Pump it Up: Data Mining the Water Table

Antriksh Agarwal Sarvesh Pandit

Abstract

The task of this project is to predict which water pumps throughout Tanzania are functional, which of those need repairs, and which do not work at all. The prediction is based on a number of variables about what kind of pump is operating, when it was installed, and how is it managed. We are comparing the performance of a couple of boosted classifiers with Support Vector Machines (SVMs) and the performance of SVMs with different kernel functions. The selected boosted classifiers for this project will be XGBoost and LightGBM. These classifiers will be evaluated using accuracy as the evaluation criteria as given on the challenge website. The ultimate goal would be to find the best classifier and suggest improvements.

1 Introduction

Water is a fundamental need of the human beings, mainly for their survival. We need water for drinking, bathing, washing, cooking and a lot of other things. Water fulfils so many of our domestic needs and is also useful for so many of our industrial needs such as producing electricity, powering plants, irrigation of agricultural lands, etc.

There are a lot of countries in the world having minimal sources of water and even lesser resources to harvest it. Such countries require serious laws and regulation of water so that the people are not dieing due to shortage of water. One such country seems to be Tanzania where lots of pumps were set up by various firms and have now been non functional for some time. Tanzania, officially the United Republic of Tanzania, is a country in eastern Africa within the African Great Lakes region. Tanzania's population in 2016 has been estimated to be 55.6 million, composing of various ethnic, linguistic and religious groups. Tanzania's water supply can be characterized as decreasing access to improved water sources, intermittent water supplies and generally a low quality of service [1]. This project aims to identify the factors which lead to a water pump not being functional.

The project aims to identify problems with various factors that affect the functioning of water pumps, how to use these factors to an advantageous outcome and how to overcome the problems of these factors. This is to be done by gaining a smart understanding of the features because of which the water pumps installed by various organizations in Tanzania have stopped functioning properly. The training dataset

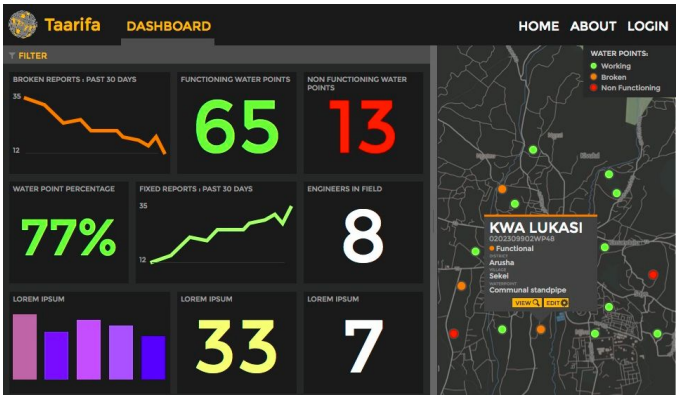

Fig. 1. Pump It Up: Data Mining the Water Table; Taarifa.com visualization of condition of water pumps. Source: Driven Data Competition [2]

divides each water pump into being functional, non functional or functional but needing repairs.

We are going to be using various supervised machine learning algorithms and evaluate them based on the data obtained from the Driven Data [3] competition Pump it Up: Data Mining the Water Table [2]. The first algorithm we would be working on is Support Vector Machines which have proven to be very useful in the supervised learning world transforming data from one dimension to another dimension where it is linearly separable. There are also various boosting algorithms like Random Forests, XGBoost, Gradient Boosted Trees etc. which perform very well on all sorts of classification problems. We would only be experimenting with XGBoost and LightGBM which are two tree boosting algorithms. The rest of the report explains these algorithms briefly and displays various distributions of data.

A smart understanding of these water pumps using the algorithms mentioned in this report, and other techniques which perform relevantly well will help us and the government of Tanzania improve the condition of water supply in the country.

2 Related Work

There has not been a lot of work trying to predict the functional or non functional water pumps using the data provided in the challenge, but a lot of work has been done using data mining techniques to identify related fields like future healthcare, bioinformatics and even pump and pipe failures.

