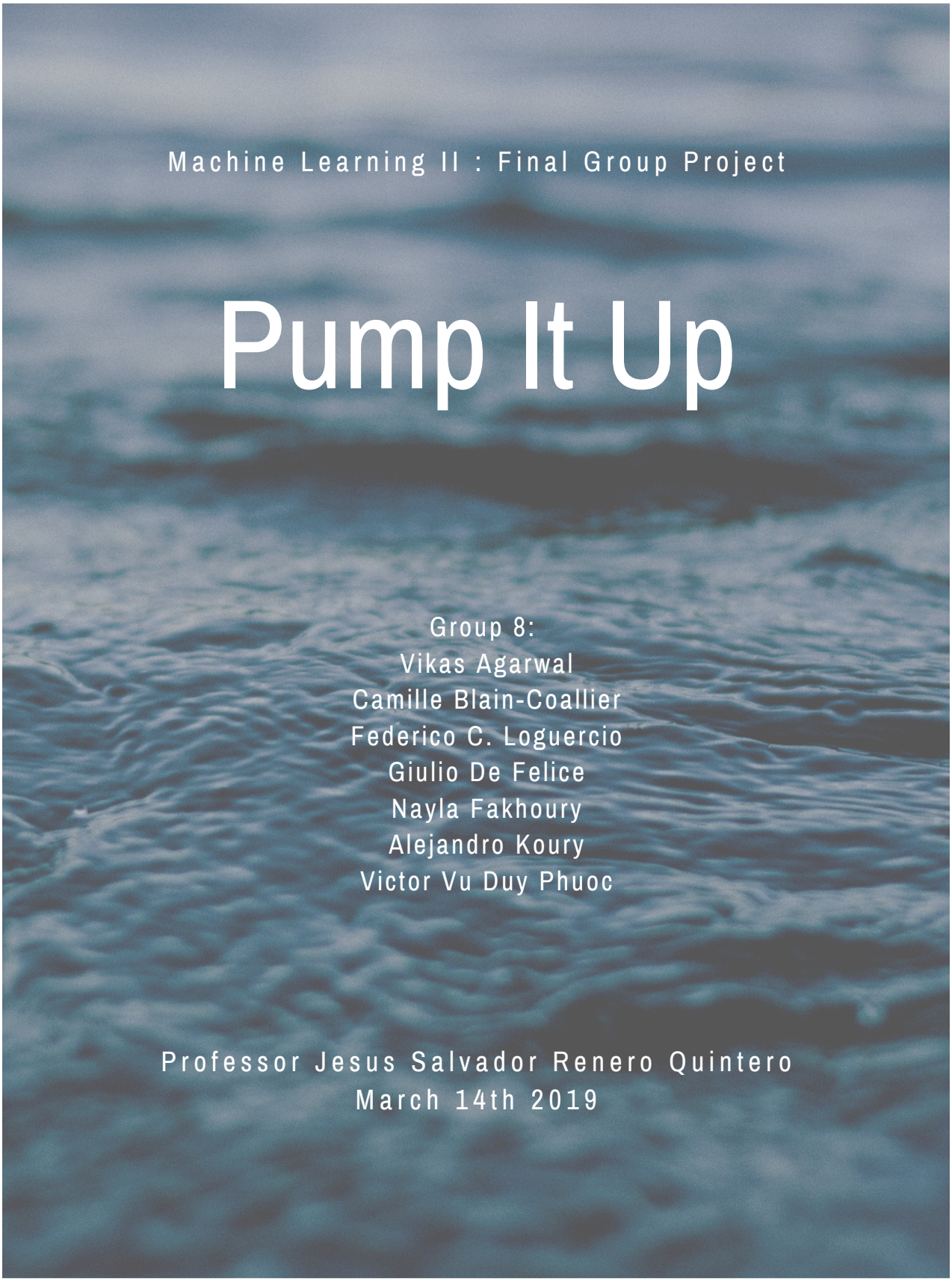


Machine Learning II : Final Group Project

Pump It Up

Group 8:
Vikas Agarwal
Camille Blain-Coallier
Federico C. Loguercio
Giulio De Felice
Nayla Fakhoury
Alejandro Koury
Victor Vu Duy Phuoc

Professor Jesus Salvador Renero Quintero
March 14th 2019

TABLE OF CONTENTS

PROBLEM STATEMENT

The goal of the case study is to create a model able to classify which pumps are functional, which need repairs, and which are non-functional. In addition, the goal is also to understand which variables are the most important in classifying a pump's functionality. The results obtained will help improve maintenance operations for water pumps and ensure that clean, potable water is available to communities across Tanzania.

