Alexia Marcous
Springboard Data Science Career Track
August 5th2019 Cohort

# Capstone Project 2 – Predicting Water Pump Status
# Summary Report

## Importance of Topic

According to the 2017 "Tanzania Economic Update" report from the World Bank, only 63% of the population has access to basic and improved water supply services[1], making the functioning of those services vital to the health of millions of people. The same report also said that "Tanzania needs to urgently improve the management of its water resources" to prevent disruption of the country's economic progress and sustain growth rates "long enough to make a significant dent on poverty".[2]

To help improve the management of water resources, a machine learning model will be created using supervised, classification algorithms to predict water pump status as functional, functional needing maintenance and non-functional. The model will also help identify which features contribute most to the prediction.

This model could be used by the Ministry of Water to make their maintenance schedule more efficient and effective. By knowing which pumps are more likely to break down, they can prioritize those pumps for maintenance. This model can also help the Ministry learn what are the factors that are important in determining the pump status, in order to start to address the root problems of pump malfunction.

## Data Source

This project will use data collected from the Tanzanian Ministry of Water, which has been organized into datasets by DrivenData. These datasets were made available by DrivenData for a Kaggle competition and on their website, where they host competitions with humanitarian impact.

## Data Wrangling

The data was in 2 datasets - one containing the target and the other containing 41 features and 59,400 rows. The target feature, status_group, has 3 possible outcomes: 'functional', 'non functional' and 'functional needs repair'.

Visual inspection of the data showed there were two features with very high frequencies of the same data at 85%. These features were dropped. There were also two features that had duplicated data, and were also dropped.

Missing categorical data was evaluated. One feature was missing in almost half of the observations, and represented the name of the group that managed the pump. This feature was dropped. Missing observations in 5% of cases for a boolean feature were assigned randomly in the same frequency as the 95% of present values.

Percentage of zero values for numeric data was evaluated. A feature with 97% zeros was dropped, and other features were evaluated as to whether zero was a valid value, as with altitude (for pumps at sea level).

Three features contained survey-specific data, and were dropped as they did not add predictive value.

Datatypes were reviewed among the 31 remaining features, with 22 categorical and 9 numeric.

Pandas Profiling was run on this reduced dataset, and after the results were reviewed. Given that most of the features in this dataset are categorical, they will need to be one-hot-encoded for use by the machine learning models, which will create x(n-1) features for x features with n values. This makes reducing cardinality important. At the same time, there are many features with varying levels of granularity for the same data. For example, there are 6 features related to the location of the pump. The curse of dimensionality seemed impending.

Removing features and reducing cardinality will reduce the variability of the data, but this could lead to an over-simplified dataset, and excluding potential relationships and patterns in the data.

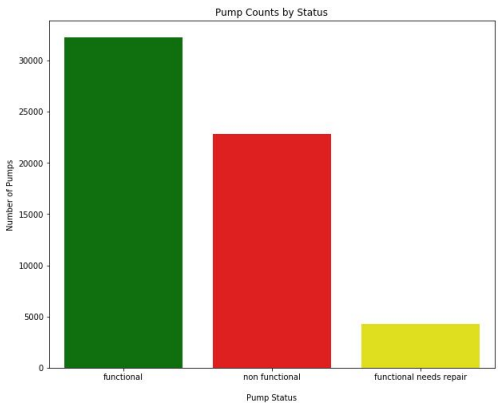
In an attempt to balance this tradeoff, two datasets were created. The first 'low-variance' dataset had a set of features with less granularity, and the second 'high-variance' dataset had features with more granularity. For example, the type of pump was represented in both datasets as 'extraction_type_class' with 7 unique values in the low-variance set, and 'extraction_type' with 18 unique values in the high-variance set.

Fortunately, this did not result in a large difference in the number of features. The low-variance dataset had 11 features, and the high-variance dataset had 13 features.

## Data Story

Visual data analysis techniques were applied to gain an understanding of the data. 4 main questions were asked:

1. **What is the distribution of pump status values?**

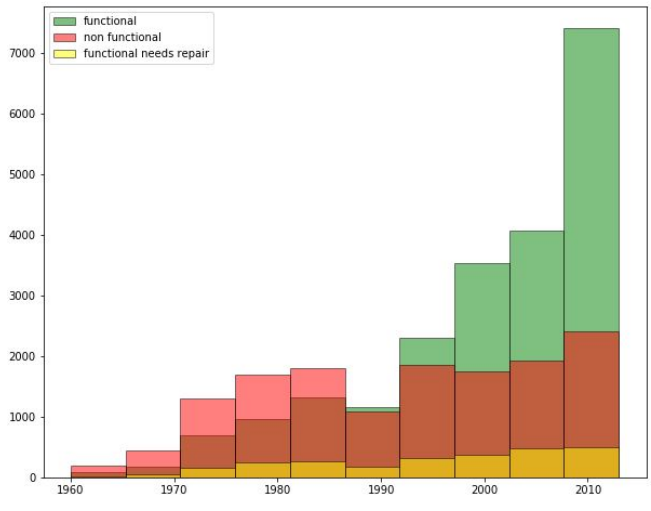
| | |
|---|---|
| As the straightforward bar chart shows, the distribution of pump status is very uneven.<br><br>54% are 'functional'<br>39% are 'non-functional'<br>7% are 'functional needs repair' |  |

2. **How does the pump status vary over features that would seem to have an impact on the functional status?**

   We will examine construction year, pump type, management, geographic location, altitude, payment_type, permit status, quantity, source, and waterpoint type?

