Daniel Siegel
June 20th, 2018

Capstone Project

Machine Learning Engineer Nanodegree

Domain Background

The Water Point Data Exchange (WPDx) is a global platform for sharing data on rural water distribution points in the developing world. It has a webform that Government agencies can use to upload their records (typically spreadsheets) and connects directly to the monitoring platforms like mWater and Akvo that non-governmental organizations use. The WPDx provides maps, and an API that can be used to query the shared database. This is a valuable resource for underfunded water ministries, which typically do not have the expertise to develop their own data sharing platform, and for the non-governmental organizations, which can now see their projects in the context of what everyone else is doing. It is also a unique dataset that is being used to develop analytic tools and guide resource allocation the water sector.

A team of students at Berkeley did some analysis on this dataset as part of a Data for Good grant. However, their project limited itself to examining each country one-by-one. By treating the whole WPDx as a single dataset, it might be possible to learn more interesting things about the nature of water point durability.

Problem Statement

Sending staff out into the field to inspect water distribution points is time-consuming and expensive. If an algorithm could predict which water points are most at risk of failure, then it would be possible to allocate maintenance and repairs more efficiently. This algorithm would classify a water distribution point as “broken” or “functional” based on attributes about its location, age, and construction.

Evaluation Metrics

The consequence of a false positive is that we will believe a point to be broken, and send a repair crew in error. This is a waste of money and resources, but still better than the consequence of a false negative. Classifying water points as functional when it is actually broken means a repair crew won’t be sent, and the people living nearby will have no access to clean drinking water. With this in mind, I will use an F-score that

has β=2 as my evaluation metric. This favors models with high recall, minimizing false negatives without ignoring the importance of precision entirely.

Analysis

Data Exploration

Each water point in the WPDx has the following features, listed here in order of their presumed importance:

- **Status_id**: Discrete. This is a label that our classification algorithm will learn to predict.
- **Install_year**: Continuous. This value is used to calculate the age of each water point.
- **Pay**: Discrete. This determines how well the site will be managed. Most countries use only a few simple bins ('no pay' vs. 'pay per month' and 'pay per bucket'), but Uganda and Swaziland record the exact cost, which probably isn't important to this analysis. Only 45% of records record the payment method.
- **Management**: Discrete. There are 7 kinds of management, including water committees, private operators, and direct Government management. Different management structures will likely lead to different levels of maintenance.
- **Adm1**: Discrete. The administrative district that the water point is in. This is useful because it tells you something about the economic and political context, as well as being a proxy for geographic location.
- **Water_Source**: Discrete. There are 226 unique values: shallow boreholes, machine-drilled borehole, lake, spring, rainwater, etc. Different countries often use slightly different names to describe the same thing, which will make learning difficult. For example, in Swaziland they say 'Borehole fitted with manual pump' and in Sierra Leone it's just 'Pump on borehole'. 62% of all water points record a water source.
- **Water_Tech**: Discrete. 576 unique values, which are often unique to their country. 89% of water points have their water_tech documented, although it is sometimes just "Other" or "Unknown."
- **Installer**: Discrete. Over 6000 unique values. Should be a very prognostic feature, but 92% of the installers have built less than a dozen water points, so it might not be possible to learn much about them. Also, Zimbabwe, Uganda, and Swaziland do not appear to be recording this information, so only 44% of water points have the installer documented.

- **Source**: Discrete. Most countries have 1 - 10 different sources contributing data to WPDx, for a total of 28 unique values. It's unclear how predictive this feature will be, but 100% of records have their source documented.
- **Location**: Adm2, lat_deg, and long_deg will definitely have strong correlations, but also might lead to overfitting, and location information is already available to our learned through the Adm1 feature.

