Pump it Up: Data Mining the Water Table

Nisha Rangnani, Nishitha Maniganahalli, Jaishree Palaniswamy

School of Information Studies, Syracuse University
Syracuse, NY
{nrangnan, nmanigan, jpalanis}@syr.edu

Abstract — Using data from Tanzanian Ministry of Water and Taarifa, we aim to predict which pumps are functional, which need repair, and which don't work at all. An understanding of which waterpoints can fail would allow improved maintenance operations and ensure that clean water is available to communities across Tanzania[1]. We employed algorithms such as Decision Tree Classifier, Random Forest Classifier, Gradient Boost Machine, Support Vector Machine and XGBoost algorithm. Along with these, we also generated rules on performing association rule mining to find top rules that lead to functional and nonfunctional pumps.

I. Introduction

Today, we have easy access to all the commonly-held services and thus, we take them for granted. However, in order to serve us and deliver each basic service, a team needs to work continuously so that our lives run smoothly. Thus, we chose to work on real world data from Taarifa and aim to accurately predict functionality of pumps for better planning and maintenance of these pumps and thus, clean water supply[4]. Taarifa is an open source platform for crowdsourced reporting and triage of infrastructure related issues. It helps to engage citizens with their local government[2].

In this paper, we aim to discuss implementing various algorithms to classify whether a pump is functional, needs some maintenance or is not functional at all. This prediction would be done by considering various factors, such as what kind of pump is operating, when it was installed, and how it is managed.

In this paper, we would be discussing the kind of data we have, how we cleaned and transformed this data, several algorithms that were implemented for classification and performance of each algorithm implemented. We'll be concluding the paper by comparing various models and interpreting their results.

I. Data cleaning & transformation

Our waterpoints dataset consisted of 59,400 rows and 41 columns. There are various attributes in the dataset which would help us classify whether the pump is functional or not. Some major attributes contributing to the classification are construction year of the pump, altitude of the well, the population utilizing the well, extraction type, quantity and quality of water[3].

As part of data preprocessing, few repetitive or unimportant columns were dropped. For example, water quality and water quality group attributes had redundant data and thus, the water quality column was dropped. The column 'funder' has several funders and as part of data transformation only top 15 funders were considered, and the rest of them were labelled as 'other'. Similar approach was taken for installers. We also checked for duplicates in the dataset and removed them.

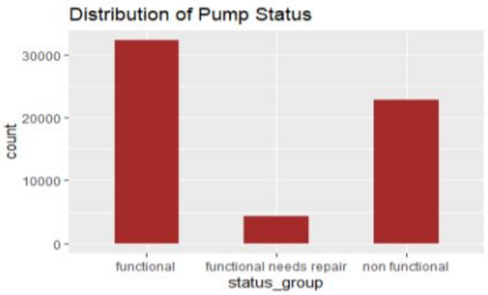
We had two columns namely construction year and date recorded, construction year tells the year when the pump was built and date recorded gave the date when the reading for that waterpoint was taken. These two columns were used to determine the age of the pump, year was extracted from the date recorded column and construction year was subtracted from it to identify age of the pump.

We also had several N/As in our dataset. We had N/As for attributes like population and height of pump, to deal with them, we replaced it with the median of the population subset of its respective region. There were cases when the data subset extracted had all N/As, in that case, we replaced those N/As with the overall median of the population attribute.

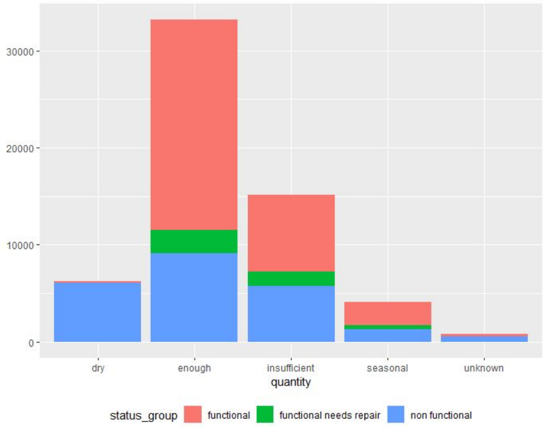
We also performed data discretization to prepare data for association rule mining. We discretized height of the pump, age and population into low, medium and high according to descriptive statistics of the overall data. Finally, data was scaled and normalized before implementing machine learning algorithms to train it.

