PUMP IT UP

TEAM LA

Amol Gupta, Shirley Hu, Gbenga Ilori, Patrick Linehan, Jessica Niles, Hari Saripalli, & Jovial Zhang

GMMA 869 | Saturday, November 13, 2021

CURRENT SITUATION

- **4M Tanzanians lack access to a safe source of water***
- **Handpumps are a critical water supply method***
- **60,000 handpumps installed across Sub-Saharan Africa every year***
- **30% to 40% do not work at any one time***
- **Women & girls spend 15-17 hours a week collecting water****
- **Loss of $1.2B initial investment with massive productivity & public health consequences***

*Purvis, 2016

**SimplePump, 2021

Image Source: The Tanzania Water Crisis: Facts, Progress, and How to Help, 2021

SOLUTION

- **Built a machine learning model to predict the operating condition of waterpoints in Tanzania**
- **Achieved 81.88% classification rate**
- **Estimated to save $445K by deploying the model**

Image Source: (HD wallpaper: Mount Kilimanjaro, n.d.)

Section 1

THE COMPETITION

8 Models
20 Submissions
Best Score: 0.8188
Rank: 1347/12696

DATA EXPLORATION

59,400 WATER PUMPS

40 attributes:

31 categorical | 6 numeric | 3 labels

PUMP STATUS

54.3% Functional

38.4% Non-Functional

7.3% Functional but needs repair

MISSING VALUES

6119 rows with NA's (10.3% of instances)

Features with 0s and 'none' in many instances

MISSING VALUES

NaNs & 0s:

Population, Construction Year, Lat/Long, GPS Height

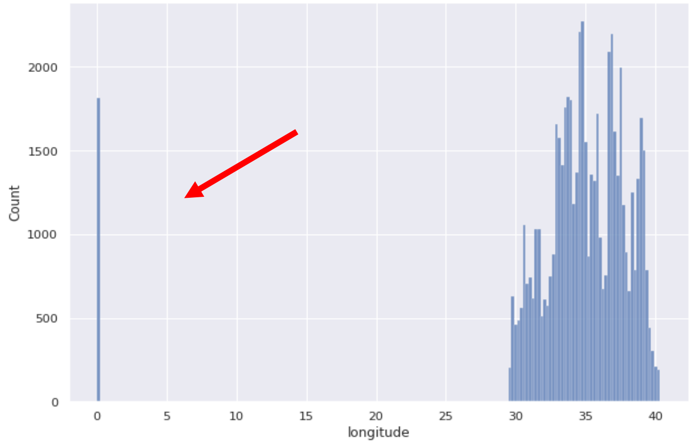
Zero longitude in Atlantic Ocean

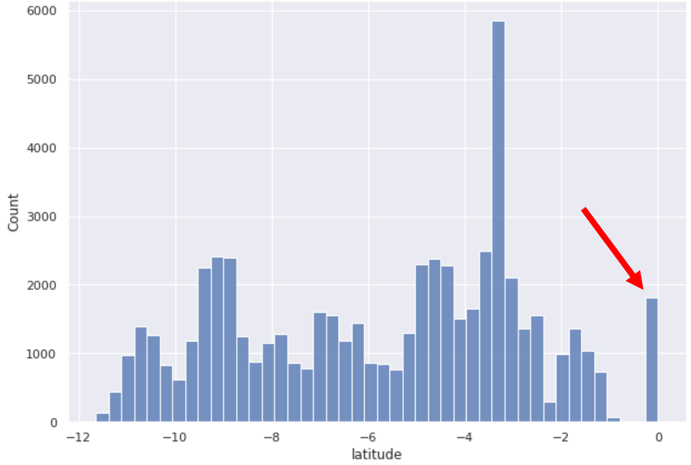
Zero latitude not in Tanzania

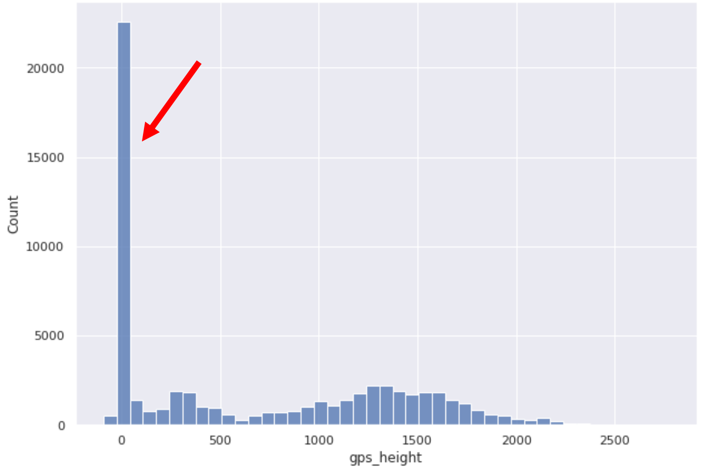
Topographic map:

elevation > 0m

Imputed NaNs & 0s with means grouped by geographic location (subvillage, ward, lga, region)







CATEGORICAL VARIABLES

Cleaning:

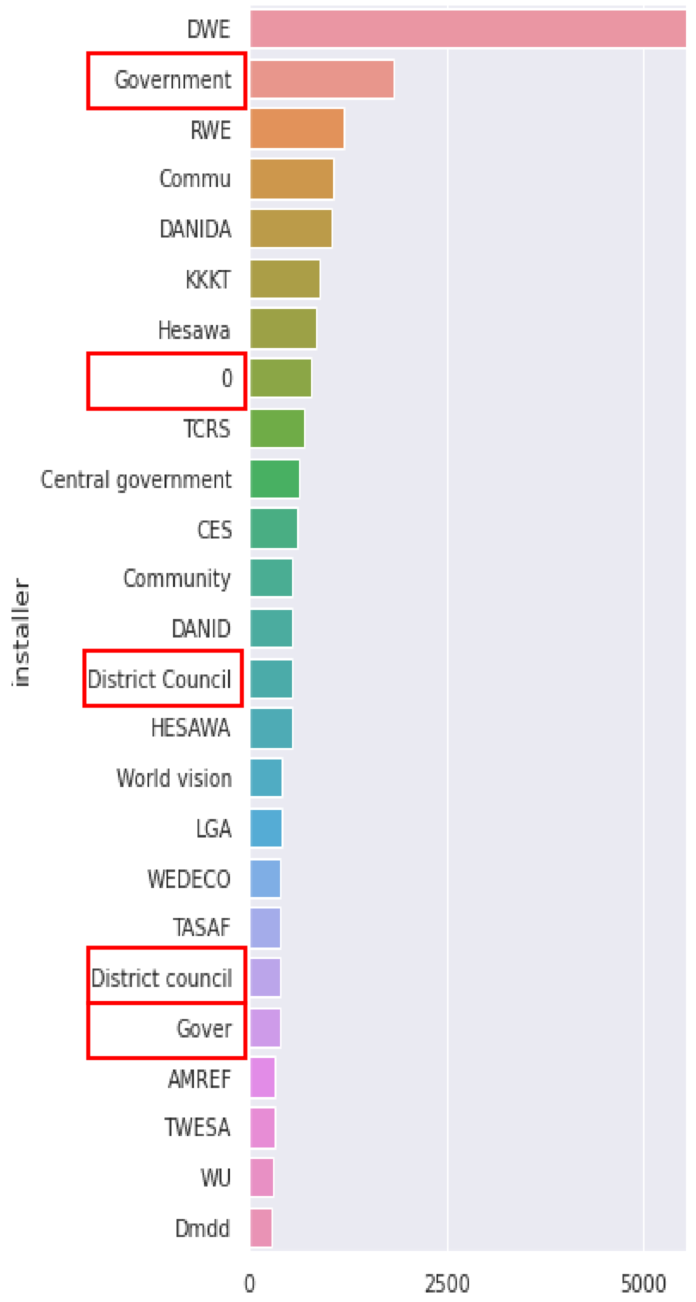
Converted to lower case

Retained top 25 & imputed the rest as “Other”

Combined similar words into one

Replaced ‘0’ & ‘none’ with most frequently occurring value

Dropped columns that are mostly similar



FEATURE ENGINEERING: LAT/ LONG

Remote Pumps:

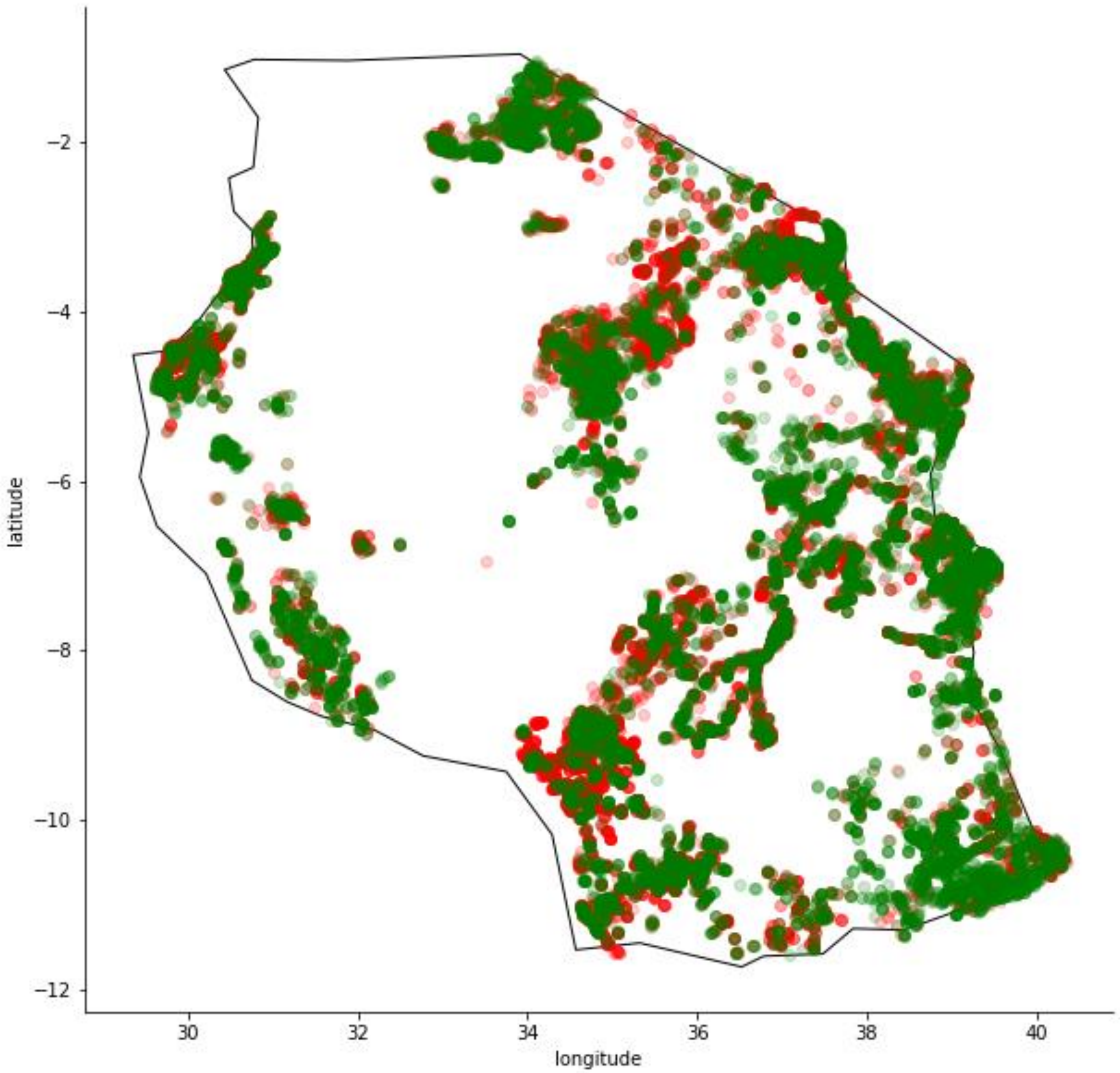
Pumps in **remote areas** (far from other pumps) have higher chance of being **non-functional**

Geo-Clusters:

Used clustering of lat / long to create **15 geo-clusters** to calculate the distance of pump from its cluster centroid

Feature Importance:

Important feature in the model & boosted accuracy slightly

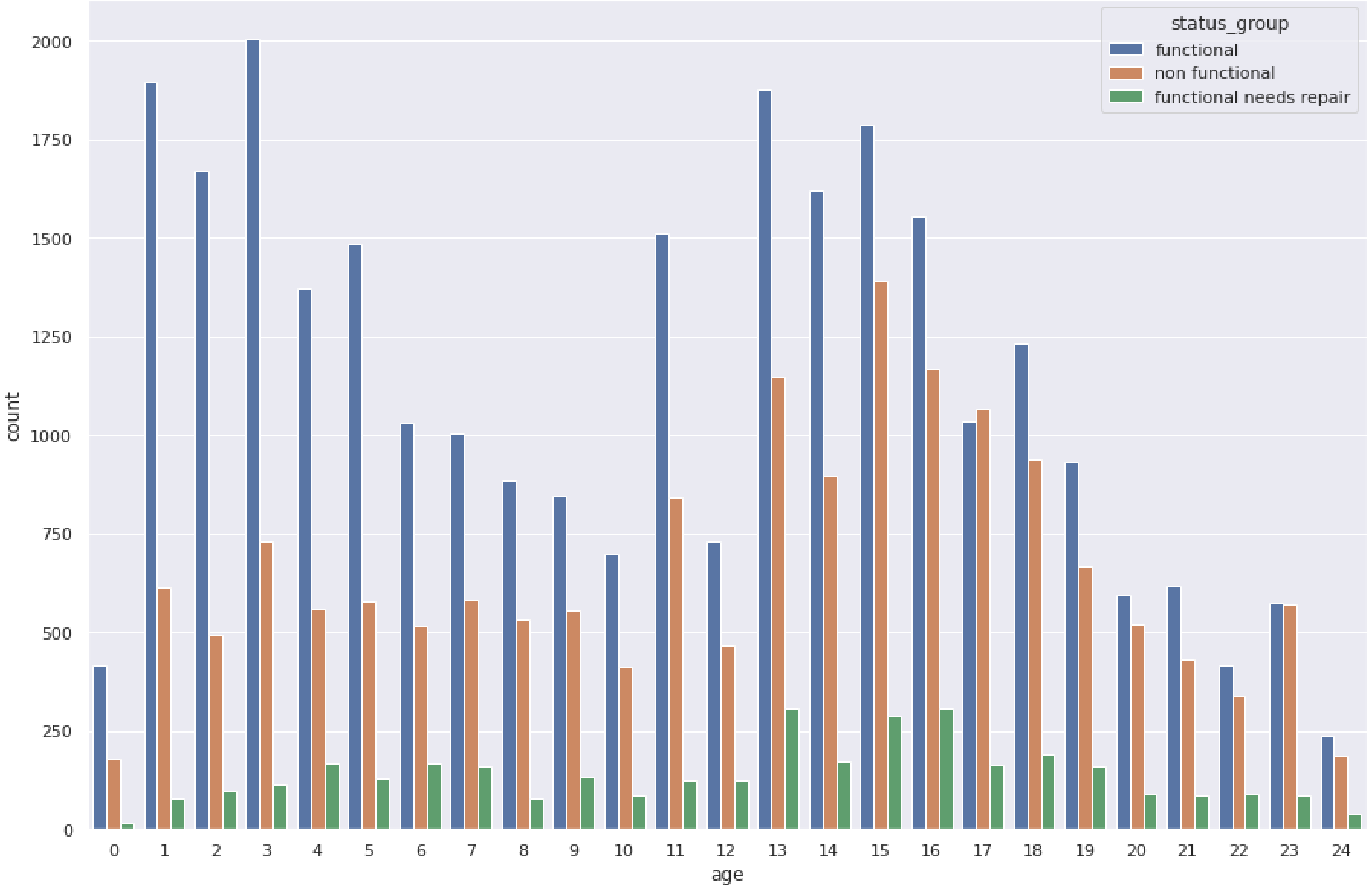


FEATURE ENGINEERING: DATE

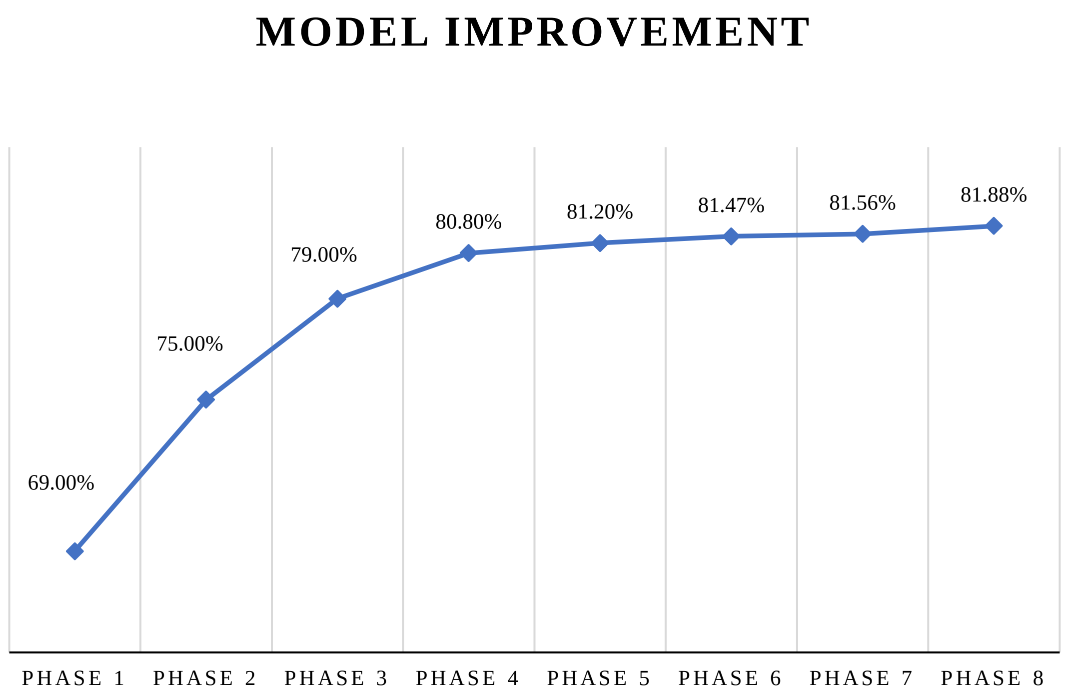
Percentage of **non-functional pumps is higher for older pumps** (more than 15 years)

