Predicting Water Pump Status

Capstone 2 Project
Alexia Marcous

Access to water in Tanzania

- Only 63% of the population has access to ‘basic and improved’ water services

- Optimizing functional status of those services is vital to the health of millions

Access to water in Tanzania

- Improvement of the management of these resources is critical to maintaining enough economic growth to overcome poverty

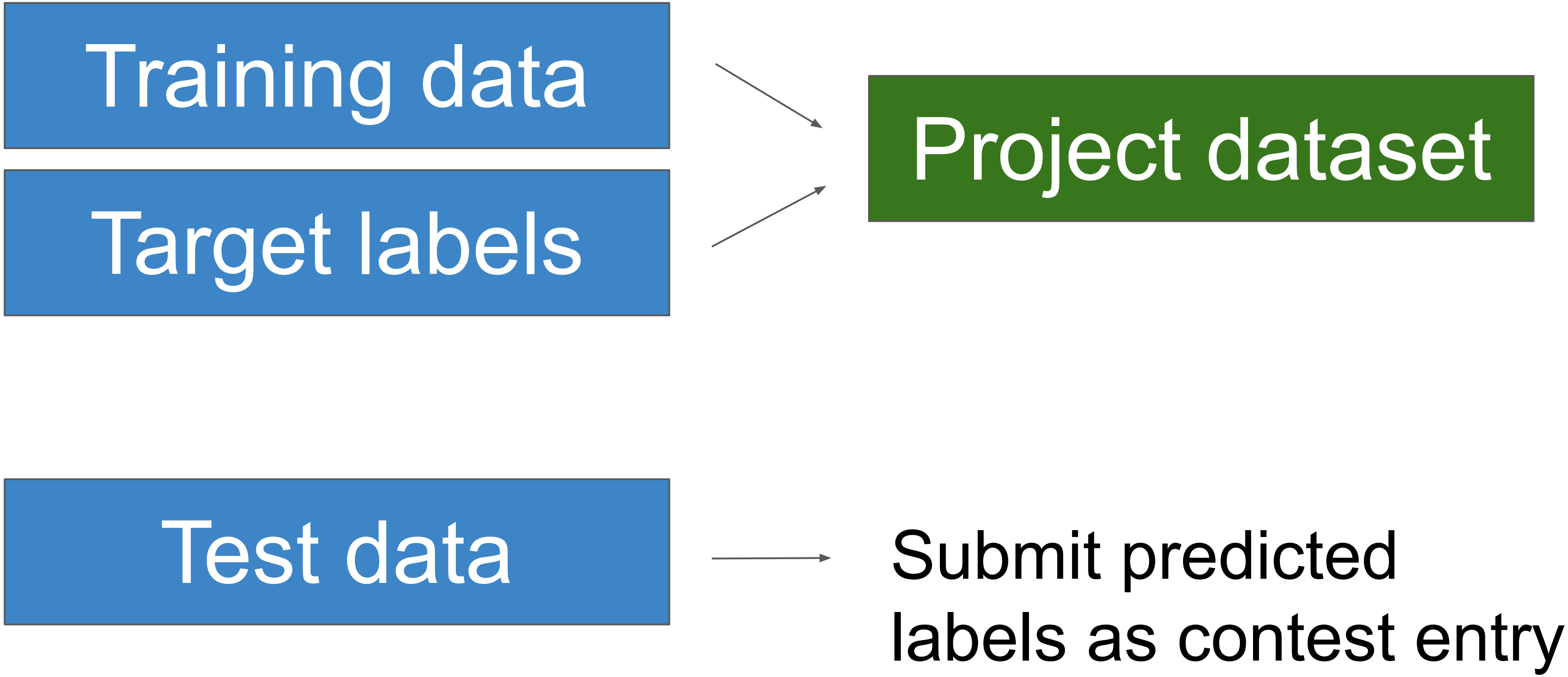
Need for prediction

- Water pumps have diversified locations, management, equipment and usage
- This makes maintenance disjointed and expensive
- Being able to predict what pumps are likely to break down can lead to
    - Better choices in pump equipment
    - More efficient and effective repair of malfunctioning pumps
    - The ability to perform preventative maintenance, reducing pump failure
    - A clearer understanding of the root causes of malfunction - the most important thing to address

Data Source

- Tanzanian Ministry of Water
- Data was organized into datasets by DrivenData
    - host competitions with humanitarian impact
    - Competition is available on their website and Kaggle
- Data downloadable from Amazon Web Service url

