PUMP IT UP:

DATA MINING THE WATER TABLE

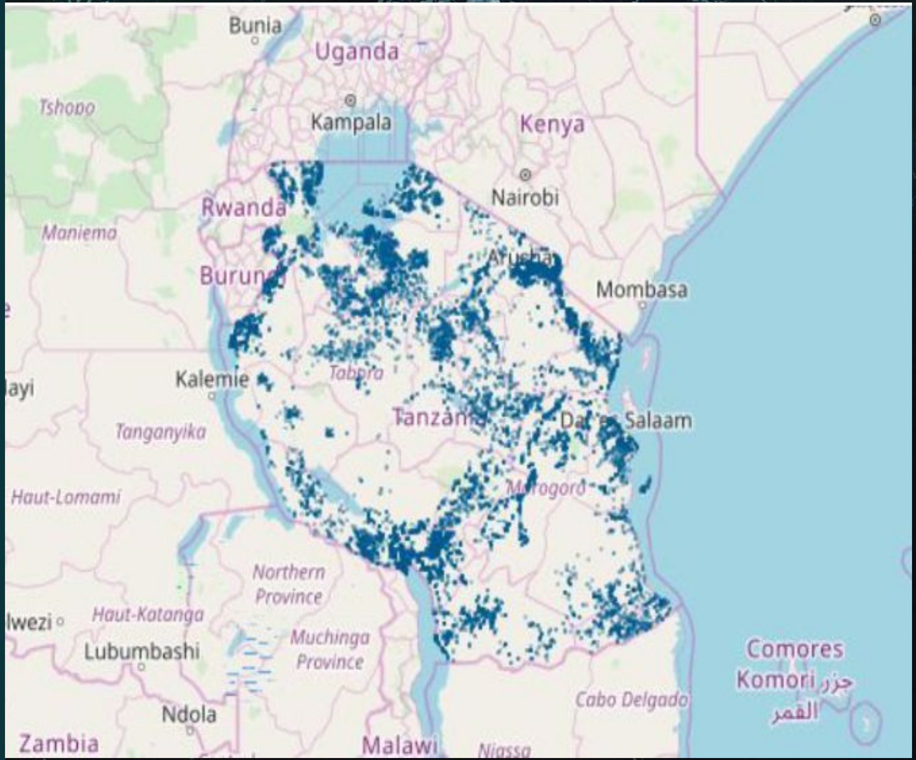
By: Rose wairimu Kimondo
DSF-PT07P3

Predicting Water Well Functionality in Tanzania

Problem

Tanzania, as a developing country, struggles with providing clean water to its population of over 57,000,000.

There are many water points already established in the country, but some are in need of repair while others have failed altogether



Main Goal

Build a classifier that predicts the condition of a water well (functional, non-function, or functional but needs repair)

Use information such as the extraction type, how it is managed, payment type, waterpoint type, the water source, whether it has a permit, and whether a public meeting was held.

Help the Government of Tanzania find patterns in non-functional wells to influence how new wells are built.

Objectives

1. Analyze the relationship between the following variables and the `status_group`(functional, non-functional, functional but needs repair) to identify patterns in non-functional wells:

- `Payment`
- `source`
- `management_group`
- `extraction_type`
- `permit`
- `public _meeting`

2. Develop a classification model to predict the condition of a well (functional, non-functional, or non-functional but needs repair)