Used construction year & date recorded to **calculate age of a pump**

Parsed dates to create a month & season column - didn't make it to the top feature list



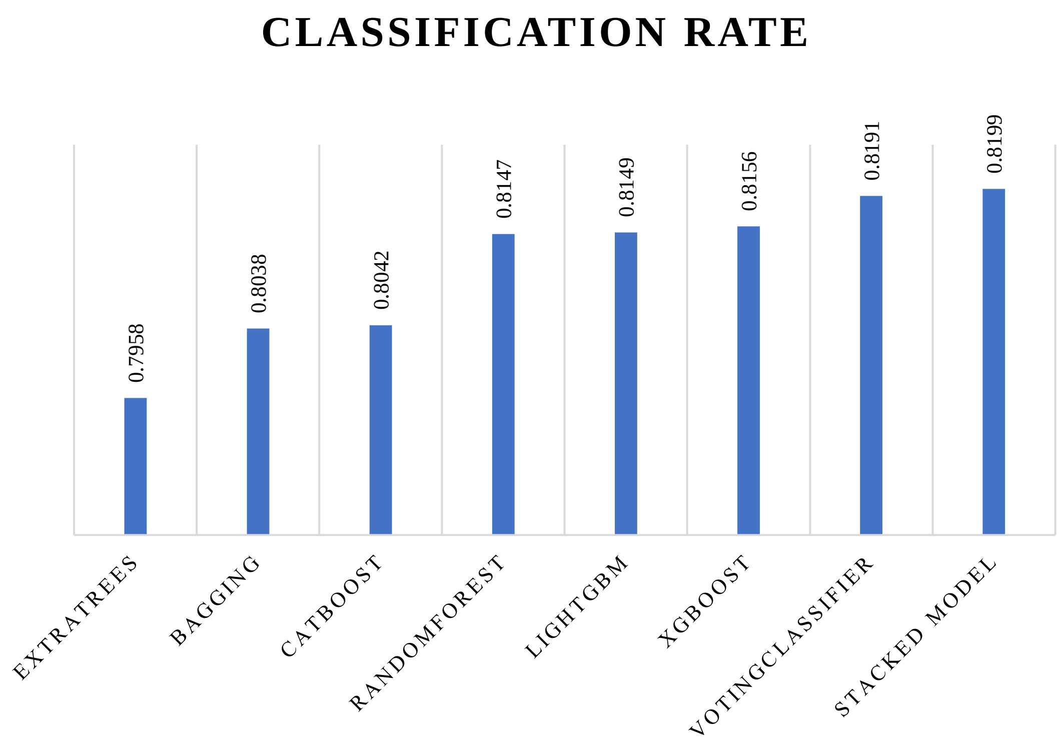
MODEL DEVELOPMENT



Phase 1	DecisionTree, No Cleaning	69.00%
Phase 2	Pipeline: SimpleImputer, AdaBoost	75.00%
Phase 3	Pipeline: SimpleImputer, Encoder, RF*	79.00%
Phase 4	Pipeline: Category Coalescer, RF*	80.80%
Phase 5	Extensive data cleaning, feature engineering, RF*	81.20%
Phase 6	Hyperparameter Tuned RF	81.47%
Phase 7	Hyperparameter Tuned: LightGBM, Catboost, XGBoost, Bagging, Extra Trees	81.56%
Phase 8	VotingClassifier, StackingClassifier	81.88%

*RF: RandomForestClassifier

MODEL SUMMARY



- Hyperparameters tuned using **GridSearchCV (3-fold)**
- **Challenge:** not knowing a good starting point
- **LightGBM:** hyperparameter tuning resulted in ~5% improvement compared to default parameters
- **Best hyperparameters:** learning rate, num_iterations, n_estimators
- **VotingClassifier** resulted in ~0.4% improvement in accuracy compared to best performing single model

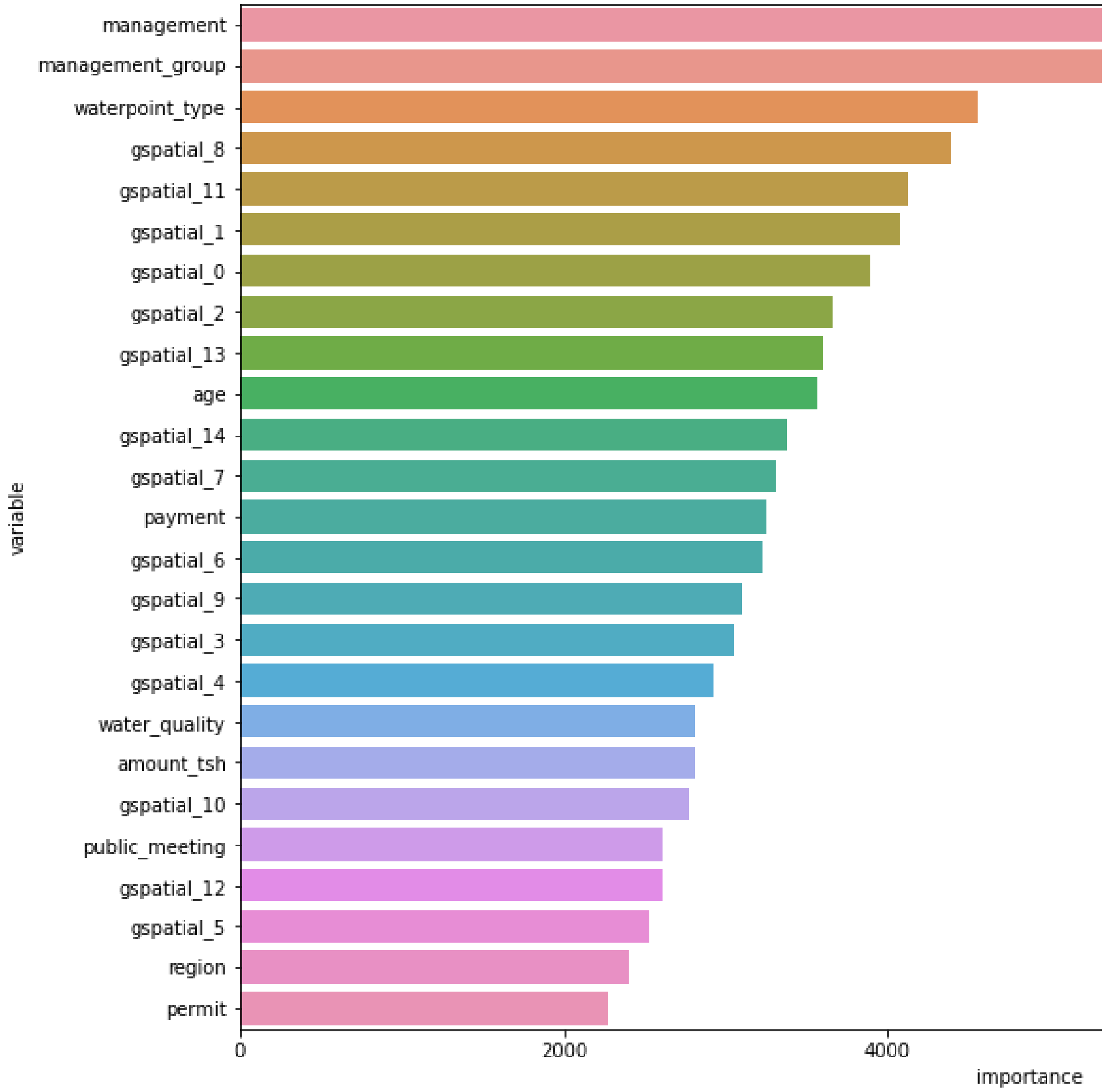
Final Submission: VotingClassifier

Classification Rate: 0.8188

LightGBM FEATURE IMPORTANCE

Top 8 features of importance:

1. How the waterpoint is managed
2. Type of waterpoint, water quality
3. Distances from the cluster centroids
4. Age of the pumps
5. Payment scheme
6. If the waterpoint is permitted or not
7. Who is the installer
8. Where is the waterpoint installed



LESSONS LEARNED

What Worked

Data Cleaning

Feature Engineering

Hyperparameter Tuning

Voting / Stacked Ensemble

Models differ in feature importance

What Didn't

DBSCAN clustering for lat/long: couldn't find optimal clusters

H2O AutoML: Default parameters - ran 6+ hours, Google Colab reached runtime limit

What Next

Optuna or hyperopt for hyperparameter optimization: GridSearchCV is not optimal

Build custom function transformers for data preprocessing to include in the model pipeline

Try H2O AutoML

Section 2

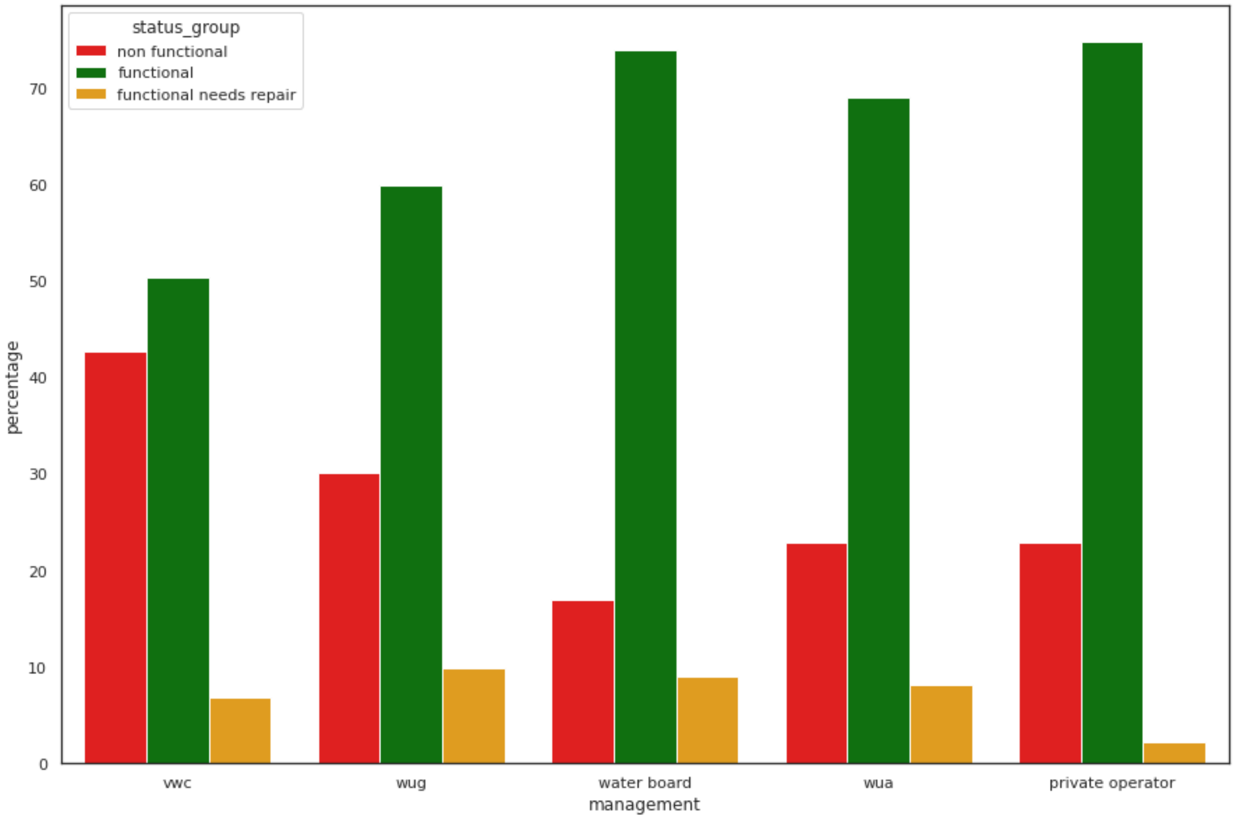
IMPLEMENTATION PROPOSAL

KEY INSIGHTS

Management

Pumps managed by VWC & WUG have higher failure rates

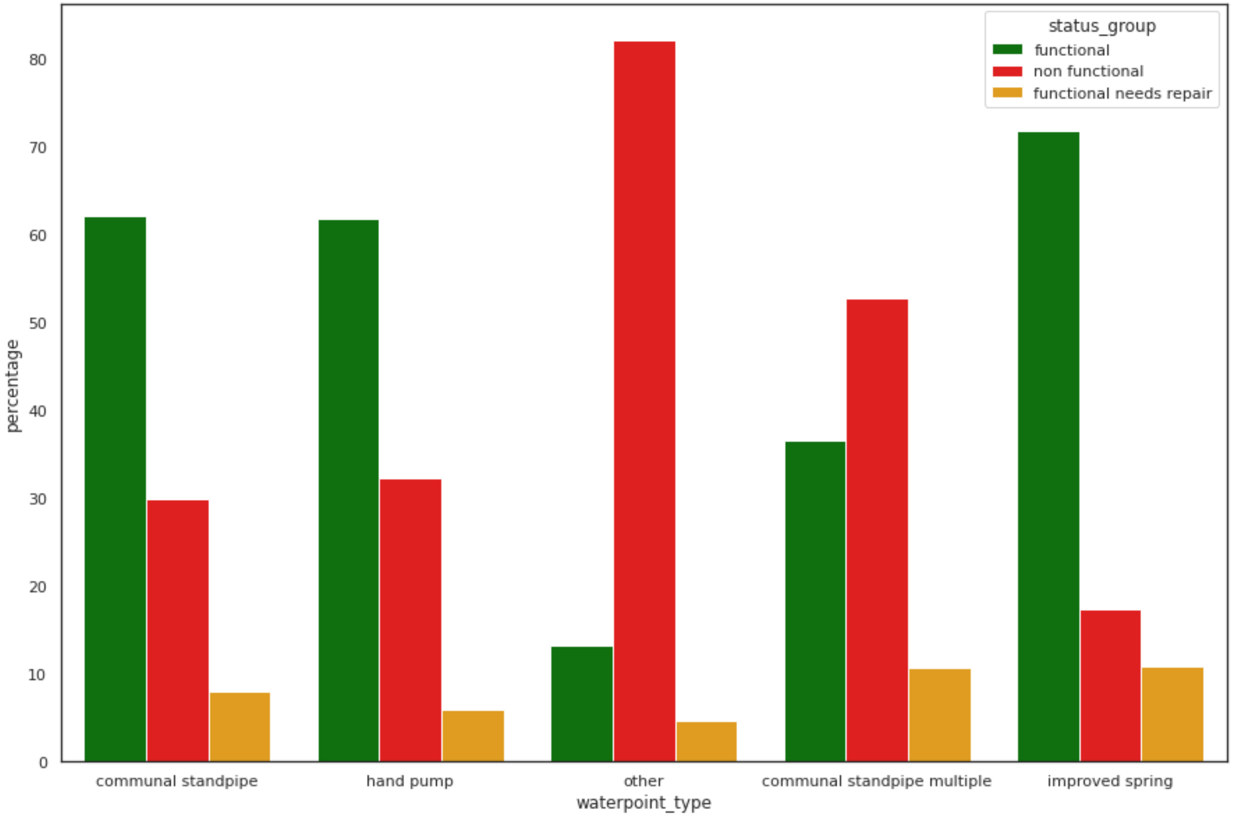
Water Board Management have higher percentage of functional pumps



