SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Masters in Big Data and Analytics 2017-2018

MACHINE LEARNING II

Report

"Pump it Up: Data Mining the Water Table"

Individual

ANDREA BLASIOLI – KAMAL NANDAN

User id in drivendata.org: ***kamalnroy*** and ***andreab***

I. EXECUTIVE SUMMARY

Using data from Taarifa and the Tanzanian Ministry of Water, we are asked to predict which pumps are functional, which need some repairs, and which are not functional. Predict one of these three classes based on a number of variables about what kind of pump is operating, when it was installed, and how it is managed. A smart understanding of which waterpoints will fail can improve maintenance operations and ensure that clean, potable water is available to communities across Tanzania.

A correct and well-planned Maintenance of physical asset is fundamental along the time in order to make sure of high service level for the final users and stakeholder and in order to have a good threshold with the cost related to the maintenance operations and the spare parts. Maintenance over the years became a science, mixing statistical and economical modelling, engineering and technology.

There are three different type of Maintenance: Preventive, Corrective and Predictive. In this practical case we want to apply Machine Learning technique to help the Tanzanian Ministry to implement **Predictive Maintenance** on its water pumps. This maintenance strategy uses data in conjunction with analysed historical trends to continuously evaluate the system health and predict a breakdown before it happens. This strategy allows maintenance to be performed more efficiently, since more up-to-date data is obtained about how close the product is to failure.

II. FEATURE ENGINEERING

In this chapter we list first the executed feature engineering which were positive in terms of results for our model and after the one we tried during our machine learning pipeline and were not useful for our model and results.

We created functions for it:
→impute median values for 0 **amount_tsh**
→impute median values for 0 **gps height**
→impute median values for 0 **population**
→transform **population** into categories, we create variables, trying to give to each category of population the same amount of pumps
→impute median values for **construction year**
→impute missing Booleans with false and convert each value to float or integer
→**Date_recorded**: Since random forest doesn't work on datetime, we break them down to month and year, also if we simply convert month into numerical values, it doesn't work well because there are big distance between Jan and December and also between 1970 to 2010, to take an example. So its better to one hot encode them after transforming the date to month and year.
→**Latitude** and **longitude**, it the values of latitude and longitude of Tanzania don't fall in this range then will be discarded.
→For the columns with many different values. Since our motive is to one hot encode the categorical columns, we need to reduce the no. of categories for each column. For this purpose we put together all the values that have counts less than 100 as "other" category for each column that contains string values.
→Use LDA to reduce the no. of dimensions. We will apply these on population, gps_height, latitude longitude because these have many different values and hence they are perfect candidates

III. MODEL SELECTION

The Problem we are facing is a classification one, we are asked to predict whether a pump is functional, needs to repair or is not functional. In Machine Learning a classification problem is a supervised

learning. Using our set of variables, we generate a function that map inputs to desired outputs. Examples of Supervised Learning are Regression, Decision Tree, Random Forest, KNN, Logistic Regression, SVM.

In our classification, we need to assign to a pump three different outputs: Functional, needs to repair and not functional; that's why the Logistic Regression is not the suitable model as it is a binary one. We went on Tree Model and we decided to use the **Random Forest model** is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

IV. FEATURE SELECTION

We will drop the following columns because of the reasons mentioned below:
1. id - Not a feature
2. amount_tsh – trials and results
3. num_private - too many different values(approx 58000), so holds no significance
4. region - almost perfect correaltion with region_code
5. quantity - very strong correlation with quantity_group
6. quality_group - very strong correlation with water_quality
7. source_type - very strong correlation with source
8. water_point_group - very strong correlation with water_point_type
9. payment - very strong correlation with payment_type
10. extraction_type_group - strong correlation with extraction_type
11. recorded_by - same values in all the rows, so its not a discriminant
12. subvillage/district_code/lga/ward - all these denote region, as such we can drop these. region is already being represented by lat/long and also by region_code
13. scheme_name - 2697 different values and 28166 empty values; so this field is almost useless

V. MODEL VALIDATION

The Random Forest Model allow us to validate our model. The **oob** function of the RF Model in Python allow us, if it is TRUE to validate what we have predicted.

The **Confusion Matrix** build in the code, is as well fundamental to understand the Accuracy of your model and feature selection.