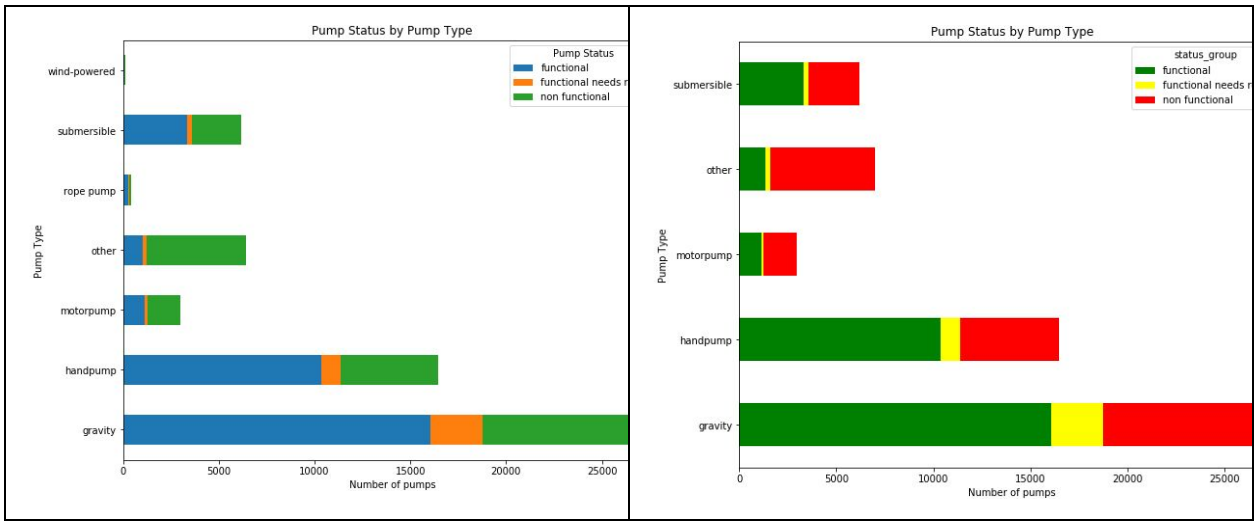
   As these features were being evaluated, opportunities for reducing cardinality were discovered. Highlights of the visualizations are presented below.

   **Pump status by construction year**

| | |
|---|---|
| From this histogram of pump status by year constructed, we can see that pumps built before 1990 are non-functional more than they are functional.<br><br>After 1990, the number of functional pumps increases, and the total number of pumps has increased dramatically.<br><br>Unfortunately, 30% of the pumps do not have a construction year recorded, so the predictive quality of this feature may not be very strong. |  |

### Pump status by pump type

The first visualization of pump types on the left revealed that the majority of the pumps were gravity pumps, and also that there were very few pumps in the 'wind powered' and 'rope pump' categories. To reduce cardinality, observations in these two categories were combined into the 'other' category, resulting in the distribution on the right.



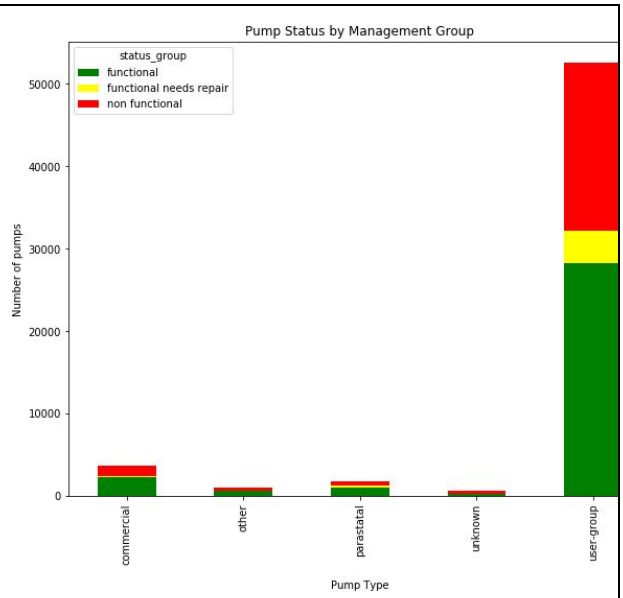

### Pump status by management group

At first glance, there didn't seem to be enough variability in the type of management groups to be of value for predictions.

However, the distribution of pump status is informative in terms of studying the impact of management on pump status. Among the most common type of management, user groups, there is high variability among the pump status. It would be interesting to learn best practices and support structures of the groups with functional pumps.