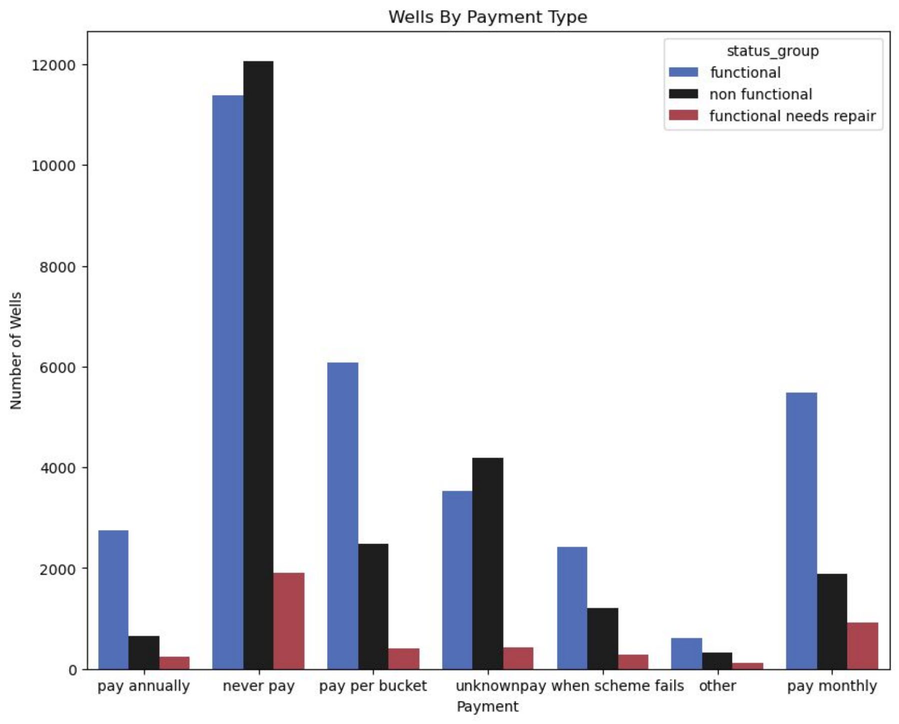
Data And Its Limitations

- Data: “Pump It Up - Data Mining the Water Table” from here
- 59,400 records with 40 features of water wells in Tanzania
-
- Many categorical features which were OneHotEncoded for modeling
- Large target classification imbalance- Wetarget encoded the labels
- Ternary classification: Three targets(functional, non-functional, functional but needs repair)
- Modeled 15 features (descriptions here) basin, month_recorded, region, extraction_type_class, management_group, payment, quality_group, quantity, source, waterpoint_type, gps_height, population, construction_age, permit, public_meeting

Payment

Wells with no fee are more likely to be non functional or need repair.

