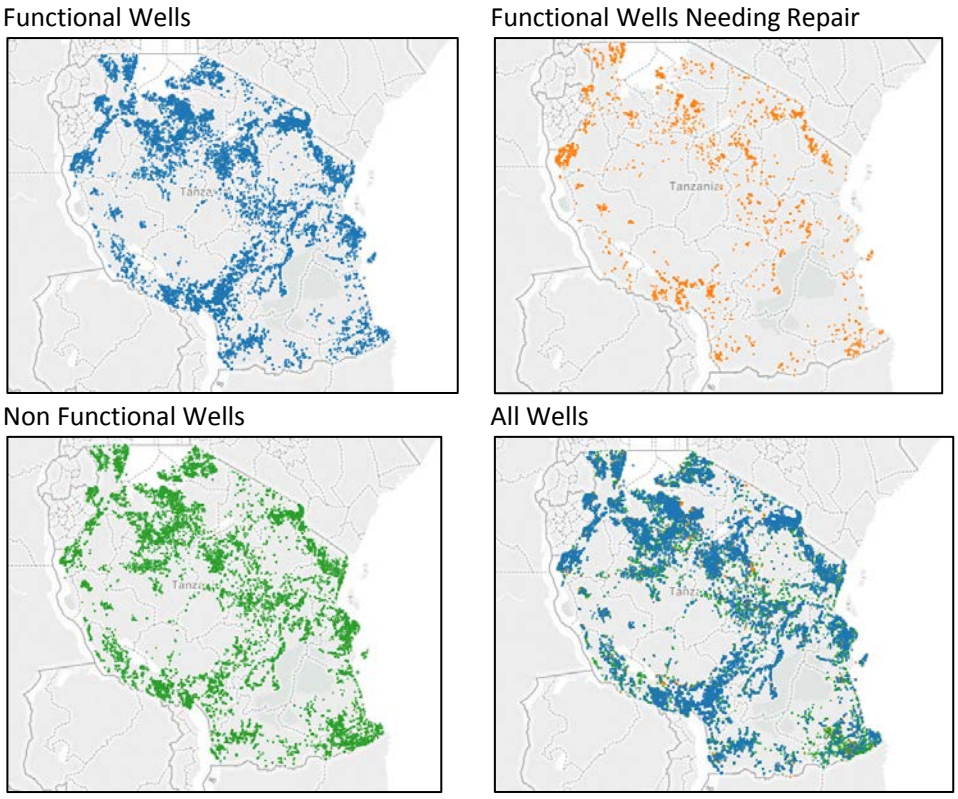
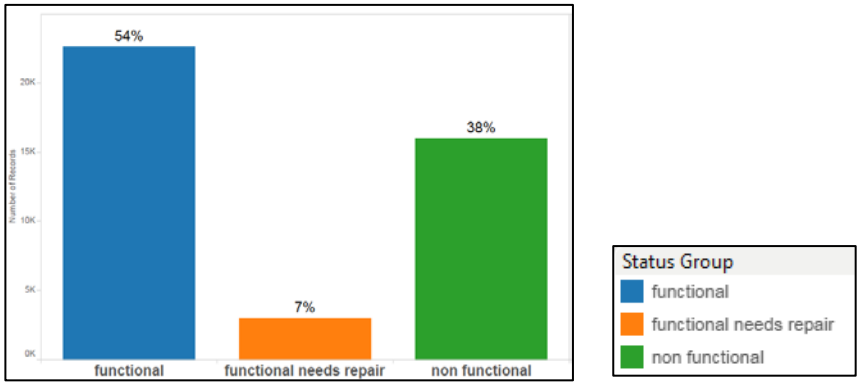


Figure 2. Wells by Functionality



Description of Data - Scope

The data to be used for this project is obtained from the *DrivenData* competition website *Pump it Up: Data Mining the Water Table* (https://www.drivendata.org/competitions/7/). Each of the 59,400 records has a unique identification number, 39 attributes to serve as predictor variables, and a response variable.

Figure 3. Description of Data

Data	Description
Data Files	2
Observations	59,400
Fields	40
Categorical Predictors	32
Numeric Predictors	6
Date Value predictor	1
Response Variable Predictor	1 with three levels

An in-depth analysis of the data is available in Pump It Up: Technical Analysis Supplement, Appendix A

Overview of the Data - Quality

A review of the data reveals quality challenges that must be addressed prior to model construction. We observe a material level of missing or erroneous data in multiple predictor variables. We also identify high cardinality (too many levels) in several fields, such that levels of categorical variables need to be reduced.

Missing data is more problematic than originally expected as a value of zero populates fields for many of the missing values. Best practice is to exclude variables with more than 5% of values missing, however, we attempt to impute values for all numeric fields except two – *Amount of Water* and *Number of Private Pumps*, which have 70% and 99% missing values, respectively. We convert *Amount of Water* into a categorical variable, and we exclude *Number of Private Pumps* entirely.

We exclude additional fields due to redundancy (*Quantity Group*, *Payment* Type), lack of differentiation between observations (*Recorded By*, *Ward*), errors (*Region Code*, *Scheme Name*), and extreme cardinality (*Subvillage*, *Waterpoint Name*).

Thus, 30 predictors are retained for analysis. With these predictors we impute missing data for numeric predictors and retain an *Unknown* category instead of imputing missing values for non-numeric variables based on comparisons of model performance.

The following sections provide additional detail about the predictors to be included in the analysis. Categorical variables are those which organize information by classes, and numeric variables are those that include actual

numbers as the observed value. We organize data in this fashion in order to apply appropriate methodologies. For example, we can apply mathematical operations to the numeric data, but not the categorical variables.

Categorical Variables

We retain 23 of the 32 categorical predictors from the original data. Each of these fields is reviewed below.

Funder - Who funded the well: high cardinality and some missing values.

Installer - Organization that installed the well: high cardinality and some missing values.

Basin - Geographic water basin: 9 classes; no missing data.

Region - Geographic location: 21 classes; no missing data.

District Code - Geographic location (coded): 20 levels; with some missing values.

Lga - Geographic location: 125 levels for Tanzanian districts. High cardinality; no missing data.

Public Meeting - True/False predictor with some missing values.

Scheme Management - Who operates the waterpoint (management); 12 levels with some missing values.

Permit - If the waterpoint is permitted: TRUE/FALSE predictor with some missing values.

Extraction Type - The kind of extraction the waterpoint uses. Three variables include this information with varying levels of granularity (7 – 18 levels); no missing data.

Management - How the waterpoint is managed. Two variables include this information with varying levels of granularity (5 or 12 levels); some observations are classified as unknown.

Payment - What the water costs; 7 levels; some observations are classified as unknown.

Water Quality - The quality of the water: Two variables include this information with varying levels of granularity (6 or 8 levels); some observations are classified as unknown.

Quantity - The quantity of water: 5 levels; some observations are classified as unknown.

Source - The source of the water: Two variables include this information with varying levels of granularity (3 or 10 levels); some observations are classified as unknown.

Waterpoint Type - The kind of waterpoint: Two variables include this information with varying levels of granularity (6 or 7 levels).

Numeric Variables

The numeric data carried forward for analysis includes seven predictors, which are reviewed below:

GPS Height - Altitude of the well: Values range from -90 to 2,770; with a material amount of missing values.

Longitude - GPS coordinate: Valid values range from 29.607 to 40.345; some missing values.

Latitude - GPS coordinate: Valid values range from -11.649 to -0.998; some missing values.

Population - Population around the well: Values range from 1 to 30,500; a material amount of missing values.

Amount of Water – Water volume at the specific pump: Values range from 0 (presumable missing) to 350,000. This predictor is carried forward as a categorical variable with seven levels.

Construction Year - Year the waterpoint was constructed: Values range from 190 to 2013; a material amount of missing values.

Recorded Year – Extracted from the date recorded: years include 2002, 2004, 2011, 2012.