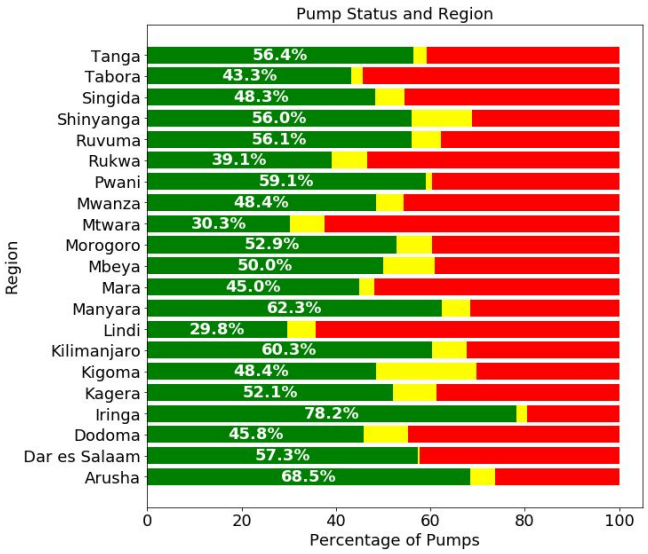
Also, despite the small number of observations, the commercial group has more than half of their pumps operating, making this another group to learn from.

The visualization also revealed an opportunity to reduce cardinality by combining the three smallest categories into the 'other' category.



3. **Which features should be chosen when there are multiple levels of granularity for the same data?**

   An example of this in this dataset, were 2 numeric, latitude and longitude, and 6 categorical features representing the geographic location of the pump: region, region_code, district_code, lga, ward, subvillage. Having so many features describing the same information, especially of the same datatype, introduces the idea of collinearity, which would have a negative on model results. Also, the lowest level of the hierarchy, subvillage, had 19,288 unique values, guaranteeing the curse of dimensionality. To address this, categorical geographic features at the highest level of granularity were visualized to determine the distribution of pump status across cardinality.

| | |
|---|---|
| Pump status by region, the top of the location hierarchy, showed that there were 21 regions, with a good amount of variability among pump status types.<br><br>Knowing that the 21 regions would be one-hot-encoded into 20 features, this feature was chosen for the low-variable dataset.<br><br>Region_code, with 27 different values, was chosen for the high-variable dataset, and latitude and longitude features were kept in this dataset. |  |

## Statistical Analysis

Given that the data from the water ministry is mostly categorical, and the goal of the project is multiclass prediction, the statistical analysis of the data was focused on testing for collinearity among features and between features and the target.

To remove any consideration of ranking, Cramer's V was run between all features, a nominal version of Pearson's Chi-Square Test for independence between categorical data. The test returns values between 0 for complete independence and 1 for complete dependence in terms of the effect a category for a feature has on the probability of a category for another feature occuring. Tests were run on the high and low-variance datasets.

These results on the low-variance dataset shows a strong correlation between basin and region. This makes sense, as pumps in the same region would draw from the same basin. There is also a strong correlation between extraction type class and waterpoint type, underscoring the repetition of some of the same data points in the two features.

Of all the features, region had the most correlations with other features. This suggests there are substantial differences among the regions of Tanzania, since as the region changes, so the other features change. This makes sense in a country as diverse as Tanzania, which contains the plains of Serengeti National Park, the mountains of Kilimanjaro National Park, and a coastline close to tropical islands.

Of interest, there is also a correlation between the extraction type class and the source, revealing that certain types of pumps are used in certain types of sources. This is probably common knowledge among pump installers, but it takes statistical analysis to identify that point to those with less contextual knowledge about the data.

The analysis of correlation on the high-variance dataset also shows a strong correlation between basin and region, this time using a feature, region_code, that had more variability. Interestingly, the correlations between region and other features are present but not as strong as in the low-variance dataset.

The same phenomenon appears with extraction type and waterpoint type. The correlation is present but not as strong as in the low-variance dataset.

This change in correlation might imply that the low-variance dataset would be the better dataset for creating the machine learning models.

With regard to correlations between the features and the target, in the high-variance dataset show slightly higher correlation to the target than the features in the low-variance dataset. This implies that the high-variance dataset would be the better set for machine learning. Both sets will be tested and compared in the machine learning phase.

## Examining Significance of Data Story Relationships

The visual data analysis revealed potential relationships between pump status and year, pump type, management group and region. The following table shows the Cramer's V test results for these features:

| Feature | Cramer's V low-variance | Cramer's V high-variance |
|---|---|---|
| Year | 0.18 | 0.18 |
| Pump Type | 0.23 | 0.25 |
| Management Group | 0.045 | 0.045 |
| Region | 0.2 | 0.2 |

The test statistics do not show strong correlations, with all p-values near zero. This underscores the importance of following up visual data analysis with statistical analysis to determine if what looks like a relationship is actually statistically significant.
The symmetry of the correlations was evaluated with Theil's U test, also referred to as the Uncertainty Coefficient. This test is based on conditional entropy - given the value of one feature, how many possible states does another feature have, and how often do they occur. The test output is in the range of [0,1], where 0 means no association and 1 is full association.

Only the 'quantity' feature demonstrated an asymmetrical relationship with a Theil's U value of 0.11. This value was the same in the high and low-variance datasets.

After wrangling, visualizing, and performing statistical analysis on the dataset, it does not appear that there are very strong correlations between any of the features and the target variable. However, machine learning algorithms may be able to uncover as-yet-unseen relationships, and having clean, organized data is the right preparation for this next step.