Here are 10 data points randomly selected from the dataset:

wpdx_id	water_source	water_tech	pay	management	installer	install_year	adm1	age_years
wpdx-00612223	PROTECTED BOREHOLE	BUSH PUMP TYPE B	NOT RECORDED	NOT RECORDED	NOT RECORDED	2001	Masvingo	15
wpdx-00055265	DAM	OTHER	NOT RECORDED	NOT RECORDED	CENTRAL GOVERNMENT	1972	Singida	34
wpdx-00277847	NOT RECORDED	KIOSK	NOT RECORDED	COMMUNITY MANAGEMENT	NOT RECORDED	1996	BUNDIBUGYO	14
wpdx-00481252	GRAVITY RIVER/STREAM	STANDPIPE	Yes	NOT RECORDED	NOT RECORDED	1993	HHOHHO	21
wpdx-00254708	NOT RECORDED	RAINWATER HARVEST TANK	NOT RECORDED	INSTITUTIONAL MANAGEMENT	NOT RECORDED	2006	TORORO	4
wpdx-00007524	NOT RECORDED	DUG WELL	NOT RECORDED	NOT RECORDED	DACAAR	1994	Nangarhar	11
wpdx-00671733	PROTECTED DUG WELL	AFRIDEV	NOT RECORDED	COMMUNITY MANAGEMENT	WORLD VISION	2009	Bo	7
wpdx-00602468	PROTECTED DEEP WELL	ELEPHANT PUMP	NOT RECORDED	NOT RECORDED	NOT RECORDED	1990	Masvingo	26
wpdx-00076212	STANDPIPE OR TAPSTAND	NOT RECORDED	No water	NOT RECORDED	OTHER	2011	Eastern	1
wpdx-00259834	NOT RECORDED	RAINWATER HARVEST TANK	NOT RECORDED	INSTITUTIONAL MANAGEMENT	NOT RECORDED	2000	KIBOGA	10

Figure 1 – Sample Rows

Exploratory Visualization

The key feature in the WPDx is status_id. This is the label our classification algorithm will predict. In total, 72% of water points are labeled as functional, 21% are labeled as broken, and 7% are unknown (Fig 2).

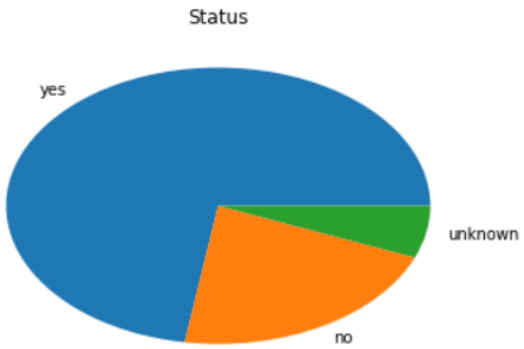

Figure 2 – Functional Status of Water Points

The most important feature to our classification algorithm is install_year. About 82% of water points have their install year documented. This ranges from 1800 – 2017, but 90% of all values are between 1980 and 2013, and half are between 1997 and 2009 (Fig 3).

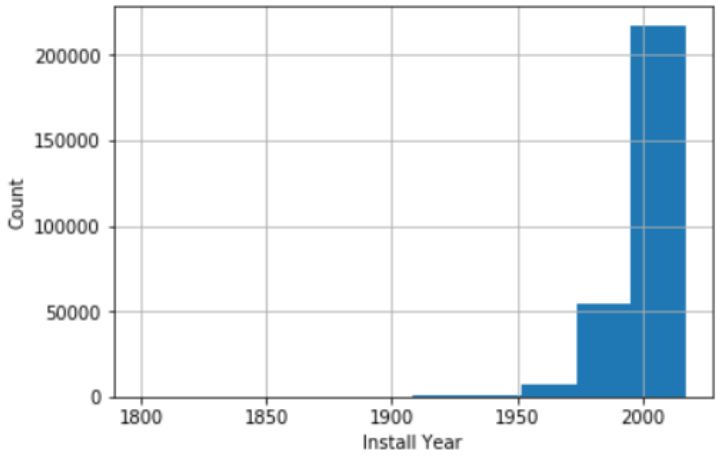

Figure 3 - Install Year Histogram

In addition to being a useful feature in its own right, the year of installation is also used to calculate the age of each water point. Over 95% are younger than 33, although there are some (probably erroneously) recorded to 216 years old (Fig 4). Most of the gauges over 100 years old are points in Swaziland with recorded install_year of 1900, so it is likely this number was used as a filler when someone didn't know the real year of installation. There are also 837 points calculated to have a negative age because the recorded install_year is after the report_date. These outliers will be removed in the data processing worfkow.