Data Source



Training data

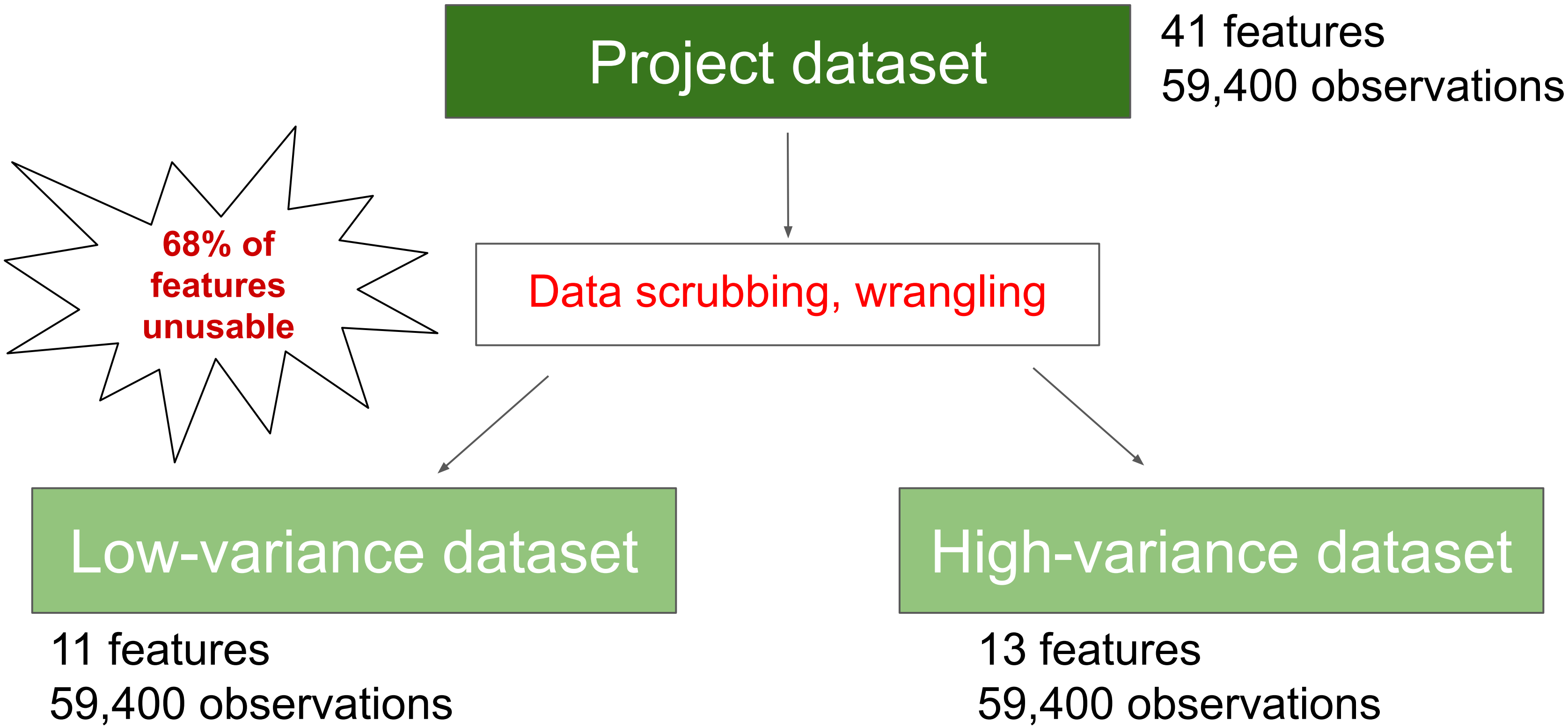
Target labels

Project dataset

Test data

Submit predicted labels as contest entry

Data Wrangling



Project dataset

41 features
59,400 observations

Data scrubbing, wrangling

68% of features unusable

Low-variance dataset

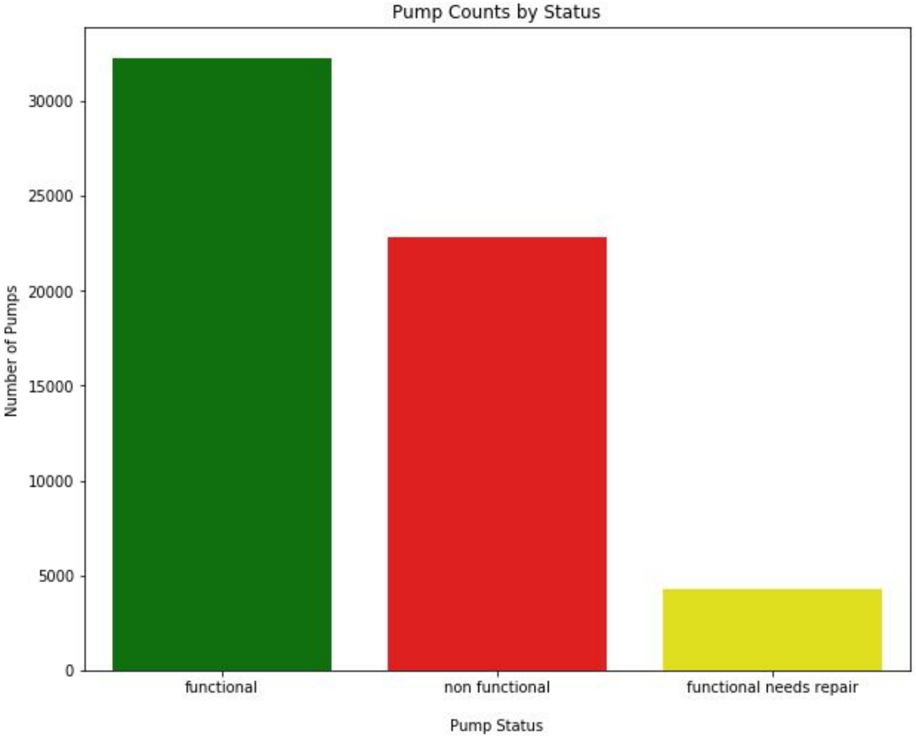
11 features
59,400 observations

High-variance dataset

13 features
59,400 observations