II. Exploratory Data Analysis

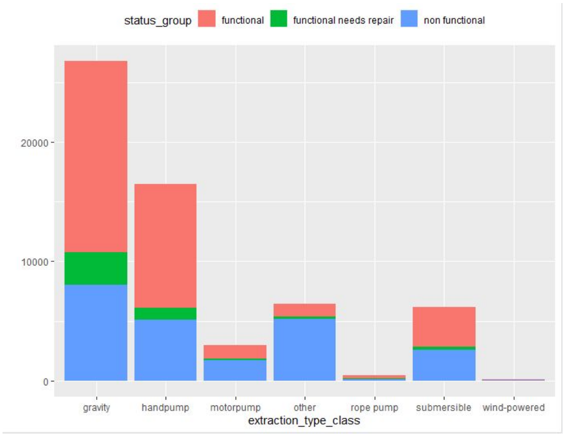
Most of our pumps were functional. Out of 59,400 pumps, around 31,000 pumps were functional, 5,000 were functional but needed repair and about 23,000 were non-functional.



On exploring the quantity of water being supplied, we can deduct that when water quantity is dry, almost all of the time the pump is non-functional. When water quantity was enough, most of the time the pump was functional.



For extraction type class, we can observe that pumps are usually functional when extraction is using hand pump and gravity. Also, most pumps are non-functional when extracted using motor pumps and other ways.



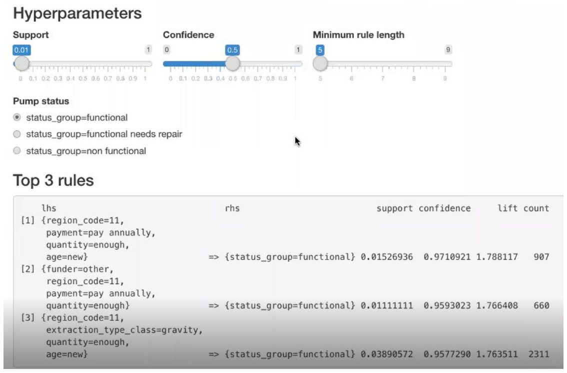
III. Model building and tuning

We implemented a list of machine learning algorithms namely Association Rule Mining, Decision Tree Learning, Random Forest Classifier, Gradient Boosting Machine and eXtreme Gradient Boosting algorithm. We applied these algorithms to Tanzanian waterpoints dataset and classified whether a pump would be functioning, not functioning or needs repair.

A. Association Rule Mining

We used the 'arules' package to implement apriori algorithm where our rhs was the combination of various values taken by the class variable - status group (functional, functional needs repair, non functional). We specified our initial hyperparameter values as support being 0.01, confidence level as 0.5 and minimum rule length being 5. The rules obtained were sorted in decreasing order of lift.

On running the apriori algorithm, we obtained following rules that lead to functional pumps: {region_code=11, payment=pay annually, quantity=enough, age=new}=>{status_group=functional}, {funder = other, region_code = 11, payment = pay annually, quantity=enough} => {status_group=functional}, {region_code = 11, extraction type class = gravity, quantity = enough, age = new} => {status_group=functional}.



Hyperparameters

Support 0.01

Confidence 0.5

Minimum rule length 5

Pump status

● status_group=functional

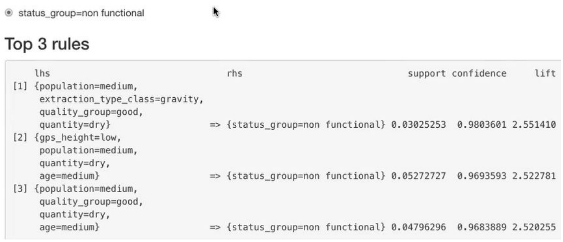
○ status_group=functional needs repair

○ status_group=non functional

Top 3 rules

```
    lhs                                 rhs                         support confidence     lift count
[1] {region_code=11,
     payment=pay annually,
     quantity=enough,
     age=new}                        => {status_group=functional} 0.01526936  0.9710921 1.788117   907
[2] {funder=other,
     region_code=11,
     payment=pay annually,
     quantity=enough}                => {status_group=functional} 0.01111111  0.9593023 1.766408   660
[3] {region_code=11,
     extraction_type_class=gravity,
     quantity=enough,
     age=new}                        => {status_group=functional} 0.03890572  0.9577290 1.763511  2311
```

Similarly, we obtained following top rules for non functional pumps: {population = medium, extraction type class = gravity, quality group = good, quantity = dry} => {status_group=non functional}, {gps height = low, population = medium, quantity = dry, age = medium} => {status_group=non functional} and {population = medium, quality group = good, quantity = dry, age = medium} => {status_group=non functional}.