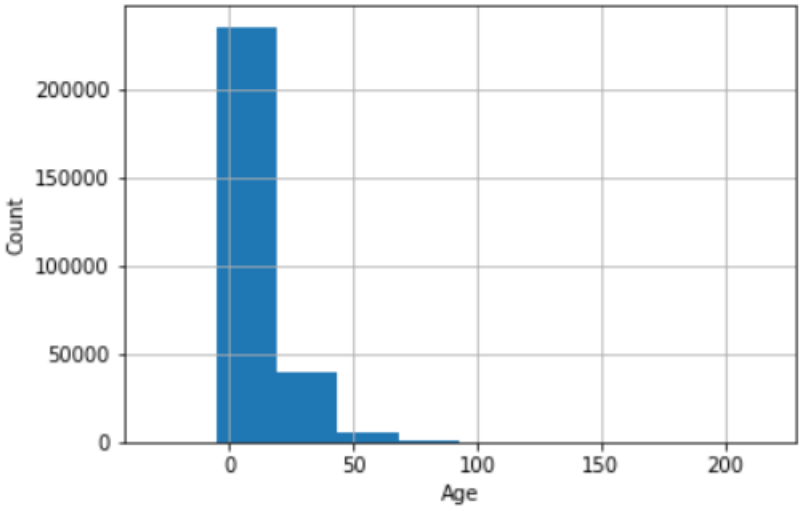

Figure 4 – Water Point Age Histogram

Algorithms and Techniques

This is a straightforward binary classification problem, but the large number of features are relatively small amount of data makes machine learning difficult. For example, the company that installed a water point is obviously relevant to its quality. But there are hundreds of different water point technologies, and thousands of different installers, 92% of which have built less than a dozen installations. It is impossible to learn anything about the quality of their work with so few data points. For this reason, all cell values that show up in the database less than 50 times (in the water_tech, water_source, and installer columns) will be reclassified at Other. This one step reduces the decision space from 5394 dimensions to 306.

A decision tree classifier was used because it is able to deal efficiently with high-dimensional classification problems. Decision trees are built from nodes and leaves. Each node performs a single operation: comparing a feature to a threshold, and sorting the output. The different outputs are sent to different nodes, and when an output can no longer be split a decision (aka leaf) is reached. [1] To start building a tree, the learner iterates through the features, finding the one that can best split the data. Then you do this again for the data in each of the child nodes.

A logistics regression will be tried as well, because it runs fast and doesn't overfit as easily as a decision tree. Logistic Regressions are used for data with high variance, because it is a simple model that won't overfit the data. They work using gradient descent to minimize the cross-entropy loss function.[2]

Finally, an AdaBoost Classifier will be tried because it can reduce the error due to bias of our decision tree without overfitting and causing more the error due to variance. It does this by fitting multiple decision trees to the training data, each time increasing the weight of data points that were classified wrong in the last iteration.[3]

---

[1] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.

[2] Cox, DR (1958). "The regression analysis of binary sequences (with discussion)". J Roy Stat Soc B. 20: 215–242. JSTOR 2983890.

[3] Friedman, Jerome; Hastie, Trevor; Tibshirani, Robert (1998). "Additive Logistic Regression: A Statistical View of Boosting". CiteSeerX 10.1.1.51.9525 Freely accessible.