## Machine Learning

### Choosing Algorithms

Algorithms were chosen based on the properties of the data and target variable. The data from the Tanzanian Ministry of Water prepared by Driven Data was formed into two datasets that were labeled and provided the target variable, pump status, making this a supervised learning problem. The target variable has three categories, 'functional', 'functional needs maintenance' and 'non functional. With more than two classes, this is a multi-class classification problem. Five algorithms were chosen for this problem type, including K-Nearest Neighbors (KNN), Logistic Regression, Random Forest Classifier, Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGBoost).

### Choosing Evaluation Metrics

Evaluation metrics were chosen based on the distribution of the data over the target, and the implication of false predictions. In this case, the data is imbalanced, with 54% of the pumps in 'functional' status, 39% in 'non functional' status and only 7% in 'functional needs maintenance' status. For this reason, F1 score was chosen over accuracy.

The F1 score was also appropriate due to a difference in the cost of false positives (predicting a functional status when the pump is not working) and false negatives (predicting a non functional status when the pump is working). A false positive could mean that the pump is ruled out for maintenance, resulting in it remaining in a non functional status, potentially causing pain and suffering in the local population. For this reason, the appropriate evaluation metrics are the true positive rate, also known as sensitivity or recall, and precision, or how "precise" the classifier is when predicting positive instances. Since the F1 score is the weighted average of precision and recall, if the F1 score is high, both precision and recall of the classifier indicate good results.

## Preprocessing

To prepare the data for the machine learning algorithms which can only analyze numeric data, categorical values for all features of type object were converted to numeric values with the pandas factorize method. Since the categorical values in the dataset were ordinal, meaning they had no ranking, the data needed to then be one-hot-encoded. This was done with the pandas get_dummies method, specifying that n-1 features be created for n values in order to avoid creating problems of collinearity. The number of features in the low-variance dataset went from 11 to 53. The number of features in the high-variance dataset went from 14 to 56.

Once the data was encoded, it was split into training and test sets, using an 80/20 split. Each set was then scaled with the StandardScaler from scikit learn.

## Model Creation and Prediction

There were 4 steps to the model creation and prediction process.

1. The 5 models were run on the low-variance dataset. Feature importances were evaluated for performance improvement.
2. The 5 models were run on the high-variance dataset. Feature importances were evaluated for performance improvement.
3. Using the dataset that produced the best results, the 5 models were optimized with cross-validation and hyperparameter tuning.
4. Using the best dataset and tuned models, principal component analysis was performed for performance improvement.

## Step 1: Low-Variance dataset

**Model results**

The algorithms were run with default parameters on the low-variance dataset . The resulting F1 scores were:

| F1 scores | 0 - Functional | 1 - Non Functional | 2 - Functional Needs Repair |
|---|---|---|---|
| KNN | 0.78 | 0.65 | 0.13 |
| Logistic Regression | 0.77 | 0.65 | 0.05 |
| Random Forest | 0.82 | 0.77 | 0.31 |
| AdaBoost | 0.78 | 0.66 | 0.04 |

| | | | |
|---|---|---|---|
| XGBoost | 0.80 | 0.68 | 0.12 |

Random Forest was the top performer on the low-variance dataset with default model parameters.

**Feature Importance**

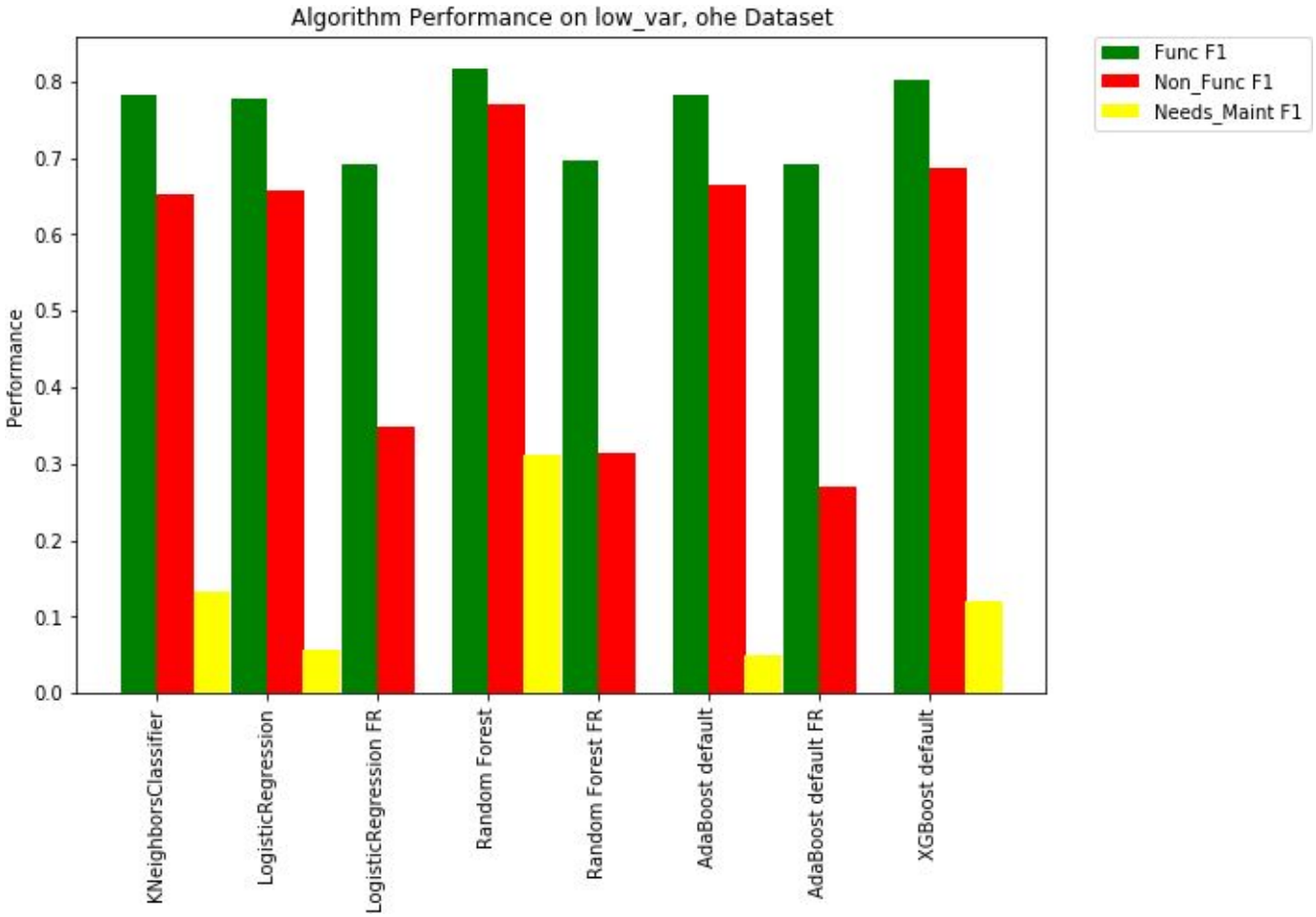
In an attempt to improve these results, feature importances were evaluated where models provided them. Features with coefficients less than a model-specific threshold were dropped from the data and the models were re-run. Of note, different features were dropped for different models. The resulting F1 scores were as follows, with changes from the full-featured dataset noted:

| | Feature drop threshold | # Features dropped | 0 - Functional | 1 - Non Functional | *2 - Functional Needs Repair |
|---|---|---|---|---|---|
| Logistic Regression | < 0.05 | 14 | 0.69 -0.08 | 0.34 -0.31 | 0 |
| Random Forest | < 0.005 | 17 | 0.69 -0.12 | 0.31 -0.46 | 0 |
| AdaBoost | = 0 | 14 | 0.69 -0.09 | 0.26 -0.40 | 0 |

* each model produced an undefined metric warning indicating there were no predicted samples for the 'Functional Needs Repair' class.

All of the models had a drop in performance following feature reduction. This could be because the thresholds were too low, something that can be evaluated in future work on this and other projects.

**Model performance on low-variance, one-hot-encoded dataset summary**




## Step 2: High-Variance dataset

**Model results**

The same algorithms were run with default parameters on the high-variance dataset. The resulting F1 scores were as follows, with changes from the low-variance dataset results noted:

| F1 scores | 0 - Functional | 1 - Non Functional | 2 - Functional Needs Repair |
|---|---|---|---|
| KNN | 0.78 | 0.63 -0.02 | 0.15 **+0.18** |
| Logistic Regression | 0.78 **+0.01** | 0.66 **+0.01** | 0.04 **+0.01** |
| Random Forest | 0.83 **+0.02** | 0.79 **+0.02** | 0.38 **+0.07** |
| AdaBoost | 0.78 | 0.67 **+0.01** | 0.09 **+0.05** |
| XGBoost | 0.80 | 0.69 **+0.01** | 0.14 **+0.02** |

Random Forest was also the top performer on the high-variance dataset with default model parameters. Of note, all of the models improved on predicting the 'Functional Needs Repair' status.