Exploratory Data Analysis - pump status distribution

54% ‘functional’
39% ‘non-functional’
7% ‘functional needs repair’

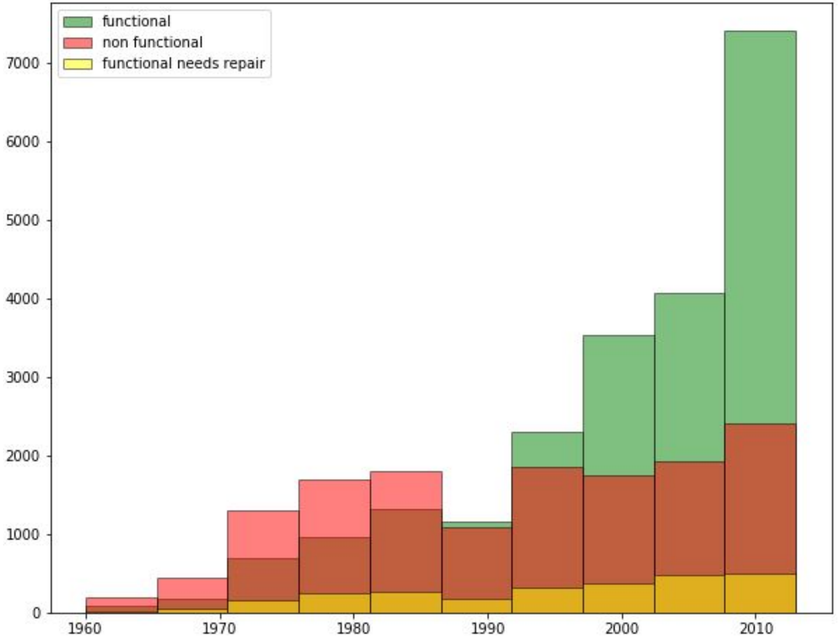


Exploratory Data Analysis - pump age

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.

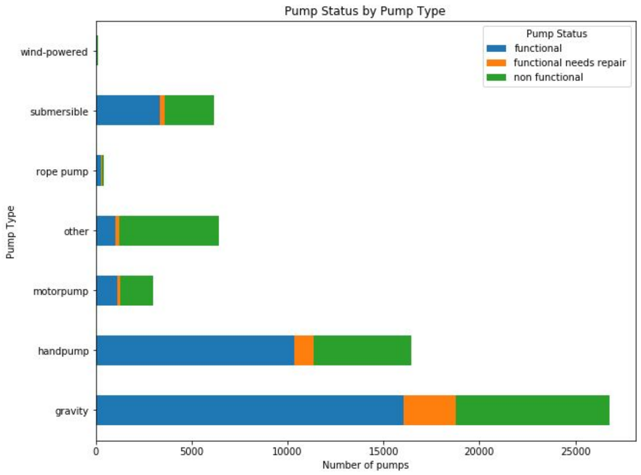
After 1990, the number of functional pumps increases, and the total number of pumps has increased dramatically.