● status_group=non functional

Top 3 rules

```
    lhs                                 rhs                             support confidence     lift
[1] {population=medium,
     extraction_type_class=gravity,
     quality_group=good,
     quantity=dry}                   => {status_group=non functional} 0.03025253  0.9803601 2.551410
[2] {gps_height=low,
     population=medium,
     quantity=dry,
     age=medium}                     => {status_group=non functional} 0.05272727  0.9693593 2.522781
[3] {population=medium,
     quality_group=good,
     quantity=dry,
     age=medium}                     => {status_group=non functional} 0.04796296  0.9683889 2.520255
```

These results make sense because we can clearly notice that when water quantity is enough, the age of pump is in early years (new); the pump is found to be in functional state. Whereas, when age is medium and water quantity is dry and when population is not low (good number of people are using the pump) then pumps tend to be non-functional.

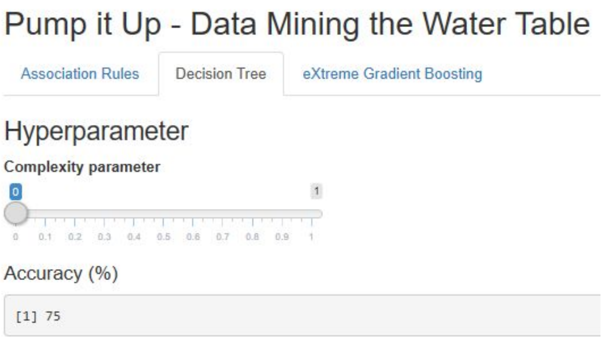
B. Decision Tree Learning

As part of data preprocessing, the dataset was split into two parts - training and testing set. 80% of the data was considered for training, while remaining 20% of the data was used for testing.

We used caret package to build our decision tree. It is a greedy algorithm and is faster as compared to other algorithms. We began with the default complexity parameter as 0.01 and increased it in an increment of 0.01. For training purposes, we used five-fold cross validation using train control and the expand grid for hyperparameter tuning.

The model provided the best result when the complexity parameter was zero. On predicting the pump functionality status for the testing dataset, we got an accuracy of 75.52%.

In our R-Shiny web application, we created the decision tree tab to specify model parameter value for Complexity parameter. On changing this parameter, the algorithm runs in the background and displays the model accuracy based on the value specified for the Complexity parameter.



Pump it Up - Data Mining the Water Table

Hyperparameter

Complexity parameter: 0

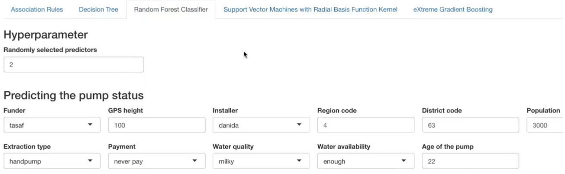
Accuracy (%)

```
[1] 75
```

C. Random Forest Classifier

The next algorithm we implemented was Random Forest. For this, we used the standardised data, having 63 features. We used three-fold cross validation for this model. We began with ten predictors and went up to forty five predictors for model training. The final number of predictors determined by the model were twenty four. This result makes sense as higher number of predictors would lead to overfitting and lower number of predictors would lead to underfitting.

After training the model, we checked the model performance against the validation data. The accuracy obtained for random forest classification algorithm is 77.69%.


Association Rules
Decision Tree
Random Forest Classifier
Support Vector Machines with Radial Basis Function Kernel
eXtreme Gradient Boosting

Hyperparameter

Randomly selected predictors: 2

Predicting the pump status

Funder: tasaf
GPS height: 100
Installer: danida
Region code: 4
District code: 63
Population: 3000
Extraction type: handpump
Payment: never pay
Water quality: milky
Water availability: enough
Age of the pump: 22

As you can see from the R-Shiny web application UI, the user could either tune the hyperparameter 'Randomly selected predictors' or they could enter values for different attributes to predict the pump status outcome. We decided to remove this tab from our final application, because Random Forest took about forty five minutes to run in the background and it does not make sense to make the user wait for so long.