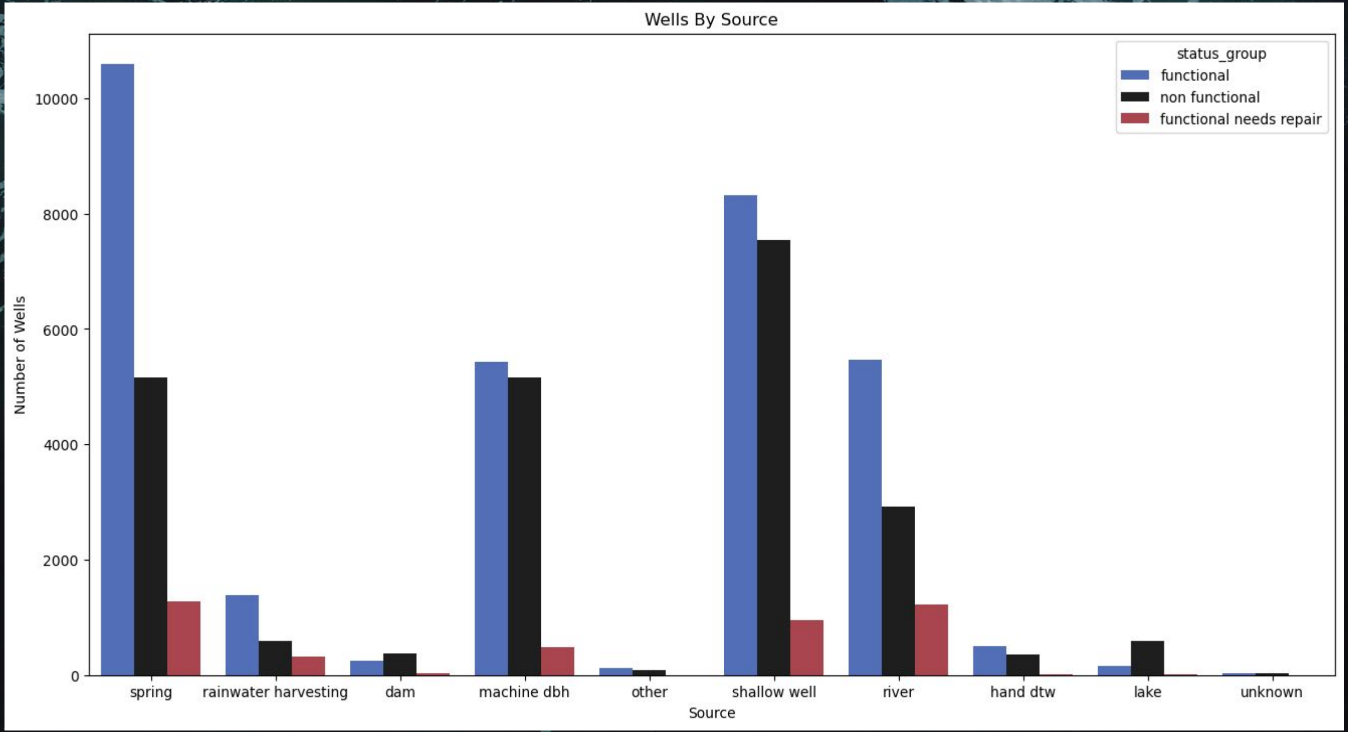
Wells by Payment Type



Water Source

Most of the non-functional wells and those that need repair have a shallow well as their water source.

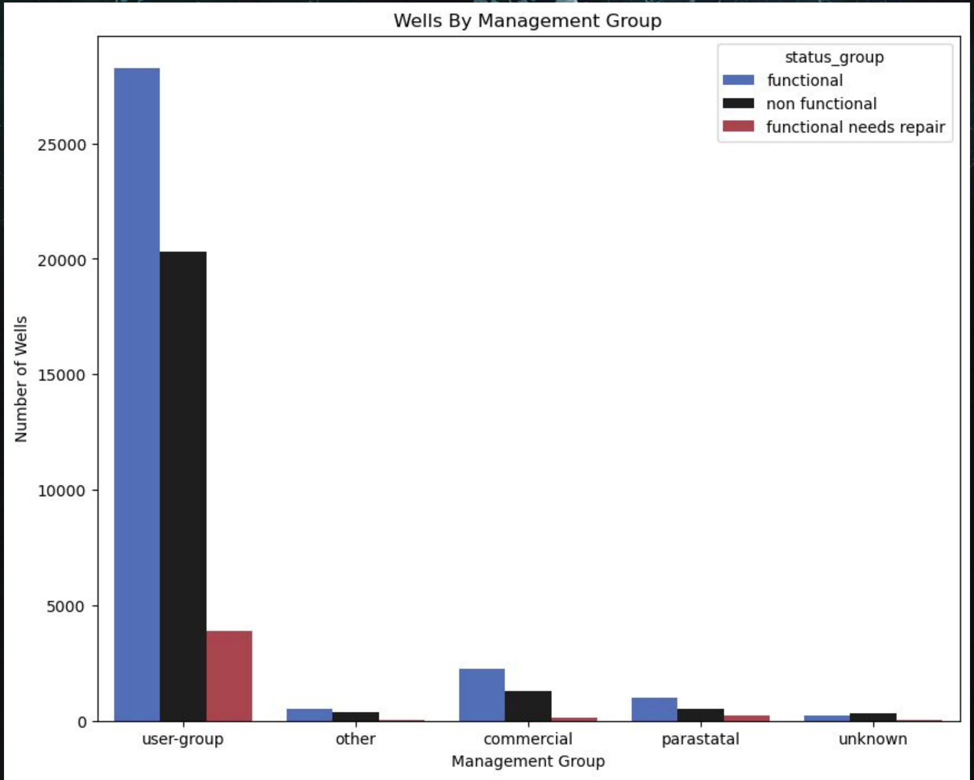
Wells by Water Source



Management Group

Most of the non-functional wells and those that need repairs are managed by the user group