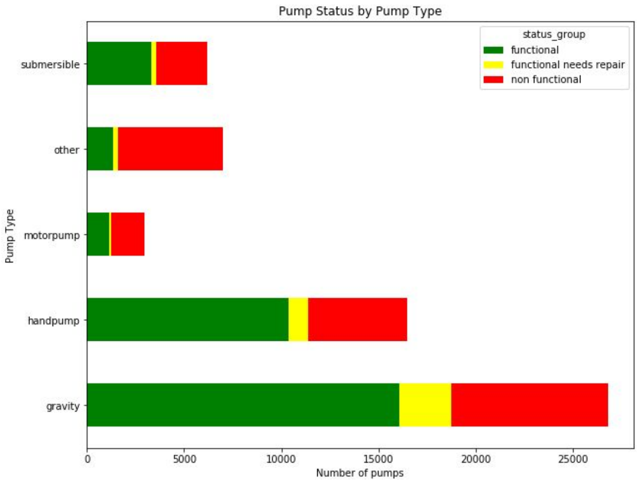
Unfortunately, 30% of the pumps do not have a construction year recorded, so the predictive quality of this feature may not be very strong



Exploratory Data Analysis - reduce cardinality

Visualizations highlight opportunities to reduce cardinality: a combination of the low-frequency groups on the left into the ‘other’ category produced the updated distribution on the right