D. Gradient Boost Machine

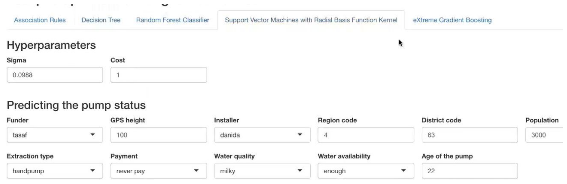
Post Random Forest, we implemented Gradient Boost Machine. We used caret package for its implementation, we also used five fold cross-validation. The final model obtained had one hundred fifty trees and an accuracy of 73.72%.

The accuracy obtained for Gradient Boost Machine was slightly lesser than other algorithms, probably because of the correlation among the data.

E. Support Vector Machine

Post Gradient Boost Machine, we implemented a non-linear support vector machine with radial bias function. We used the kernlab package to implement this. This model used a five-fold cross validation. For hyperparameter tuning purposes, we varied the penalty cost between zero and one, along with sigma values for errors.

Non-linear support vector machine is highly complex, yet it provided us with an accuracy of 74.37%.


Association Rules
Decision Tree
Random Forest Classifier
Support Vector Machines with Radial Basis Function Kernel
eXtreme Gradient Boosting

Hyperparameters

Sigma: 0.0988
Cost: 1

Predicting the pump status

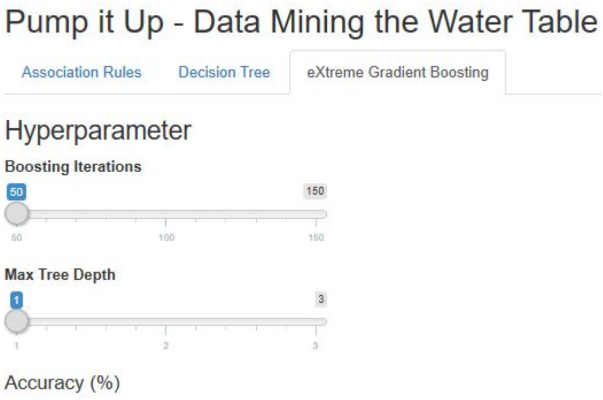
Funder	GPS height	Installer	Region code	District code	Population
tasaf	100	danida	4	63	3000

Extraction type	Payment	Water quality	Water availability	Age of the pump
handpump	never pay	milky	enough	22

From the above snapshot from our R-Shiny web application, we had created the SVM tab where the user could choose to alter hyperparameters Sigma and/or Cost to get the model accuracy. The user could also enter in values for several attributes to predict working status of the pump. We decided to remove this tab due to the amount of time it took to run the model, and it did not make sense to include an algorithm that made the user wait for about thirty minutes.

F. eXtreme Gradient Boosting

Finally, we ran extreme gradient boosting to predict our class variable pump status group. We used caret package to implement extreme gradient boost. We ran the model for several values of maximum number of iterations and maximum tree depth. The model provided maximum number of iterations as 150 and maximum tree depth as 3. On testing the model against the validation data, we received the accuracy of 75.16%.



Pump it Up - Data Mining the Water Table

Hyperparameter

Boosting Iterations: 50 – 150

Max Tree Depth: 1 – 3

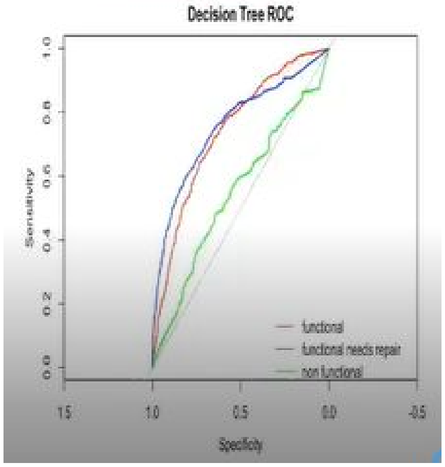
Accuracy (%)

If you zoom in to the R-Shiny web application snapshot for eXtreme Gradient Boosting tab, the user can choose to alter values for following hyperparameters - Boosting iterations and max tree depth. The algorithm takes some time to run and it displays the accuracy according to the hyperparameters specified.

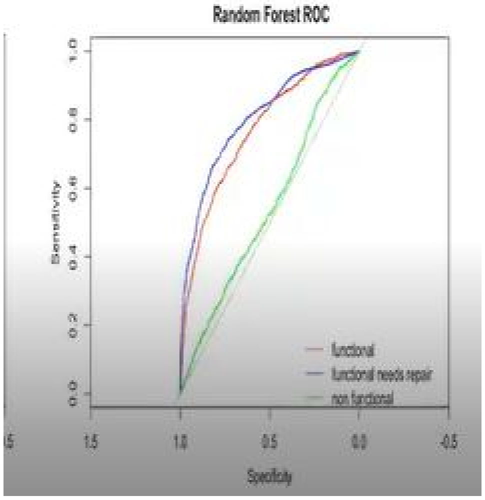
Model Performance Evaluation & Comparison

We implemented six different types of algorithms and we took the two best performing models - Decision Tree Learning and Random Forest Classifier and compared the model performance based on time complexity and accuracy determined by the Receiver Operating Characteristics curve.