Response Variable

The response variable is a categorical variable with three levels. Over half of the records indicate that the water pump is *functional*, with 38% identified as *non-functional* and 7% identified as *functional* needs repair.

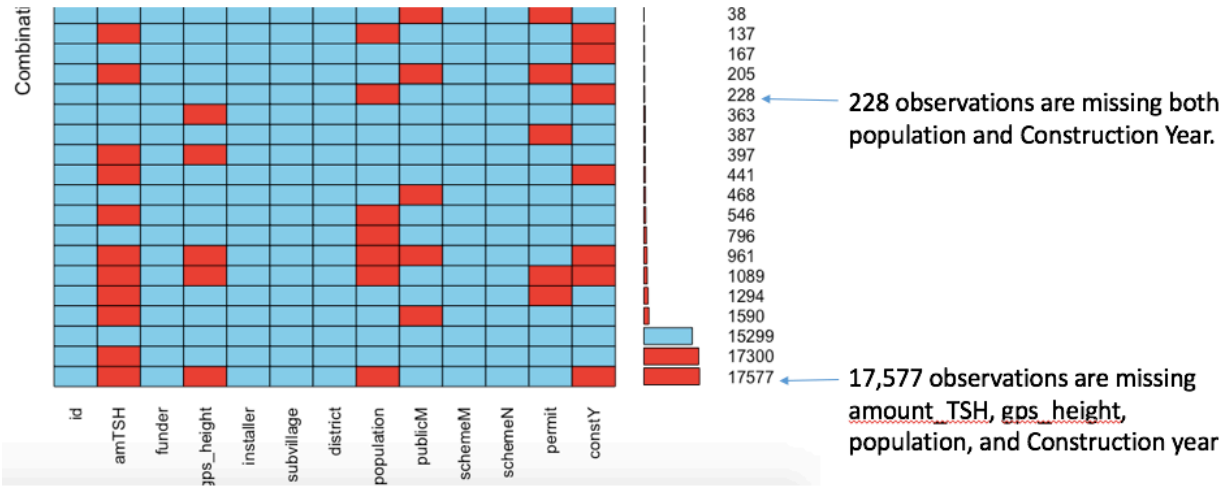
Figure 4. Response Variable Levels

Response Level	Count	%
functional	32,259	54%
functional needs repair	4,317	7%
non functional	22,824	38%
Grand Total	**59,400**	**100%**

Missing Data Patterns

As part of assessing the data, we examine combinations of missing data. The following chart provides a partial view into missing combinations. Understanding the nature of the patterns will influence the imputation method selected.

Figure 5. Missing Data Patterns



An in-depth analysis of the data is available in Pump It Up: Technical Analysis Supplement, Appendix B

Description of Data Transformation

Approach

Based on our data review, we identify the need to perform three types of transformations. First, we must perform imputation for missing data. Second, we need to address the issue of high cardinality for several of the predictor variables. Finally, we need to standardize the data prior to modeling.

Data Imputation

We review potential data imputation techniques and identify two viable options – MICE and missForest imputation.

- MICE (multivariate imputation by chained equations) is a methodology that can handle different data types at the same time (e.g. categorical and numeric) and can be used when multiple predictor variables have missing values. The MICE algorithm applies a regression strategy to estimate missing values for one predictor at a time based on values of the other predictors. This process is repeated until convergence is achieved.
- The R package "missForest" is also suitable for imputing mixed type data. This non-parametric approach uses a random forest based on actual observed values to predict the missing values. An advantage to this approach is that it can be performed using parallel processing to expedite processing time

Figure 6. Fields Considered for Imputation

Field	% Missing
Funder	7%
GPS Height	34%
Installer	7%
Longitude	3%
Latitude	3%
District Code	<1%
Population	36%
Public Meeting	6%
Scheme Management	7%
Permit	5%
Construction Year	35%

Performance

Figure 7. Performance of Data Imputation Techniques

Technique	R on PC	Microsoft Azure	Accuracy
MICE	Approximately 6 hours	36 seconds	
missForest (tree-based)	Approximately 6 hours	N/A	Slightly more accurate than MICE

To determine which imputation methodology to implement, we perform both the MICE and missForest imputations on the predictor variables, train a random forest model on each dataset, and evaluate results using a multivariate ROC[1] technique to obtain an AUC (area under the curve) metric. While results are similar, the dataset with missing variables predicted using missForest has slightly more favorable performance. As such, we continue our analysis with the dataset using the missForest imputations. Additionally, a comparison of model performance with imputation applied to all missing values versus only those of numeric predictors reveals that missing categorical values should not be imputed, but should be classified as *Unknown* instead.

High Cardinality

We noted in our data overview that a number of potential categorical predictors have a very large number of levels, referred to as high cardinality. With so many levels, some of these predictors cannot be included in potential models, which would result in losing any valuable information these predictors would hold. To address this challenge we apply a transformation to several of the predictors to combine levels with smaller numbers of observations. Specifically, we group levels with less than a specified percentage of the observations to achieve a suitable number of levels. Predictors transformed using this process include Funder, Installer, LGA, District Code, and Scheme Management.

Data Standardization

Once we address missing data and high cardinality, we also standardize the data, as some of the proposed modeling options are sensitive to scale. For numeric data we center the values by subtracting the sample mean, and then we scale the data by dividing by the standard deviation. For categorical predictors we use dummy coding, which creates an indicator for each level of each categorical variable. This adjustment is necessary for some machine learning techniques that cannot accommodate categorical variables without this transformation, such as neural networks and support vector machines.

An in-depth description of the data transformations is available in Pump It Up: Technical Analysis Supplement, Appendix C

[1] Standard ROC techniques were not appropriate they apply to response variables with only two levels.

Visualizing Data / Exploratory Data Analysis

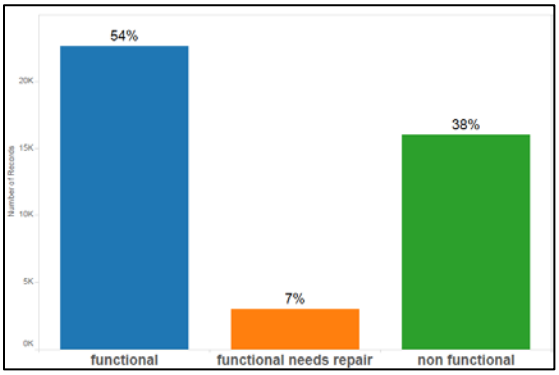
With the data quality check complete, we partition the data using a 70/30 split to create a training data set (41,581 records) for model construction and a test data to be used for model validation (17,819 records). To ensure reproducible results, a default seed value is selected for creating the split dataset.

For this report, the exploratory data analysis (EDA) is enhanced through the creation of an EDA dashboard using Tableau business intelligence software. The dashboard compares all predictor variables with the response variable and can be accessed and manipulated by all team members. This allows each team member to perform their own EDA and choose the variables they would like to analyze. All of the figures included in this section of the report are created using the dashboard.

Response Variable

We begin the exploratory data analysis with an assessment of the response variable *status group.* Within the training data set, 54% of the wells are considered functional, 7% functional needs repair and 38% non-functional (Figure 8).