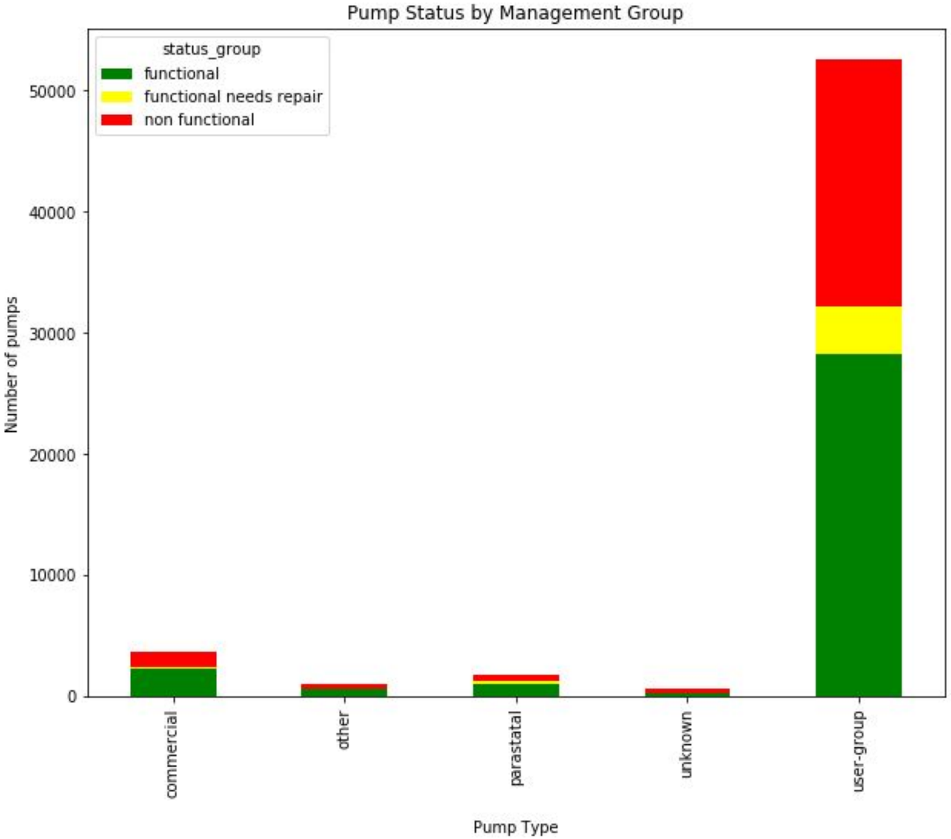




Exploratory Data Analysis - management group

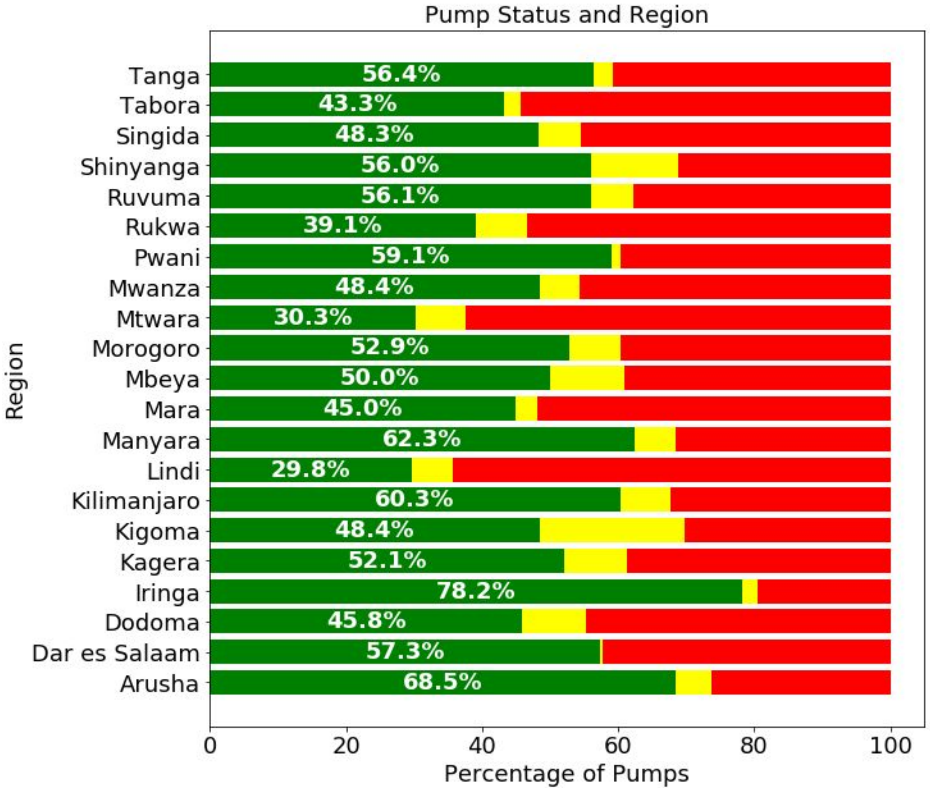
A intended outcome of this project is to improve pump management.

The visualization clearly shows variance among the most frequent management group. Efforts could be taken to encapsulate and teach the best practices of the successful groups.



Exploratory Data Analysis - feature selection

- 2 numeric and 6 hierarchical, categorical features describing geographic location
- Keeping too many meant introducing collinearity
- Choosing wrong level meant ensuring the curse of dimensionality with one hot encoding
- Visualizing distribution helped select the correct level without sacrificing too much variance



Statistical Data Analysis - Correlation

- Cramer's V - nominal version of Pearson's chi-squared test for independence between categorical features
- Low variance dataset

Insert heat map for low variance dataset

Statistical Data Analysis - Correlation

High variance dataset

Insert heat map for high variance dataset

Weaker correlations between features, stronger to target

Statistical Data Analysis - Correlation

- 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

Machine Learning - Algorithms and Evaluation Metric

- Algorithms for Classification Problems
    - K-Nearest Neighbors (KNN)
    - Logistic Regression
    - Random Forest Classifier
    - Adaptive Boosting (AdaBoost)
    - Extreme Gradient Boosting (XGBoost)
- Metric - F1
    - Unbalanced data
    - Cost of inaccurate classification

Machine Learning - Preprocessing

1. Pandas Factorize
2. Pandas get_dummies, N-1 to avoid collinearity
3. Test/Training split with 80/20 ratio
4. Standard Scalar

	Original number of features	**One hot encoded number of features**
Low-variance	11	53
High-variance	14	56

Machine Learning - Approach

1. The 5 models were run on the low-variance dataset.
    a. Feature importances were evaluated for performance improvement.
2. The 5 models were run on the high-variance dataset.
    a. 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 best performing tuned model, principal component analysis was performed for performance improvement.

Machine Learning - Step 1: Low variance dataset

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	0.81	0.77	0.31
AdaBoost	0.78	0.66	0.04
XGBoost	0.80	0.68	0.12

