Pump It Up: Prediction of Tanzanian wellpoint status based on available information from inspection of wellpoint

Report by: Brian Wilson

1. Project Statement

The project is based on the Pump It Up competition on DrivenData.com1. Data is provided for wellpoints in Tanzania. These are well locations where people come and collect drinking water. They want to be able to predict which waterpoints are functional and which are non-functional. They also want to be able to differentiate between functional but out of service and nonfunctional.
From the project partners Tanzanian Ministry of Water and Taarifa:

"Using data from Taarifa and the Tanzanian Ministry of Water, can you predict which pumps are functional, which need some repairs, and which don't work at all? This is an intermediate-level practice competition. Predict one of these three classes based on several 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."[1]

2. Datasets

The provided dataset includes records of just under 60,000 inspections with accompanying status labels (classes) that were performed on wellpoints. The inspections took place between 2002 and 2004, though the vast majority of the data is from between 2012 and 2014. The columns provided are as follows:

- `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)

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

- `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

lat, long, date recorded, funder, installer, lga, ward, extraction_type, source, waterpoint_type

3. EDA/Cleaning/Processing

The data provided was relatively messy and required a considerable amount of cleaning. This included imputation of missing and/or incorrect values as well as categorization of variables into useful buckets. We applied a Chi[2] Test to compare the independent variables distribution to the dependent variables distribution. This gave us an idea if the categories within each column matched with the overall distribution of classes. Where there was a significant difference we can hypothesize a stronger predictive variable of on or another.

a. Classes (dependent variable)
    i. The classes are unevenly distributed with the overall dataset having the following split:
        1. Functional 54.3%

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

2. Functional needs repair 7.3%
3. Non functional 38.4%

b. Filled NaN values
    i. Installer: Unknown
    ii. Permit: False
    iii. Funder: Unknown
    iv. Public_Meeting: False
    v. Scheme Management: Unknown
    vi. Scheme_name: Unknown
c. Water / Equipment Columns
    i. Source / Source Type / Source Class
        1. These three variables give the same information about what kind of water source is being pulled from (lake, dam, etc.) at varying levels of detail. Source gives the most granular data, while source Class gives the least. As source only includes 10 total categories we chose to eliminate the other columns as we would rather keep as granular data as possible.
    ii. Extraction Type, Extraction Type Group, Extraction Type Class
        1. These three variables give the same information about what kind of extraction device is being used to pull the water from the source (handpump, windmill, etc.) at varying levels of detail. Extraction Type gives the most granular data, while Extraction Type Group gives the least. As Extraction Type only includes 19 total categories we chose to eliminate the other columns as we would rather keep as granular data as possible. We will be wrapping some of the smaller categories into a general 'other – handpump' category, which will eliminate 3 categories.
    iii. Water Quality / Quality Group
        1. These two columns give the same information with water quality being more granular. Quality Group has been dropped.
        2. Created an ordinal categorical column that ranks by amount of each class in relation to general percentage. Ex: more functional than expected earns a higher rank for that category.
    iv. Quantity / Quantity Group
        1. These two columns give the same information with Quantity being more granular. Quantity Group has been dropped.
        2. Created an ordinal categorical column that ranks by amount of each class in relation to general percentage. Ex: more functional than expected earns a higher rank for that category.
        3. Created a column that combines the quantity and quality ranking columns
    v. Waterpoint Type / Waterpoint Type Group
        1. These two columns give the same information with Waterpoint Type being more granular. Waterpoint Type Group has been dropped.

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

d. Geographic Columns
    i. Latitude and Longitude
        1. Upon inspection it was noted that a number of lats and longs had been entered as zeros. This was corrected by noting there basin and selecting lat and long within the IQR of the basin values.
    ii. Subvillage
        1. The subvillage column includes the name of the subvillages where the wellpoints are located. This column's data is very messy with over 300 missing values and many subvillages represented by only 1, 2 or 3 letters.
        2. We imputed values based upon a KNRegressor model based on a handful of other categories for entries that had 3 or less characters.
        3. NLP could be used in the future to better categorize these errors. As there are over 19,000 subvillages, we will be only be keeping separate categories for subvillages that appear more than 200 times. All others will be placed in an other category.
        4. We also created a new ordinal categorical column that ranks by size.
    iii. Region Code and Region
        1. These two columns appear to be providing the same information, one in numeric form and one in string form. We will keep the numeric form (region code)
    iv. LGA
        1. This column represents geographic regions. It includes a number of labels that include the word urban or rural. We grouped any categories with less than 500 wellpoints into an other category to reduce the number of categories. We are also going to create an urban_rural column that labels either urban, rural or unknown based on the LGA label including either word. This could be enhanced with mapping based on population counts in the future.
    v. Ward/Basin
        1. These two columns represent geographic regions. Basins will be left alone, while wards will be broken into small, medium, etc. categories based on amount of wellpoints in order to reduce the number of categories.
        2. We also created a new ordinal categorical column that ranks by size.
e. Administrative Columns
    i. Payment / Payment Type
        1. These two columns give the same information with payment type being more granular. Payment has been dropped.
        2. We also created a new ordinal categorical column that ranks by size.
    ii. Data Recorded / Construction Year
        1. The date recorded was converted into two columns a year column and a month of the year column.