Waterpoint Type

Communal standpipe & hand pumps type waterpoints have higher percentage of functional pumps

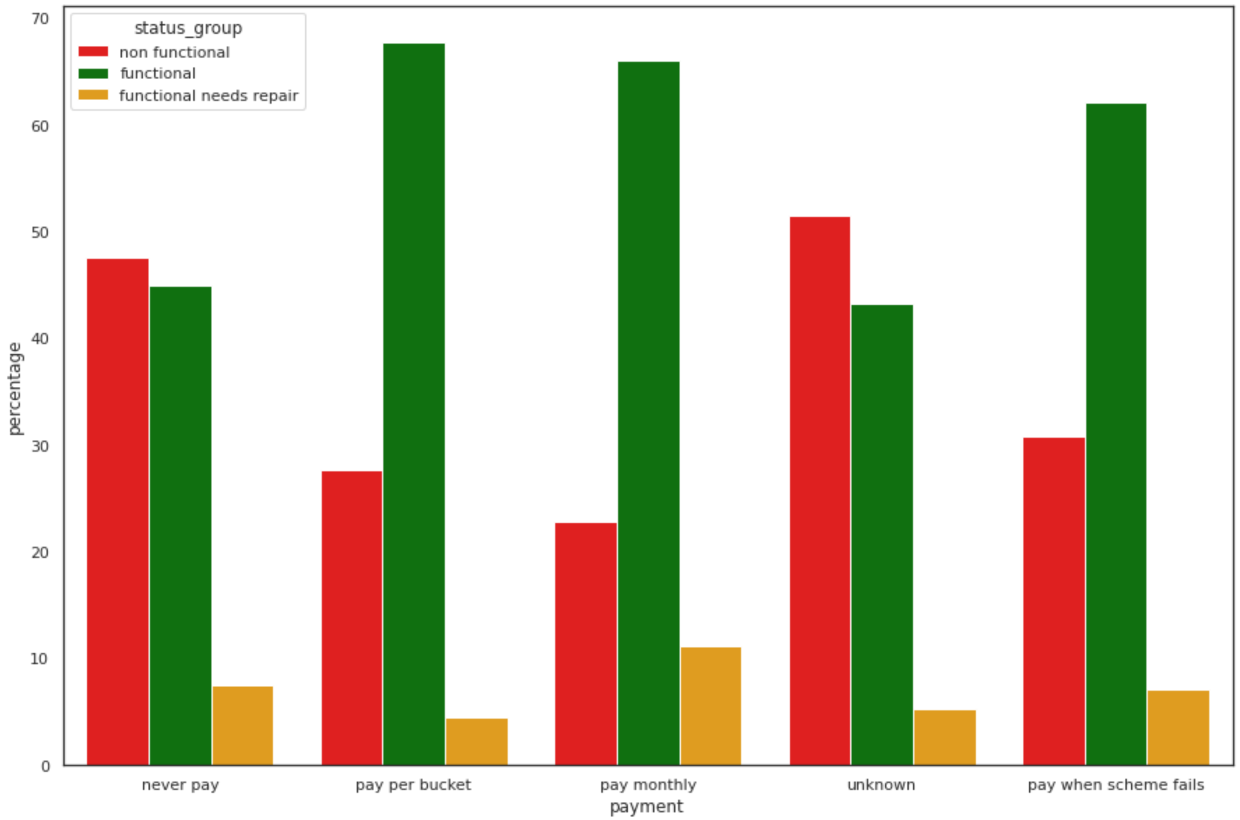
'Other' and 'communal standpipe multiple' needs further investigation



KEY INSIGHTS

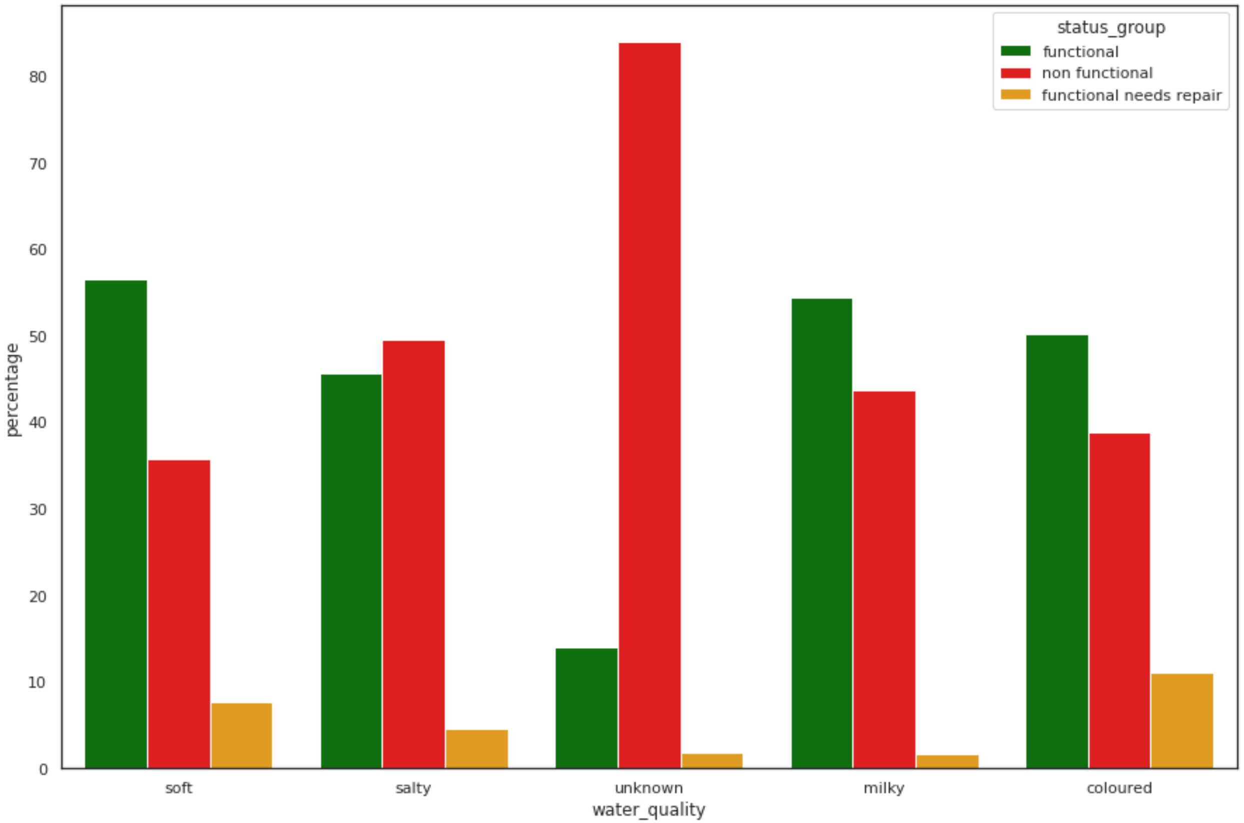
Payment

Free / never pay water points have higher percentage of non-functional pumps



Water Quality

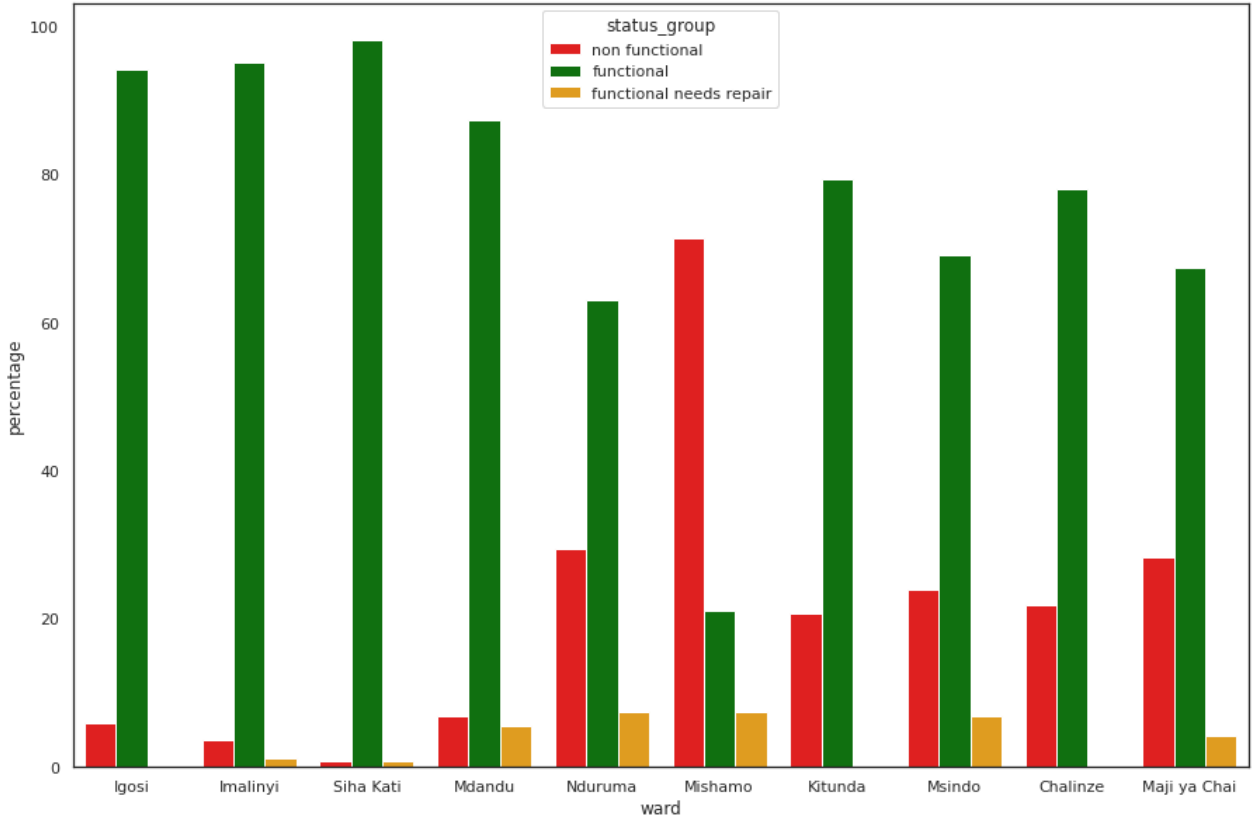
'Soft' water quality is better for waterpoint operation



KEY INSIGHTS

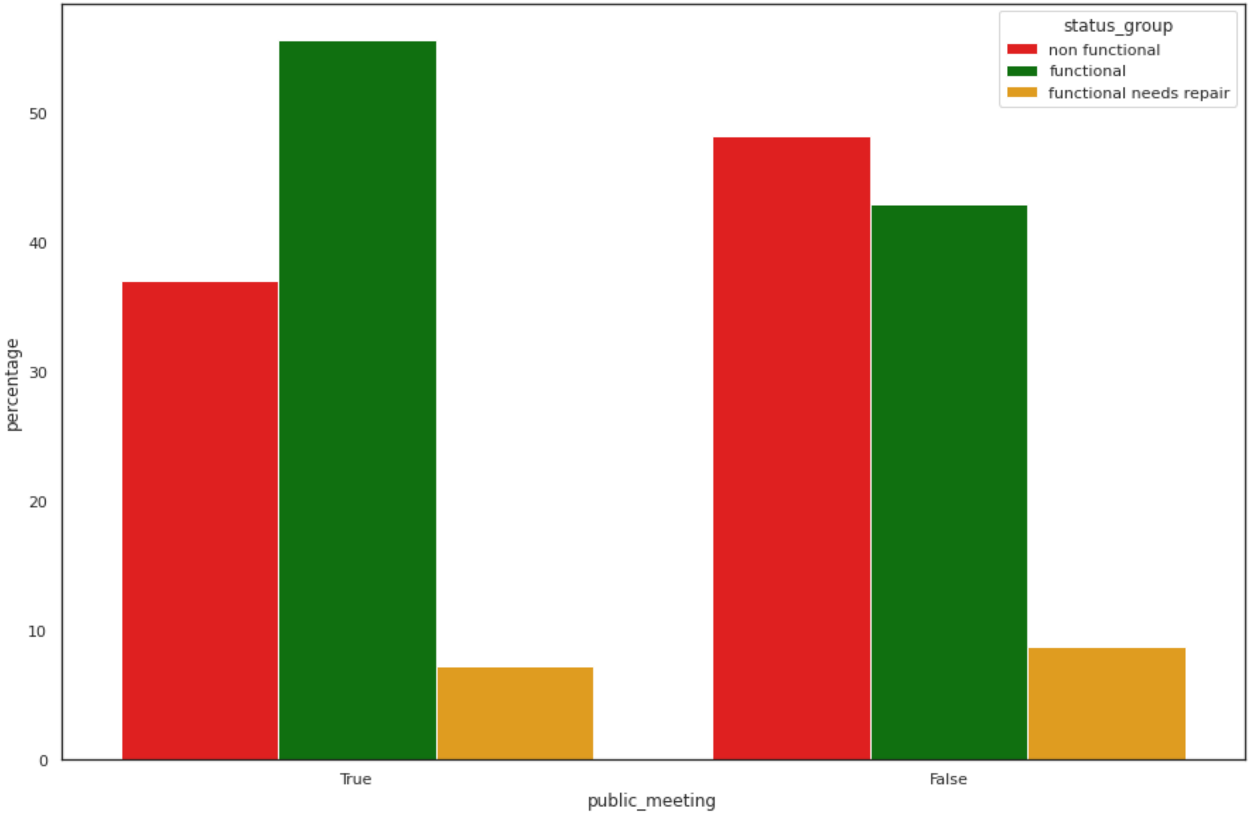
Ward

Certain regions need more maintenance
(Mishamo has high percentage of non-functional pumps)