Machine Learning - Step 1: Low variance dataset

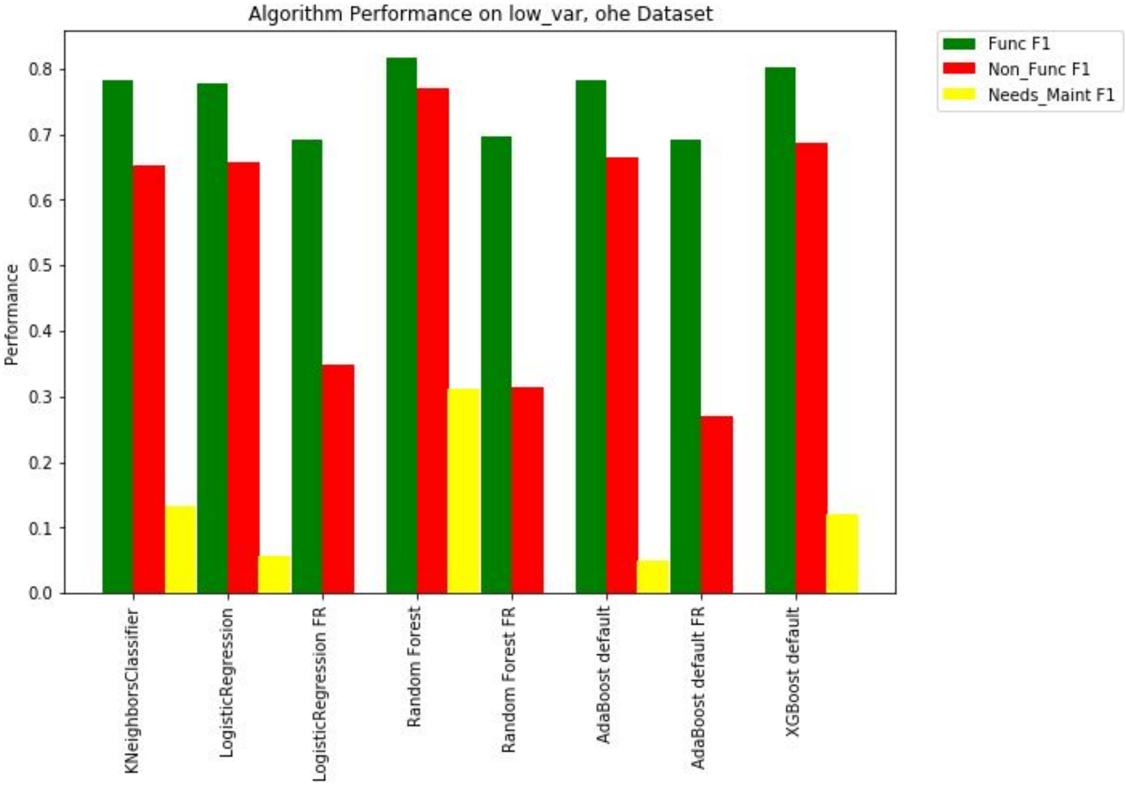
Feature evaluation

	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.

Machine Learning - Step 1: Low variance dataset

Results Summary



Machine Learning - Step 2: High variance dataset

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

Machine Learning - Step 2: High variance dataset

Feature evaluation

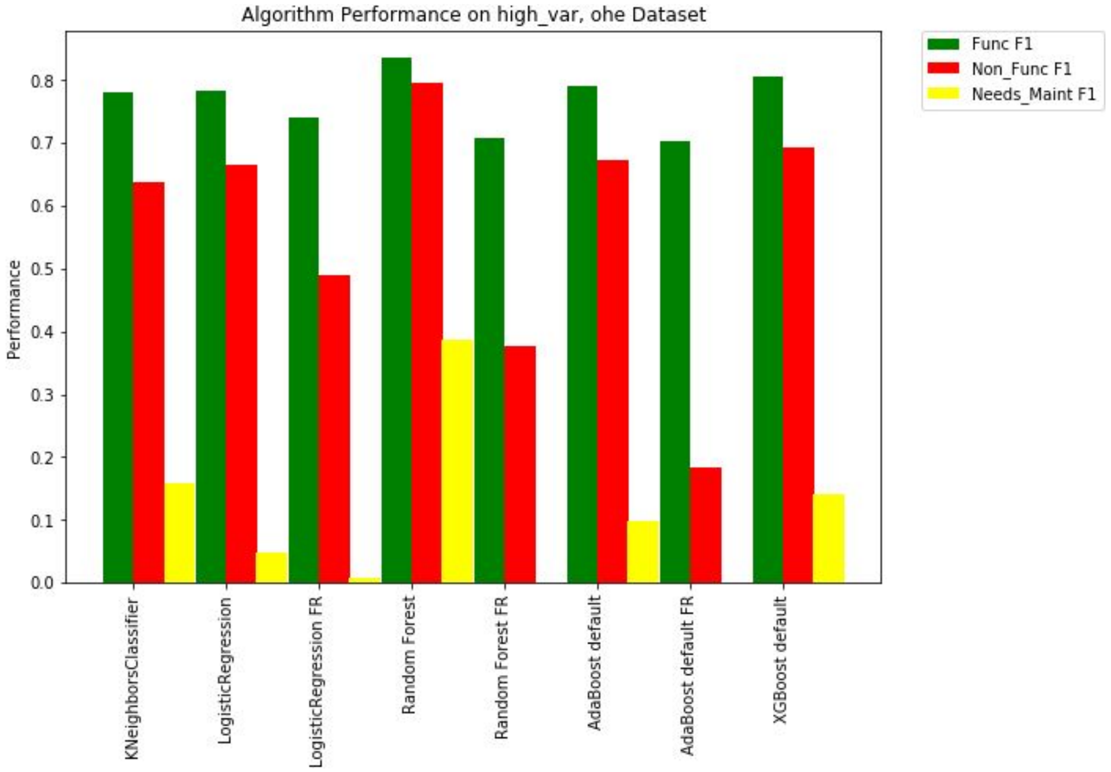
	Feature drop threshold	# Features dropped	0 - Functional	1 - Non Functional	*2 - Functional Needs Repair
Logistic Regression	< 0.05		0.74 -0.08	0.48 -0.24	0.006
Random Forest	< 0.005		0.70 -0.13	0.37 -0.42	0
AdaBoost	= 0		0.70 -0.08	0.18 -0.49	0