ANALYSIS

DATA EXPLORATION

At first glance, the water pump Training Set is comprised of around 60k rows and the Test Set is comprised of around 15k rows (for a total of around 75k), both characterized by 41 variables (including the target variable), with features ranging from numerical to categorical. Considering the high number of variables, here are some of the original fields that will be analyzed when creating our model:

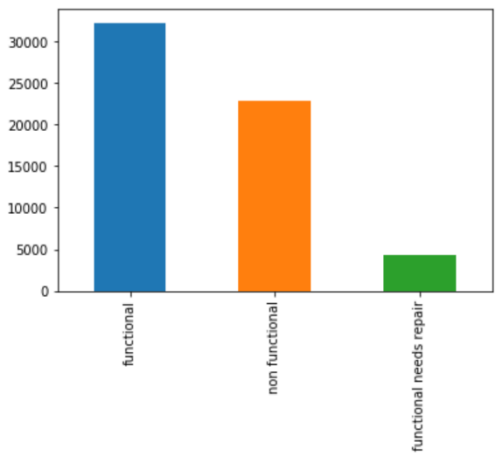
- Longitude and Latitude: GPS coordinates of the pump (float64)
- Basin: The body of water which the pump is feeding from (object)
- Quantity: The quantity of water in the pump's respective basin (object)
- Waterpoint_type: The kind of waterpoint from which the water comes out (object)
- Construction_year: Year the waterpoint was constructed (n64)

In this experiment, we want to classify "status_group" which is the functionality state of the water pumps. A pump's functionality can have three status: functional, non-functional and functional but needs to be repaired.

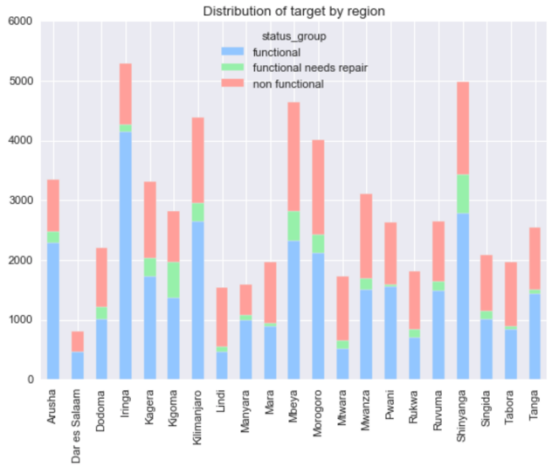
EXPLORATORY ANALYSIS

Before analyzing the dataset, the variable types were corrected to reflect their true value; converting all object variables to category or numeric and parsing dates.

When having a quick overview at the target variable ("status_group"), we can see that there is an uneven distribution of the different classes. In addition, considering the small amounts of pumps that need repair, we can assume that the communities aren't capable of predicting maintenance requirements before the pumps become non-functional. Thus, we will create various models to classify the three types of functionality in the target variable.



The following graph shows the distribution of functional/non-functional/functional needs repair pumps per region in Tanzania. It is evident that the proportions heavily depend on the region.



The distribution of missing values across the levels of the target variable are displayed in the table below. For example, while overall 38.4% of missing data is for pumps needing repair, 45.9% of missing data in scheme management is for pumps which need repair. Missing values for certain variables appear to contain information regarding whether a pump is functional or not.

	overall	scheme_name	scheme_management	installer_cleaned	funder_cleaned	public_meeting	subvillage
functional	0.543081	0.514379	0.483105	0.547294	0.544979	0.503299	0.552561
non functional	0.384242	0.414471	0.459376	0.332422	0.334801	0.449910	0.444744
functional needs repair	0.072677	0.071150	0.057519	0.120284	0.120220	0.046791	0.002695

DATA CLEANING AND PREPARATION

Initially, since the data had many missing values and incorrect values, an intense data cleaning process was performed to obtain data that would allow us to create an accurate model. Particularly, the variables funder and installer where full of spelling mistakes. Before cleaning the two, only 15% of pumps had the same entity for funder and installer. After cleaning both and standardising the spelling, that value went up to 45%.