A Support Vector Machine is also able to efficiently classify high-dimensional data, but it cannot efficiently determine prediction probabilities, making it inappropriate for this analysis for reasons that will soon become clear.

Benchmark

The simplest approach to solving this problem is to calculate the average age of a broken water points, which turns out to be 9 years old. If we assume any water point older than this is broken, we have a classifier with Recall = 0.5, accuracy = 0.6 and F2-score = 0.42 (Fig 5).

However, even using a simple age-based model like this, it is possible to achieve a higher F2-score. By assuming that all water points are broken, the accuracy drops to 28%, but 100% recall gives this model F2-score = 0.6 (Fig 5). The relationship between cutoff age, accuracy and various F-scores is shown here:

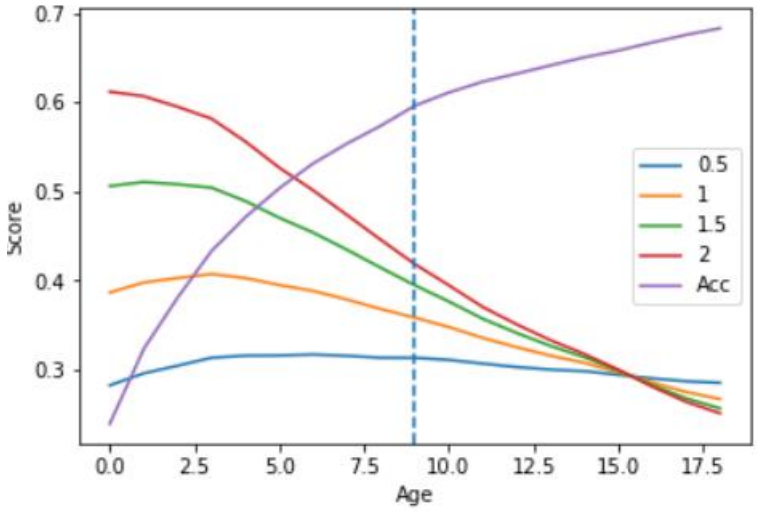

Figure 5 – Evaluation Metrics of Benchmark Model

This shows that it is relatively easy to build a model that gets high accuracy or high recall. My hope is that, by taking characteristics other than age into account, it will be possible to build a classifier that has both accuracy and F2-score higher than 0.6.

Methodology

Data Preprocessing

The main goal of this data preprocessing workflow is to reduce the number of unique values in our data's text fields, thus reducing the overall dimensionality of the final training dataset (Fig 7). Key steps are as follows:

1. Capitalize the text fields, so that entries like 'World Vision' and 'world vision' are no longer treated as different.
2. Calculate the age of each water point by subracting the install_year from the report_year.
3. Removing all rows where status_id is unkown, install_year is unkown, or age is calculated to be negative leaves 281154 water points.
4. Reclassify all null cells as NOT RECORDED. All other versions of null (None, unkown, don't know, etc.) are replaced with NOT RECORDED as well.
5. Any installer, water_source, or water_tech that is referenced less than 50 times in the database is reclassifed as Other.
6. Special attention is paid to the "Pay" field:
    - All forms of No (Never, no payment, 0, free, etc.) are binned together.
    - Any number other than 0 is reclassified as Yes.
    - 'Yes but only in case of breakdown', 'Only if there is a breakdown','Pay when scheme fails', and 'Only after system breakdown' are all reclassified as "On Breakdown.