There has been some work conducted on trying to predict failures of pipes by Tran and Ng [8] where they compare SVM with Back-Propagational Neural Network (BPNN) concluding that SVM outperforms BPNN. This suggested that we use SVM instead of experimenting with Neural Networks. A similar work on attempting to assess deterioration of stormwater pipes in Guelph, Ontario has been done by Harvey and McBean [9]. They use classification trees on a dataset gathering insight on the influence of construction year, diameter, length and slope on pipe condition.

Another research closely resembling the one this project aims to work on has been conducted previously by Arymurthy [10] where they are comparing the performance of Random Forests, XGBoost, Gradient Boost Machine (GBM) and SVM. They also perform a data analysis trying to exclude unnecessary features. They conclude that XGBoost outperforms any other classifier in the domain. They also identify *quantity, waterpoint_type, longitude and latitude (coordinate location), gps_height, wpt_nama, subvillage, ward, construction_year and date_recorded* as the 10 most important features.

3 Dataset

The dataset for the task has been provided in the challenge Pump It Up: Data Mining the Water Table [2] posted on the Driven Data [3] website.

The features provided in the dataset have been described on the challenge webpage and are shown with their description in Fig. 2. There are 40 columns in the training dataset with over 59,000 instances of data. The labels for each of the instances have been provided in a separate file. Each of these instances have a classifying label which classifies the data point to either “functional”, “non functional” and “functional needs repair”. The labels mean the following:

- functional - The waterpoint is operational and there are no repairs needed
- functional needs repair - The waterpoint is operational, but needs repairs
- non functional - The waterpoint is not operational

4 Preprocessing Techniques

In this project, python will be used as the coding language. To begin with, training dataset contains all the attributes without any labels and a separate file containing just the labels. The decision to combine the two by using the id attribute was taken, which resulted in all the attributes with their labels in a single table.

For generating test data, 20% of the entire dataset was used. The remaining 80% was used for generating training and validation datasets. 10 fold cross-validation technique was used for the creating training and validation dataset.

After visualizing the data on the bar graph, it was found out that the categories with similar names were recognized differently due to different cases (uppercase/lowercase) of some part or whole of the category or some abbreviations (e.g. Gover and Government, Commu and Community). Our assumption is that if the name (in it's exact given form) is different, then the categories are different, else they are already being recognized as same.

The missing values in the dataset were identified as 0 which can be seen in Fig. 11 for the attribute “construction_year”.

Some of the features (such as features showing Geographic location, management and management_group, extraction_type_group and extraction_type_class) had identical results and therefore only one of them seemed useful for the experiments and for reducing the complexity.

- `amount_tsh` - Total static head (amount water available to waterpoint)
- `date_recorded` - The date the row was entered
- `funder` - Who funded the well
- `gps_height` - Altitude of the well
- `installer` - Organization that installed the well
- `longitude` - GPS coordinate
- `latitude` - GPS coordinate
- `wpt_name` - Name of the waterpoint if there is one
- `num_private` -
- `basin` - Geographic water basin
- `subvillage` - Geographic location
- `region` - Geographic location
- `region_code` - Geographic location (coded)
- `district_code` - Geographic location (coded)
- `lga` - Geographic location
- `ward` - Geographic location
- `population` - Population around the well
- `public_meeting` - True/False
- `recorded_by` - Group entering this row of data
- `scheme_management` - Who operates the waterpoint
- `scheme_name` - Who operates the waterpoint
- `permit` - If the waterpoint is permitted
- `construction_year` - Year the waterpoint was constructed
- `extraction_type` - The kind of extraction the waterpoint uses
- `extraction_type_group` - The kind of extraction the waterpoint uses
- `extraction_type_class` - The kind of extraction the waterpoint uses
- `management` - How the waterpoint is managed
- `management_group` - How the waterpoint is managed
- `payment` - What the water costs
- `payment_type` - What the water costs
- `water_quality` - The quality of the water
- `quality_group` - The quality of the water
- `quantity` - The quantity of water
- `quantity_group` - The quantity of water
- `source` - The source of the water
- `source_type` - The source of the water
- `source_class` - The source of the water
- `waterpoint_type` - The kind of waterpoint
- `waterpoint_type_group` - The kind of waterpoint

Fig. 2. Feature names and their descriptions as provided on the challenge website

5 Experiment

For this project, the three primary techniques to be experimented with are Support Vector Machine, XGBoost and LightGBM. A brief explanation of all the three techniques is provided in this section.