Public Meetings

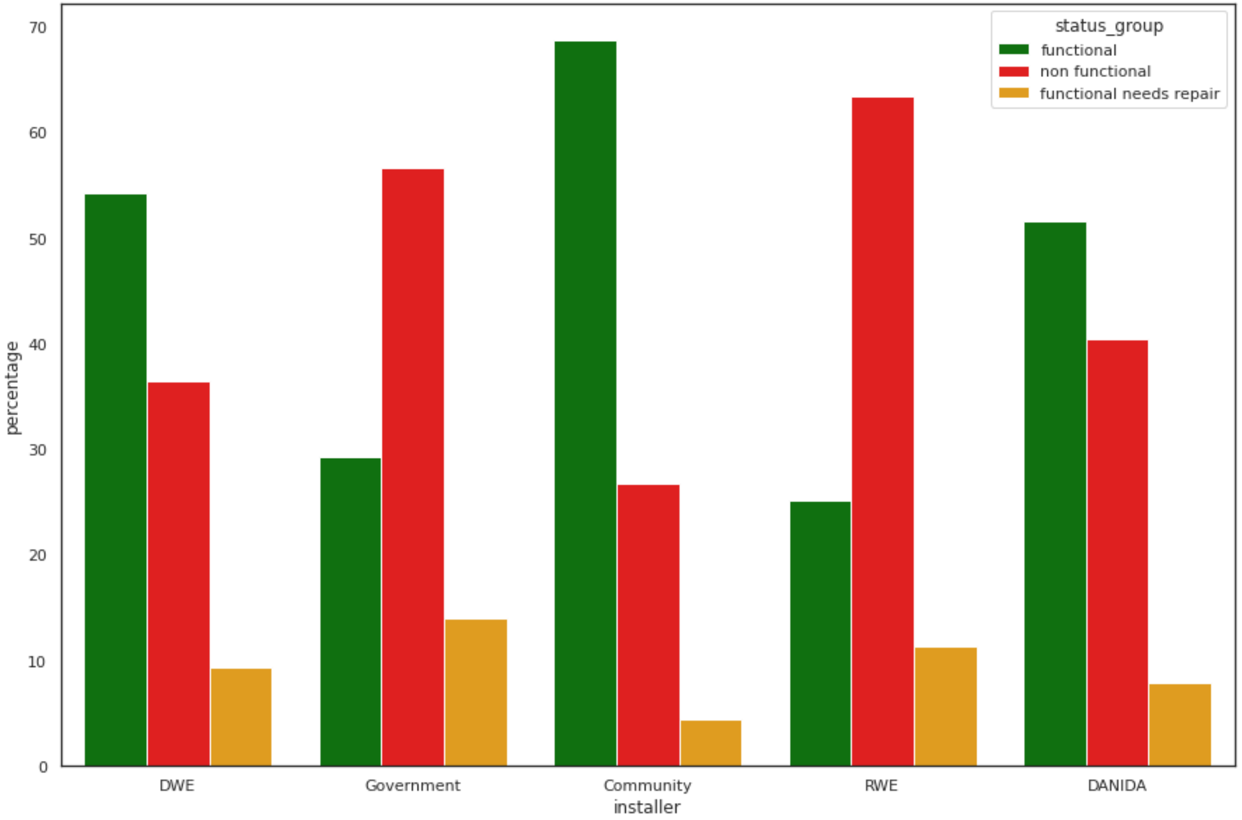
Hold public meeting about waterpoint



KEY INSIGHTS

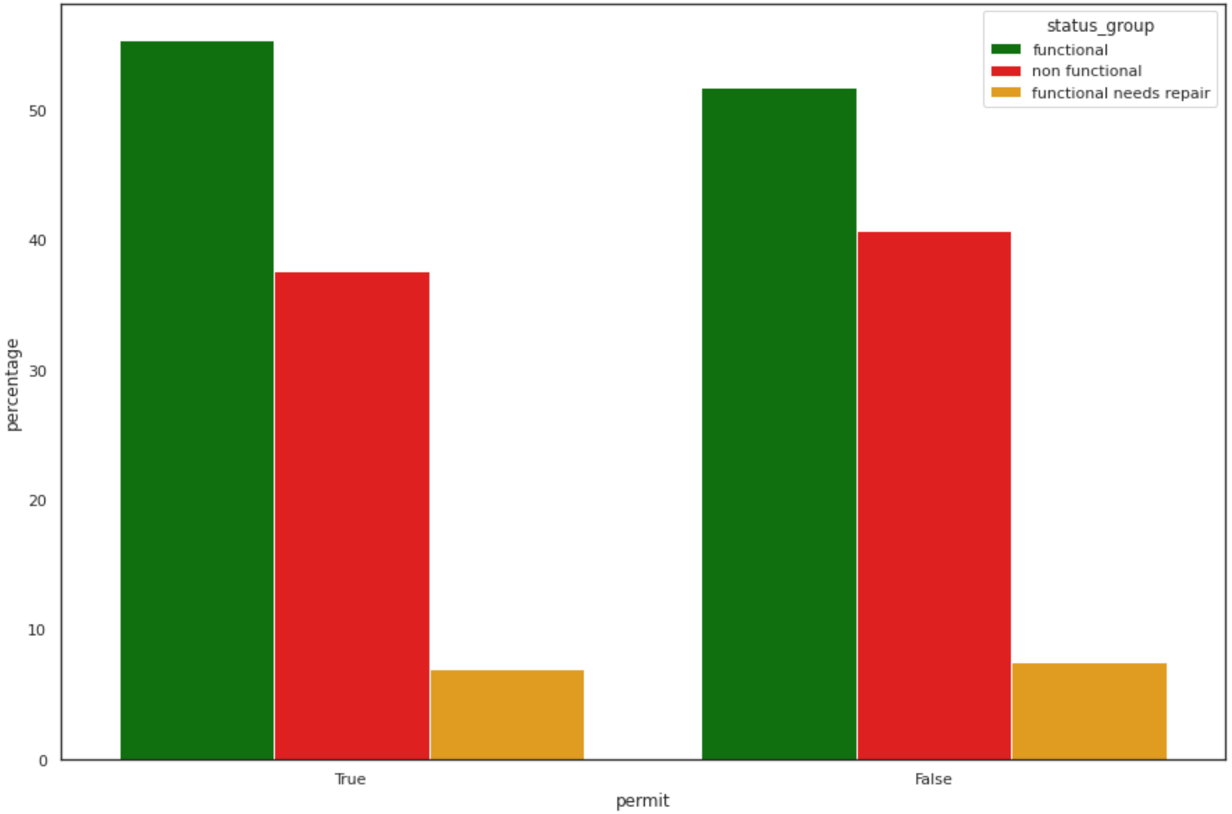
Installer

DWE & community installed pumps have higher functional rate



Permit

Permitted pumps have slightly higher functional rate



COSTS

Cost of False Negative: loss of water supply

Increased cost of water
Human life, health, and wellbeing
Economic – agriculture, livestock
Time – economic, education

Cost of False Positive: pump is not faulty

Waste of time and limited resources (wages, transportation costs)

Repair vs. Replace:

The cost to **repair a pump ($10)** is significantly less than the **cost to replace (~$1,000)***

Value of Model:

Estimated **immediate value of ~$445K** and **medium/life term value of ~$3.7M** if the model is implemented

*(George Joseph, 2019)

IMPLEMENTATION PROPOSAL

“Pump Playbook”

Predictive model built into interactive web platform

Mobile-friendly

Data collection verification & standardization

Pump Management:

Pump installation & maintenance education & employment program for women & girls

ARCHIBUS

Best practices for installation & maintenance:

- *Management:* Water Board are effective managers
- *Pump Age:* Older pumps are more likely to fail
- *Public Meetings:* Host public meeting for waterpoint management
- *Permit:* Go through permit process
- *Region:* Certain regions require more maintenance
- *Waterpoint Type:* Focus on Communal Standpipe Multiple

Aligned with the UN Sustainable Development Goals:
Universal coverage of safe water by 2030**

Estimated that every $1 invested in water & sanitation programs yields up to $12 in economic returns*

*The Water Project, 2021
**UNDP, 2021

CONCLUSION

Machine learning model achieved 81.88% classification rate

Estimated to save $445K by deploying the model

- Implement “Pump Playbook” maintenance program led by women & girls
- Focus on: Management, Pump Age, Public Meetings, Permit, Region, Waterpoint Type, Water Quality, & Payment

The Result?
Improved economic output & public health

APPENDIX

PRESENTATION OVERVIEW

The Competition

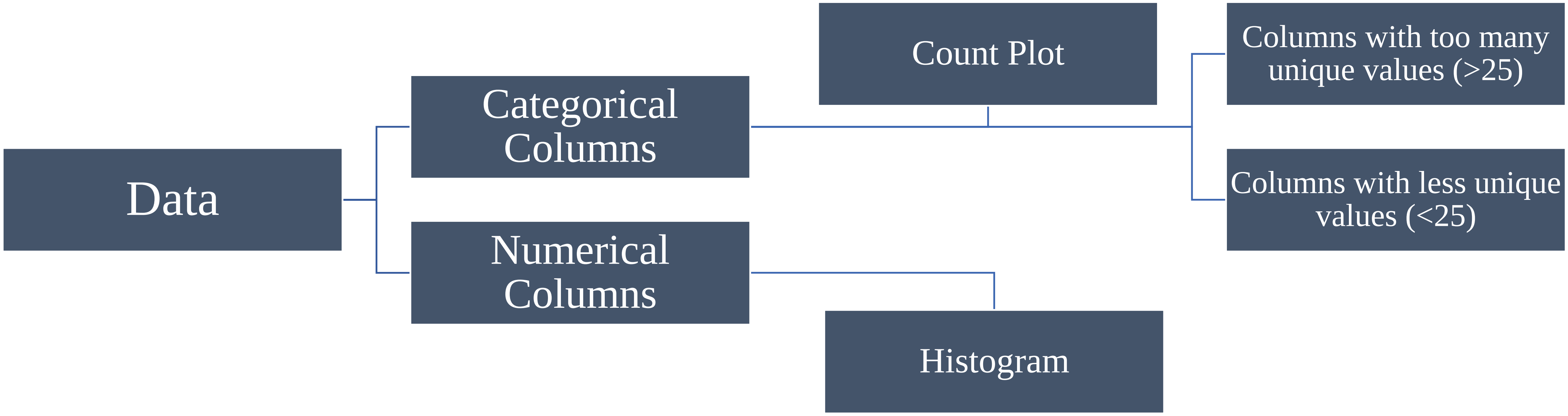
1. Background & Current Situation
2. Exploratory Data Analysis & Model Development
3. Model Performance & Submission Review

Implementation Proposal

1. Application & Social Impact
2. Recommendations to Tanzanian Government

Image Source: Wallpaper Flare

EDA WORKFLOW



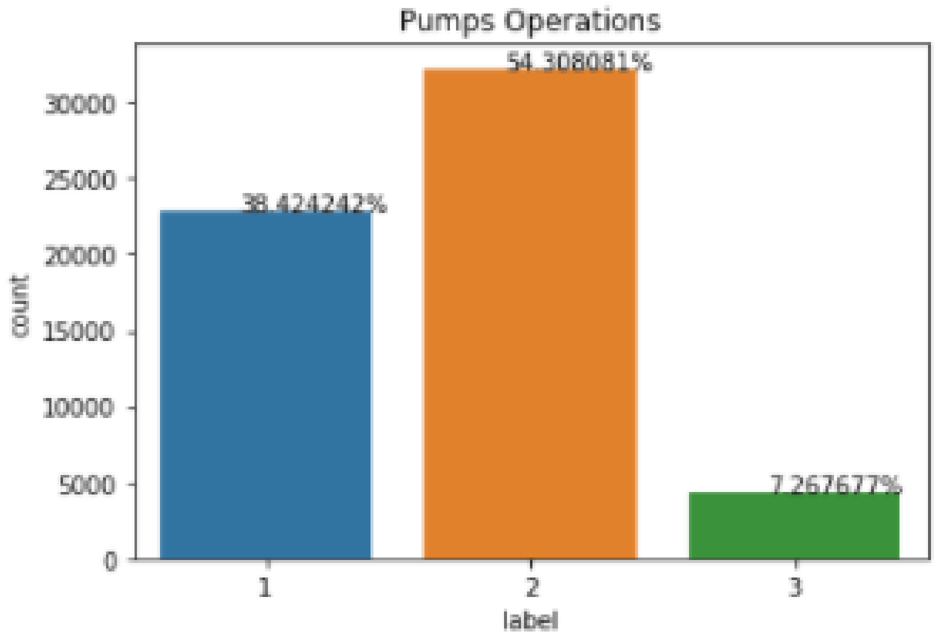
DATA EXPLORATION

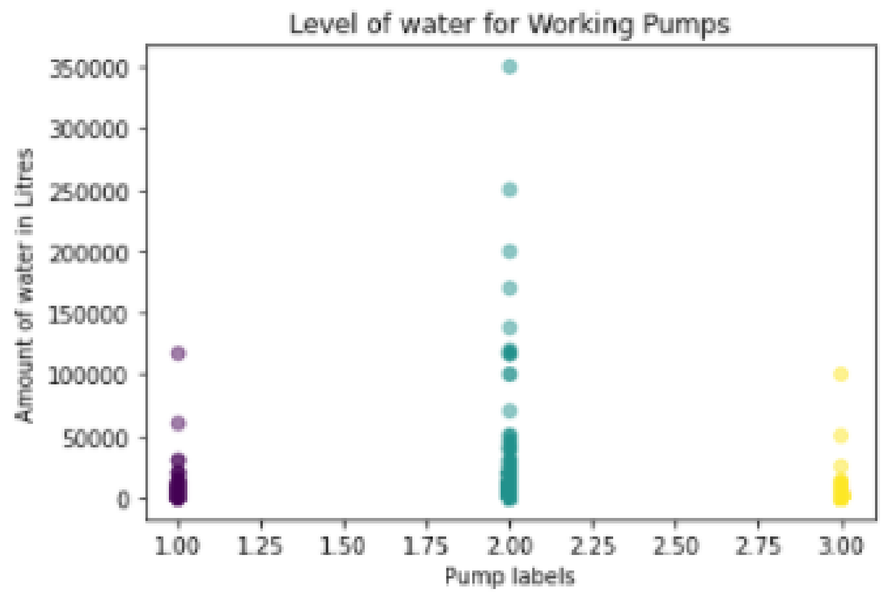
Data Distribution

- Non-functional pumps: 38%
- Functional pumps: 54%,
- Functional needs repair pumps: 7%

Water Level

If the amount_tsh is greater than 150K litres of water, the water pump is working and this amount is sufficient for the water pump to work smoothly



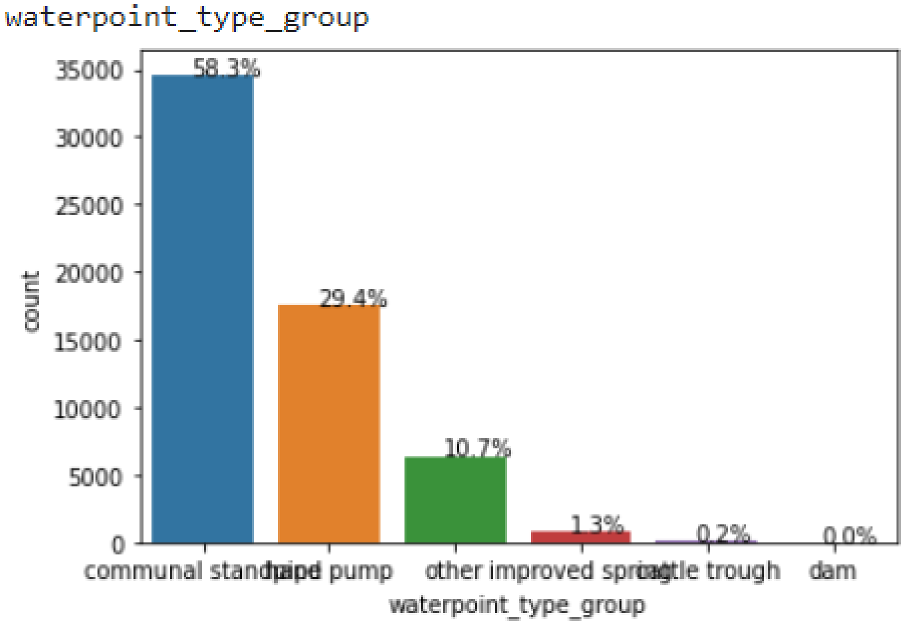


DATA EXPLORATION

Waterpoint Type

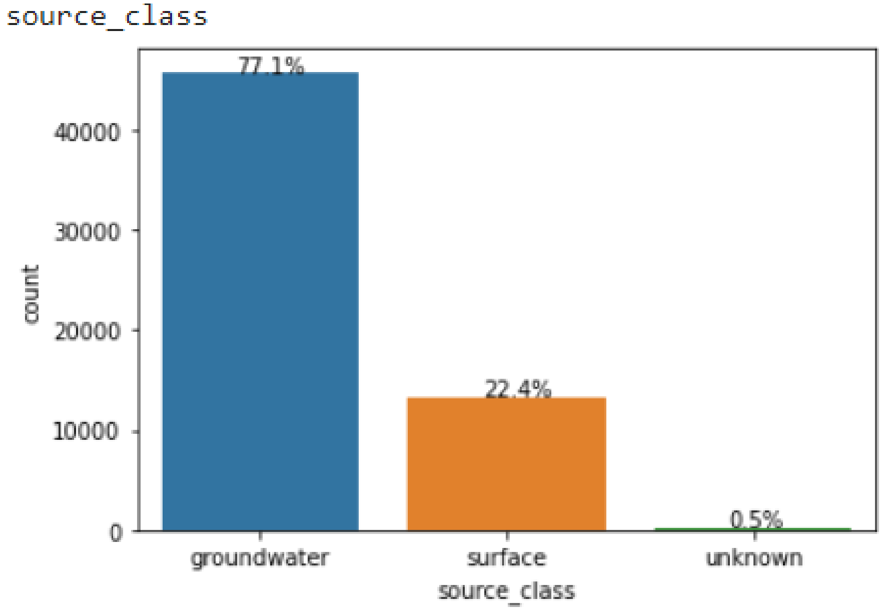
~90% of the waterpoints are "Communal Standpipe" or "Standard Pumps”.

Almost all pumps are used as a communal water supply for neighbourhoods which lack individual housing water service.



