Pump it or Leave it?
A Water Resource Evaluation in Sub-Saharan Africa

Jacqueline Fortin Flefil, Marios Andreas Galanis, Vladimir Kozlow

December 13, 2018

1 Abstract

In Sub-Saharan Africa, an estimated 184 million people rely on hand pumps for their water supply[1]. The goal of this study is to develop an algorithm that can predict hand pump sustainability in low-income countries based on a minimum of data collected on the field. Predicting the sustainability of a hand pump at a given point in time can help shorten the time for NGOs to provide support and organize targeted maintenance operations in remote areas. Using the Taarifa dataset, we trained, compared and optimized different machine learning algorithms to predict three categorical features of the dataset that were identified as possible indicators of hand pump sustainability: functionality of the hand pump, quantity of water delivered, and quality of water delivered. Logistic Regression, Gaussian Discriminant Analysis, Support Vector Machine, Decision Trees and Neural Networks algorithms were tested on our dataset. We then optimized Logistic Regression, Random Forest, and Neural Networks, ultimately combining them with a Voting Ensemble Classifier. The Random Forest algorithm had the best performance when looking at F1 and MCC scores. However, the Voting Ensemble method yielded better distributed results across all classes.

2 Introduction

Our project is a tool for development agencies and governments to understand the state of water resource infrastructures in underdeveloped and vulnerable regions of the world. In 2015, an estimated 184 million people living in Sub-Saharan Africa relied on hand pumps for their water supply[1] and more than 300 million people lacked access to an improved water source[2]. Historically, development agencies have been supporting those populations by providing infrastructures, such as hand pumps, but very little attention had been directed to their sustainability and their maintenance [3],[4]. Functionality rate of hand pumps in selected Sub-Saharan countries was 36% in 2009, and is respectively 15% and 25% one year and two years after construction in 2016[5].

Our goal is to apply machine learning techniques to evaluate the sustainability of a water scheme using data that is already being collected by managing agencies. We look at different aspects of sustainability, including whether a water point is functional or not, the quantity of water it outputs, and its water quality. These predictions can shorten the time required for agencies to provide support and organize maintenance operations. Ideally, this project can inform the water sector and help improve the lives of those that rely on such hand pumps for daily tasks and basic human needs.

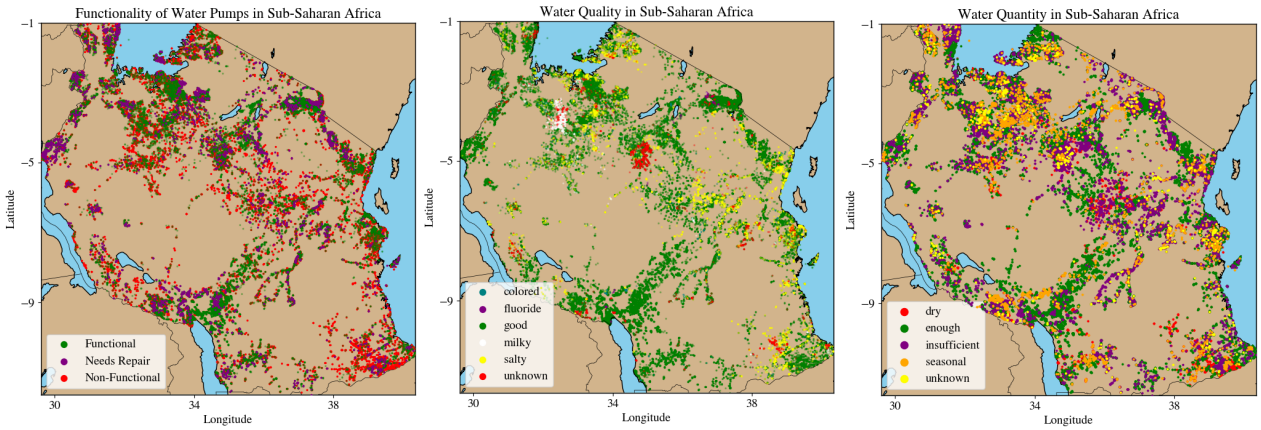
3 Related work

The Taarifa dataset that we used in this study, and variants of it, has been extensively explored in the field of access to Water Sanitation and Hygiene services in developing countries. Most of those studies however do not use machine learning methods to analyze the dataset. A recent study from 2017[6] used a Bayesian network to analyze correlations in the data. Some groups have used Machine Learning methods but did so for other purposes: a study from 2013[7] used STATA to perform Multivariate Logistic Regression but only looked at relationships between features and non-functionality of the hand pump.