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

2. The construction year column included a large number of zeros. We imputed values based upon a KNRegressor model based on a handful of other categories. Trials showed the model predicted correct years within approx. 4yrs. The construction year was then converted to a years old column taken from date recorded to date built.

iii. Public Meeting / Permit
   1. Both of these columns had values that were filled as previously noted. They denote whether a public meeting was held to approve construction and whether a permit was issued for construction.

iv. Installer / Funder / Scheme Name / Scheme Management / Management Group / Management
   1. These columns note different entities involved in the construction / management of the wellpoints. They each include many categories and so as with previous columns have had smaller categories grouped together into 'other categories. In future a grouping by Kmeans could be useful or if more information was known about well-run / poorly-run entities it could be used to better segment this group.
   2. Created a column that gives a true false for the installer also being the funder

f. Numeric Columns
   i. GPS Height
      1. This column included many incorrect data points that were set to zero or below. We imputed values based upon a KNRegressor model based on the known latitude and longitudes. Trial showed this to be very accurate.

   ii. Amount TSH
      1. Many of the values of this column were zero and others were well above the expected range (2000 max as a conservative estimate). We set all values above 2000 to 2000 and left the zero values as is as there are many factors that could determine this columns values. We attempted imputation of values based upon a KNRegressor model using on a handful of other categories, but results were poor.

   iii. Population
      1. The original population category was kept and a new column was also created for segments of population amounts. This could act a stand in for the urban vs. rural split.

4. Modeling

DrivenData has set accuracy as the metric for use in this competition so we will compare the accuracy of our models. We will also take a look at confusion matrices to see which classes the models do well with and which they do not. Numerical features

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

with the exception of Latitude and longitude were scaled and categorical features were One Hot Encoded for use with all models. PCA was considered, but not found to increase accuracy consistently. See model metric CSV for parameters used for each model.

Overall tree models (XGB, RF) performed much better than linear models (LR, SGD). While the best linear model, Logistic Regression, was able to achieve above 75% accuracy, this was below an out of the box tree model, Random Forest. I expect that with further parameter tuning both XGBoost and Random Forest could see further gains. This would unfortunately require considerably more computing power than I have available at this time.

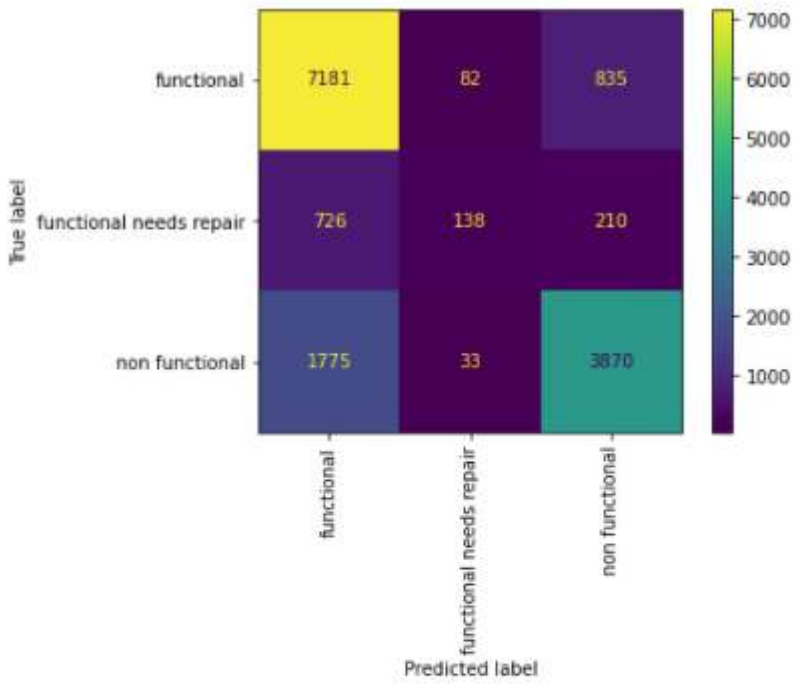
All of the models did best with functional/non functional split (F1 scores near 0.80) and had difficulties with functional needs repair (F1 scores of 0.20 to 0.40). It was this difference that allowed the tree models to have an overall better score as they were more typically around 0.40 while the linear models struggled to exceed 0.30.