Water Source

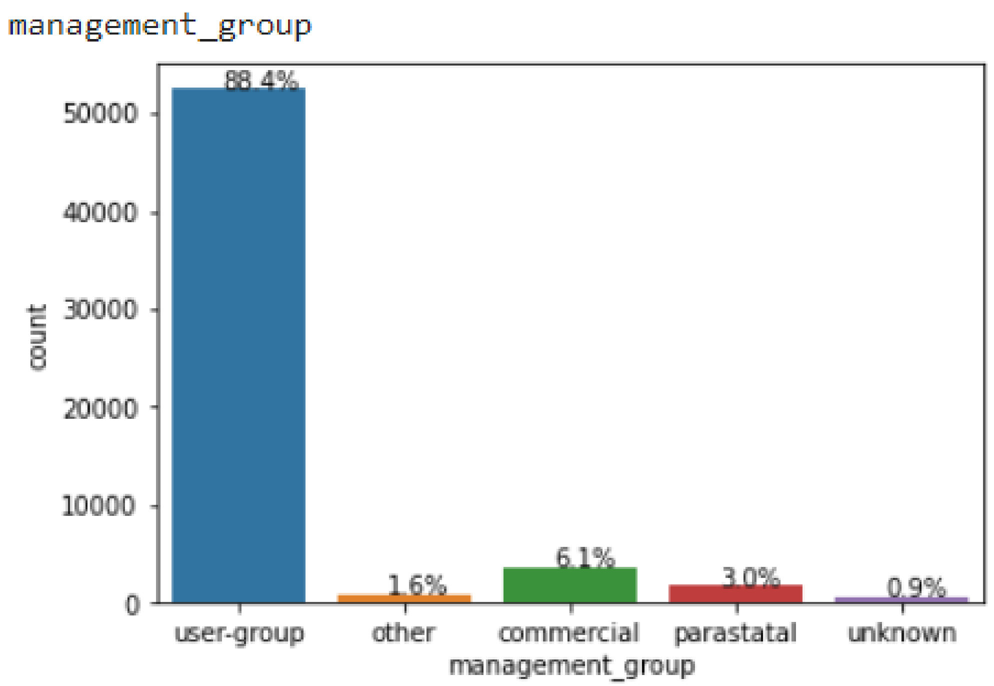
"Groundwater" accounts for the highest source of water at about 77% and the next is for the surface at about 22%.



DATA EXPLORATION

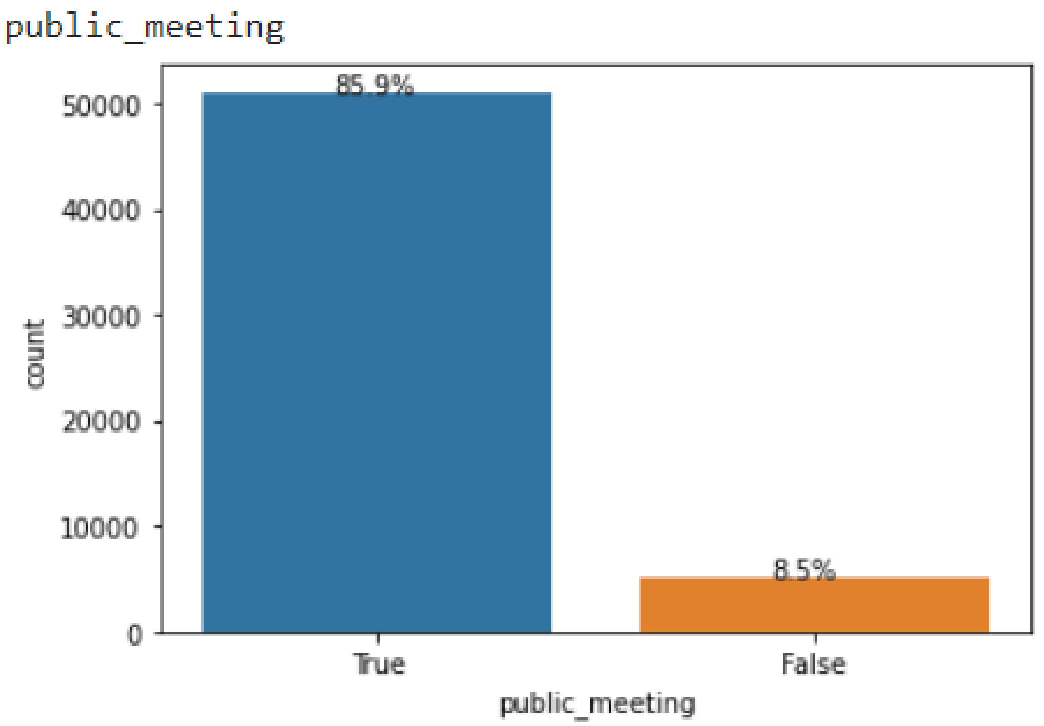
Management Group

User groups manage ~90% of all pumps.



Public Meetings

86% of pumps held public meetings.

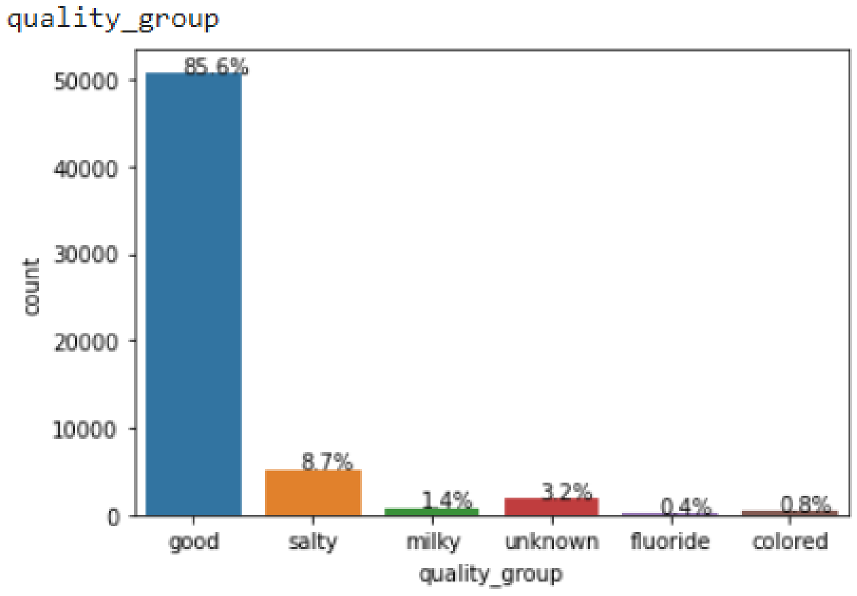


DATA EXPLORATION

Water Quality

85% of pumps have "Good“ water quality.

The rest of the pumps are split into salty and other categories.

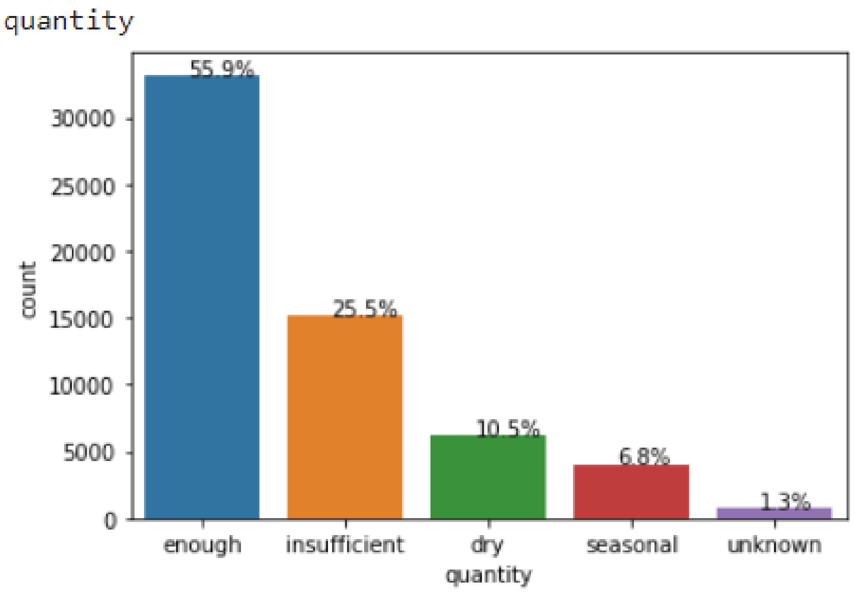


Water Quantity

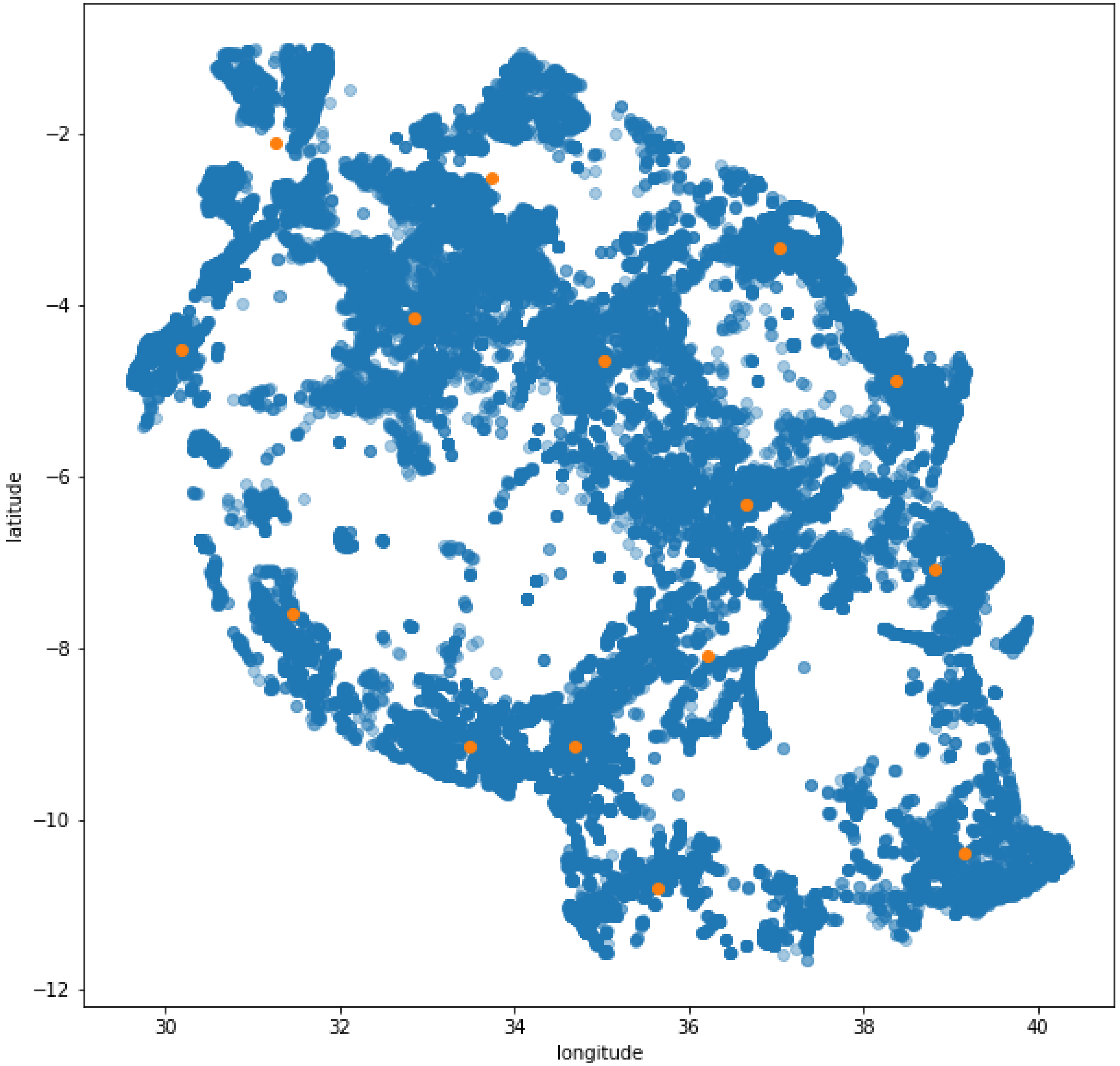
30K L is required for functional pumps.

Almost 56% of pumps have a high enough quantity.

~25% have insufficient water levels, with quantities below 15K L.