4 Dataset and Features

4.1 Dataset

The dataset used in this study was collected by the Tanzania Ministry of Water, aggregated by Taarifa, and downloaded through Kaggle.com. Figure 1 shows the features that will be classified. This dataset contains data for 59,400 hand pumps, each with 40 features. Some of the features are binary/categorical, and some numerical. These include the location of the water pump, water source type, date of construction, the population it serves, and whether there were public meetings for the point.



(a) (b) (c)

Figure 1: Maps of Tanzania's hand pump (a) functionality, (b) water quality, and (c) quantity.

4.2 Dataset Split

To train and test our algorithms, we initially split the dataset by randomly assigning 25% of it to the test set and 75% of it to the training set. However, by doing so we became aware that there was a class imbalance problem, where some of the classes for all three classification problems had a relatively small number of points (less than 10% of the total for functionality which has three classes). We therefore decided to apply the synthetic minority over-sampling technique (SMOTE), as described in Chawla et al. (2002)[8], on the training set to ensure our algorithms would have enough training data from each class.

4.3 Data Processing

We first performed a feature screening and decided to use only 24 of the 40 features. Our screening process excluded 16 features for the following reasons:

- Irrelevance: some of the features were deemed irrelevant to our project and we decided to exclude them to reduce the computational cost of our algorithm.
- Redundancy: some of the categorical features had exact or almost exact duplicates and we decided to only keep one out of the two or three identical features. In these cases, we kept the most granular feature. In particular, this reduced the number of geographical features.

We then transformed most of the remaining categorical features into binary variables through a One Hot Encoding (OHE) process. Finally, we imputed values where data was missing, and replaced those data points with the mean (continuous) or mode (categorical/binary) of the feature that was missing. This allowed us to keep over 24,000 data points that were missing at least one feature.

5 Methods

We tested the following algorithms on each of the three classification problems tackled, and performed them both for the original train/test split (with class imbalance), and the SMOTE resampled split. Ultimately, all models were trained on the set that was resampled from 75% of the original data, and tested on the remaining 25%. We tested Logistic Regression (LR), Gaussian Discriminant Analysis (GDA), Support Vector Machine (SVM), Decision Trees (DT) and Neural Networks (NN) algorithms. Based on preliminary results, both in terms of computational time and accuracy of the results, we decided to only optimize the LR, DT and NN algorithms. Models were optimized using grid search cross-validation (5-fold) to fine tune hyperparameters and final results were obtained using 5-fold cross validation on the test set. Algorithms were evaluated and optimized based on the micro F1 score and on the Matthews Correlation Coefficient (MCC) because of the class imbalance of our test set. The voting ensemble method was used to optimize our final results. LR, DT, NN and VC algorithms optimization is described below:

- **Logistic Regression (LR)**: Logistic regression was chosen because it is a robust learning algorithm that makes few, and usually reasonable, assumptions about the data. The penalty factor and the type of regularization were optimized. Best performance for this algorithm on all three classification tasks was achieved for an L2 regularization with penalty factor of 1.0.
- **Decision Trees (DT)**: A decision tree model seemed a particularly promising idea given the number of features used in our algorithm, especially after the OHE process. Having many features, none of which have

an obvious effect on the output alone, means that the causal relationship between the features and the output might come from different combination of the features that cannot be modelled well by algorithms that rely on assumptions about data distributions. We tried the Random Forest (RF), AdaBoost, and Bagging and after tuning the parameters of each algorithm, RF performed the best.

- **Neural Networks (NN)**: Our last algorithm was a NN because of the NN's ability to generalize and to respond to unexpected patterns. During training, different neurons are taught to recognize various independent patterns and then the combination of all the neurons manages to capture all those different patterns and combine them in one final output node. Since our dataset likely contained unexpected and difficult interactions, this was a promising choice.
- **Voting Ensemble Classifier (VC)**: None of the methods discussed above provided exceptional accuracy for all classes, so the last optimization step was to combine the best three methods (RF, NN, LR) into one ensemble method, the Voting Classifier, which provided great results. We mostly used "hard" voting (i.e. majority class), but for water quantity we used "soft" voting, which averages output probabilities from the three input models.