It is possible a combination of models could be used for different subsets of the data. A detailed look at which subsets had good vs. bad performance was not performed yet and could help to direct further feature engineering.

The following shows a confusion matrix and classification report for each models best parameters:

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

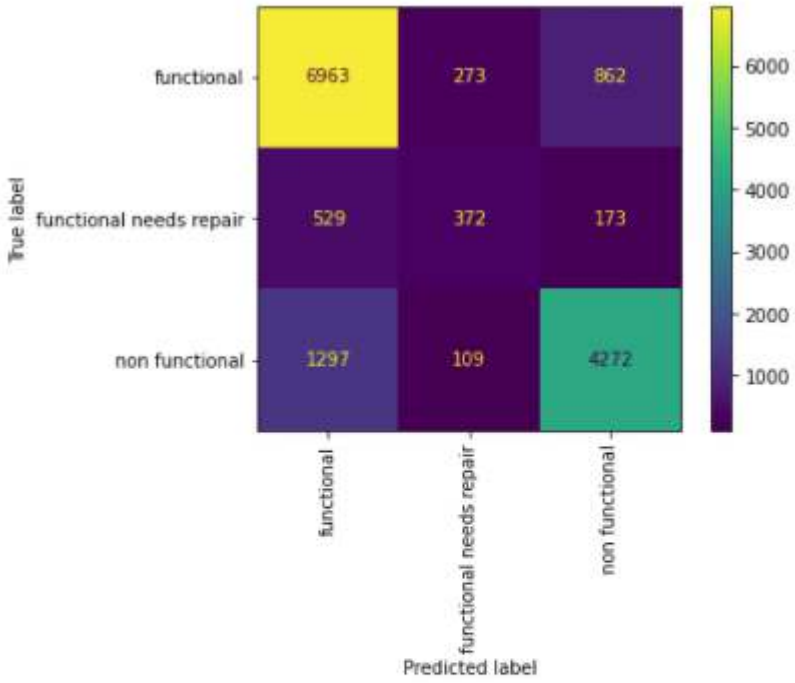
a. Logistic Regression



	precision	recall	f1-score	support
functional	0.74	0.89	0.81	8098
functional needs repair	0.55	0.13	0.21	1074
non functional	0.79	0.68	0.73	5678
accuracy			0.75	14850
macro avg	0.69	0.57	0.58	14850
weighted avg	0.74	0.75	0.73	14850

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

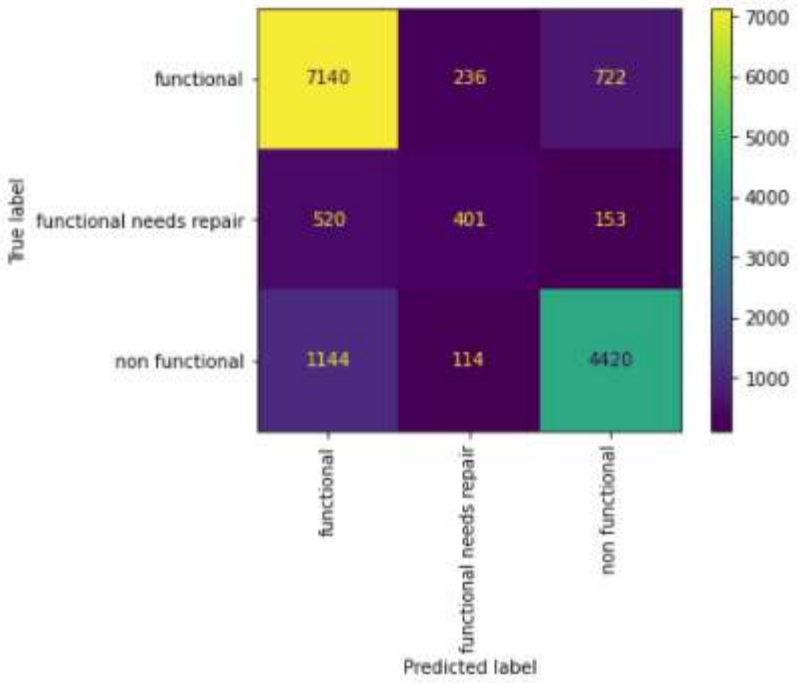
b. KNN



	precision	recall	f1-score	support
functional	0.79	0.86	0.82	8098
functional needs repair	0.49	0.35	0.41	1074
non functional	0.80	0.75	0.78	5678
accuracy			0.78	14850
macro avg	0.70	0.65	0.67	14850
weighted avg	0.78	0.78	0.78	14850

