**Project: Pump it Up: Data Mining the Water Table**

**Introduction**

This project is based on the competition Driven Data® had published about water pumps [1] in Tanzania, a large country that suffers from access to good quality water. The information for the competition was obtained by the Tanzania Ministry of Water using an open source platform called Taarifa [2].

The data that had been collected has lots of features related to water pumps, it has data, in general, related to geographic locations, organizations that build and manage them, also there is some data about the Region, Local government areas, also type of extraction, type of payment, quantity, among other. With all this data, the purpose of the competition is to identify the functionality of water pumps, in terms of three possibilities functional, non-functional, and functional but it needs to repair.

This document analyses the elements related to the functionality of water pumps. Predictive models had been built in order to approximate a precise solution that let the Ministry make good decision about water pumps spread across regions in Tanzania. The metric used for this competition is “Accuracy” for calculating the precision of the model. It had been used process to balance data found in the target feature. Finally, some conclusions of the study were raised.

The result of the project was uploaded in order to score the predictions, which generated a response that is within 14% of the global participant results in this competition. The result was great, because it had been 0.8125, comparing with the first place 0.8294. It is a very good result considering the difference between this results.

**Dataset description**

The data had been collected using an open source web app named “Taarifa”. The data for the training has 59.400 rows and 40 columns without the label that comes in a separate file, in the case of testing data it has 14.850 rows. The description of each column is found in [1]. The name of the target is “status\_group” that has three possible values functional, non-functional, and functional but it needs to repair.

**Exploratory Data Analysis**

The exploratory data analysis was performed by analyzing the context of the problem and the characteristics of the variables as observed in the directory of the project, in this case “Exploration.ipynb” in the notebooks directory of the project. The target variable has three possible outcomes:

* Functional
* Non-functional
* Functional but it needs to repair.

It’s interesting to see these three outcomes, because they describes three potentials problem that may give some insight for a proper decision making. It can be seen that the dataset is not balanced, “Functional” has the most cases followed by “Non-functional” and “Functional but it needs repair “with less cases.

Because of the metric used for this competition is “Accuracy”, the data imbalance has to be solved, because it can be an issue that limit the prediction power of the model used. The initial strategy had been to analyze individually every feature and also in related group of features depending on its characteristics.

The feature *amount\_tsh* that is the total static head (amount water available to waterpoint), is skewed that is why log had been used to smooth it. In the case of the features *date\_recorded, gps\_height*, *longitude* and *latitude* were used as such, just in the case of date it had been taken the month and year of it.

It had been created a couple of features as part of the feature engineering part. The first case was related to date *construction\_year*, which had been subtracted from *date\_recorded*, in order to identify how old that wells are, and also in the case of amount of well water related to population it had been divided *amount\_tsh* by *population*, in these cases some processing had been done at the beginning. By the way, it had been found that the older the water pump the more likely it will require repair.

In the case of *funder* and *installer*, they have lots of different values in each one of them, that is why by looking at its frequencies, it had been decided to reduce to just the 20 most frequent values in these cases, a reference that helped us in this case is [2]. And also it had been noticed that one missing *funder* is also missing *installer,* that is why the null imputer was similar in these cases.

In the case of the following features related to location:

* *basin*
* *subvillage*
* *region*
* *region\_code*
* *district\_code*
* *lga*
* *ward*

It is found that most of them are very closely related, that is why some of these features had been deleted as it is observed down below in this document.

It had been identified that there is high correlation among groups of categorical features, in lots of cases the correlation raised to one. The main reason why this happens is that the information is repeated, for example, in the case of *waterpoint\_type* and *waterpoint\_type\_group,* basically they have same information, that is why it had been decided to delete the following features:

* *Region*
* *Lga*
* *Ward*
* *Extraction\_type\_group*
* *Management\_group*
* *Payment*
* *Water\_quality*
* *Quantity\_group*
* *Source\_type*
* *Waterpoint\_type\_group*

With the rest of categorical features that is a relevant quantity, it had been decided to apply one-hot encoding in the cases in which the number of different feature values was below 50 different feature values, which are the following:

* *funder\_cat*
* *basin*
* *installer\_cat*
* *public\_meeting*
* *scheme\_management*
* *permit*
* *extraction\_type*
* *extraction\_type\_class*
* *management*
* *payment\_type*
* *quality\_group*
* *quantity*
* *source*
* *source\_class*
* *waterpoint\_type*

**Imbalance data**

One of the most important problems found in this dataset was the imbalance data fount in the target feature *status\_group*, specifically in the case of the performance metric that was “Accuracy”, in this particular case, the metric needs to have a balanced target. That is why it had been used the function SMOTE which balances the target values in order to have the same quantities in every target feature value. In this case it had been used oversampling for the processing.

**Performance Metric**

The metric of the competition was defined as “accuracy”. This metric needs the dataset target be balanced in order to get the best results. But in the case of this competition the target values were imbalanced but solved with SMOTE.

**Modeling**

It had been used just one model for the results, every model output had been sent to Data Driven. it had been used Random Forest, Grading Boosting Machine and XGBoost. The one that worked the best was RandomForest, with which it had been obtained the highest score in the submission to Data Driven®. This model can help the Tanzanian government to make good decision about wells and water pumps, and also have these model to predict future events in the case new wells were built.

**Data:**

The original data was obtained from the [DrivenData 'Pump it Up: Data Mining the Water Table'](https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/) competition. Basically, there are 4 different data sets; submission format, training set, test set and train labels set which contains status of wells. With given training set and labels set, competitors are wanted to build predictive model and apply it to test set to determine status of the wells and submit.

**Bibliography**

[1] Tanzania Ministry of Water & Data Driven. (n.d.). *Pump it Up: Data Mining the Water Table*. Pump It Up: Data Mining the Water Table. Retrieved 2015, from https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/

[2] Tanzania Ministry of Water & Data Driven. (n.d.). *Pump it Up: Data Mining the Water Table*. Retrieved 2015, from

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

[3] Gumusbas, E. (n.d.). *Tanzania Water Well Prediction*. Github.Com. Retrieved 2020, from https://github.com/ezgigm/Project3\_TanzanianWaterWell\_Status\_Prediction