5.1 Support Vector Machines (SVM)

SVM is a supervised learning model with associated machine learning techniques for classification and regression. SVM is a model which will split the data in the best possible way by separating the two groups with the widest possible margin. SVM works as a constraint optimization problem where the constraints are the data points and we want to maximize the margin of classification between any two classes.

The equation of the hyperplane for linearly separable classes for an SVM is $w \bullet x + b = 0$ such that

$$w \bullet x_i + b < -1 \quad : if\ y_i = -1 \tag{1}$$
$$w \bullet x_i + b > +1 \quad : if\ y_i = +1 \tag{2}$$

where w is the weight vector, x_i is the ith instance provided in the dataset, b is the bias term, and y_i is the actual class value.

SVM relies on kernel functions for transformation of dimensionality from one dimensional space to a higher dimensional space. A few of the most common functions that have been used with SVM are shown in Eq. 3, Eq. 4 and Eq. 5 representing the polynomial kernel, hyperbolic tangent kernel and the Gaussian radial basis function kernel, respectively. This project will be experimenting with all of these kernel functions to see which one gives the best accuracy.

$$k(q, q') = (1 + q.q')^k \tag{3}$$
$$k(q, q') = tanh(aq.q' + b) \tag{4}$$
$$k(q, q') = exp(-\| q - q' \|^2 / \sigma^2) \tag{5}$$

The Scikit-Learn [7] implementation of the multi-class SVM, with the already implemented kernel functions for the purpose, would be used for this project.

5.2 Extreme Gradient Boosting (XGBoost)

XGBoost [5] is an optimized open source gradient boosting library, developed at Distributed Machine Learning Common (DMLC), designed to be highly efficient, flexible and portable. It implements machine learning algorithms under Gradient Boosting framework. This library has been designed to run on a single machine as well as distributed clusters over Apache Hadoop. It has been developed by Chen and Guestrin [11] at the University of Washington. This library has been developed for Python, R, Java and Scala programming languages.

5.3 Light Gradient Boost Machine (LightGBM)

Light Gradient Boost Machine is a gradient boosting framework which is based on decision trees. Since it is based on decision trees, the split is done leaf-wise rather than the conventional depth-wise or level-wise split. The leaf-wise split results in better accuracy. Advantages of using LightGBM are faster training speed, lower memory usage, better accuracy, compatibility with large datasets.

There has been no research paper written on this implementation but this implementation has proven to be fast and more accurate than a lot of other machine learning gradient boosted trees.

6 Preliminary Result

The dataset provided has 40 columns and 59,400 rows/instances. To have a brief understanding of the dataset, decision to visualize the data on a bar graph was taken for a better understanding of the features and their correlation with the output. Various relations between some attributes/features given in the dataset w.r.t. the required status_group of each of the pumps were obtained. The following results can be seen from the visualization of data.

Out of the three labels (functional, non-functional, functional needs repair), over 30,000 instances have label as "functional", about 23,000 instances have label as "non-functional" and a little less than 5,000 instances have label as "functional needs repair". This distribution can be seen in Fig. 3.