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

Machine Learning - Step 2: High variance dataset

Results Summary

The high-variance dataset produced better results.



Machine Learning - Step 3: CV and HP tuning

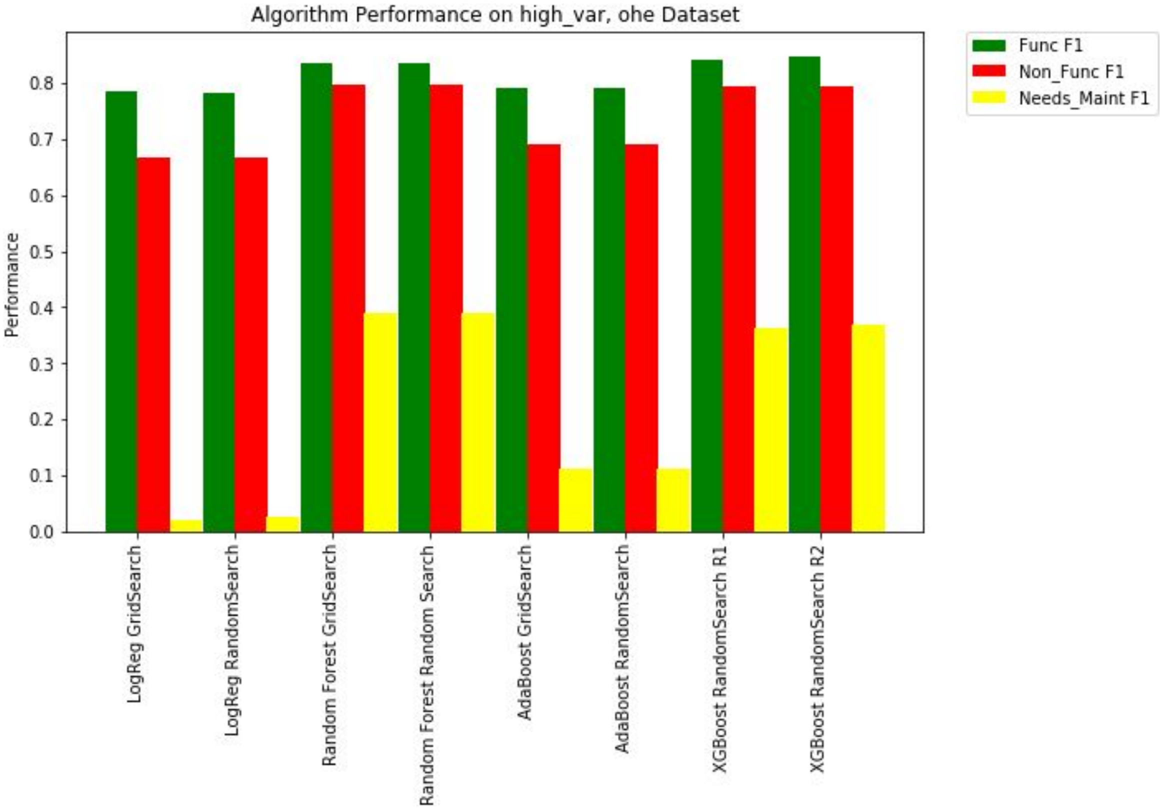
F1 scores	Search Type	0 - Functional	1 - Non Functional	2 - Functional Needs Repair
Logistic Regression	Grid	0.78	0.66	0.02 -0.02
Logistic Regression	Random	0.78	0.66	0.02 -0.02
Random Forest	Grid	0.83	0.79	0.38
Random Forest	Random	0.83	0.79	0.38
AdaBoost	Grid	0.79 +0.01	0.68 +0.01	0.11 +0.02
AdaBoost	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