The data preparation done in the “Data Preparation” section of the notebook consists of the following steps:

1. Append new datasets to the whole data frame, which included external information about precipitation (“pr”) and temperature (“tas”) in Celsius.
2. Verify presence of missing values: missing values are present (will be analyzed further on).
3. Convert feature data types to their proper types (“Converting columns to their true type”).
4. Fill the zeros in “construction_year” with the mean and create dummy variables indicating previous zeros.
5. Correct skewness and standardize the data through the PowerTransformer from the sklearn library.

There was no need to remove outliers because, when analyzing the dataset after the data preparation, all variables that contained outliers were corrected (outliers were identified as data outside 1.5 times the interquartile range).

BASELINE MODEL (BENCHMARK)

A baseline was estimated using logistic regression, including all the variables in the dataset at this point. Categorical variables where label encoded for this purpose.

Classification report

	precision	recall
functional	0.56	0.88
functional needs repair	0.00	0.00
non functional	0.47	0.17

Baseline Accuracy Score: 0.5481

FEATURE ENGINEERING

The feature engineering done in the "Feature Engineering" part of the notebook consists of the following steps:

1. Group certain "extraction_types" to decrease the number of categories for this variable.
2. Remove under represented values for the variables "installer" and "funder" after being cleaned. Thus, we dropped the values that totalled to less than 1% of the total levels within those two specific variables.
3. Create a new variable named distance. This variable was created to add further information to the dataset in relation to the distance between the pump and (latitude, longitude) = (0,0). This variable simply attempts to consolidate some of the information contained in the coordinates into one continuous variable. To give this variable more interpretability, a different version was also added indicating the distance between the pump and the respective basin from which it sourced the water.
4. Create 300 geographical clusters to regroup variables based on latitude and longitude. This adds a more granular regional segmentation of pumps compared to the region variable, while being significantly cleaner and more evenly distributed than the existing variables such as subvillage.
5. Create two date range variables which are "days_since_recorded" and "time_since_construction". These two variables will allow us to assign a numeric value that we can use in our model instead of a timestamp.
6. Create indicator variables to represent the missing values found in the exploratory data analysis: we created Boolean variables for the different categories, determining the presence of missing values. These variables are based on the previous finding that missing data in certain variables contain information on the likelihood of a pump being functional.
7. One Hot encode the remaining categorical variables, while excluding variables which would either be duplication (like payment and payment_type), and variables with too many levels (such as ward)
8. Encode the categorical variables with very large amounts of levels to a continuous variable indicating the fraction of functional pumps for that level. For example, each "subvillage" is turned into the fraction of pumps in that subvillage which are functioning. This feature, however, would not be the most accurate estimate of a unit's likelihood to have functioning pump; it yields the same "probability of functioning" for a subvillage with one pump which is functioning as for a subvillage with 100 pumps which are functioning. Thus, a correction like Laplace Smoothing was attempted, adding 1 to both the numerator and the denominator of the fraction.

Following the feature engineering steps, we performed recursive feature elimination (RFE) using a Random Forest Classifier with 10 trees, which enabled us to select the most important features of the model. This decreased our dataset from 183 variables to 91 variables. Neither the continuous encodings nor the distance from the basin were chosen.

From there, we decided to apply a randomized and cross validated Grid Search to the Random Forest Classifier (RFC) and then, to the Gradient Boosting Classifier (GBC), tuning their hyperparameters. Here are the following results for the standard RFC and GBC models (where the number of estimators = 50) as well as the applied Grid Search on both (tuned models):

	Model	Function	CV Score	Holdout Score
0	random_forest	base_score	0.80 +/- 0.00066	0.795960
1	random_forest_tuned	base_score	0.81 +/- 0.0034	0.818631
2	gradient_boosting	base_score	0.74 +/- 0.0021	0.746240
3	gradient_boosting_tuned	base_score	0.79 +/- 0.002	0.791695

The low standard deviation of the accuracy from the cross-validation score demonstrates that the performance of the model is stable, at least on the training set. Further transformations were attempted including PCA, binning continuous variables by quantiles and performing genetic programming, which created feature-based combinations of features, keeping the best ones and further elaborating them. All of them worsened the model.