The parameters involved in our final models are as follows:

Neural Network	# Layers/Neurons	Activation Function
Functionality	130, 5	Sigmoid
Water Quality	120, 30	Sigmoid
Water Quantity	60, 15	Sigmoid

(a)

Random Forest	# Estimators	Max. Depth
Functionality	70	25
Water Quality	75	60
Water Quantity	45	45

(b)

Figure 2: Optimized parameters for (a) RF and (b) NN models per output task

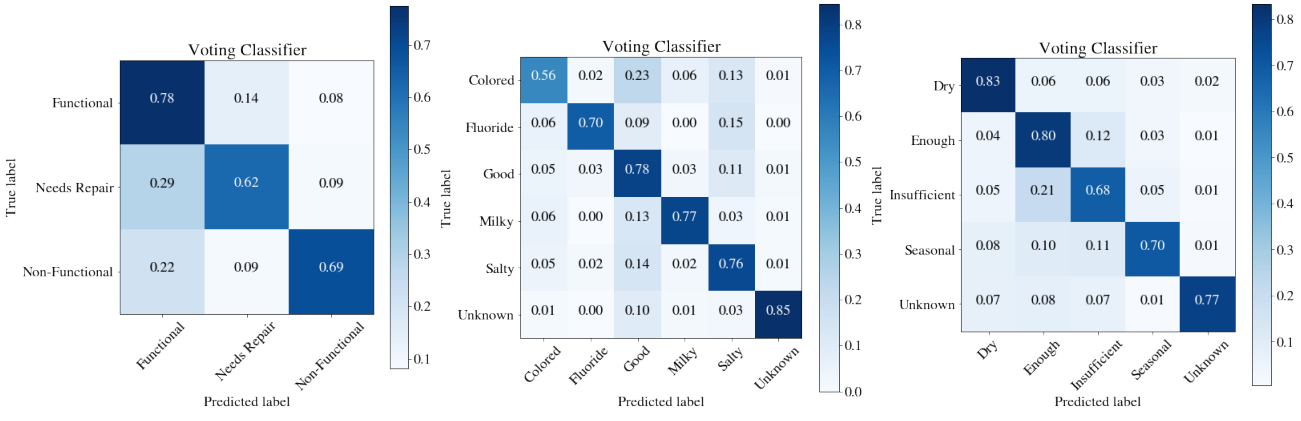
6 Results and Discussion

All three of our classification algorithms provided us with probabilities of the samples being classified into one of the classes. Since we are dealing with multiclass classification outputs, our algorithms assign each point to the class that holds the highest probability. In order to evaluate our models' performances, we produced confusion matrices to compare predicted and true values. Since the class imbalance was preventing our algorithms from learning the characteristics of less represented classes well enough to predict them, we found the SMOTE resampled dataset to produce better results than the regular dataset split. The results are presented in three forms: confusion matrices which give an idea of how the accuracy of the predictions is distributed between classes, and an evaluation of the prediction through two scores that balance precision and recall. Precision and recall are defined by:

$$precision = \frac{\sum TruePositives}{\sum TruePositives + \sum FalsePositives} \tag{1}$$

$$recall = \frac{\sum TruePositives}{\sum TruePositives + \sum FalseNegatives} \tag{2}$$

Results of the three Voting Classifiers are presented in Figure 3.



(a) (b) (c)

Figure 3: Classification algorithm results for Voting Classifiers.

In order to evaluate our predictions taking into account the imbalance of our test set (unlike the training set, it was not modified by the SMOTE method), we decided to use the micro F1 score instead of the macro F1 score. The micro F1 score is the harmonic mean of the precision and recall of all examples. It thus takes into account equally the prediction obtained for each example, disregarding classes. The macro F1 score on the contrary, is the harmonic mean of the F1 score of each class and thus weights each class (but not each example) equally.