Decision Tree

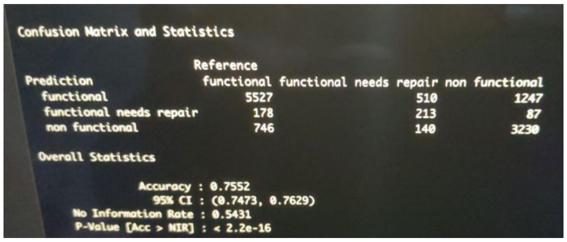


Random Forest



Below are the confusion matrices produced for the top two performing algorithms, decision tree and random forest respectively.

Decision tree correctly classified 8,970 records and misclassified 2,908 records out of a total of 11,878 test records.



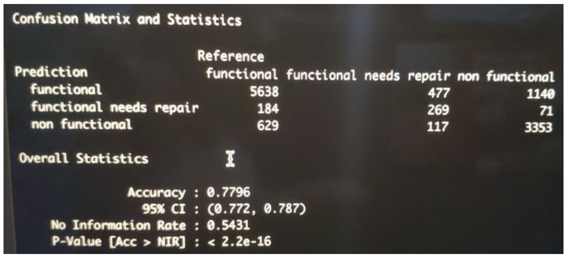
```
Confusion Matrix and Statistics

                          Reference
Prediction                functional functional needs repair non functional
  functional                    5527                     510           1247
  functional needs repair        178                     213             87
  non functional                 746                     140           3230

Overall Statistics

               Accuracy : 0.7552
                 95% CI : (0.7473, 0.7629)
    No Information Rate : 0.5431
    P-Value [Acc > NIR] : < 2.2e-16
```

Random Forest correctly classified 9,260 records and misclassified 2,618 records out of a total of 11,878 test records.



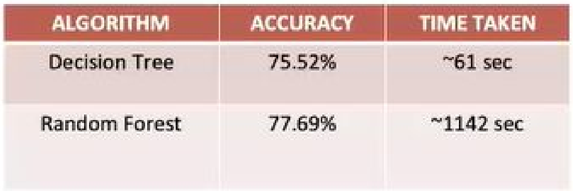
```
Confusion Matrix and Statistics

                          Reference
Prediction                functional functional needs repair non functional
  functional                    5638                     477           1140
  functional needs repair        184                     269             71
  non functional                 629                     117           3353

Overall Statistics

               Accuracy : 0.7796
                 95% CI : (0.772, 0.787)
    No Information Rate : 0.5431
    P-Value [Acc > NIR] : < 2.2e-16
```

Thus, to compare:



ALGORITHM	ACCURACY	TIME TAKEN
Decision Tree	75.52%	~61 sec
Random Forest	77.69%	~1142 sec

Random Forest Classifier gave the highest accuracy of 77.69% but took up a lot of time to run, about 1142 seconds. Whereas, Decision tree had an accuracy of 75.52%, but it was 20 times faster than Random Forest. So, finally choosing which model to go ahead with depends on the user and their requirement. It is a trade-off between time complexity and higher accuracy.

IV.R-Shiny Web Application

We developed an R-Shiny web application for our users to view various machine learning algorithms, tune their hyperparameters and view the obtained accuracies.

The web application could be accessed through following link:
https://nishithavenkatesh.shinyapps.io/pumpitup/

We have only included Association Rules, Decision Tree and eXtreme Gradient Boosting in our final application because other algorithms like Random Forest and Support Vector Machine were taking very long to run and the end user had to wait for over thirty minutes because of the size of the dataset.

Acknowledgement

We would like to thank Prof. Ying Lin for introducing us to various data mining and machine learning techniques. You not only made us familiar with real world applications from various domains, but we also got to learn about several challenges involved in these applications and how to deal with them. We got to explore various open-source state-of-the-art software packages and develop skills to deal with data problems from several domains.

References

https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/ : 1

https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/24/ : 2

https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/25/ : 3

https://taarifa.wordpress.com/ : 4