Before:

	water_source	water_tech	pay	management	installer	install_year
Type	object	object	object	object	object	float64
Completeness	62.1938	89.2313	44.6991	40.9729	44.3735	82.4318
Unique	196	514	372	13	5239	131

Figure 6 – Summary of Attributes

After:

	water_source	water_tech	pay	management	installer	install_year	age_years
Type	object	object	object	object	object	int64	int64
Completeness	100	100	100	100	100	100	100
Unique	59	104	5	10	166	131	154

Figure 7 -Summary of Attributes After Data Preprocessing

Implementation

The Python module Scikit-Learn[4] is used to split the records into training and test datasets, which are used to fit a Decision Tree classifier, an AdaBoost classifier, and a Logistic Regression. Each one was trained on 1% of the training dataset and 10% of the training dataset before using all the points, in order to better understand how the algorithms learn. Default hyperparamerers were used for the initial run. The Decision Tree classifier was selected for refinement because it fit the data well without overfitting. Once trained on the full training dataset, the decision tree is able to classify water points in the testing dataset with 81% accuracy, but the F2-score is only 0. 516. Then the training data was split with Scikit-Learn's Grid Search algorithm to compare different values for max_depth and min_samples_split. The Grid Search showed no improvement in classification skill with regularization, suggesting that the algorithm is not suffering from overfitting.

Refinement

The problem seems to have more to do with error due to bias. In this case, we have high precision (0.7), but low recall (0.48). The goal of this project is to have high recall, so that no towns go without clean drinking water. With that in mind, I changed the script so that instead of using the classifier's predict() method, I use predict()_proba to get the probabilities, and classify any point with more than 20% chance of being broken as non-functional. This reduces precision to 0.48, but increases recall to 0.74 (Fig 9). The overall F2-score is 0.67, which is in the range I was hoping to achieve.

```
Unoptimized model
------
Accuracy score on testing data: 0.8264
F-score on testing data: 0.5169
Precision score on the testing data: 0.6967
Recall F-score on the testing data: 0.4855

Optimized Model
------
Final accuracy score on the testing data: 0.7419
Final F-score on the testing data: 0.6677
Precision score on the testing data: 0.4748
Recall F-score on the testing data: 0.7431
```

Figure 9 – Evaluation Metrics for Decision Tree Classifier on Testing Dataset

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Results

Model Evaluation and Validation

Afganistan

One of the most prolific contributors to the WPDx, Afghanistan has 51,284 water points in the database, all submitted by the Danish Committee for Aid Afghan Refugees. Management and payment method are not recorded, which reduces the amount of available information, but also reduces the dimensionality of the classification problem. The decision tree classifer was able to predict when water points are broken in the testing dataset with 68% accuracy and a 0.68 F2-score. The most useful features turned out to be water_source, adm1, and water_tech. This is similar to the result we had on the global dataset.

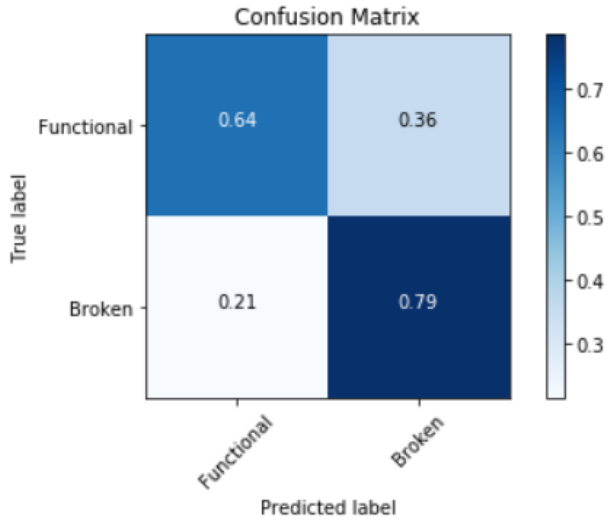

Figure 10 – Confusion Matrix for Classification in Afghanistan

Liberia

In Liberia there are 34,281 documented water points, most submitted by WASH Liberia. However, only 30% of them have their install_year recorded. This means most of the points were thrown out, leaving very little data for our classifer to learn from. Accuracy on the testing dataset was 59% and the F2-score is only 0.4889, based almost entirely on who the installer is. I tried predicting their status using the classifier that was trained

on all seven datasets combined. It was able to perform almost as well (Acc: 0.52, F2: 0.46), but does not appear to have learned anything useful from the other six datasets.

