markp-rankin / Water-Well-Classification

Project 3

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security Insig

main

markp-rankin Add files via upload ... 3 minutes ago 20

View code

README.md

Identifying Broken and Non-Functioning Water-points in Tanzania

Phase 3 Project by Mark Patterson , December 2020

Background

Basic access to clean water for drinking and sanitation remains a serious problem for 10% of the world's population. Tanzania, in sub-Saharan Africa, is number 7 in a recent list of countries without ready access to safe drinking water. In fact, 43% of the countries 57 million people lack such access.

Although there has been an exponential increase in the number of water-points (wells, springs, surface water, rainwater collection, etc.) constructed in the past 20 to 30 years, many of them are either no-longer functional, or are in need of repair. Having a predictive model that can classify existing water points as either functional or not, can help the government, NGOs and donors to prioritize and fix or reclaim poorly or non-functioning water-points.

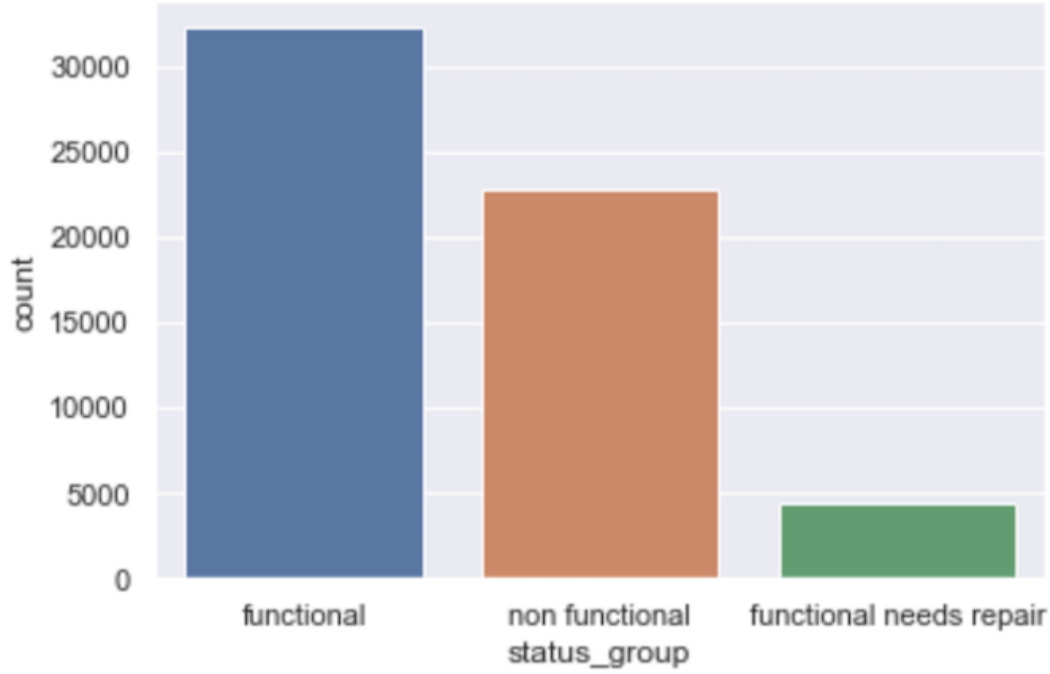
Source Data

The data set was provided by the Tanzanian Ministry of Water and a non-profit group Taarifa. It consists of information on water-points established between 1960 and 2013. The data set contains 59,400 records with 41 variables. The target variable, “status_group” contains 3 classes of values:

- 0 = functional (55%)
- 1 = non-functional (38%)
- 2=functional but needs repair (7%)

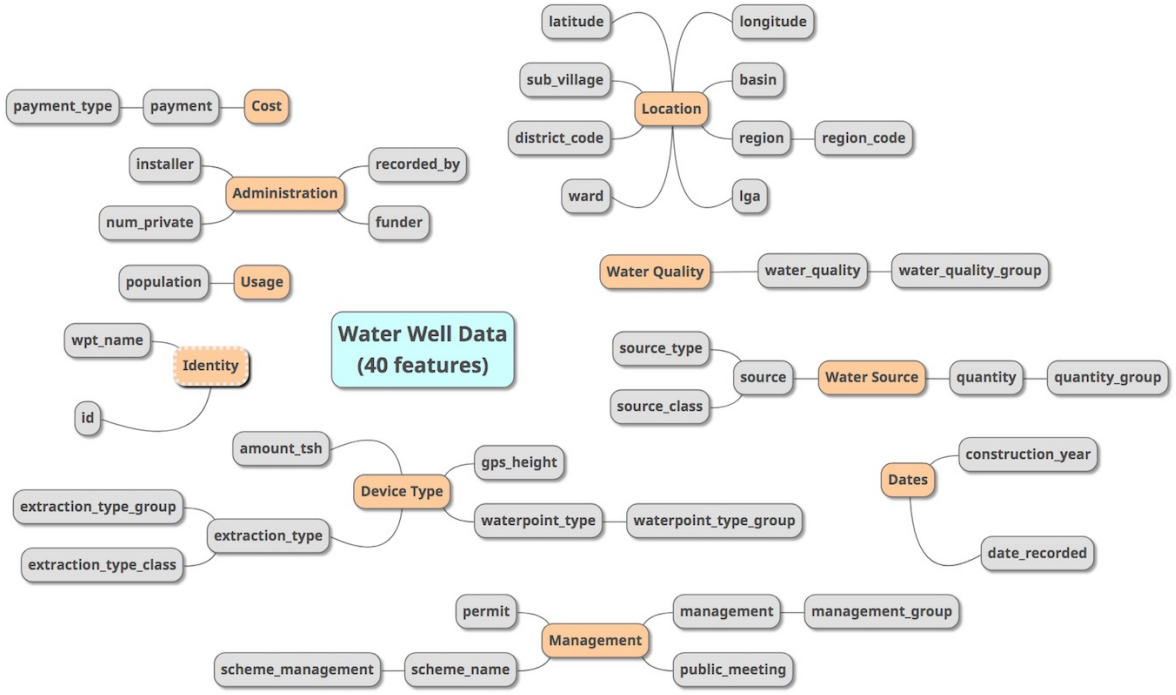
```
functional                 32259
non functional             22824
functional needs repair     4317
Name: status_group, dtype: int64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a257c6f98>
```