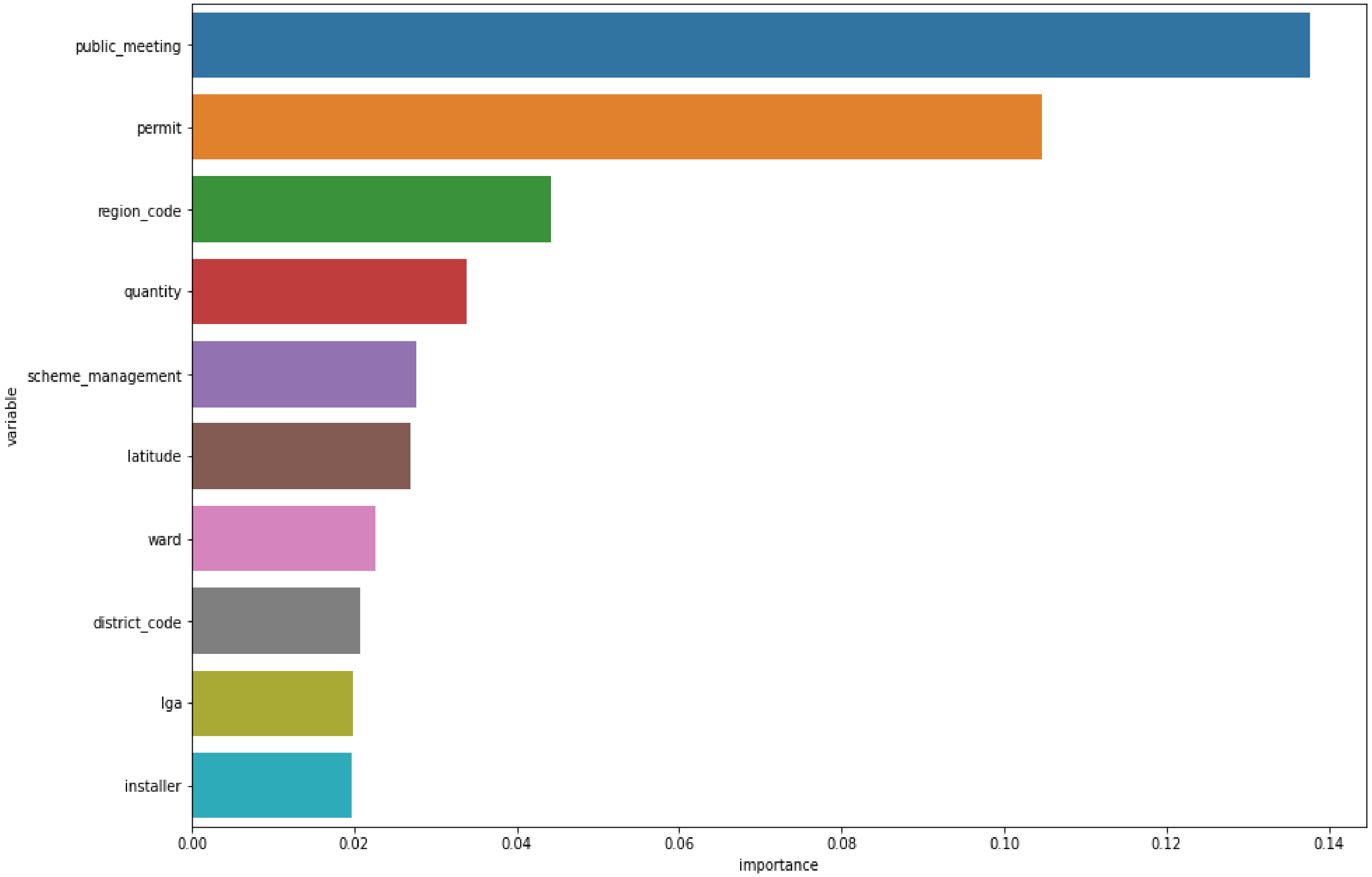
- KMeans clusters of lat/long



XGBoost FEATURE IMPORTANCE

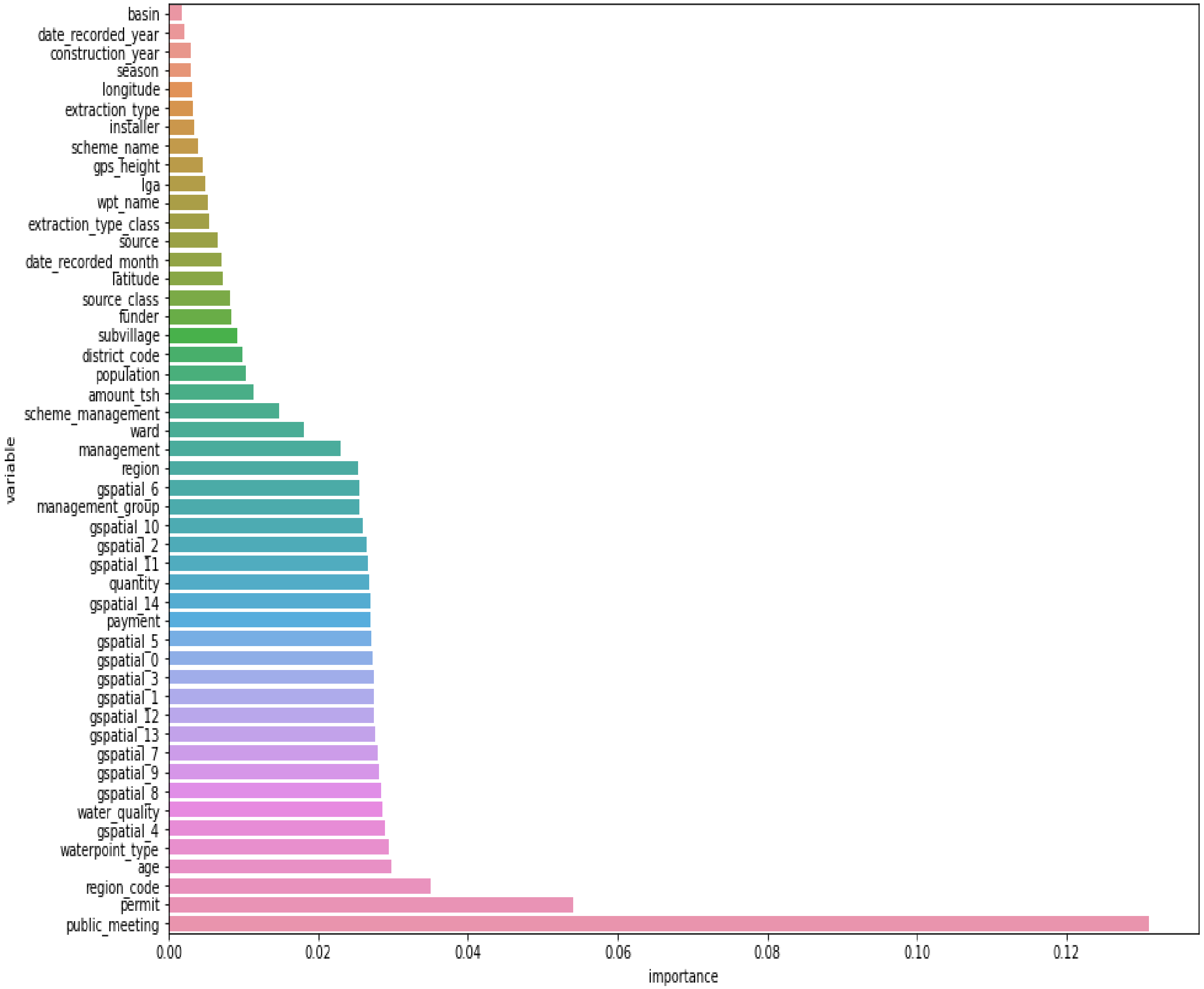
XGBoost has different feature importance for the same data

Does not include the geo spatial features that were added



RandomForest FEATURE IMPORTANCE

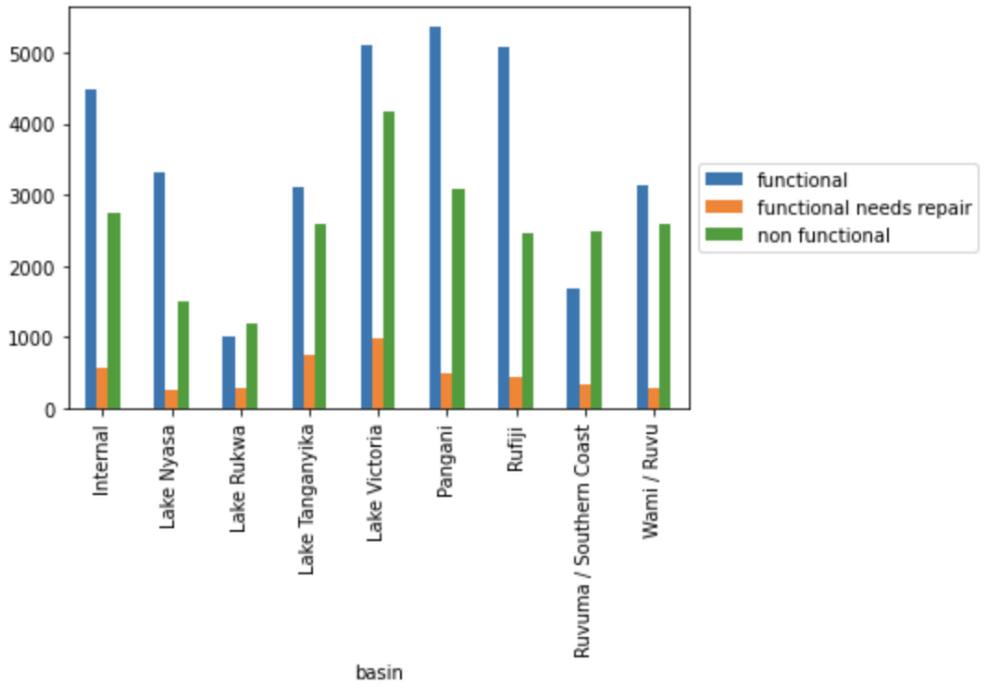
RandomForest has different feature importance for the same data



BASIN PERFORMANCE

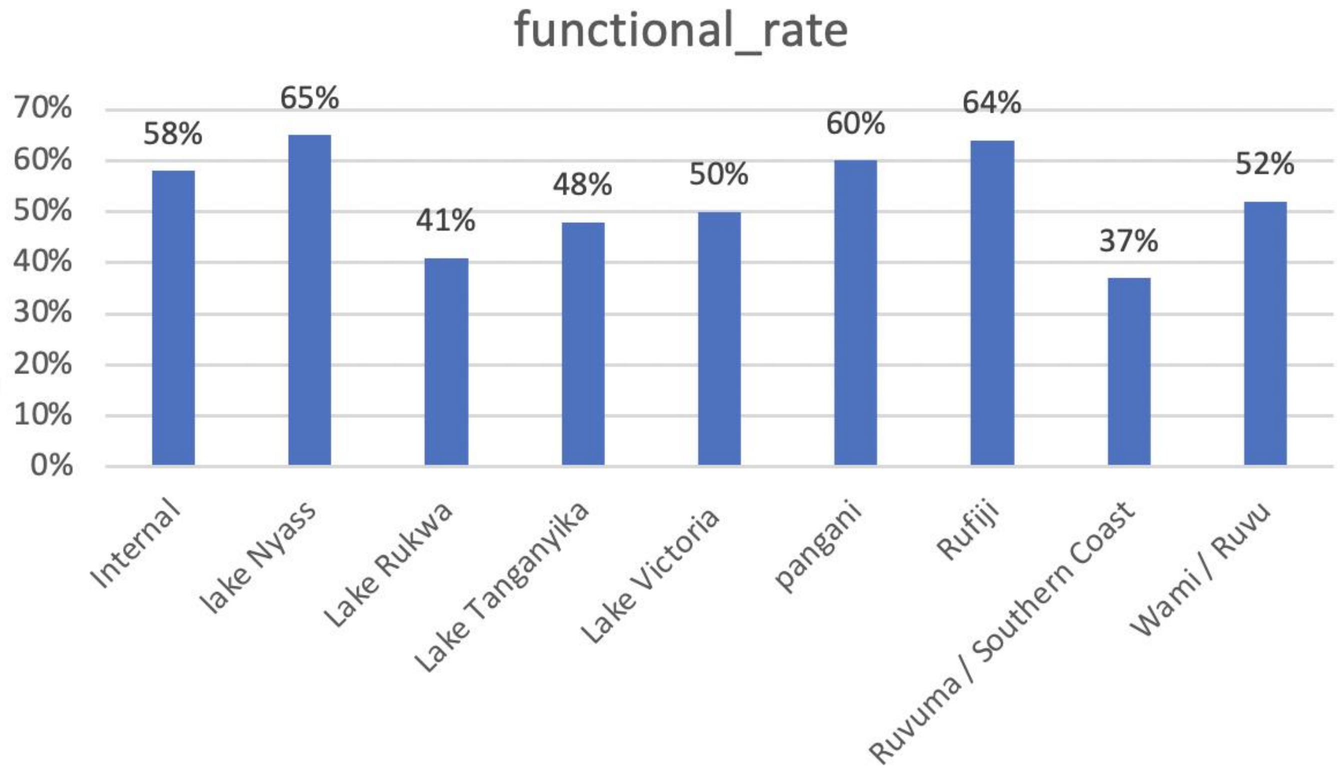
Top Basins

Top three basins with more functional water pumps: Lake Victoria, Pangani, Rufiji



Functional Rates

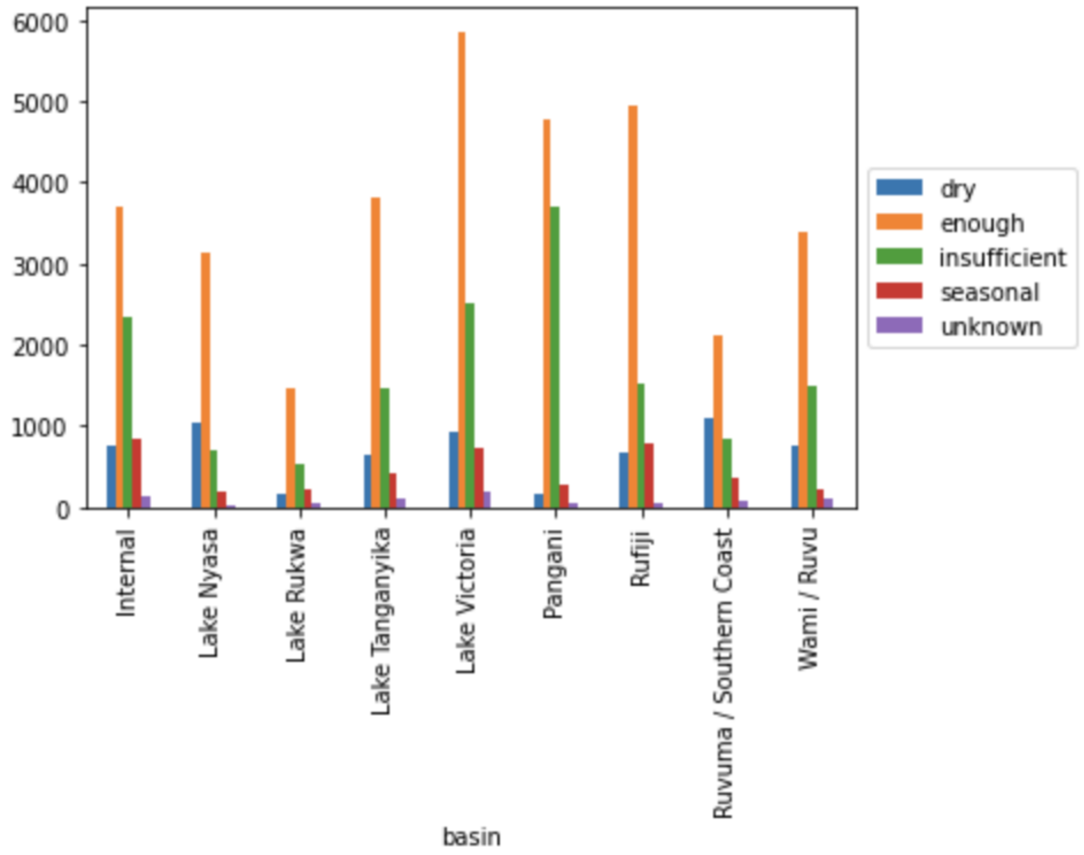
Top overall functional rate among basins: lake Nyass, Rufiji, Pangani



WATER QUANTITY

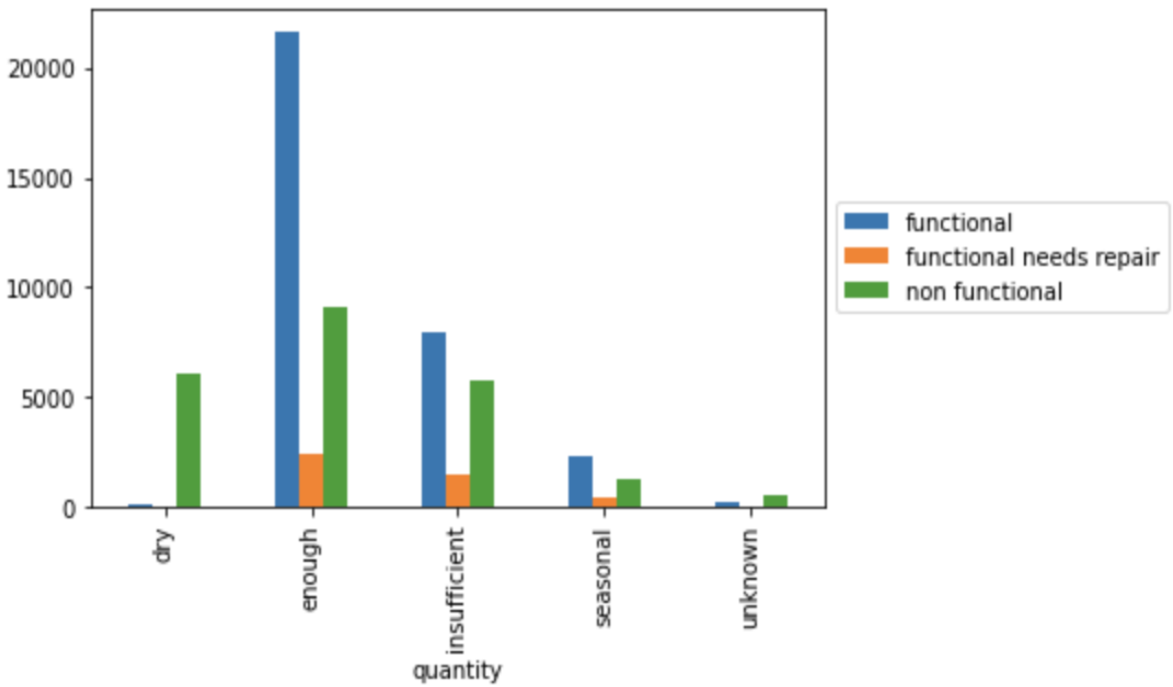
Top Basins

Top three basins with more water quantity:
Lake Victoria, Pangani, Rufiji



Water Quantity

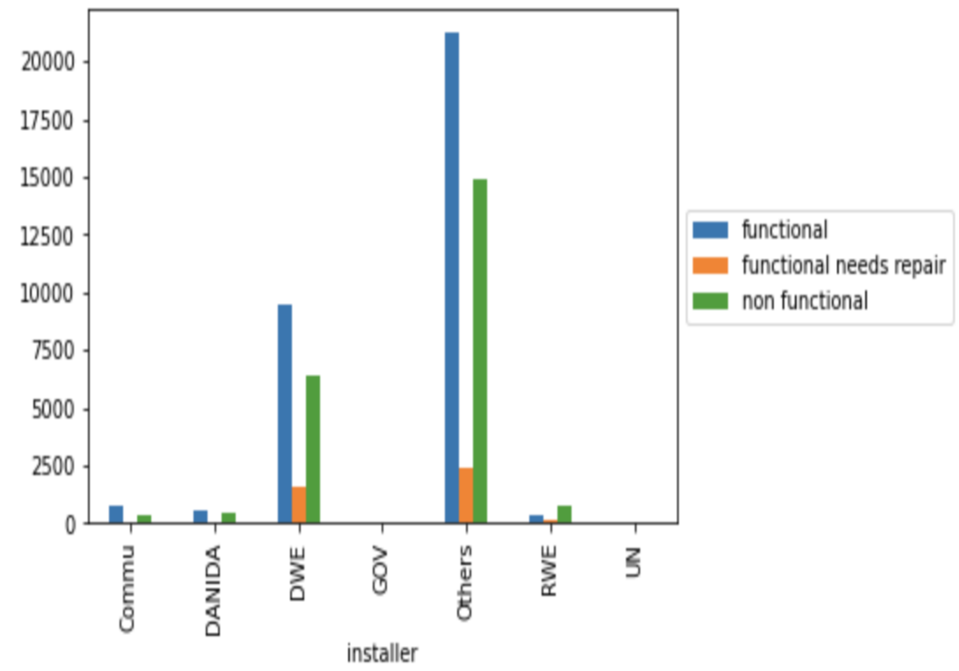
Basin with enough water has more functional water pumps