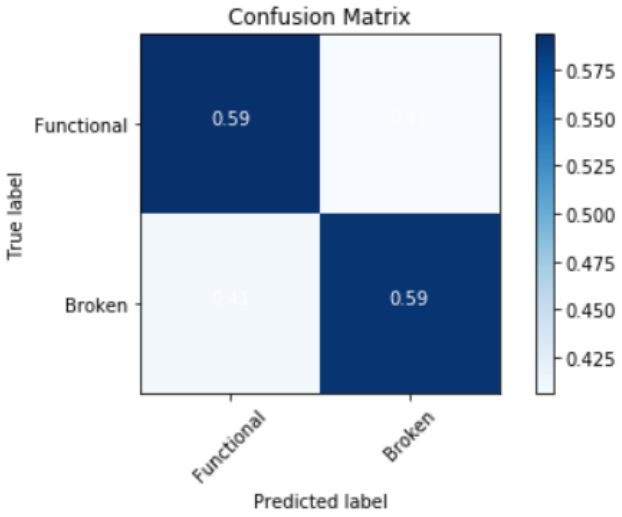

Figure 11 – Confusion Matrix for Classification in Liberia

Sierra Leone

Sierra Leone is one of the best documented countries, with 47,220 points in the database. The learner was extremely effective in this region, classifying water points in the testing dataset with an accuracy of 89% and F2-score = 0.81. Those are some of our best results, and they are almost entirely based on who the installer was. Oddly, this is the region in which the globally-trained classifier was least effective, with only 61% accuracy and an F2-score of 0.21

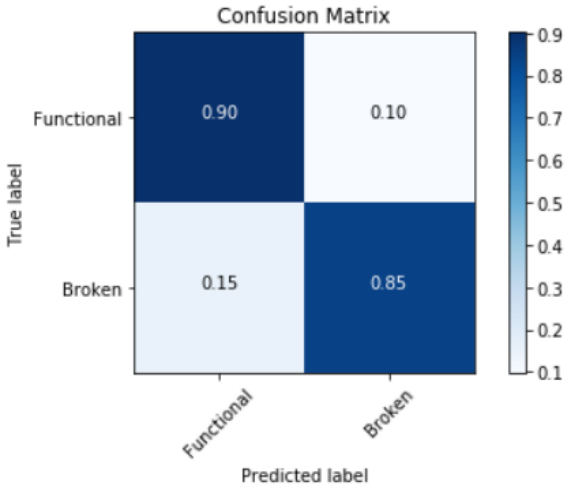

Figure 12 – Confusion Matrix for Classification in Sierra Leone

Swaziland

The best results we were able to achieve are in Swaziland: 92% accuracy on the testing dataset and an F2-score of 0.85. This is partly because Swaziland has a very well standardized terminology for cataloguing water points, leading to there being fewer unique values in the database. In Swaziland the classification problem only has 33 features for 22,969 data points – definitely enough to learn from. It also helps that Swaziland has the best maintained water distribution. Over 85% of Swaziland's water points are functional, which probably makes it easier for our classifier to find the broken ones.

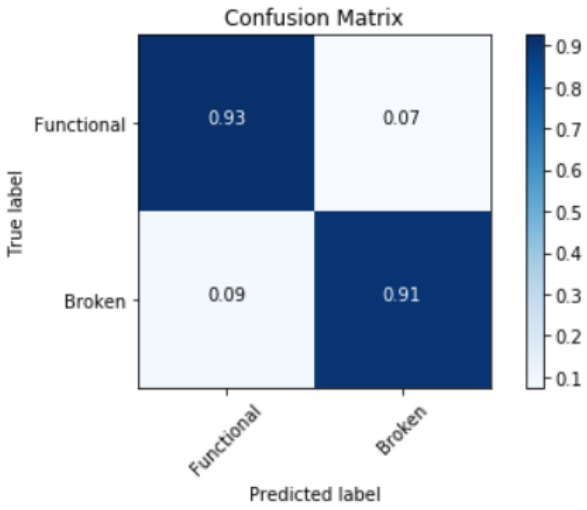

Figure 13 – Confusion Matrix for Classification in Swaziland

Tanzania

Tanzania has very thorough records on its water points – every attribute except 'management' is recorded for almost every point. The learner is able to classify points in this region's test dataset with 72% accuracy and an F2-score of 0.78. Install_year and water_tech turned out to be the most important features here, which is different than what we saw in other regions.