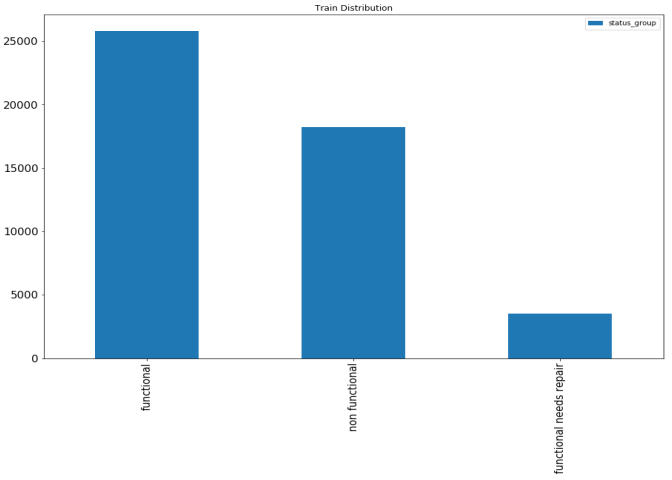

Fig. 3. Distribution of labels for training data

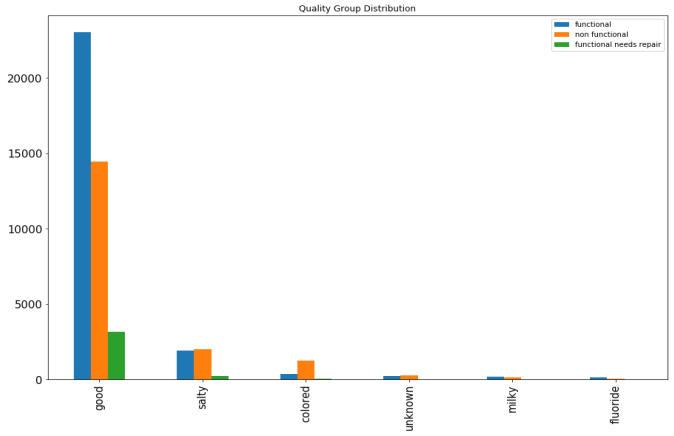

Fig. 4. Distribution of labels over all instances for the parameter quality_group

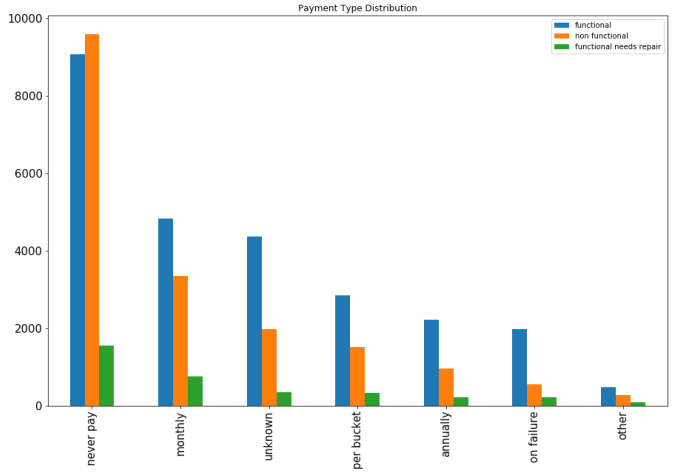

Fig. 5. Distribution of labels over all instances for the parameter payment_type

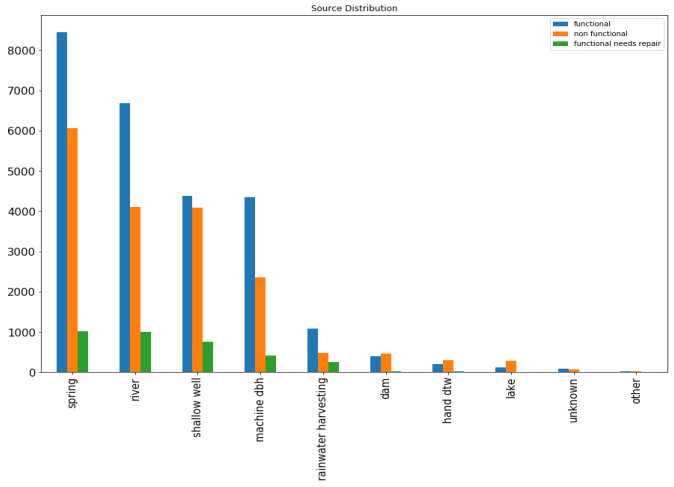

Fig. 6. Distribution of labels over all instances for the parameter source

For “quality_group” which is one of the feature, looking at the result, we came to a conclusion that if the value for quality_group attribute is “good” the chances of it to be classified with the label “functional” are more whereas the chances of it to be classified with the label “non- functional” are more for rest of the values. This can be seen in Fig. 4.

For the feature “payment_type”, if the value for the this attribute is “never pay” then there is a good chance that the instance will be classified as “non-functional”, for all the other values the instance to be classified as “functional” has a greater chance. This can be seen in Fig. 5.