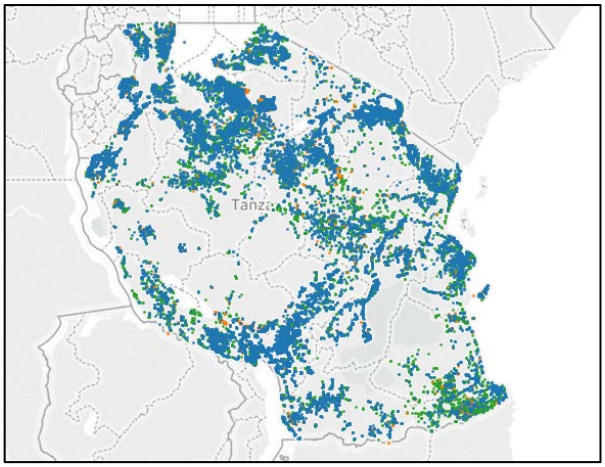
Figure 8. Status Group Proportions



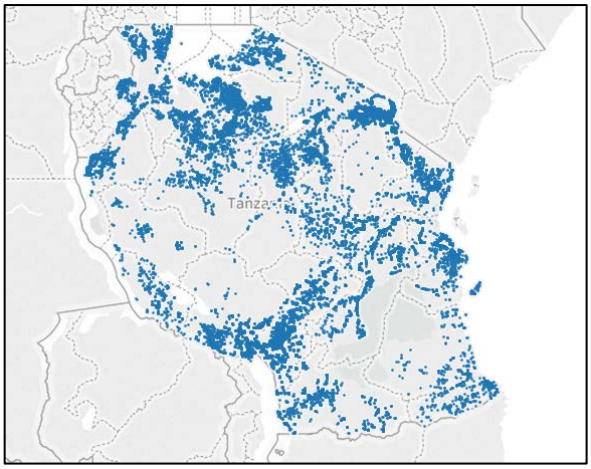
Understanding the geographic location of the wells is very important when it comes to efficiently deploying teams to fix the wells. To further the teams' understanding of the location of the wells as well as investigate the geographic distribution of the wells by *status group,* well maps are created (Figure 9). As seen in Figure 9, wells within each st*atus group* are distributed across the country.

Figure 9. Well Maps by Status Group

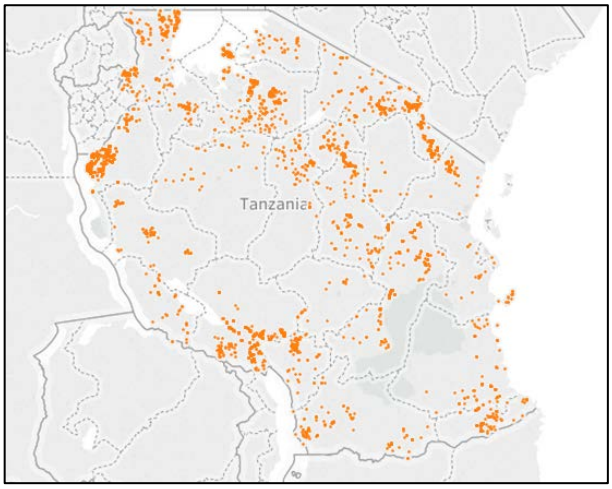
All Status Groups



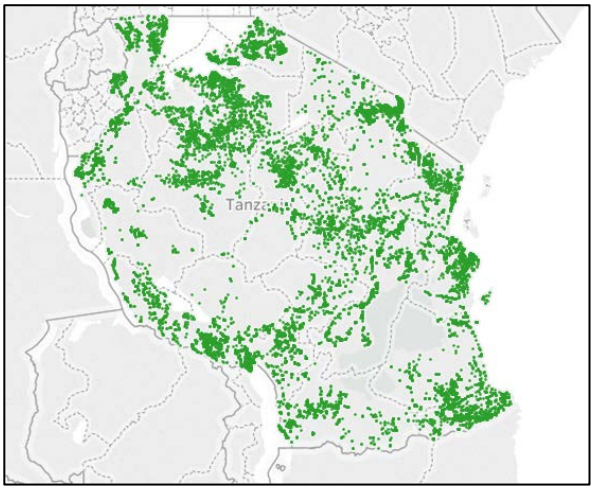
Functional Wells

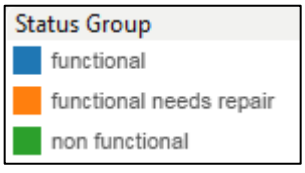


Functional Needs Repair Wells



Non Functional Wells





That dashboard containing the wells maps can be accessed via the link below.

https://public.tableau.com/profile/rob.herold#!/vizhome/PumpItUp-WellsMap/WellMap

Categorical Variables

The next step in the exploratory data analysis is an assessment of how select categorical predictors interact with the response variable *status group.* The dashboard with the categorical variables analysis can be accessed via the link below.

https://public.tableau.com/profile/rob.herold#!/vizhome/PumpItUp-InteractiveEDA/InteractiveEDA

The first predictor we review is *quantity group*, which classifies available water amount. We observe an intuitive pattern showing that the "dry" level has predominantly "non-functional" water pumps; the "enough" level has the greatest proportion of "functional" pumps (Figure 10). Based on this exhibit, the distribution of *status group* varies meaningfully by level of *quantity group*

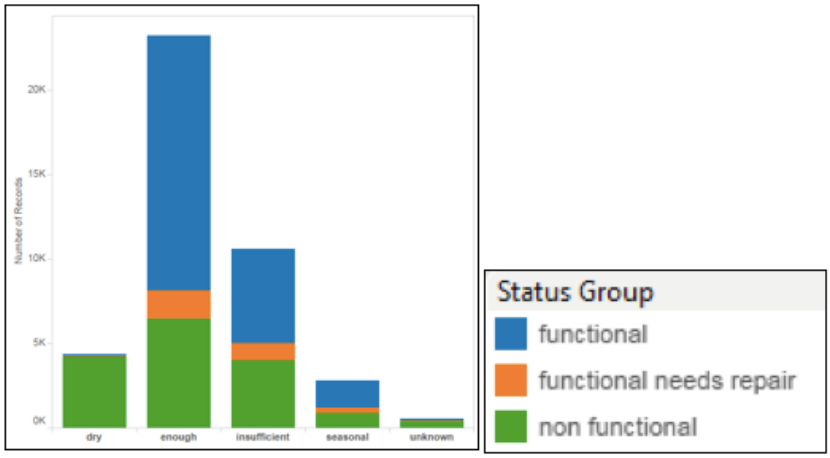


Figure 10. Status Group Proportions by Quality Group Level

The distribution of the predictor *quality group* demonstrates that the most common water quality is "good" followed by "salty." The distribution of water pumps by *status group* within these classes suggests that the greatest proportion of water pumps within the "good" class are "functional," while the majority of water pumps in the "unknown" class are "non-functional" (Figure 11).

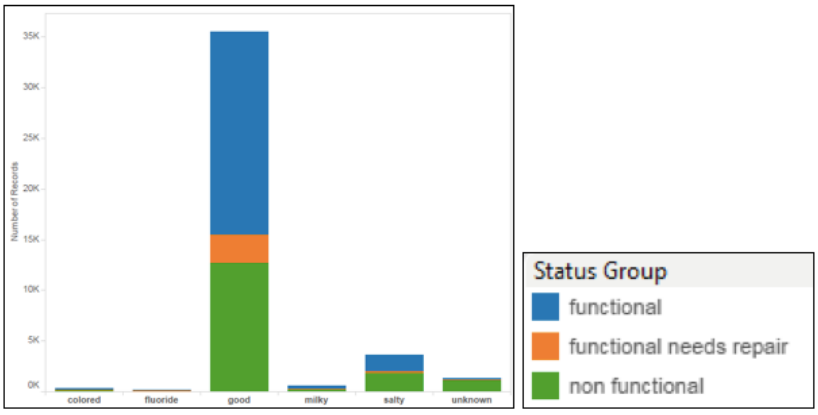


Figure 11. Status Group Proportions by Quality Group

An analysis of the predictor *waterpoint type group* reveals relatively similar proportions of water pump *status group* for “communal standpipe” and “hand pump,” while “other” types of water pumps are primarily “non-functional” (Figure 12).