RESULTS AND CONCLUSION

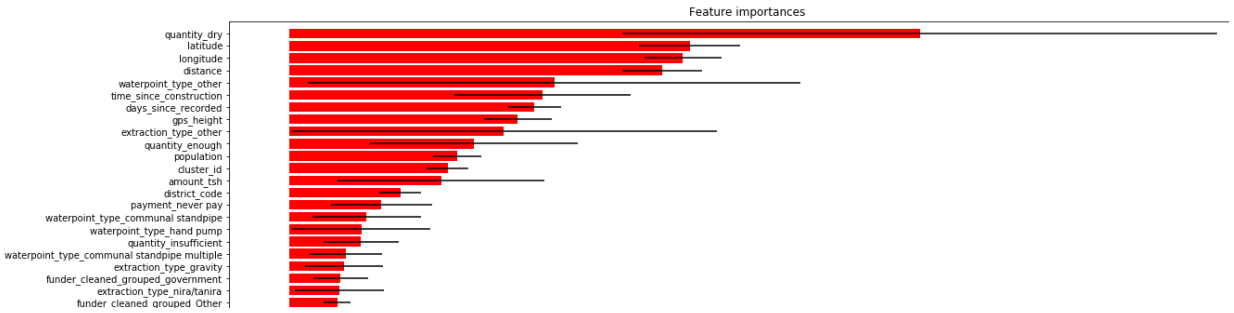
Multiple models were submitted to the competition. Both the tuned and untuned versions of the RFC and GBT were considered, as well as a One vs Rest (OVR) classifier (which fits a RFC to each different class of the target variable). Finally, a stacked ensemble model, comprised of the best two previous models (tuned RFC and GBT) plus the OVR was fit. The stacked model used the predictions of the previous models as input to classify the target variable on the test set. Ideally the stacked model would be able to benefit from uncorrelated errors of the different models, achieving an even higher accuracy. It also was attempted to fit a model to a reduced set of 46 variables, based on feature importance, as well as using the entire dataset before RFE. The following scores were obtained:

Model	Result
Tuned Random Forest	0.8232
Stacked using RF, GBT and OVR	0.8189
Stacked using tuned RF, GBT and OVR	0.8182
Stacked using tuned RF, tuned GBT and OVR	0.8207
Stacked using tuned RF, tuned GBT and OVR, using a further reduced dataset (46 variables)	0.8182
Stacked using tuned RF, tuned GBT and OVR, using all features (no RFE)	0.8172

In conclusion, the tuned Random Forest produced the highest accuracy at 82.32%, reaching rank 197 in the competition. The stacked model using tuned RF, tuned GBT and OVR obtained the second highest accuracy at 82.07%. Even though the pure Random Forest performed best in the competition, it might be advisable to proceed with the stacked model instead. It only performed slightly worse, while being significantly more

stable. Depending on the random seed set, the RF was achieving scores between 0.8201 and 0.8232, whereas the stacked model was only ranging from 0.8200 to 0.8207.

Plotting the variable importance below, it can be seen that when the basin is dry, together with location-related variables, that these are the main features for which the model's decisions are based on. This has important implications for the maintenance of the pumps, as it indicates that as soon as one malfunctioning pump is detected, nearby pumps are also likely to be defected.



IMPROVEMENTS

For the final random forest, a classification matrix was created from which we were able calculate the proportion of true positives among all positives (precision) and the proportion of true positives that have been retrieved over the total amount of relevant instances. As we can see, our model is good at selecting the number of relevant items over all relevant elements for functional pumps, as well as true positives over all positives for functional and non-functional. The model isn't as accurate for the pumps that need repair because of the imbalanced regarding pumps needing repair. To produce a better model, the dataset would require more pumps that need repair or an advanced sampling technique to rebalance the dataset. Alternatively, an entirely separate model being fit and optimized for that class could improve the respective performance.

Classification report

	precision	recall
functional	0.80	0.91
functional needs repair	0.67	0.33
non functional	0.86	0.78

It should be noted that accuracy may not be the best metric to evaluate the performance of this model. Accuracy is somewhat flawed when the target is unbalanced; it can inappropriately inflate model performance. From a business perspective, precision or recall of functional pumps may be more important. If it is costly to travel to a certain pump to check if it is not working and if it is necessary to replace it, it may be more important to achieve high precision for that class. Whereas if it is crucial to identify as many non-functional pumps as possible, recall of that class might be the way to go.

In the future, to better classify the status of the pumps, it would be useful to gather further information related to population and GDP. These economic measures may influence our model because richer regions may have more resources to build better water pumps, hence reducing the amounts of repairs needed. This would allow us to increase the level of performance of the classification model, while avoiding bias within certain variables.