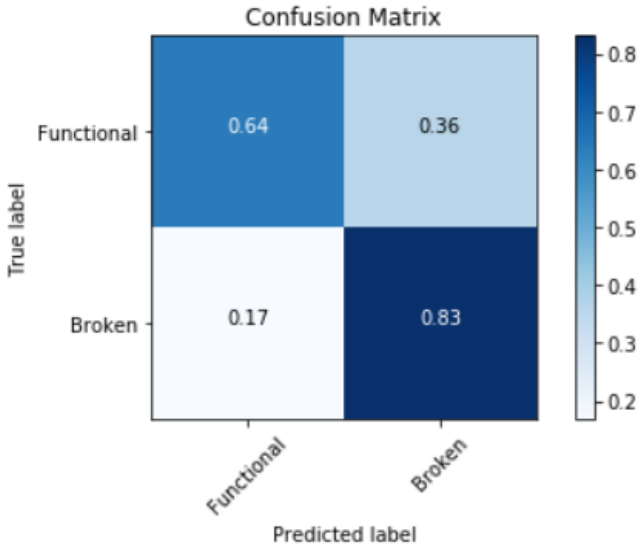

Figure 14 – Confusion Matrix for Classification in Tanzania

Uganda

Uganda has the most water points of any country in the WPDx database: 114,722. Unfortunetely, it also suffered most from the curse of dimensionality, with 169 features left even after the data preprocessing workflow. This made learning difficult, and the classifier was only able to achieve 70% accuracy on the testing dataset and an F2-Score of 0.57. This might have something to do with the fact that 'Installer' turns out to be the most useful feature, even though only 1% of water points here have that feature documented. Other than that the classifier mostly just works based on age.

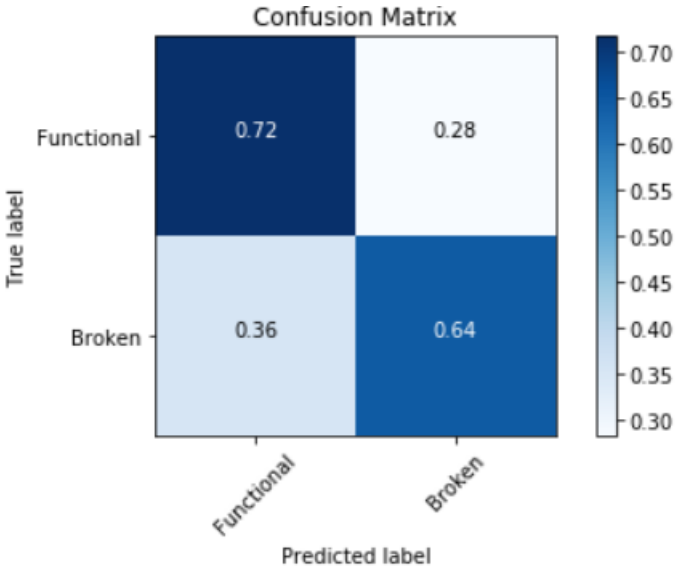

Figure 15 – Confusion Matrix for Classification in Uganda

Zimbabwe

Zimbabwe also did not document who installed each water point, and this seems to be limiting the learner. In many regions Installer was the most useful feature. It only classifies with an accuracy of 63% on the testing dataset with an F2-score of 0.5, pretty much based only on the age of the water point, which explains why the metrics are so close to that of our benchmark model.