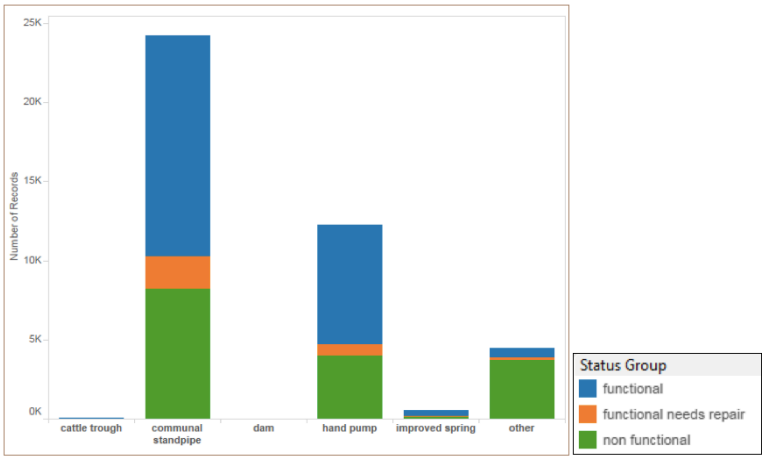


Figure 12. Status Group Proportions by Waterpoint Type Group

The functionality of water pumps appears to vary by *payment* class as well, with “never pay” and “unknown” having higher levels of “non-functional” pumps compared to the paying classes (Figure 13).

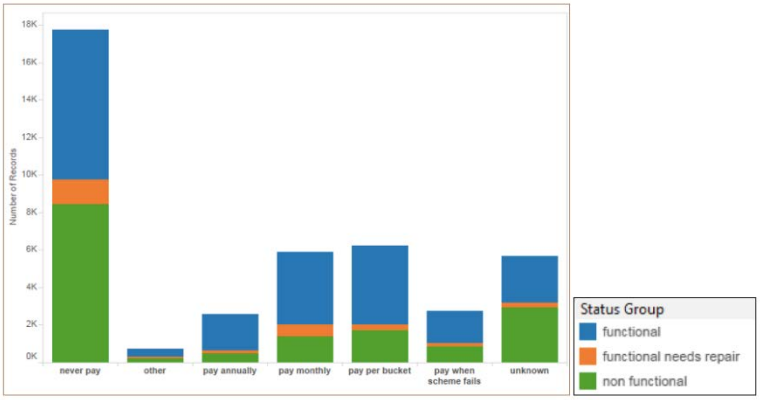


Figure 13. Status Group Proportions by Payment Class

Water pump status appears to be somewhat similar across the various geographical *basins*, however, there may be a tendency for Lake Victoria to have a higher portion of water pumps that need repair, while the Ruvama/Southern Coast basin appears to have a greater proportion of “non-functional” water pumps (Figure 14).

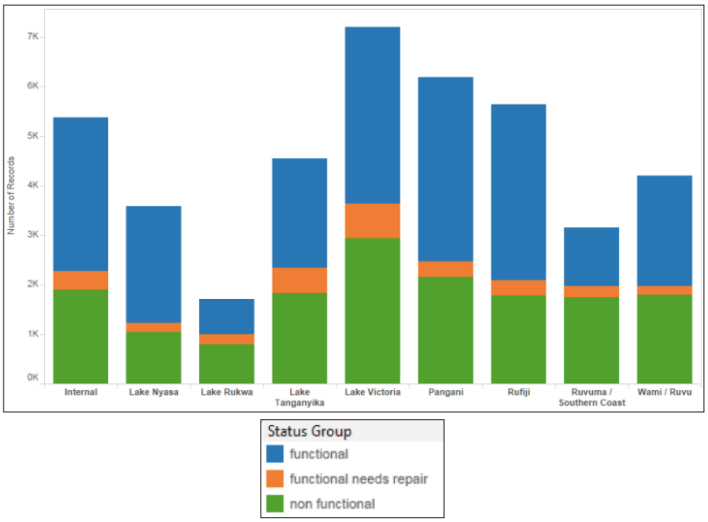


Figure 14. Status Group Proportions by Basin

Figure 15 shows the geographic location of each basin.

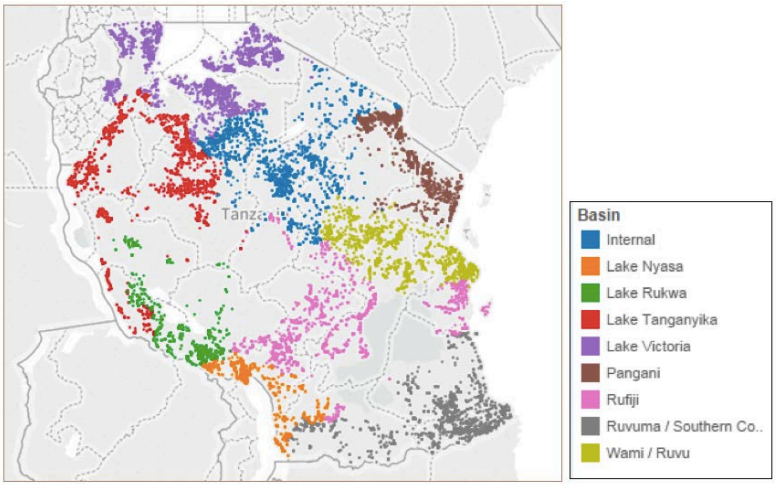


Figure 15. Status Group Proportions by Basin

A review of water pump status by *extraction type class* suggests that pumps that use "gravity" have a relatively higher portion of pumps that are in need of repair when compared with other types. The extraction type "other" has disproportionately more "non-functional" water pumps (Figure 16).

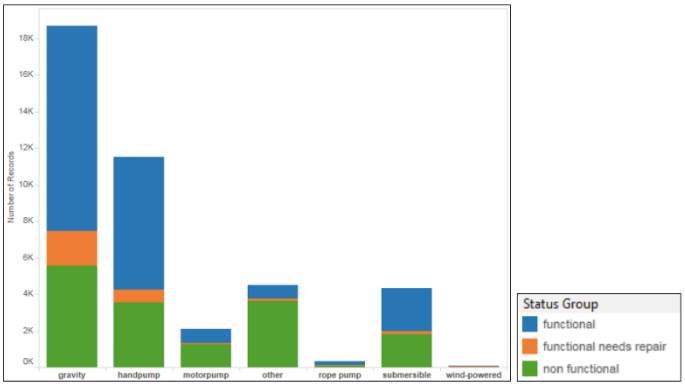


Figure 16. Status Group Proportions by Extraction Type Class

An assessment of water pump *status group* by *source type* suggests that a disproportionately large number of "borehole" and "shallow well" water pumps are "non-functional." The water pumps with the "spring" *source type* have a more favorable distribution, such that water pumps are more likely to be "functional" or "functional in need of repair" (Figure 17).

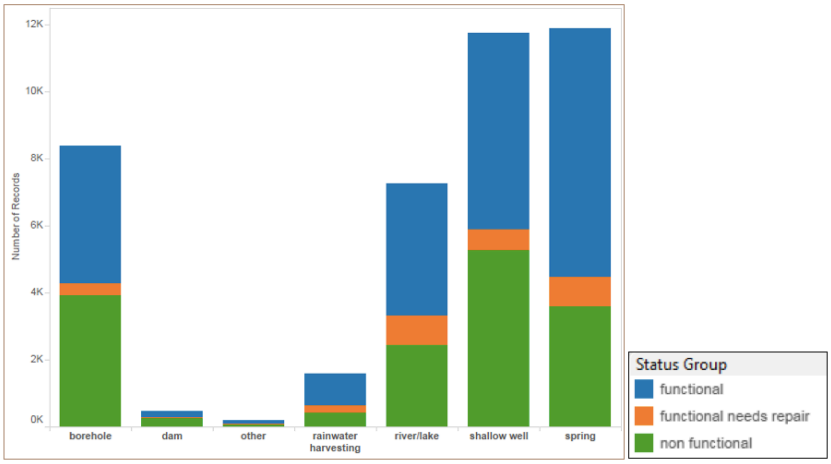


Figure 17. Status Group Proportions by Source Type

The distribution of *construction year* appears to vary by water pump *status group* such that “functional” water pumps are more likely to be more recently constructed, whereas “non functional” and ”functional needs repair” pumps are less likely to have been constructed after the year 2000 (Figure 18).