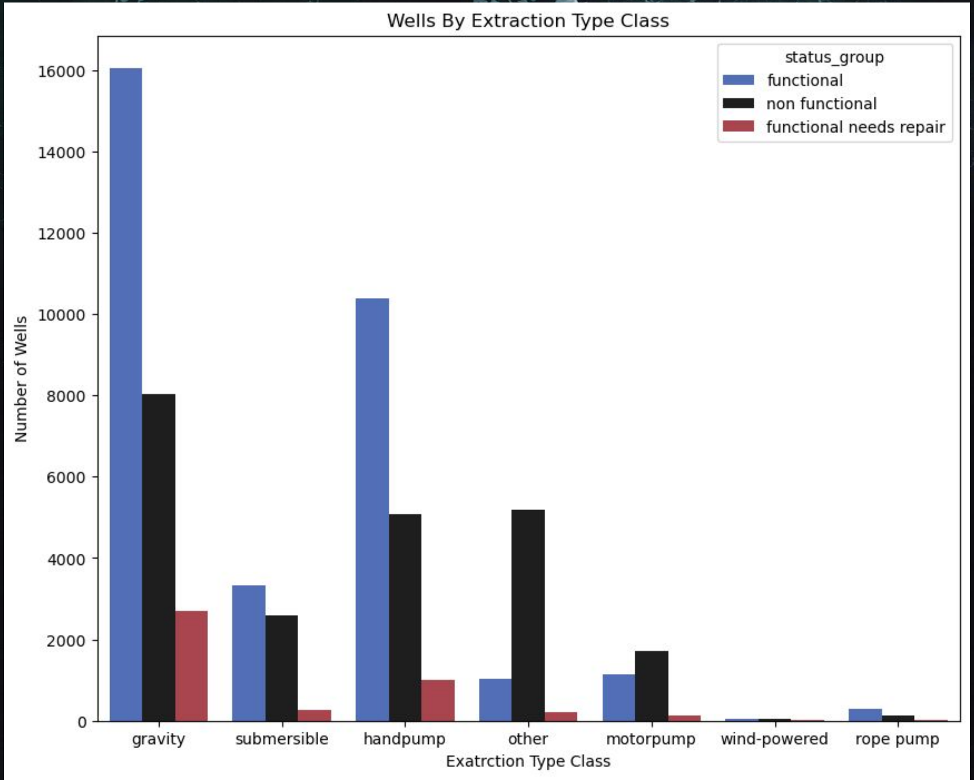
Wells by Management Group



Extraction Type Class

Most of the non-functional wells and those that need repair use gravity as the extraction type.

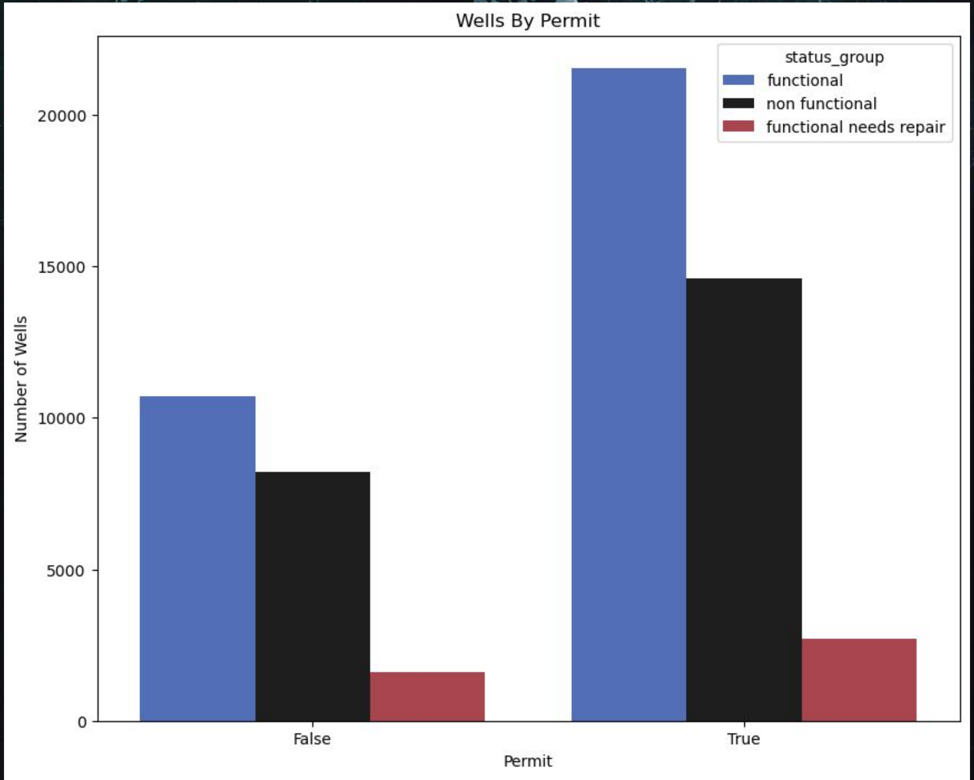
Wells by Extraction Type Class



Permits

Most of the non-functional wells and those that need repair are permitted.