INSTALLER PERFORMANCE

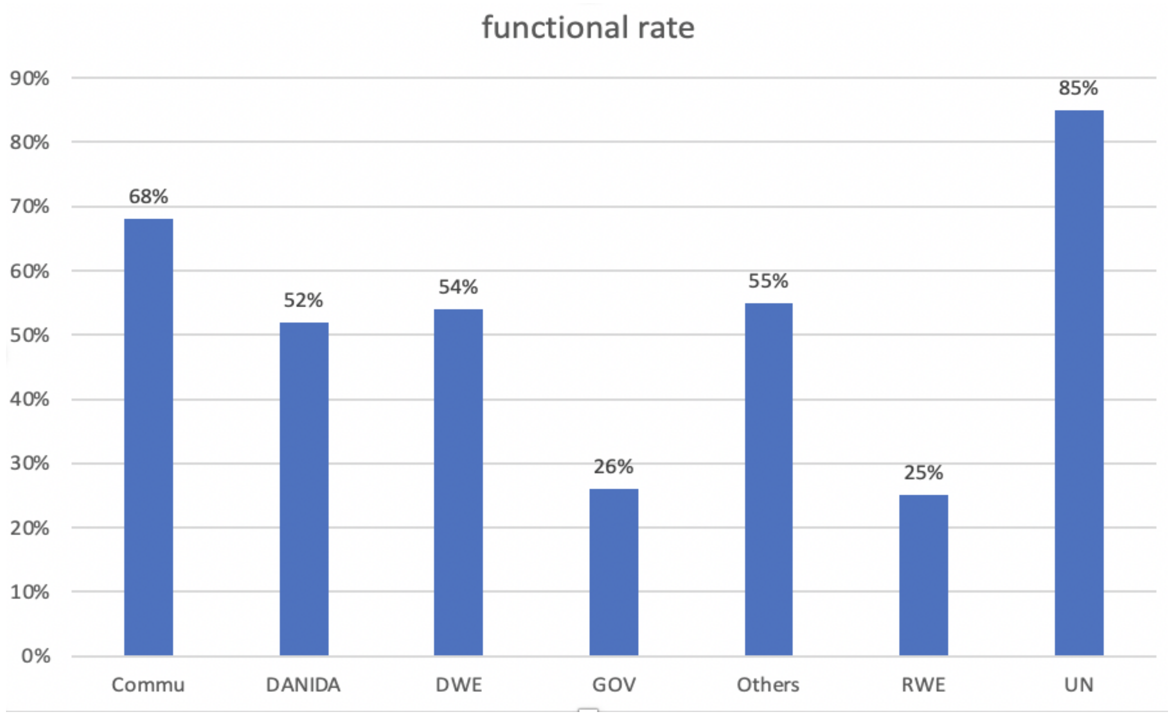
Top Installers

Top three installer with more functional water pumps: Others, DWE, Community



Functional Rates

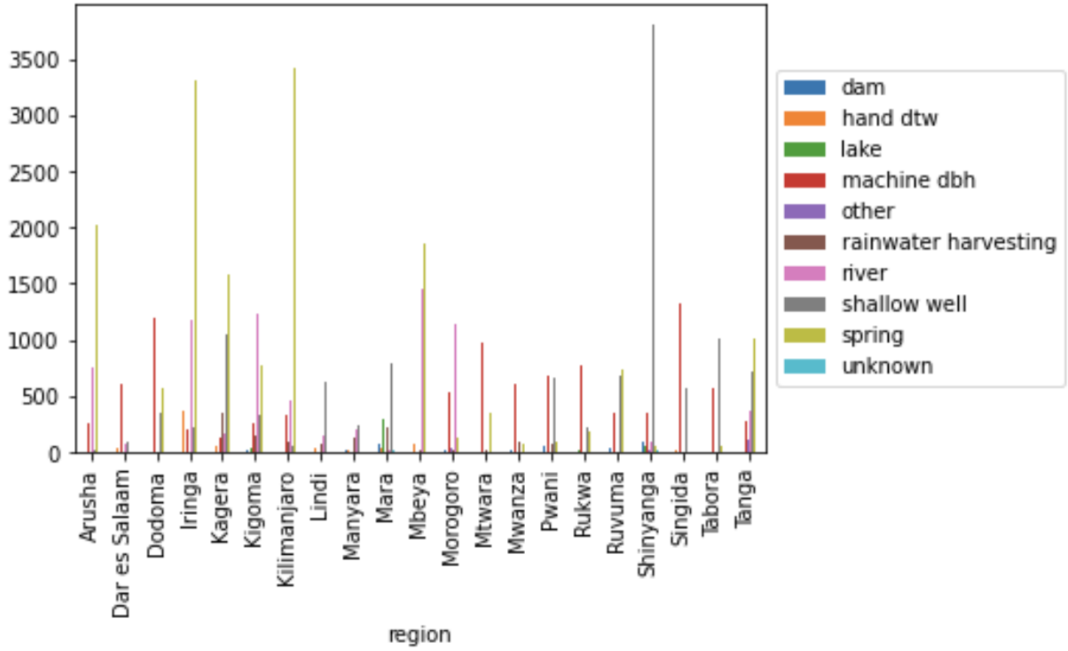
Top overall functional rate among installers: UN, Commu, DWE, Others



REGION WATER SOURCE & QUANTITY

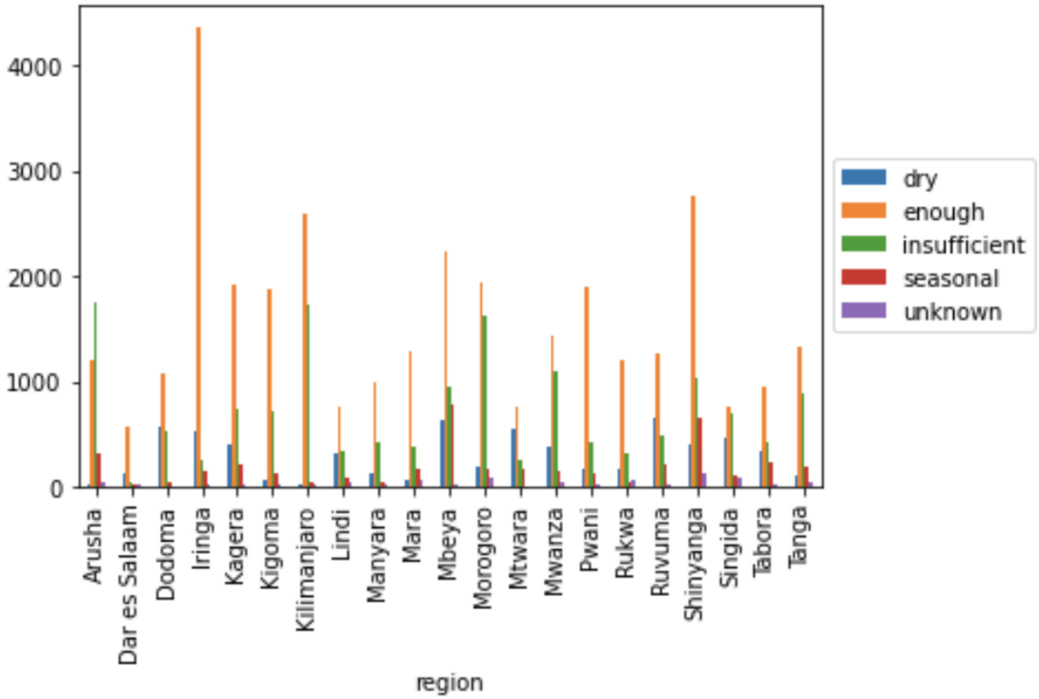
Top Water Sources

Top three water sources in regions: Spring, Shallow Well, Machine dbh



Top Region

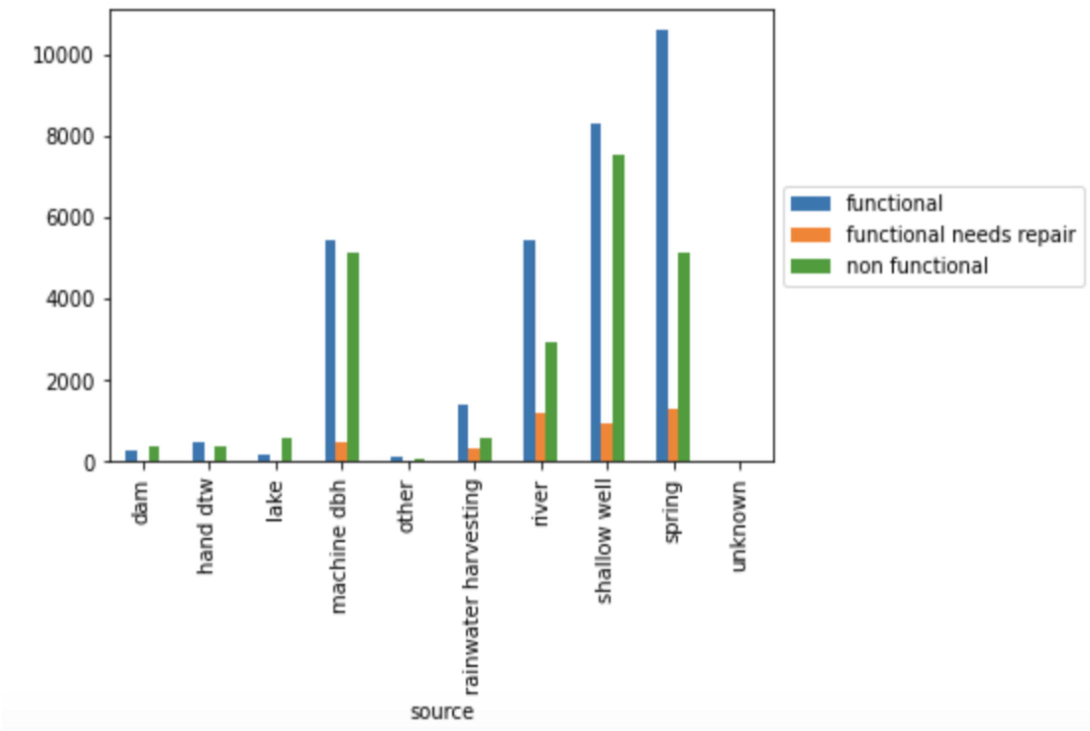
Top region with enough water quantity: Iringa, Shinyange, Kilimanjaro



WATER SOURCE

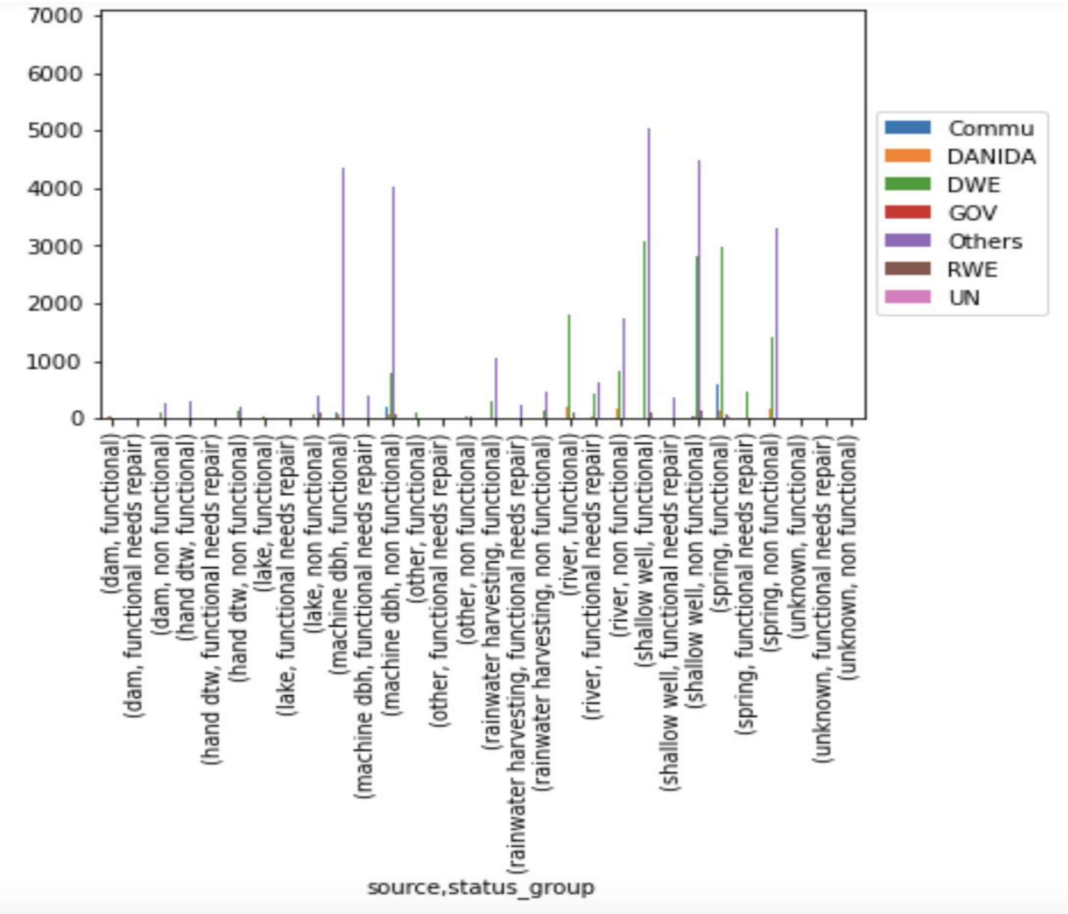
Water Source

Water Source with the most functional water pumps: Spring, Shallow Well, Rainwater Harvesting, Machine dbh



Status Group

DWE performed the best in constructing functional water pumps with different water sources



FINANCIAL VALUE OF MODEL

Confusion Matrix

Actual \ Predicted	Functional	Needs Repair	Not Functional	
Functional	7316	123	524	
Needs Repair	621	340	115	*42%*
Not Functional	1242	73	4496	

Repair and Replacement Costs ($)

Actual \ Predicted	Functional	Needs Repair	Not Functional
Functional	0	5	5
Needs Repair	1000	15	15
Not Functional	1000	1000	1000

Model Prediction Costs ($)

Actual \ Predicted	Functional	Needs Repair	Not Functional
Functional	-	615	2,620
Needs Repair	621,000	5,100	1,725
Not Functional	1,242,000	73,000	4,496,000

- The Confusion Matrix shows specificity of 42% for the 'Needs Repair' class.
- Specificity for the 'Needs Repair class is more useful in practical terms for the following reasons:
  - Research shows it is much cheaper to repair a pump than to replace a bad one ($10 vs $1000)
  - Ability to predict which pumps need repair before they need replacement, is the main value of the model
  - If the model predicts that a pump that needs repair is not functional, it will trigger a check on the pump that will likely result in the right assessment
  - Consequently, this category is also valuable.
  - We assume average logistics cost of $5 (50% of repair cost). For future work, we can calculated a weighted cost based on average distance of each well.