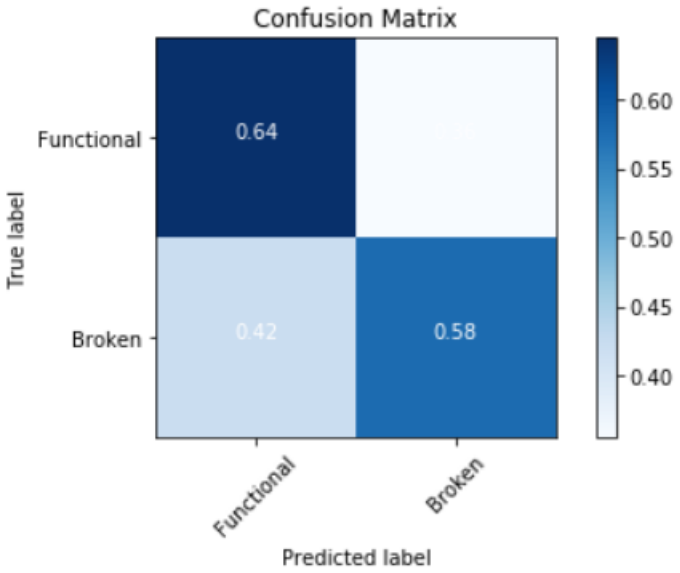

Figure 16 – Confusion Matrix for Classification in Zimbabwe

Justification

The benchmark model was only able to achieve an accuracy of 60%, with F2-score equal to 0.42. It was possible to nudge the F2-score as high as 0.6, but only by dropping the accuracy down to 28%. Our model was able to achieve an F2-score of 0.67, without sacrificing accuracy. In fact, it labeled broken water points with 74% accuracy. Recall is also about 74%, so most broken points will be inspected, and precision is 47%, so about half of all water points inspected will actually be broken. It is important to be thorough, but it is also important to allocate resources efficiently, and this model seems to me a good compromise between the two.

	Benchmark Model	ML Model
Accuracy	0.6	0.74
Precision	0.3	0.47
Recall	0.47	0.74
F2	0.42	0.67

Figure 17 – Comparison with Benchmark Model

Conclusion

Reflection

For many of the individual countries, Installer turned out to be the most useful feature for classification. However, this was not the case for our global learner trained on all 7 datasets:

feat	adm1	age years	install year	installer	management	pay	water source	water tech
imp	0.216159	0.11219	0.150601	0.068052	0.037803	0.080465	0.206229	0.128501

Figure 18 – Feature Importance

It's a pretty even split between water_source, adm1, and the age of the water point. Management and pay, which I expected to be extremely predictive, turned out not to be useful at all (Fig 18). This is probably because so few submitters recorded these data.

This analysis show definitively that attributes about how a water point was built, and most importantly who built it, are useful in estimating its lifespan. In every single region I was able to train a classifier that works better than the benchmark model, which is based on age alone. The results are similar to those that the team at Berkeley was able to achieve.[5]

However, the purpose of this study was to see if training the leaner of multiple countries worth of data would continue to improve the performance. At first if might appear so - the classifier that was trained on all seven datasets performed well. However, in not a single region did the globally-trained classifier perform better than a learner trained on data from that region alone. So it appears that learning from data in other countries not improve performance locally.

[5] http://159.89.145.240/solution

Improvement

The terminology used to describe water points in not consistent internationally. For example, in Swaziland they say ‘Borehole fitted with manual pump’ and in Sierra Leone it’s just ‘Pump on borehole.’ With much more meticulous feature engineering, it might be possible to standardize all the terminology and build a learner that can learn across national borders.

Other than this, the main source of error is due to bias, so it’s possible that a more powerful algorithm could better model the data. Neither Random Forest or AdaBoost was able to improve on basic Decision Tree, but its possible something like XGBoost or LightGBM could.