For the attribute “source”, the distribution can be seen in Fig. 6, the bar graph obtained was much more complex than the rest of the attributes and therefore, to derive a conclusion only by looking at the bar graph was not possible

For the feature “quantity”, whose distribution is in Fig. 7, if the value of the feature is “enough” then it will have higher chances of it to be classified as “functional”, else the chances of it to be classified as ”non-functional” will be more.

For “installer” attribute (distribution in Fig. 8.), There are categories with similar names, but which are recognized different due to different cases (lowercase/uppercase, e.g. World vision and World Vision) of some part or whole of the category or some abbreviations (e.g. Gover and Government, Commu and Community). Our assumption is that if the name (in it's exact given form) is different, then the categories are different, else they are already being recognized as same. These are the results which we obtained for installer attribute.

The test data which comprises of about 12,000 instances, over 6000 instances are classified with the label “functional”, about 4000 instances have the label “non-functional” and about 1000 instances have the label as “functional needs repair”. The distribution of test data has been shown in Fig. 9.

For the attribute waterpoint_type, if the value of the attribute is “communal standpipe”,“hand pump” or “improved spring” then the probability of it to be classified with the label “functional” is high.

For other values, probability of it to be classified as “non-functional” is high. This distribution can be seen in Fig. 10. For the attribute contruction_year, if the value is 0 then the year of construction is not provided. The values for construction_year ranges from 1960 to 2013.

7 Conclusion and Future Work

We have successfully visualized the dataset provided to us. The next steps include getting familiar with the libraries (Scikit-Learn, XGBoost, and LightGBM) in the machine learning, python programming space and producing results based on the data and the attribute selection process after the visualization of the useful factors.

Our aim is to produce data over a 10-fold cross validation of the training set after splitting the provided training set into test and train set. The 10-fold cross validation will serve as our train and validation set for iterative improvement.