	F1 score (micro)					
Prediction	**Functionality**		**Quantity**		**Quality**	
Set used	Train	Test	Train	Test	Train	Test
LR	65.2%	64.3%	68.6%	57.0%	77.0%	59.8%
RF	86.2%	76.8%	91.9%	78.9%	97.9%	87.4%
NN	74.0%	70.4%	79.6%	66.4%	91.0%	73.3%
Voting	76.6%	73.5%	93.7%	77.0%	93.3%	78.3%

Figure 4: Micro-averaged F1 Scores for train and test datasets

While the micro F1 score is a good measure of the overall accuracy of our predictions, it is not a good evaluation metric for the less represented classes. To deal with this problem, we decided to use the Matthews Correlation Coefficient (MCC) as defined below for the multiclass case:

$$MCC = \frac{\sum_{i,l,m=1}^{k} c_{ii}c_{ml} - c_{li}c_{im}}{\sqrt{\sum_{k=1}^{n}(\sum_{k=1}^{n} c_{lk})(\sum_{\substack{f,g=1 \\ f\neq k}}^{k} c_{gf})}\sqrt{\sum_{i=1}^{k}(\sum_{i=1}^{k} c_{il})(\sum_{\substack{f,g=1 \\ f\neq k}}^{k} c_{fg})}} \tag{3}$$

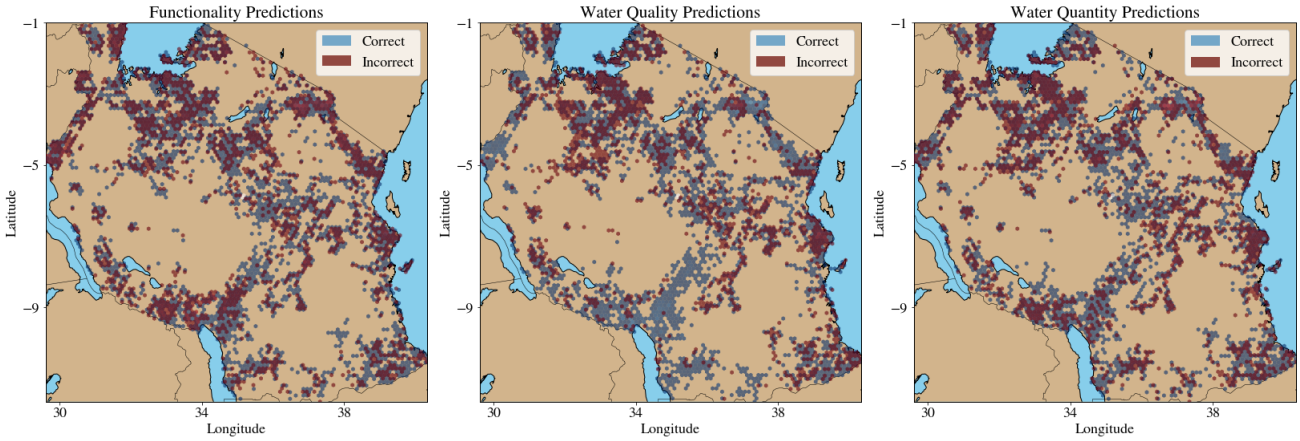
Because it takes into account the ratios of the four confusion matrix categories (true/false positives, true/false negatives) the MCC is a good evaluation metric for imbalanced datasets.

	MCC					
Prediction	**Functionality**		**Quantity**		**Quality**	
Set used	Train	Test	Train	Test	Train	Test
LR	0.476	0.441	0.611	0.392	0.731	0.348
RF	0.774	0.589	0.978	0.696	0.973	0.578
NN	0.577	0.473	0.742	0.499	0.888	0.428
Voting	0.664	0.537	0.920	0.618	0.932	0.497

Figure 5: MCC for all algorithms

Upon evaluating the MCC for all our algorithms, we found a similar pattern to the micro-averaged F1 score in that RF always had the highest scores, followed by the voting classifier, as shown in in Figure 5. Overall, the micro F1 score provided a good evaluation of the tested algorithms in terms of number of good predictions but the MCC is a

better evaluation metric for the distribution of those predictions across classes. Our results showed that the Voting Ensemble Classifier consistently yielded the highest results for *all* classes, which we considered more valuable than excellent performance in some classes but poor performance in others. Therefore, the Voting Ensemble Classifier was our chosen final algorithm for all three output predictions. For all cases, we were happy to verify that our algorithm performed significantly better than a random classifier (corresponding to a MCC of 0).