**Feature Importance**

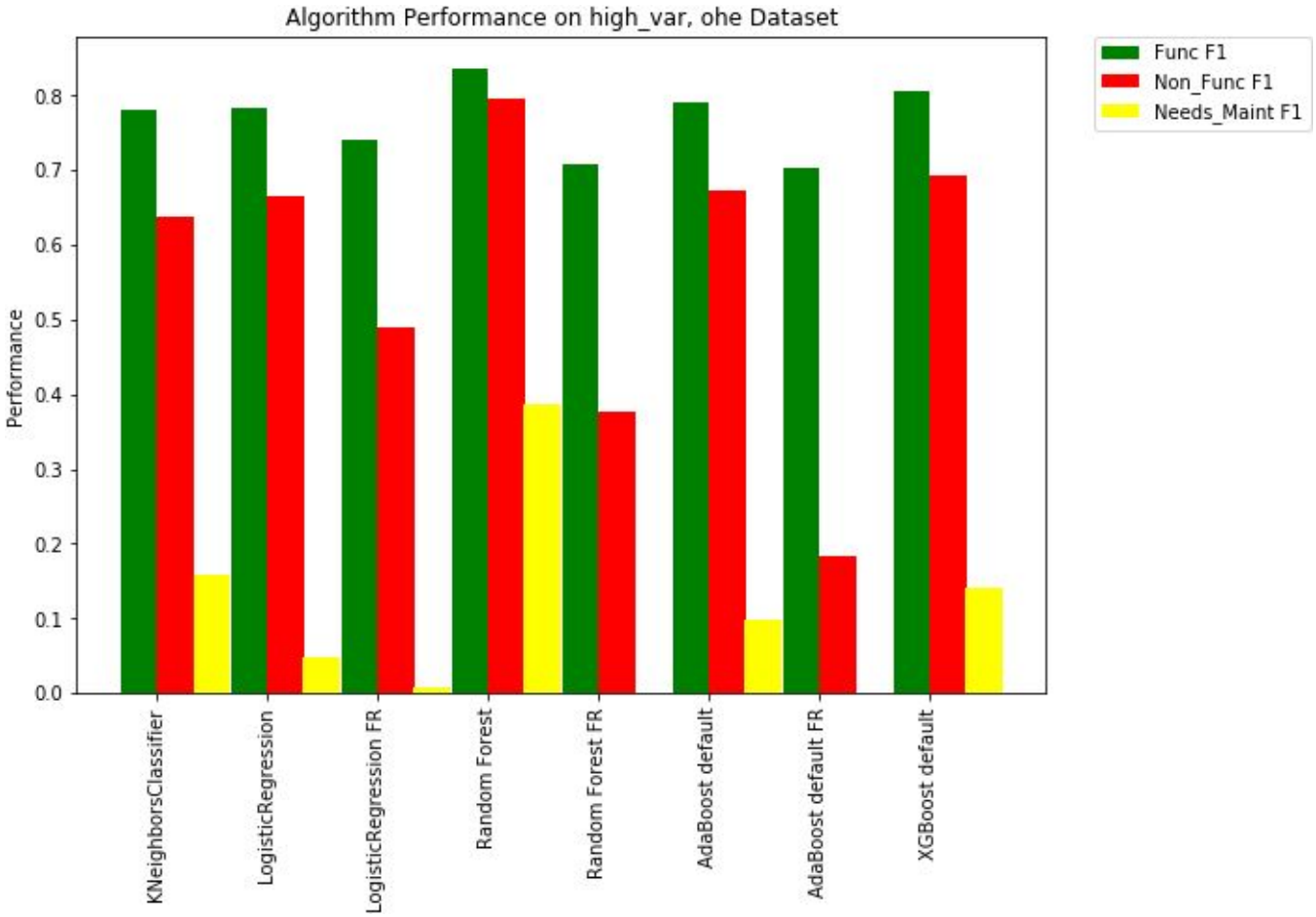
As with the low-variance dataset, an attempt was made to improve these results by evaluating feature importances where models provided them. Features with coefficients less than a model-specific threshold were dropped from the data and the models were re-run. Of note, different features were dropped for different models. The resulting F1 scores were as follows, with changes from the full-featured high-variance dataset noted:

| | Feature drop threshold | # Features dropped | 0 - Functional | 1 - Non Functional | *2 - Functional Needs Repair |
|---|---|---|---|---|---|
| Logistic Regression | < 0.05 | | 0.74 -0.08 | 0.48 -0.24 | 0.006 |
| Random Forest | < 0.005 | | 0.70 -0.13 | 0.37 -0.42 | 0 |
| AdaBoost | = 0 | | 0.70 -0.08 | 0.18 -0.49 | 0 |

* The Random Forest and AdaBoost models produced an undefined metric warning indicating there were no predicted samples for the 'Functional Needs Repair' class.

As with the low-variance dataset, all of the models had a drop in performance following feature reduction. This could also be because the thresholds were too low, something that can be evaluated in future work on this and other projects.