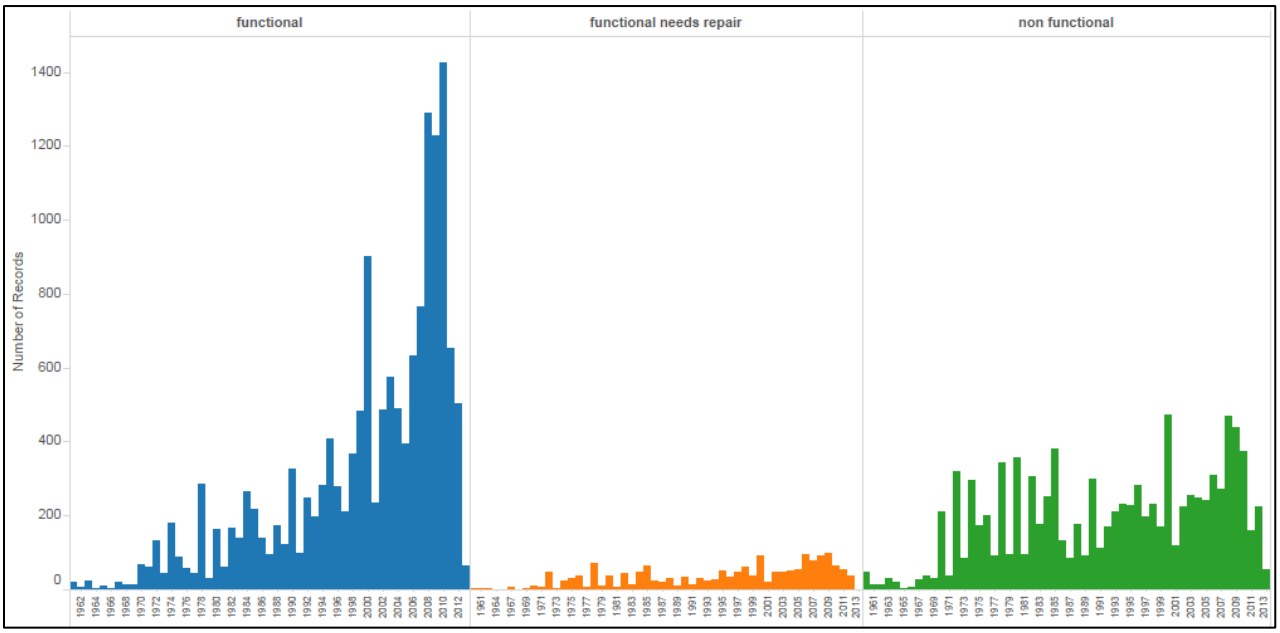


Figure 18. Construction Year Distribution by Status Group

The entire exploratory data analysis for this project is available in Pump It Up: Technical Analysis Supplement, Appendix D

Analysis of Data – Model Construction

With data preparation and exploratory data analysis complete we construct multiple models to identify which performs most favorably with regard to accuracy metrics (ROC area under the curve and prediction accuracy). We construct models with R, Azure software and ANGOSS software in order to compare performance. The following sections provide comparisons of results and descriptions of the models constructed.

R Models

Models constructed and evaluated with R software include Random Forest, Deep Learning Neural Network, Support Vector Machine, Gradient Boosting, Bagging, and Multinomial Regression. The table below displays the results for the most accurate version of each model type (Figure 19). Based on both the Multiclass ROC Area under the Curve metric and predictive accuracy for the test data, we identify the Random Forest as the most favorable modeling option.

Figure 19. R Model Performance

Model	Accuracy – Training Data	Accuracy – Test Data	Area under the Curve – Test Data
Random Forest	**0.8058**	**0.8080**	**0.7680**
Gradient Boosting	0.9131	0.8013	0.7621
Bagging	0.5431	0.6964	0.6474
Deep Learning Neural Network	0.7950	0.7688	0.7268
Support Vector Machine	0.8553	0.7893	0.7491
Multinomial Regression	0.7501	0.7477	0.6976

The following sections offer a brief description of each model, observations pertaining to results, and key performance metrics.

Random Forest

The machine learning technique, Random Forest, is a tree-based approach to modeling that uses random subsets of the predictor variables to generate decision trees. These trees are then combined to establish a complete model. We use R scripting in the open source H2O package to develop models and generate evaluation metrics.

The optimal Random Forest model constructed includes 19 of the predictor variables and indicates that *Quantity Group* is by far the most important predictor. The chart below displays each of the predictors included in the model, organized by most to least important (Figure 20).

Figure 20: Random Forest Variable importance Plot

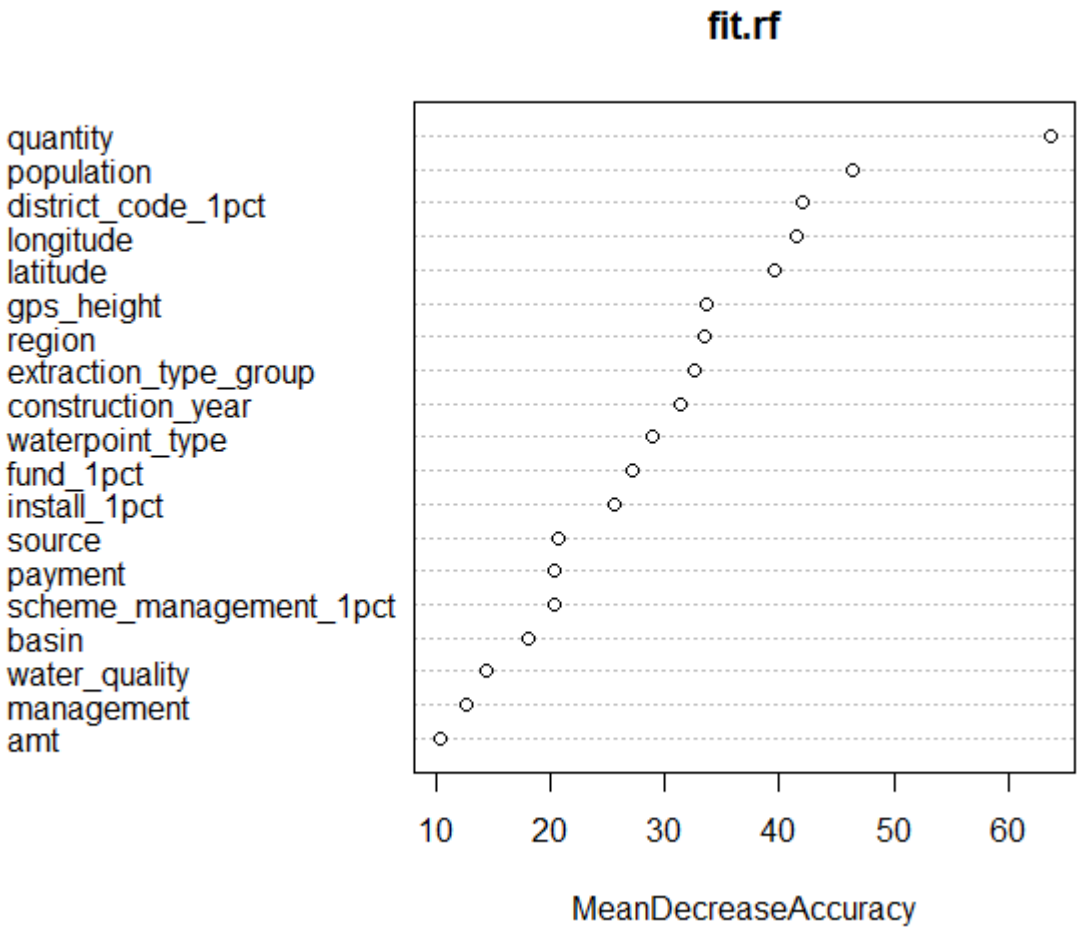


Figure 21. Random Forest Model Results

	Model 1	Model 2	**Model 3**
Accuracy – Training Data	0.7972	0.7967	**0.8058**
Accuracy – Test Data	0.8055	0.8057	**0.8080**
Area under the Curve – Test Data	0.7677	0.7665	**0.7680**

Gradient Boosting

We construct our next set of models using another type of tree-based approach called Gradient Boosting. While similar to the Random Forest approach, Gradient Boosting includes a process that tries to identify the most favorable combination of trees. As such, we expect to see more overfitting with the Gradient Boosting approach, which means that the model is more likely to reacting to noise in the training data. We use R scripting in the open source H2O package to develop models and generate evaluation metrics.