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

c. Random Forest

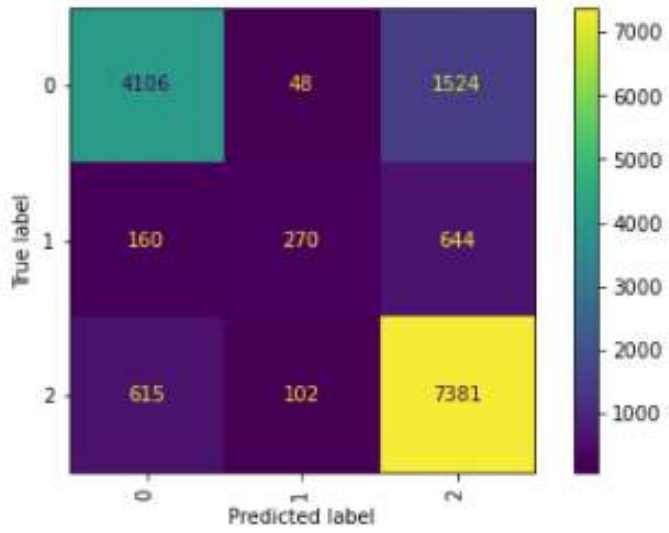


	precision	recall	f1-score	support
functional	0.81	0.88	0.84	8098
functional needs repair	0.53	0.37	0.44	1074
non functional	0.83	0.78	0.81	5678
accuracy			0.81	14850
macro avg	0.73	0.68	0.70	14850
weighted avg	0.80	0.81	0.80	14850

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

d. XGBoost

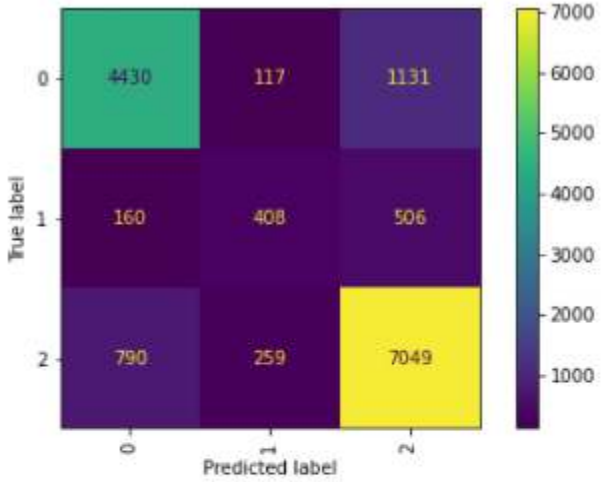
'functional':2, 'functional needs repair':1, 'non functional':0



	precision	recall	f1-score	support
0	0.84	0.72	0.78	5678
1	0.64	0.25	0.36	1074
2	0.77	0.91	0.84	8098
accuracy			0.79	14850
macro avg	0.75	0.63	0.66	14850
weighted avg	0.79	0.79	0.78	14850

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

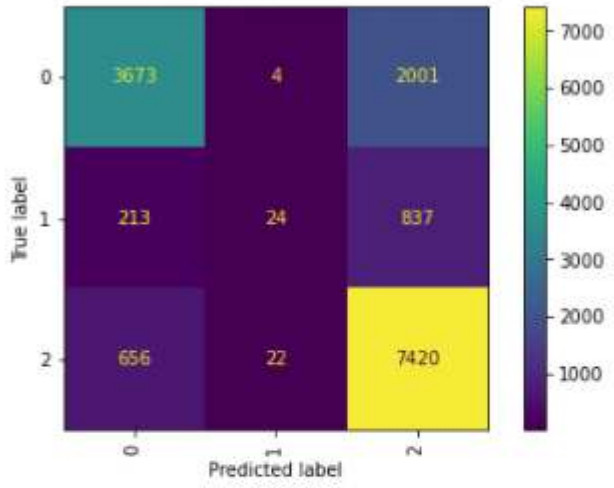
e. ADABoost



	precision	recall	f1-score	support
0	0.82	0.78	0.80	5678
1	0.52	0.38	0.44	1074
2	0.81	0.87	0.84	8098
accuracy			0.80	14850
macro avg	0.72	0.68	0.69	14850
weighted avg	0.80	0.80	0.80	14850

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

f. SGDClassifier



	precision	recall	f1-score	support
0	0.81	0.65	0.72	5678
1	0.48	0.02	0.04	1074
2	0.72	0.92	0.81	8098
accuracy			0.75	14850
macro avg	0.67	0.53	0.52	14850
weighted avg	0.74	0.75	0.72	14850

5. Conclusions

The achievement of over 80% accuracy should allos for the Tanzanian government to use this model as an accurate way to plan ahead for maintenance issues as well as choose which areas to look deeper at for regular up keep and or replacement. The year over year simulation available on Github shows a continued improvement of the model as new information from new years are add in so the tool will only continue to grow in it's ability to predict wellpoint statuses.

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)

[1] Competition: Pump it Up: Data Mining the Water Table (drivendata.org)