**Model performance on high-variance, one-hot-encoded dataset summary**




## Step 3: Cross validation and hyperparameter tuning

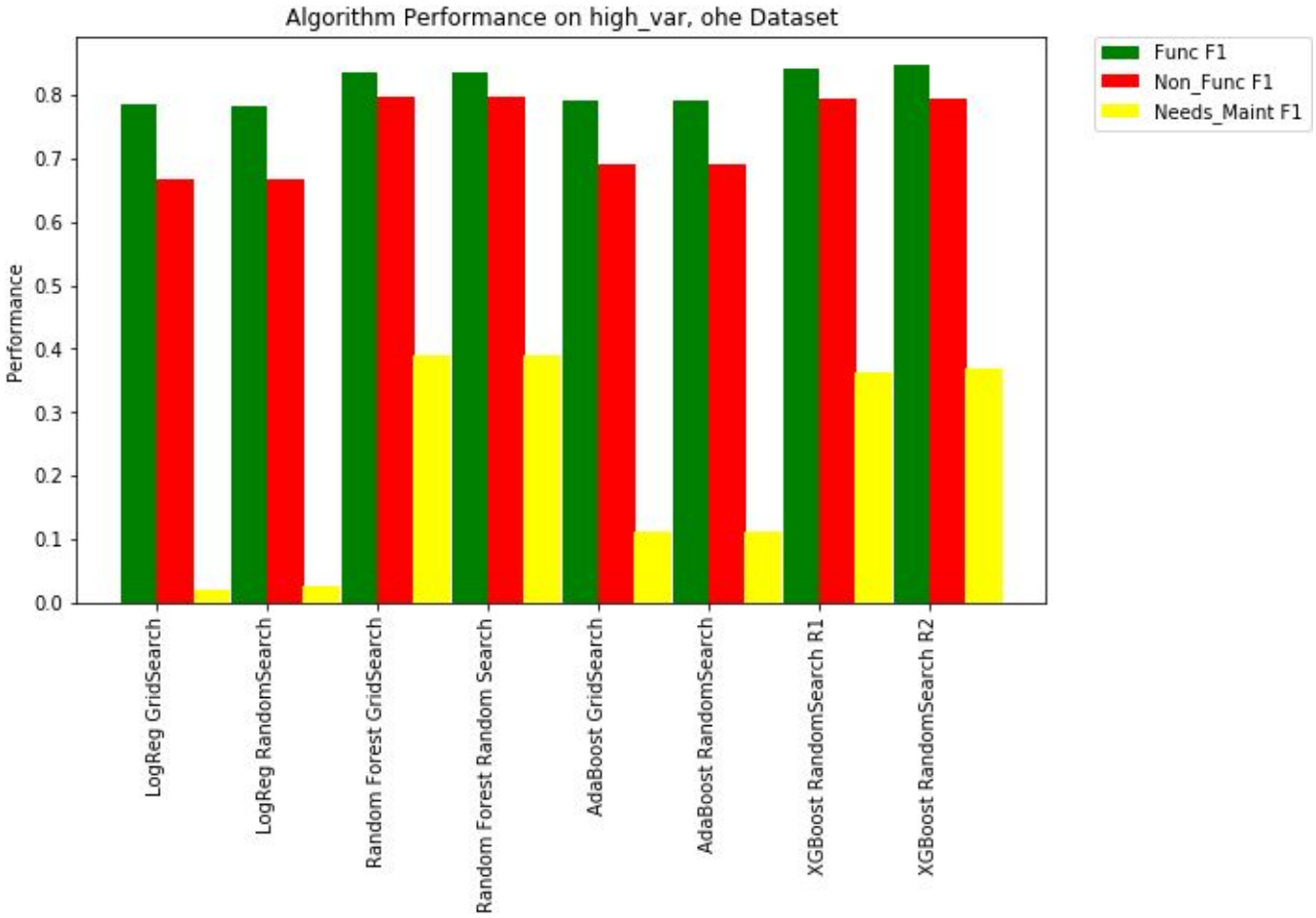
With the better of the two datasets identified, the next step towards improving the model results was to perform cross validation and hyperparameter tuning on the top 4 highest performing models. 5-fold cross validation ran the models through more training and testing scenarios by subsetting the existing data into new test and training sets. Hyperparameter tuning tested multiple scenarios of inputs to the learning process. Both grid search, an exhaustive exploration, and random search, a partial exploration of the matrix of possible hyperparameters were run on the logistic regression, random forest and Adaboost models. Given that hyperparameter tuning on XGBoost required multiple rounds due to the number of parameters available for tuning, only random search was performed for expediency. The results were as follows:

| F1 scores | Search Type | 0 - Functional | 1 - Non Functional | 2 - Functional Needs Repair |
|---|---|---|---|---|
| Logistic Regression | Grid | 0.78 | 0.66 | 0.02 -0.02 |
| Logistic Regression | Random | 0.78 | 0.66 | 0.02 -0.02 |
| Random Forest | Grid | 0.83 | 0.79 | 0.38 |

| | | | | |
|---|---|---|---|---|
| Random Forest | Random | 0.83 | 0.79 | 0.38 |
| AdaBoost | Grid | 0.79 +0.01 | 0.68 +0.01 | 0.11 +0.02 |
| AdaBoost | Random | 0.79 +0.01 | 0.68 +0.01 | 0.11 +0.02 |
| XGBoost Round 1 | Random | 0.84 +0.04 | 0.79 +0.10 | 0.36 +0.22 |
| XGBoost Round 2 | Random | 0.85 +0.05 | 0.80 +0.11 | 0.37 +0.23 |

Three rounds of tuning was planned for XGBoost, but the second round took 38 hours, so additional tuning was left for future work. XGBoost produced the highest-scoring model for the first two classes, and Random Forest remained in the lead for the 'functional needs maintenance' class.