FINANCIAL VALUE OF MODEL

- Based on our best model's performance metrics and assumptions made, we estimated a current model value of ~$445K and life term value of ~$3.7M if the model is implemented.
  - Current Model Value is estimated using Confusion Matrix, assuming only 42% of existing pumps needing repair will be correctly predicted by the model
  - Lifetime (cumulative) Model Value assumes all pumps currently functional will eventually need repair and assumes a 42% accurate model prediction rate for this class
- Other Recommendations / Costs
  - Hiring one person per location to monitor state of pumps – initiate "Pump Playbook" program.
  - Install sensors on pumps (smart water pumps) in very remote areas where communication may be difficult
  - Implement a preventive maintenance program based on age of the pumps (future work)

Current Model Value

TOTAL COST	$'K	Comments
Ex Model	6,887	Cost of not using the model
With Model	6,442	Predicted costs if model is used
Model Value	445	Based on current state of pumps

Lifetime Model Value

TOTAL COST	$'K	
Ex Model	14,850	Assumes all will pumps eventually be replaced
With Model	11,174	Assumes model will detect 42% of pumps currently functional and needing repair
Model Value	3,676	Long term value based on current number of pumps

Based on our best model performance metrics and assumptions made, we have estimated an **immediate model value of ~$445K** and **medium/life term value of ~$3.7M** if the model is implemented

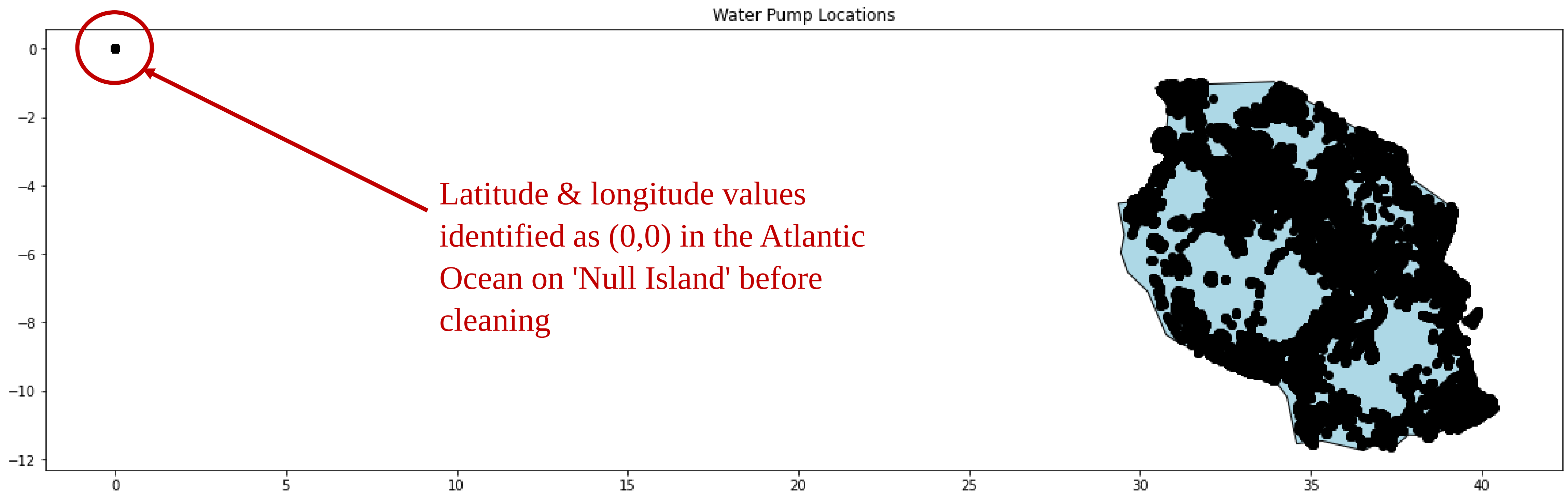
FINANCIAL ESTIMATES

No education: Average monthly wage of US$300 = $1.25/hour = $10/day*

Water engineer: Average monthly wage of US$623 = $3.90/hour = $31/day*

* TakeProfit.org, 2021

GEOPLOT: SHOWING LATITUDE / LONDITUDE DATA BEFORE CLEANING



IMPLEMENTATION CASE STUDY

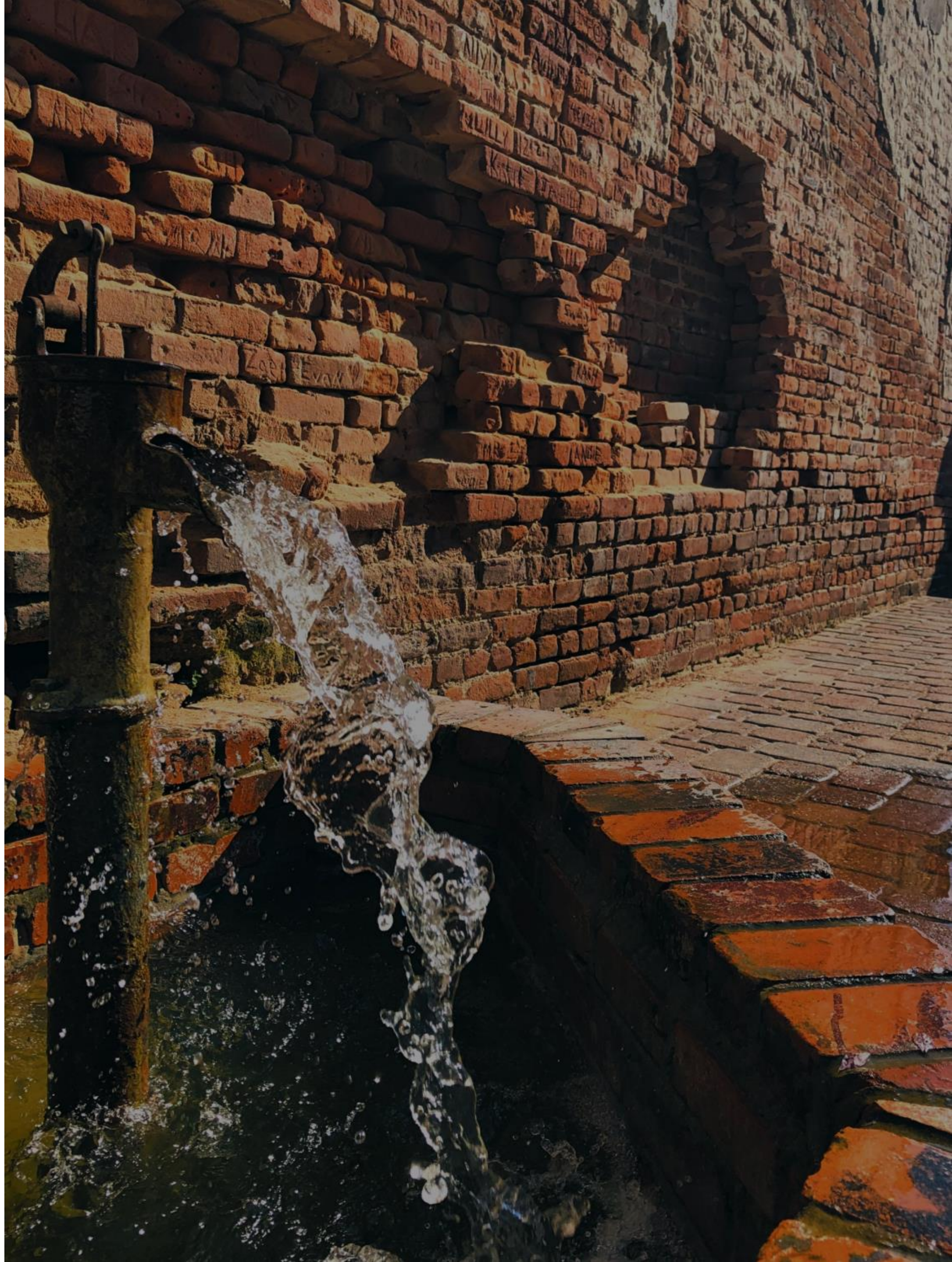
WaterWatchers

IBM & the city of Tshwane in South Africa piloted a crowdsourced app known as WaterWatchers

Users report water supply information via SMS

IBM found that the city was losing almost $30M in wasted water annually*

* Sustainable Brands, 2013



COUNTRY & POPULATION INFORMATION

Population

2020 total population of Tanzania: 59.73M*

- 21,381 records contain “0” value for population → this is unlikely → impute with mean
- Total population captured in dataset: 10,686,653 (from original data) + 3,848,580 (imputed) = 14,535,233 people (24.28% of Tanzanian population)
  - Average population around each well: 180 people
  - Assume ~50% average women = 7.3M women

Economy

- 2020 GDP of Tanzania: US$62.41B*
- Employ 1 lead woman per 20 well = 2,970 jobs @ US$8,600/year = US$25.5M per year
- Estimated that every $1 invested in water and sanitation programs yields up to $12 in economic returns = US$306.5M in yearly economic returns + 15-17 hours (2 days a week) per water-collecting woman

* World Bank, 2021
** The Water Project, 2021

FUTURE WORK

Build **custom function transformers** to be included in the model pipeline

GridSearchCV is not optimal – we are curious to try **Hyperopt library for hyperparameter optimization** and compare the results

NLP to identify and combine similar words in the categorical columns

Improved data collection

(see next slide)

FUTURE WORK: DATA COLLECTION

- Cost of implementing ML-derived preventative maintenance model compared to other models
- Performance of water pumps
- Payment tracking/ Payment Source
- XGBoost in Python, ML Enabled Smart Sensor System can be applied to monitor execution of water projects

LightGBM

```
[ ] from lightgbm import LGBMClassifier

    lgbm = LGBMClassifier(boosting_type = 'gbdt',
                          num_leaves = 200,
                          learning_rate = 0.05,
                          min_data_in_leaf = 20,
                          max_depth = 50,
                          objective = 'multiclass',
                          num_class = 3,
                          metric = 'multi_error',
                          bagging_fraction = 0.5,
                          num_iterations = 200)

    lgbm_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('lgbm', lgbm)])
    lgbm_pipeline.fit(X_train,y_train)
    y_pred_lgbm_pipeline = lgbm_pipeline.predict(X_test)

    print("Accuracy of LGBM    = {:.4f}".format(accuracy_score(y_test, y_pred_lgbm_pipeline)))
```

```
/usr/local/lib/python3.7/dist-packages/lightgbm/engine.py:118: UserWarning: Found `num_iter
  warnings.warn("Found `{}` in params. Will use it instead of argument".format(alias))
Accuracy of LGBM    = 0.8149
```

```
              precision    recall  f1-score   support

           0       0.80      0.91      0.85      7963
           1       0.60      0.33      0.42      1076
           2       0.87      0.78      0.82      5811

    accuracy                           0.81     14850
   macro avg       0.75      0.67      0.70     14850
weighted avg       0.81      0.81      0.81     14850
```

	0	1	2
0	7242	154	567
1	601	355	120
2	1219	87	4505

CatBoost

```
from catboost import CatBoostClassifier

cat = CatBoostClassifier(depth = 10,
                        iterations = 500,
                         learning_rate = 0.05,
                        random_state = 42)

cat_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('catboost', cat)])
cat_pipeline.fit(X_train,y_train)
y_pred_cat_pipeline = cat_pipeline.predict(X_test)
```

```
print("Accuracy of Catboost   = {:.4f}".format(accuracy_score(y_test, y_pred_cat_pipeline)))
```

```
Accuracy of Catboost   = 0.8042
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred_cat_pipeline))
pd.DataFrame(confusion_matrix(y_test, y_pred_cat_pipeline))
```

	precision	recall	f1-score	support
0	0.78	0.91	0.84	7963
1	0.65	0.26	0.37	1076
2	0.86	0.75	0.80	5811
accuracy			0.80	14850
macro avg	0.76	0.64	0.67	14850
weighted avg	0.80	0.80	0.79	14850

	0	1	2
0	7278	91	594
1	653	279	144
2	1364	62	4385

RandomForest

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(criterion = 'gini',
                            n_estimators = 1000,
                            random_state = 123,
                            min_samples_split = 5,
                            max_depth = 20)

rf_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('rf', rf)])
rf_pipeline.fit(X_train,y_train)
y_pred_rf_pipeline = rf_pipeline.predict(X_test)
```

```
print("Accuracy of RF = {:.4f}".format(accuracy_score(y_test, y_pred_rf_pipeline)))
```

```
Accuracy of RF = 0.8147
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred_rf_pipeline))
pd.DataFrame(confusion_matrix(y_test, y_pred_rf_pipeline))
```

```
              precision    recall  f1-score   support

           0       0.79      0.92      0.85      7963
           1       0.64      0.31      0.42      1076
           2       0.88      0.76      0.82      5811

    accuracy                           0.81     14850
   macro avg       0.77      0.67      0.70     14850
weighted avg       0.81      0.81      0.81     14850
```

	0	1	2
0	7318	120	525
1	630	336	110
2	1299	67	4445

XGBoost

```
xg = XGBClassifier(nthread=2,
                   num_class=3,
                   min_child_weight=3,
                   max_depth=15,
                   gamma=0.5,
                   scale_pos_weight=0.8,
                   subsample=0.7,
                   colsample_bytree = 0.8,
                   objective='multi:softmax')

xg_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('xgboost', xg)])
xg_pipeline.fit(X_train,y_train)
y_pred_xg_pipeline = xg_pipeline.predict(X_test)
```

```
print("Accuracy of XGB   = {:.4f}".format(accuracy_score(y_test, y_pred_xg_pipeline)))
```

```
Accuracy of XGB   = 0.8156
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred_xg_pipeline))
pd.DataFrame(confusion_matrix(y_test, y_pred_xg_pipeline))
```

	precision	recall	f1-score	support
0	0.80	0.90	0.85	7963
1	0.59	0.34	0.43	1076
2	0.86	0.78	0.82	5811
accuracy			0.82	14850
macro avg	0.75	0.67	0.70	14850
weighted avg	0.81	0.82	0.81	14850

	0	1	2
0	7190	164	609
1	589	361	126
2	1164	86	4561

VotingClassifier

```
from sklearn.ensemble import VotingClassifier

est_list = [('rf', rf), ('xgboost', xg), ('extra trees', xt), ('bagging', bag), ('catboost', cat)]
vclf = VotingClassifier(estimators = est_list, voting='soft')

vote_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('voting', vclf)])

vote_pipeline.fit(X_train,y_train)
y_pred_vote_pipeline = vote_pipeline.predict(X_test)
```

```
print("Accuracy of VOTING = {:.4f}".format(accuracy_score(y_test, y_pred_vote_pipeline)))
```

```
Accuracy of VOTING = 0.8191
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred_vote_pipeline))
pd.DataFrame(confusion_matrix(y_test, y_pred_vote_pipeline))
```

```
              precision    recall  f1-score   support

           0       0.80      0.92      0.85      7963
           1       0.63      0.32      0.42      1076
           2       0.88      0.77      0.82      5811

    accuracy                           0.82     14850
   macro avg       0.77      0.67      0.70     14850
weighted avg       0.82      0.82      0.81     14850
```

	0	1	2
0	7316	123	524
1	621	340	115
2	1242	73	4496

Stacking

```python
from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression

est_list = [('rf', rf), ('xgboost', xg), ('extra trees', xt), ('bagging', bag), ('catboost', cat)]

sclf = StackingClassifier(estimators = est_list,
                          final_estimator = LogisticRegression())

stacking_pipeline = Pipeline(steps = [('preprocess', preprocessor), ('stacking', sclf)])

stacking_pipeline.fit(X_train,y_train)
y_pred_stacking_pipeline = stacking_pipeline.predict(X_test)
```

```python
print("Accuracy of STACKING = {:.4f}".format(accuracy_score(y_test, y_pred_stacking_pipeline)))
```

```
Accuracy of STACKING = 0.8199
```

```python
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred_stacking_pipeline))
pd.DataFrame(confusion_matrix(y_test, y_pred_stacking_pipeline))
```

```
              precision    recall  f1-score   support

           0       0.81      0.91      0.85      7963
           1       0.64      0.33      0.43      1076
           2       0.86      0.79      0.82      5811

    accuracy                           0.82     14850
   macro avg       0.77      0.67      0.70     14850
weighted avg       0.82      0.82      0.81     14850
```

	0	1	2
0	7238	129	596
1	594	350	132
2	1153	70	4588

REFERENCES

Average Salary Survey. (2021). Tanzania. Retrieved from https://www.averagesalarysurvey.com/Tanzania.

DrivenData. (2021). Pump it Up: Data Mining the Water Table. Retrieved from DrivenData: https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/.

Guardian. (2016). How do you solve a problem like a broken water pump?. Retrieved: https://www.theguardian.com/global-development-professionals-network/2016/mar/22/how-do-you-solve-a-problem-like-a-broken-water-pump.

Joseph, G. (2019). Why Do So Many Water Points Fail in Tanzania? An Empirical Analysis of Contributing Factors. World Bank Group.

Lifewater.org. (2021). The Tanzania Water Crisis: Facts, Progress, and How to Help. Retrieved from: https://lifewater.org/blog/tanzania-water-crisis-facts/.

Purvis. (2016). 13 Ways to Provide Water and Sanitation for Nine Billion People. The Guardian. Retrieved from https://www.theguardian.com/global-development-professionals-network/2015/jul/14/water-sanitation-scarcity-population-growth-summary.

SimplePump. (2021). Reliable Clean Water in Developing Nations. Retrieved from https://simplepump.com/reliable-clean-water-in-developing-nations/.

Sustainable Brands. (2013). IBM's Water Watchers Gives South African Citizens Power Over Water Challenges. Retrieved from https://sustainablebrands.com/read/ict-and-big-data/ibm-s-water-watchers-app-gives-south-african-citizens-power-over-water-challenges.

TakeProfit.org. (2021). Tanzania. Retrieved from https://take-profit.org/en/statistics/wages/tanzania/.

Water.org. (2021). Tanzania. Retrieved from: https://water.org/our-impact/where-we-work/tanzania/.

The Water Project. (2021). Water Crisis. Retrieved from https://thewaterproject.org/why-water/water-crisis.

World Bank. (2021). Tanzania. Retrieved from https://www.worldbank.org/en/country/Tanzania.

UNDP. (2015). Sustainable Development Goals. Retrieved from https://www.undp.org/sustainable-development-goals.

United Nations. (2007). Bringing Water to Africa's Poor. Retrieved from https://www.un.org/africarenewal/magazine/october-2007/bringing-water-africa%E2%80%99s-poor.

Zientala, P. (2017). Water Point Mapping in Tanzania Using a Machine Learning Approach. Department of Geography, University of Southhampton, UK.