We run several versions of Gradient Boosting, modifying variables included in the models and customizing model features (also called model tuning) to achieve the most favorable results. In the table below, we see that Model 3 performs best on the test data, however, the more accurate results for the training data are consistent with our expectation that we might observe overfitting.

Figure 22. Gradient Boosting Model Results

	Model 1	Model 2	**Model 3**
Accuracy – Training Data	0.8509	0.8492	**0.9131**
Accuracy – Test Data	0.7973	0.7974	**0.8013**
Area under the Curve – Test Data	0.7385	0.7391	**0.7621**

Bagging

The third tree-based method we attempt is Bagging (a bootstrap aggregation method.) This method is thought to be useful in improving the stability of predictions. We use the R package 'adaboost' to develop models and generate evaluation metrics for this technique. We view results as inferior relative to other modeling options and limit results displayed to two models.

Figure 23. Bagging Model Results

	Model 1	**Model 2**
Accuracy – Training Data	0.5431	**0.5431**
Accuracy – Test Data	0.6481	**0.6964**
Area under the Curve – Test Data	0.5926	**0.6474**

Deep Learning Neural Network

The Deep Learning Neural Network is another machine learning technique viewed as potentially valuable in solving a classification problem such as ours. This technique is intended to simulate decision-making the way the human brain processes data. We use R scripting in the open source H2O package to develop models and generate evaluation metrics.

Figure 24. Deep Learning Neural Network Model Results

	Model 1	Model 2	**Model 3**
Accuracy – Training Data	0.7754	0.7967	**0.7950**
Accuracy – Test Data	0.7599	0.7681	**0.7688**
Area under the Curve – Test Data	0.7189	0.7261	**0.7268**

Support Vector Machine

Another machine learning technique that is suitable for classification problems is the Support Vector Machine. This technique establishes boundaries to separate classes of the response variable (*Status* Group) most

effectively. We create an initial model and then test adjustments to specific model features (also known as model tuning). We use the R package 'e1071' to develop models and generate evaluation metrics.

Figure 25. Support Vector Machine Model Results

	Model 1	**Model 2**
Accuracy – Training Data	0.8175	**0.8553**
Accuracy – Test Data	0.7826	**0.7893**
Area under the Curve – Test Data	0.7311	**0.7491**

Multinomial Regression

Our final modeling approach is the Multinomial Regression, a statistically-based approach solving a multi-class classification problem such as ours. We try multiple versions of this approach as well, but revising variables included in the model and adjusting model features does not yield meaningful improvements in results.

Figure 26. Multinomial Regression Model Results

	Model 1	Model 2	**Model 3**
Accuracy – Training Data	0.7497	0.7491	**0.7501**
Accuracy – Test Data	0.7478	0.7475	**0.7477**
Area under the Curve – Test Data	0.6980	0.6976	**0.6976**

Azure Models

Microsoft's cloud solution, Azure, includes powerful machine learning capabilities. It uses some of the best-in-class algorithms through a drag and drop interface, allowing data scientists to deploy machine learning models in hours instead of days. As with most new products, there are some limitations. At the present time, it only has four multi-class classification models. Microsoft offers a workaround to this by supporting emended R and the many available R libraries. The advantage to native Azure libraries is that they are optimized for Azure, and support parallelism (using many CPUs at the same time to complete the work faster).

Two of Azure's distinguishing features are

• **Tuning of Hyper Models** – Azure can run many iterations of its built in machine learning model, automatically tuning the model, selecting optimal parameters. It cannot do this with ones used from R.

• **Model Evaluation** – Azure can compare the output of two different models, giving the option of several comparison criteria depending on the types of models selected (AUC, F-Score, Comparison, etc.). It is limited to comparing two machine learning models at a time.

Comparing Azures 4 built in Multiclass models

Four of Azure's built in multi-class models (Figure 27) were run using the same training and test data as the other experiments in R. The Multiclass Decision forest has the best outcome, with the Multiclass Neural network coming in third place. The Multiclass Neural Network has the potential to come in first, but would require running many more automated iterations and more CPU run time, driving up costs. This is a case where greater accuracy costs more (in Azure billing). This isn't an issue with small data sets, like the Well Data. It does become an issue with very large complex data sets.

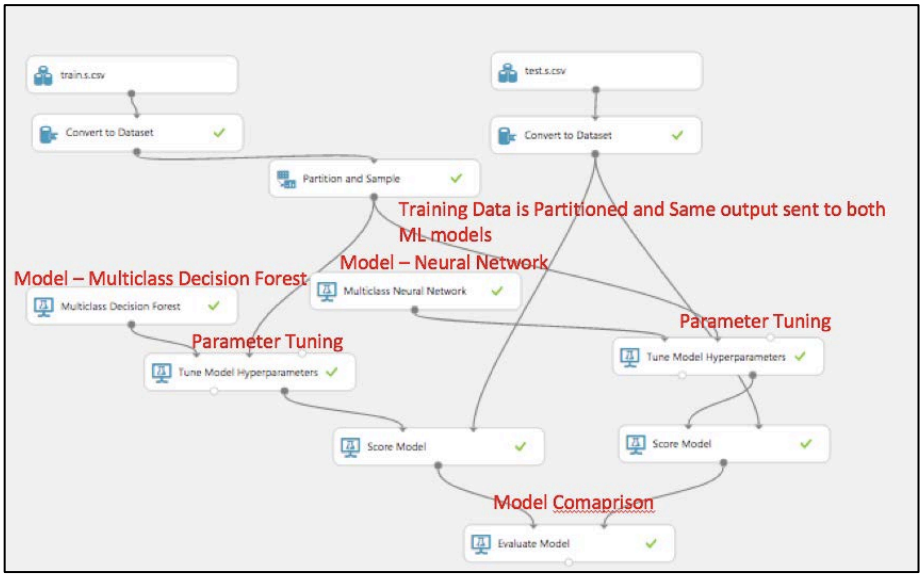
Figure 27. Comparison of Azure multi-class models

Model	Run Time for Selecting Best Parameters (minutes) based on 5 automatic passes	Evaluation - Overall accuracy	Model benefits	Model Cons
Multiclass Decision Forest	1:26	0.791122	Fast	
Multiclass Decision Jungle	11:00	0.780122	Less Memory	Requires More CPU
Multiclass Neural Network	4.12	0.779056	Greater accuracy is possible with many more interations	requires more CPU due to number of hyperparamaters and introduction of custom network topologies.
Multiclass Logistic Regression	0.28	0.743813	Speed	May sacrifice accuracy, depending on what you are trying to solve. In csome cases, it may be perform better, so iut should never be dismissed.

Automated Model Evaluation in Azure

Building and comparing models in Azure is relatively simple and quick compared to building code by hand in R. The figure below shows advantage of the machine learning hyper parameter tuning and model comparison Azure using two different models (Multiclass Decision Forest and Multiclass Neural Network) while taking (Figure 28). As noted earlier, the Azure Evaluate Model function only allows the comparison of two models. To compare all four models, the losing model is deleted and replaced by another model. Azure only requires re-running the new model, and caches the results from previous ML model runs.

Figure 28. Model Comparison in Azure



Automated Hyper Parameter Tuning Lessons Learned

Hyper Parameters are like virtual dials on a machine. You can keep adjusting them, rerunning the algorithm to try and get better outcomes. Not all models will materially benefit from automated hyper parameter tuning, where Azure randomly selects combinations of hyper parameter values, selecting the best outcome for the attempted "dial settings". When choosing the number of attempts, the randomly selected values does not always perform better than the default values.