← **38 hours!**

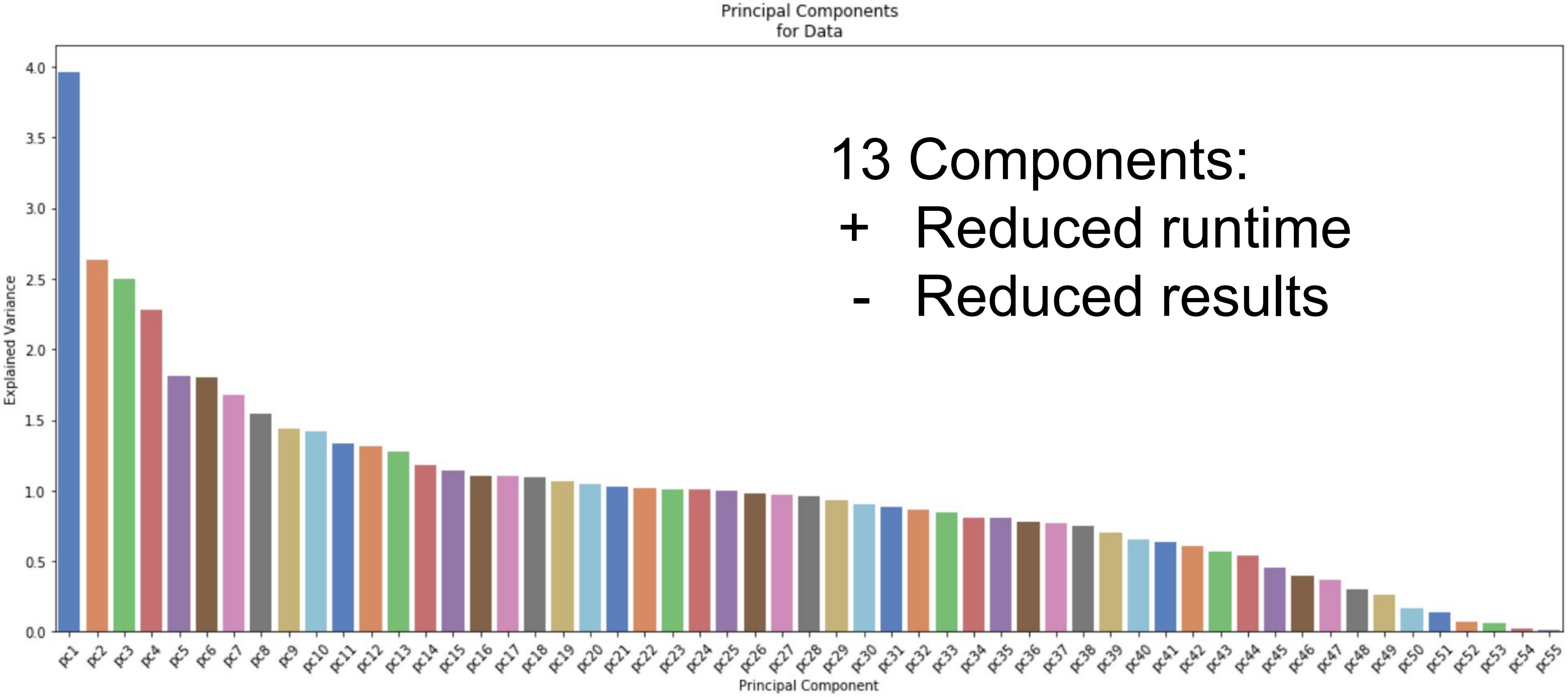
Machine Learning - Step 3: CV and HP tuning

Results Summary

Mixed results between two models: Random Forest and XGBoost



Machine Learning - Step 4: PCA



13 Components:

- + Reduced runtime
- - Reduced results

Machine Learning - Results Summary

	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

- Goal of maintenance improvement: focus on **recall** of ‘non functional’ and ‘functional needs maintenance’ class

Summary

- Saw improvement of results
- Feature selection is critical
- Created a model that can be useful to the Tanzanian Ministry of Water
- Future work
  - Different permutations of features
  - Additional rounds of hyperparameter tuning
  - Review of other preventative maintenance ML projects for best practices
  - Connect to data from other sources to include features like annual rainfall, and complete features that were included but were mostly missing, such as population