EDA

Initial data exploration indicated that many of the variables were different groupings of similar concepts. To get a better handle on this, a mind-map was created. This helped to identify 15 redundant or potentially less useful features. The remaining features were visualized to examine differences in distributions for each of the target classes. For example, it can be seen that older water-points constitute a larger proportion of non-functional ones.



Other issues identified and addressed during data exploration and cleaning included:

- 7 columns that contained null values were dropped.
- Outlier values were examined and in some cases dropped (longitude for example).
- About 5 of the continuous features had a large percentage of 0 values. In some cases these may be valid values, but in others (like construction year), they clearly are not. Rather than cutting these 18,000 records, these values were transformed to the mean year of 1997. Another case of many 0 values was the feature “population.” There were approximately 18,000 with a value of 0. I decided to cut these records.
- Several features had a class of “other” or “unknown” that had a large count. These provide little value as it is not clear if they represent missing values or something else. These were dropped after initial modeling,

At the end of EDA, the dataset contained *35,303 records* and 17 variables. The 9 categorical variables were one-hot encoded, resulting in a total of *75 features.*

Modeling Process

To address this classification problem, 4 supervised machine learning models were used:

- KNN
- Decision Trees
- Random Forest
- XGBoost

All models were run on the initial data set with the unbalanced target classes. The worst performance (in terns of accuracy and recall for class 1 (non-functional) was obtained from a KNN model (test accuracy of 0.58, and class 1 recall of 0.57). Note that the standard scaler was applied to the data prior to modeling.

An iterative approach was taken that included the following refinements to the models and the data set.

1. Initial “vanilla” modeling with all 4 models without taking the class imbalance into account.
2. Modeling with all 4 variables after SMOTE was applied to the target variable (creating synthetic data for the 2 under-represented classes).
3. Several rounds of hyper-parameter tuning (via GridSearchCV) for the Random Forest and XGBoost models. Several iterations of the Decision Trees model with varying max_depth values (3 to 15).
4. Reclassify the target variable into *2 classes*, leaving the majority class as is (55%) and combining the other 2 classes together (45%). This makes sense if we now consider class 1 as being water-point needing attention (rebuild or repair), and makes the class imbalance less problematic so that SMOTE is not needed. 5. Several rounds of Decision Tree models were run with this data-set.
5. Finally, the original data was adjusted (cut 4 original features that were less important; cut 9 “other” and “unknown” classes from original variables, and did not “drop first” wen one-hot encoding. This resulted in a data set of *38 columns* (down from 75 in the first modeling rounds). This was run along with the 2 class target variable on several rounds of Decision Trees (max_depth 3 to 18) and XGBoost.

Final Model

After numerous rounds of modeling and hyper-parameter tuning, an XGBoost model provided the best performance. For the 2-class target variable, this model had an accuracy of 0.82 and a class 1 recall of 0.77. The class 0 recall was 0.87. This means that 82% of the time, this model will predict correctly a true positive (an observation is in class 1 = a non-functioning or in-need of repair water-point) or a true negative (an observation is in class 0 = a functional water-point). The recall measure for class 1 tells us that there is about a 23% chance of having a false negative where we will predict that a water-point is in need of repair or refurbished when in fact it is functioning.

```
------------------------------------------------------------------------
------------------------------------------------------------------------
Classification Report - TEST  - XGBoost-C-Optimized2
------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.82      0.87      0.84      4839
           1       0.82      0.77      0.79      3987

    accuracy                           0.82      8826
   macro avg       0.82      0.82      0.82      8826
weighted avg       0.82      0.82      0.82      8826

------------------------------------------------------------------------
Confusion Matrix - TEST - XGBoost-C-Optimized2
------------------------------------------------------------------------
Predicted     0     1   All
True
0          4186   653  4839
1           926  3061  3987
All        5112  3714  8826
------------------------------------------------------------------------
```

Feature importance was evaluated at various points in the modeling process. In general consistent results were obtained. The features contributing the most to the model included:

- Age (construciton_year)
- Quantity of water (quantity_dry, quantity_enough, amount_tsh)
- Location (longitude, latitude)

	features	importance
0	quantity_dry	0.154350
1	longitude	0.137392
2	waterpoint_type_group_other	0.130257
3	latitude	0.107340
4	construction_year	0.102898
5	gps_height	0.072590
6	population	0.039401
7	amount_tsh	0.039079
8	quantity_enough	0.025860

Conclusions

This classification model, although not perfect, could help government agencies, local communities, and NGOs identify and prioritize which water-points need attention. By having such a predictive tool, water-points could proactively be added to a list of projects for both immediate repair or reclamation and anticipate future work and work load. This could lead to more systematic planning and organizational optimization to save costs and address more projects in a more timely way. Making these water-points operational will help provide safe, clean drinking water to millions of people throughout Tanzania.

Releases

No releases published

Create a new release

Packages

No packages published
Publish your first package

Languages

- Jupyter Notebook 100.0%