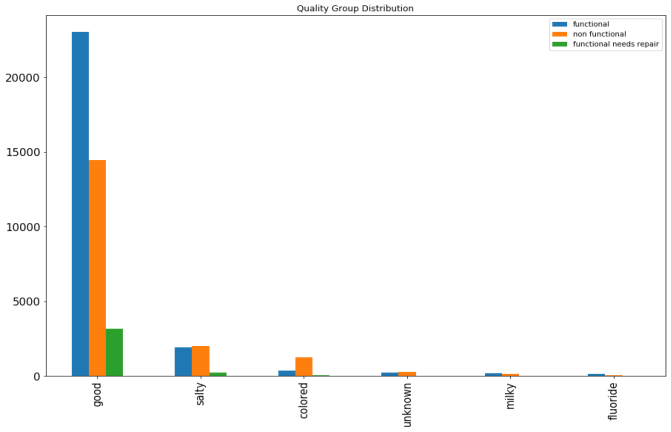

Fig. 7. Distribution of labels over all instances for the parameter quantity

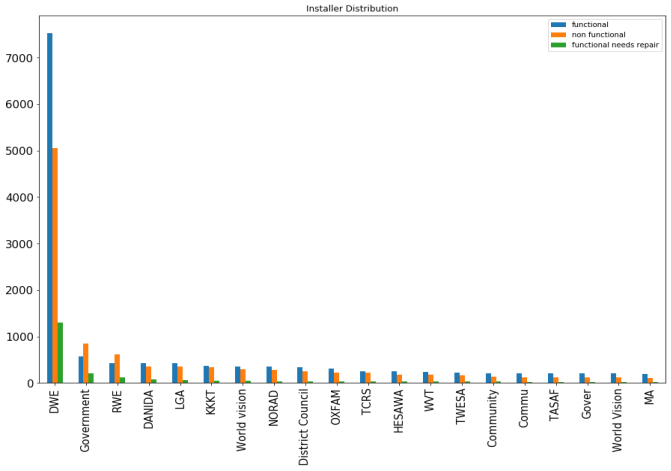

Fig. 8. Distribution of labels over all instances for the parameter installer

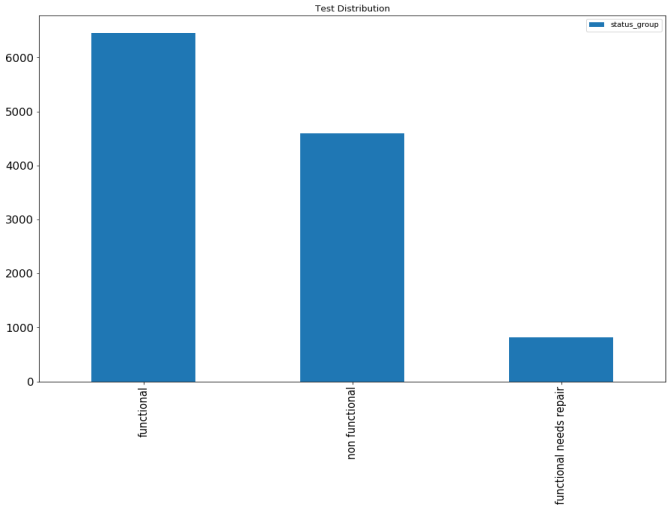

Fig. 9. Distribution of labels for test data

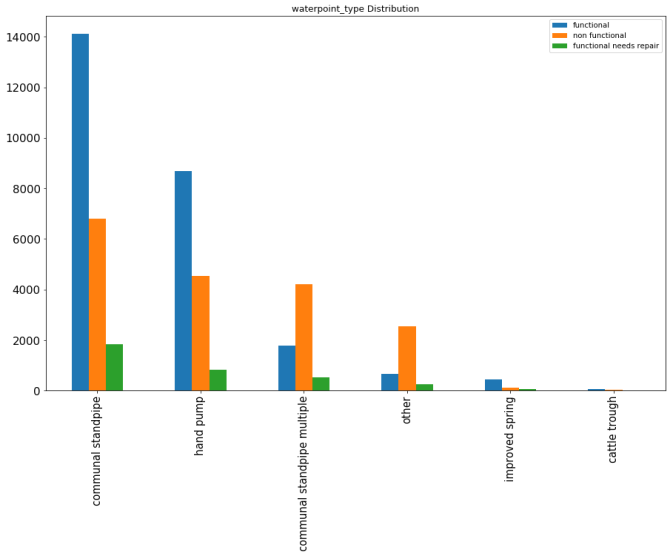

Fig. 10. Distribution of labels over all instances for the parameter waterpoint_type

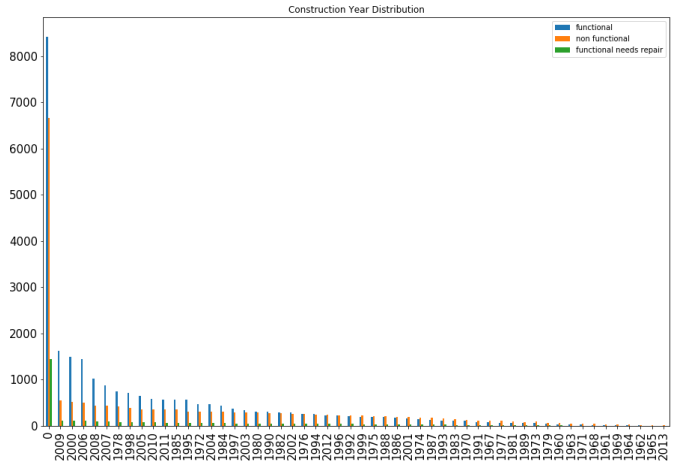

Fig. 11. Distribution of labels over all instances for the parameter construction_year

8 References

[1] https://en.wikipedia.org/wiki/Tanzania

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

[3] https://www.drivendata.org/

[4] https://github.com/dmlc/xgboost

[5] https://lightgbm.readthedocs.io/en/latest/Features.html

[6] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12.Oct (2011): 2825-2830.

[7] Tran, Huu Dung, and A. W. M. Ng. "Classifying structural condition of deteriorating stormwater pipes using support vector machine." *Pipelines 2010: Climbing New Peaks to Infrastructure Reliability: Renew, Rehab, and Reinvest*. 2010. 857-866.

[8] Harvey, Richard, and Edward McBean. "Understanding Stormwater Pipe Deterioration Through Data Mining." *Journal of Water Management Modeling* (2014).

[9] Arymurthy, Aniati Murni. "Predicting the status of water pumps using data mining approach." *Big Data and Information Security (IWBIS), International Workshop on*. IEEE, 2016.

[10] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016.