**Model performance on high-variance, one-hot-encoded dataset with hyperparameter tuning and cross validation summary**



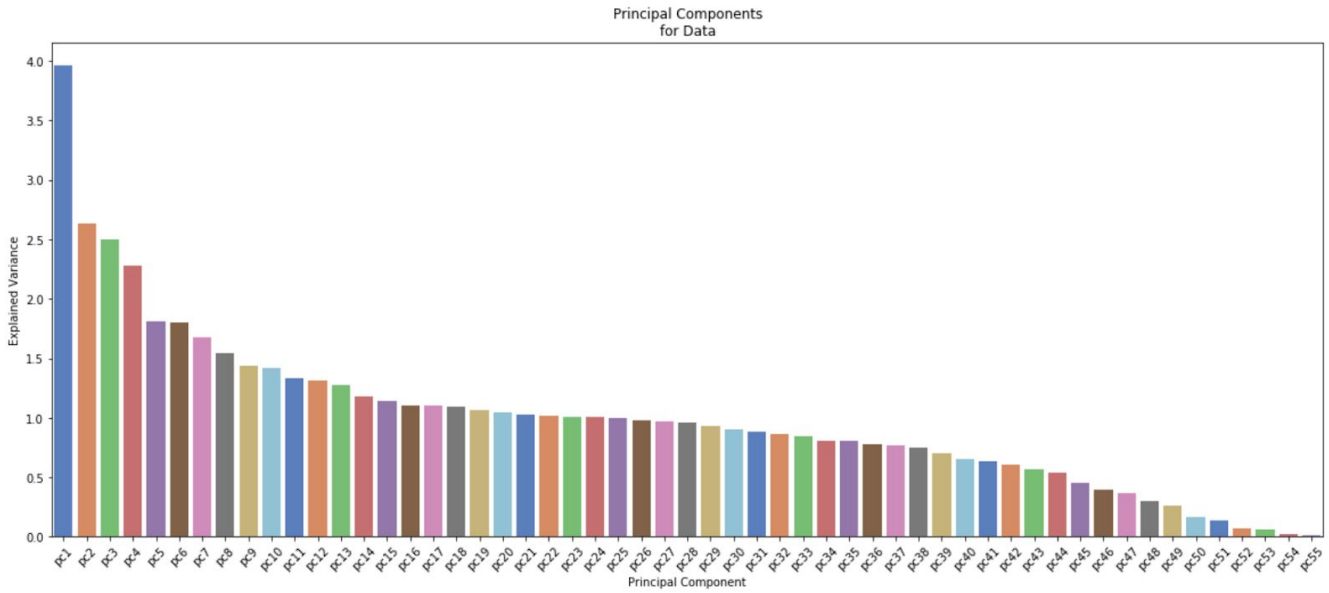

## Step 4: Principal Component Analysis

With the better of the two datasets and the best of the tuned models identified, the next and final step towards improving the model results in this project was to perform principal component analysis (PCA).

This is a dimension-reduction technique, aimed at including fewer relationships between variables in an attempt to prevent overfitting by capturing the variance of the data in as few components as possible.

The drawback to PCA is that it reduces the interpretability of the results. We no longer have an understanding of which features are relevant. This is not in alignment with the original goals of the project, but if more water pumps are kept operable by a better predictive outcome, that still achieves the overall goal.

A chart of the principle components ordered by explained variance shows the results from fitting the PCA instance on the training data:




Test and training sets were reduced to 13 components, the first leveling-off point, indicating diminishing returns with additional components. Given the 38-hour duration of the best-performing model, the next-best performing classifier, the tuned RandomForest model, was used first to determine if there was any improvement to be gained using PCA.

The model ran quickly on the reduced data but the results are not as good as the model produced before, for F1 scores of all classes. This could be because the cutoff of principal components was too aggressive. The model was run again using 38 components, but produced results worse than with 13 components.

Since performing dimension reduction with feature extraction via PCA did not improve the results on the next-best model, we will not attempt it on the top-performing model, XGBoost. Given the impractical duration of tuning that model further, we will leave that for future improvements.

## Summary

This 4-step approach revealed that the high-variance dataset produced better results, which were not improved with feature reduction via exclusion by importance threshold. The results were improved with hyperparameter tuning and cross validation, although with drastic increase in run time. Principal Component Analysis improved run time, but sacrificed results.

Of note, the XGBoost model took 38 hours to run in the second round of hyperparameter tuning. Utilizing PCA for feature extraction did improve performance on the Random Forest model, but lead to sub par results.

No one model performed the best across all three target classes on measures of F1 scores. The following is a summary of the two top-performing models:

| | Random Forest | | | XGBoost | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| 'Functional' | 0.81 | 0.87 | 0.84 | 0.80 | 0.89 | 0.85 |
| 'Non Functional' | 0.82 | 0.77 | 0.80 | 0.83 | 0.77 | 0.80 |
| 'Functional needs maintenance' | 0.48 | 0.33 | 0.39 | 0.55 | 0.28 | 0.37 |

With the goal of helping to improve the management of Tanzania's water resources and ensure access to potable water for its citizens, we can evaluate these models by reviewing the recall scores to determine how helpful they are in terms of knowing when a pump is broken or needs maintenance. Of all the pumps that were actually non functional, both models predicted that correctly 77% of the time. Of all the pumps that needed maintenance, the Random forest model predicted that correctly 33% of the time, while the XGBoost predicted that correctly 28% of the time. Based on this evaluation criteria, we can declare the Random Forest model to be the most useful.

## Future Work

Future work on this project could include bringing in data from other sources to fill in missing data, such as population, or augment the data, such as annual rainfall. It would be interesting to review other preventive maintenance programs to learn best practices. Work could also be done to help address the imbalance in the target classes, such as bootstrapping the data, using weights in model creation, or evaluating the impact of combining classes. Additional tuning could also be done to attempt to improve results further.

References

1. "Water Stress Could Hurt Tanzania's Growth and Poverty Reduction Efforts – New World Bank Report", accessed on 1/3/2020 at https://www.worldbank.org/en/news/press-release/2017/11/06/water-stress-could-hurt-tanzanias-growth-and-poverty-reduction-efforts---new-world-bank-report

2. Ibid.