Two methods are attempted:

- **Random Grid**: A set number of random settings are attempted (in this case, 50 or 200)
- **Entire Grid**: Each and every combination is tested

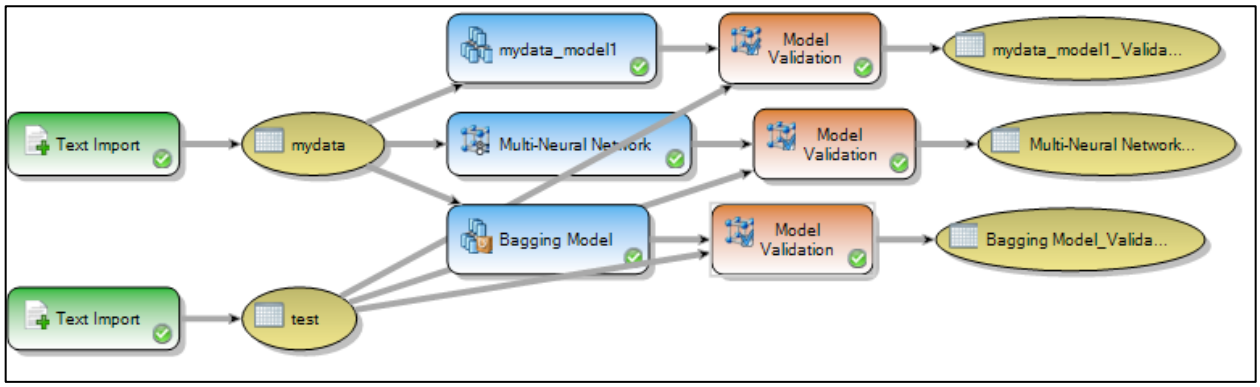
As you can see, the time to run many permutations can be quite lengthy, increasing cloud computing CPU chargeback fees. Frequently, there are diminishing returns when increasing permutations, as seen through the incremental improvement and increased Azure utilization feeds. To mirror what would be expected in a cost sensitive production environment, CPU utilization within the cloud was limited to 1 hour (Figure 29).

Figure 29. Model Comparison in Azure – Automatic Hyper-Parameter Tuning

Model	5 Iterations for Automatic Hyper-Paramater Tuning		50 Iterations for Automatic Hyper-		200 Iterations for Automatic Hyper-Paramater Tuning		Entire Grid	
	Run Time (Minutes)	Evaluation - Overall accuracy	Run Time (Minutes)	Evaluation - Overall accuracy	Run Time (Minutes)	Evaluation - Overall accuracy	Run Time (minutes)	Evaluation - Overall accuracy
Multiclass Decision Forest	1:26	0.791122	4:57	0.796341 ▲	19:47	0.798979 ▲	26:56:00	0.80156 ▲
Multiclass Decision Jungle	11:00	0.780122	>60min*	N/A	>60min*	N/A	>60min*	N/A
Multiclass Neural Network -Deep Neural networks (DNN) - Deep Learning	4:12	0.779056	12.32	0.780908▲	47:44:00	0.779898 ▼	7:57	0.781413 ▲
Multiclass Logistic Regression	0:28	0.743813	3:31	0.743252 ▼	18:39	0.742915 ▼	4:21	0.743869 ▲
*All Experiments limited to a maximum of 1 hour of CPU time								

ANGOSS Models

Figure 30. Model building in ANGOSS



In addition to R and Azure, our team also evaluated ANGOSS software. Utilizing ANGOSS allows the team to quickly come up with R code to input into the system. There are certainly downfalls to using this system, as the team is required to work within the parameters set forth by ANGOSS. Three test models were run in ANGOSS to see how they compare to the overall models that were run in R. Below you can see test data outcomes for a Random Forest, Bagging model, and a Deep Learning Neural Network model.

Figure 31. ANGOSS model comparison

Model	Accuracy –ANGOSS	AUC –ANGOSS	Accuracy – R	AUC –R
Random Forest	**0.7553**	**0.6994**	**0.8080**	**0.7680**
Bagging	0.7468	0.6846	0.6964	0.6474
Deep Learning Neural Network	0.736	0.6796	0.7688	0.7268

Looking at Figure 31, it is clear that ANGOSS falls short of R when it came to accuracy. R affords the freedom to tweak models as seen fit, while the ANGOSS models are not as adaptable. With the three models constructed, we can follow the same trajectory of Random Forest doing the best, followed by Bagging and concluding with Deep Learning Neural Network. You can see that while the Bagging technique performs worse than the Random Forest approach in ANGOSS, the ANGOSS Bagging technique outperforms the Bagging technique in R.

The R code that was easily exported from ANGOSS does provide base code to perform more sophisticated modeling, however, there are many packages available in R to help form the appropriate models without the need for a jump start on code from ANGOSS. Furthermore, using the H20 package in R provides additional efficiency for several of the model types, making it slightly easier to test multiple model features and less cumbersome to extract the output.

Conclusions and Recommendations

This portion of the report includes the final results of our analysis. It lays out the final recommended model for predicting the functionality of the wells, the tool that is recommended for future modeling problems and an overview of the data visualization tool and dashboards used in the report.

Predictive Model Conclusion and Recommendation

Predictive Model Recommendation: Random Forest Model created using R

For this project a total of 13 different model and modeling tool combinations were evaluated. Six different types of models were constructed using R, four different types of models were constructed using Azure and 3 different types of models were constructed using ANGOSS. All of the models were evaluated based on the Multiclass ROC Area under the Curve metric. Based on this metric, we identify the Random Forest constructed using R as the most favorable modeling option for this data set.

The optimal Random Forest model constructed includes 19 of the predictor variables and indicates that *Quantity Group* is by far the most important predictor (Figure 20).

Modeling Tools Conclusion and Recommendation

Modeling Tool Recommendation: R

Recommending a predictive modeling approach requires that many factors be considered during the planning process. Economics, skill sets, timeliness and accuracy should all be considered, as well as what the data itself naturally lends itself to. Wall Street firms may want to focus on the greatest accuracy and speed, while government agencies, like the Water Ministry will want a more nuanced balance between economics and accuracy. It is largely up to the key stakeholders to determine which of these factors are the most important in creating a predictive model. For example, the team never even entertained the idea of using SAS for analysis due to its costly licensing.

The process that the team ran through shows that R is an economical and a very powerful tool. This program produced the best results for the modelling process that the team engaged in. There are downfalls however. It does require specialized skill sets if highly accurate results are required, and it also suffers from performance issues when imputing missing data. Imputing missing data can be done through an overnight process, with results being made available when the staff arrives at work in the morning. Being able to wait for an overnight batch process to impute data, can greatly reduce the cost of a solution.

When Azure's built in algorithms are used, accuracy is nearly as high as accuracy in R, dropping less than one percentage point (Best R model = 0.8080 , Best Azure model =0.8016). Azure outperforms R in processing speed, such that imputation of missing data is handled in minutes, contrasted with the overnight processing in R. A drawback to Azure is that it has a limited set of built in machine learning algorithms. This challenge can be mediated, however by embedding R script within Azure to match R's accuracy. The biggest drawback to Azure is its recurring cost model, which runs contrary to Tanzania's Water Ministry's desire to be cost efficient.

ANGOSS requires a license fee and recurring yearly maintenance fees. Like Azure, it has a limited set of machine learning algorithms, requiring the need to include R code to achieve greater accuracy. In our analysis

ANGOSS underperformed, especially in comparison to the models created using R. As such, we do not recommend using ANGOSS in the future, as it is costly and does not deliver top accuracy marks.