(a) (b) (c)

Figure 6: Maps of correct and incorrect classifications for (a) Functionality, (b) Water Quality, and (c) Water Quantity.

Figure 6 shows the spatial distribution of correct (blue) and incorrect (red) predictions for the three outputs. For functionality and water quantity, there does not seem to be a clear relationship between geographic location and accuracy, however it seems that qualitatively the water points closer to bodies of water have a higher probability of being incorrect. For water quality, there is a more clear relationship in that the water points in the Northwest and East Coast seem to be misclassified more often, whereas points in the middle of Tanzania and further Southwest seem to be better classified.

7 Future work

There is definitely a lot to be done to keep improving the performance of these classification tasks. One possible future step would be to keep optimizing our models using specific strategies: trying different activation functions for each layer in the NN, performing feature selection for RF, testing more kernels for SVM, or using a nonlinear combination of the features for LR as these might increase the accuracy of our predictions. Because of time constraints, we were only able to search a grid of parameters that were reasonably but arbitrarily chosen for each algorithm, so given more time we would likely implement a more consistent and formulaic way of choosing a grid to search for optimal parameters.

Furthermore, it would be useful to explore the input features some more and see which ones contribute the most to a successful prediction. A summary of how much variance is explained by each feature would help decide which measurements are crucial for future work.

Another path would be to look at differences in predictions between countries or geographic regions to test the robustness of our algorithm. We also think there is value in going deeper into the results and seeing where (other than geographically) the algorithm fails, i.e. see if there are similar characteristics for the examples for which we are predicting incorrectly.

Finally, adapting the model to predict *when* a pump will fail would make it a more applicable tool for managing agencies.

8 Team Member Contributions

Strategy development: Team effort.
Data processing: Team effort.
Prediction of functionality: Jacqueline Fortin Flefil.
Prediction of the quality of water: Marios Andreas Galanis.
Prediction of the quantity of water: Vladimir Kozlow.
Debugging: Team effort.
Final report: Team effort.

9 Project Code

The project code can be found on `https://github.com/jackieff/cs229project`

10 References

[1]MacArthur, J. (2015) hand pump Standardisation in Sub-Saharan Africa: Seeking a Champion. RWSN Publication 2015-1 , RWSN , St Gallen, Switzerland

[2] World Health Organization, WHO/UNICEF Joint Water Supply, & Sanitation Monitoring Programme. (2015). "Progress on sanitation and drinking water: 2015 update and MDG assessment". World Health Organization.

[3] Carter, R. C., Ross, I. (2016). "Beyond 'functionality' of hand pump-supplied rural water services in developing countries". Waterlines, 35(1), 94-110.

[4] Fisher, M. B., Shields, K. F., Chan, T. U., Christenson, E., Cronk, R. D., Leker, H., Samani, D., Apoya, P., Lutz, A., ... Bartram, J. (2015). "Understanding hand pump sustainability: Determinants of rural water source functionality in the Greater Afram Plains region of Ghana". Water resources research, 51(10), 8431-8449.

[5] Banks, Brian Furey, Sean. (2016). What's Working, Where, and for How Long: A 2016 Water Point Update. 10.13140/RG.2.2.31354.49601.

[6] Cronk, R., Bartram, J. (2017). Factors influencing water system functionality in Nigeria and Tanzania: a regression and Bayesian network analysis. Environmental science technology, 51(19), 11336-11345.

[7] Foster, T. (2013). Predictors of Sustainability for Community-Managed hand pumps in Sub-Saharan Africa: Evidence from Liberia, Sierra Leone, and Uganda. Environmental Science Technology, 47(21), 12037–12046. https://doi.org/10.1021/es402086n

[8] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer. (2002). "SMOTE: synthetic minority over-sampling technique". Journal of artificial intelligence research, 321-357.

[9] Pedregosa, Fabian, et al. (2011) "Scikit-learn: Machine learning in Python." Journal of machine learning research, 2825-2830.

[10] John D. Hunter. (2007). "Matplotlib: A 2D Graphics Environment". Computing in Science Engineering, 9, 90-95

[11] Travis E, Oliphant. (2006). "A guide to NumPy", USA: Trelgol Publishing.

[12] Wes McKinney.(2010). "Data Structures for Statistical Computing in Python" Proceedings of the 9th Python in Science Conference, 51-56.