Figure 32. Modeling Tool Evaluation

Feature	R	Azure	ANGOSS
Economics	Free	Recurring costs	License & yearly maintenance fee
Skill Set	Specialized	Average	Average
Timeliness	Low	High	Moderate
Accuracy	High	Medium if using out of the box features, high if using specialized resources, writing custom extensions in R	Medium if using out of the box features, high if using specialized resources, writing custom extensions in R

When considering these approaches, the team currently recommends that the ministry leverage R because of the superior accuracy and absence of recurring costs. All products require specialized R skills to achieve optimal results. Azure should be reconsidered in 18-24 months as it begins to support more optimized built-in machine learning algorithms. As machine learning algorithms become more commoditized in Azure, the ministry should be able to use Azure to solve many business problems in a timely and accurate manner, without relying on highly specialized R skills sets.

Below is a chart that sums up the software packages that were used for data imputation and model building for this project. The recommendations on when to use these programs is laid out in the left column.

Figure 33. Software Package Summary

Business Need	Technology	Based on the Ministry's business model, needs, nature and scale of data
Imputing Missing Data		
Need to quickly impute data, where a high degree of accuracy is not required	**Azure**	**Azure's built in data imputation features, like MICE, can quickly impute missing data.**
Need a high degree of accuracy when immediacy is not required	**R**	**R has a wealth of pre-built libraries, like decision trees, which can be used to impute missing data with a higher degree of accuracy. However, it will take more time to develop (labor) and will require a much longer run time. Re-running the imputation methods with refreshed data can take several hours to a day.**
Model building		
Need reasonably accurate response within hours	**Azure**	**Reasonably accurate models may be developed in hours. The model may be quickly run with refreshed data in minutes.**
Need a high degree of accuracy when immediacy is not required	**R**	**Highly accurate models may be developed. Development time and run time can be lengthy. Run time will be longer than Azure due to Azure having greater processing power (hardware).**
Not recommended	**ANGOSS**	**While having greater functionality than Azure, it lacks the richness of R, limiting the developer to work within built in features. As with native R, re-executing imputation or re-running models with new data will take considerable time.**

Data Visualization Conclusion and Recommendation

Data Visualization Recommendation: Tableau

The team also evaluated and recommends the Tableau visualization tool. With over 59,000 wells, it is very important to be able to visualize well status based on geography or other factors, such as elevation, water table, age of well, etc. Through the use of Tableau the team was able to create several working dashboards that allow the government of Tanzania to accurately see where working wells are, as well as those that are non-functional or in need of repair.

Dashboard Overview

- **Exploratory Well Map** – Used to understand the geographic location of the wells as it relates to the functionality. Used to investigate the geographic distribution of the wells by status group.
- **Interactive EDA** - Enhances EDA by allowing each team member to perform their own EDA and choose the variables they would like to analyze. The dashboard compares all predictor variables with the response variable and can be accessed and manipulated by all team members.
- **Random Forest Model Evaluation Map** – Used to evaluate the recommended predictive model based on geography in order to identify areas that had higher or lower than average prediction success.
- **Well Maintenance Team Deployment Map** – This map is part of the final project deliverable. It can be used by the Tanzanian government to identify the geographic location of wells that are functional but are in need of repair and the non functional wells in order to coordinate well maintenance teams.

Links to the data visualizations and descriptions of the dashboards are available in Pump It Up: Technical Analysis Supplement, Appendix E

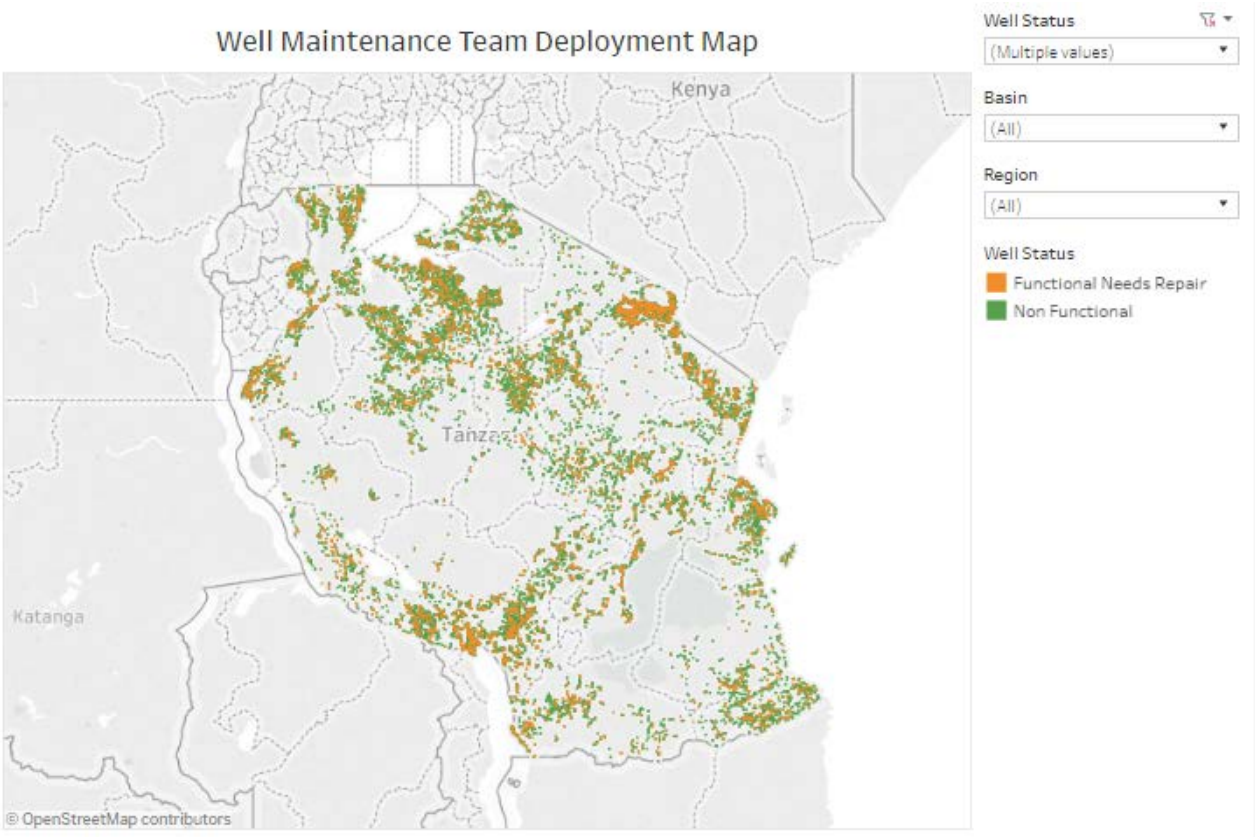
Recommendations for Next Steps

Deployment Strategy

While the final predictive model achieves nearly 81% accuracy in the classification of well status, there is still uncertainty in determining where to deploy maintenance teams. With the current model results we group the “non-functional” and “functional needs repair” wells and determine that we achieve 92% accuracy when the decision to send a crew to a well is binary (i.e. the well is functioning versus needing some form of maintenance). As such, we recommend deploying crews to wells that are either classified as

"non-functional" or "functional needs repair" in order to proactively restore or maintain water supplies. A map of wells within these classifications is presented below in Figure 34.

Figure 34. Map of "Non Functional" and "Functional Needs Repair" Wells



Database Improvement

During the course of our analysis we encountered a material proportion of missing data. We recommend that efforts are made to improve data quality in order to more effectively leverage information. First, we recommend that missing values be determined and populated when feasible. When it is not possible to obtain correct values, it would be advantageous to clearly denote which entries are unavailable rather than populating with zeros.

Predictive Model Refresh

As the recommended predictive model is derived from current data sources, we recommend a periodic review of the model to ensure that the algorithm used to predict well status is modified when appropriate. Furthermore, we recommend a review of the Azure software option in 18 – 24 months to determine whether it would be a more suitable option at that time.

Appendix

Supplemental Documents Referenced:

Pump It Up: Technical Analysis Supplement, October 30, 2016

Data Sources:

DrivenData competition website Pump it Up: Data Mining the Water Table (https://www.drivendata.org/competitions/7/