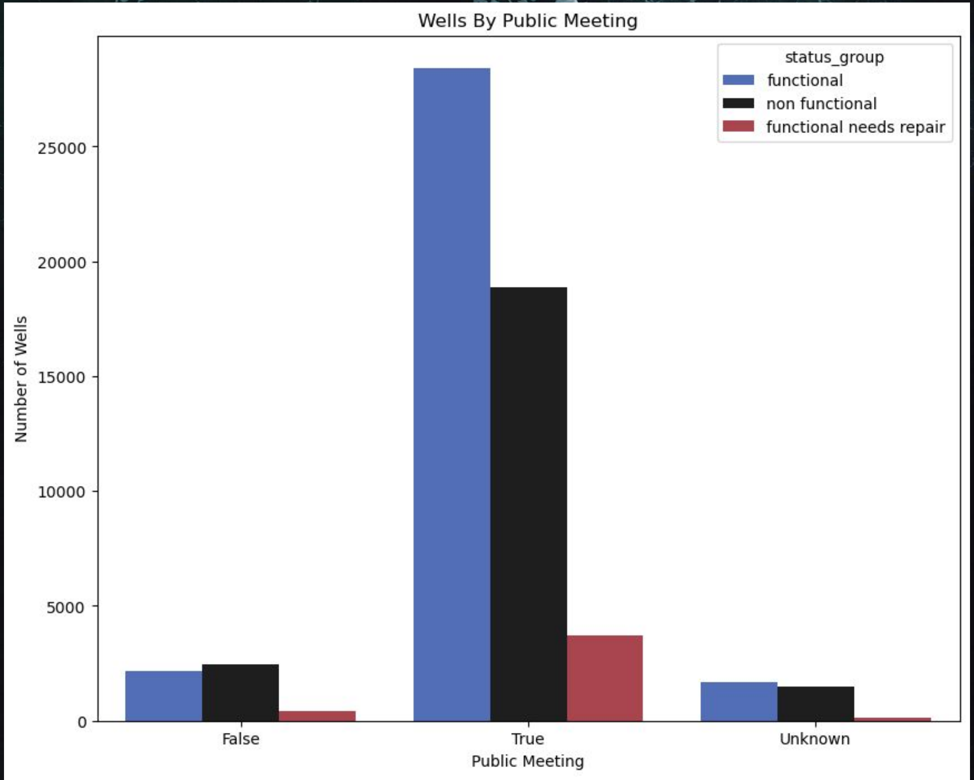
Wells by Permit



Public Meeting

Most of the non-functional wells and those that need repair had a public meeting held.

Wells by Public Meeting



Conclusion: Patterns in Non-Functional Wells and Those that Need Repairs

Wells likely to be non-functional or needing repair:

Payments: Wells where no payments are made
Water source: Wells with a shallow well as the water source
Management Group: Wells managed by the user group
Extraction type: Wells with gravity as the extraction type
Permit: Wells that are permitted
Public meeting: Wells where a public meeting was held

Models

Model	Type	Accuracy	Precision	Recall	F1 score
Model 1	Logistic Regression - Baseline model with default parameters and MinMax Scaling	73%	72%	73%	70%
Model 2	Logistic Regression with SMOTE OverSampling	62.3%	74%	62%	66%
Model 3	Logistic Regression with Standard Scaling	73%	72%	73%	70%
Model 4	Decision Tree(Default parameters Gini Criterion and random state 42)	75.7%	75%	76%	75.4%
Model 5	Decision Tree (Default parameters except entropy criterion and max depth = 14)	75.8%	75%	76%	74.3%

Conclusion: Best Model

Decision Tree (Default parameters except entropy criterion and max depth = 14)

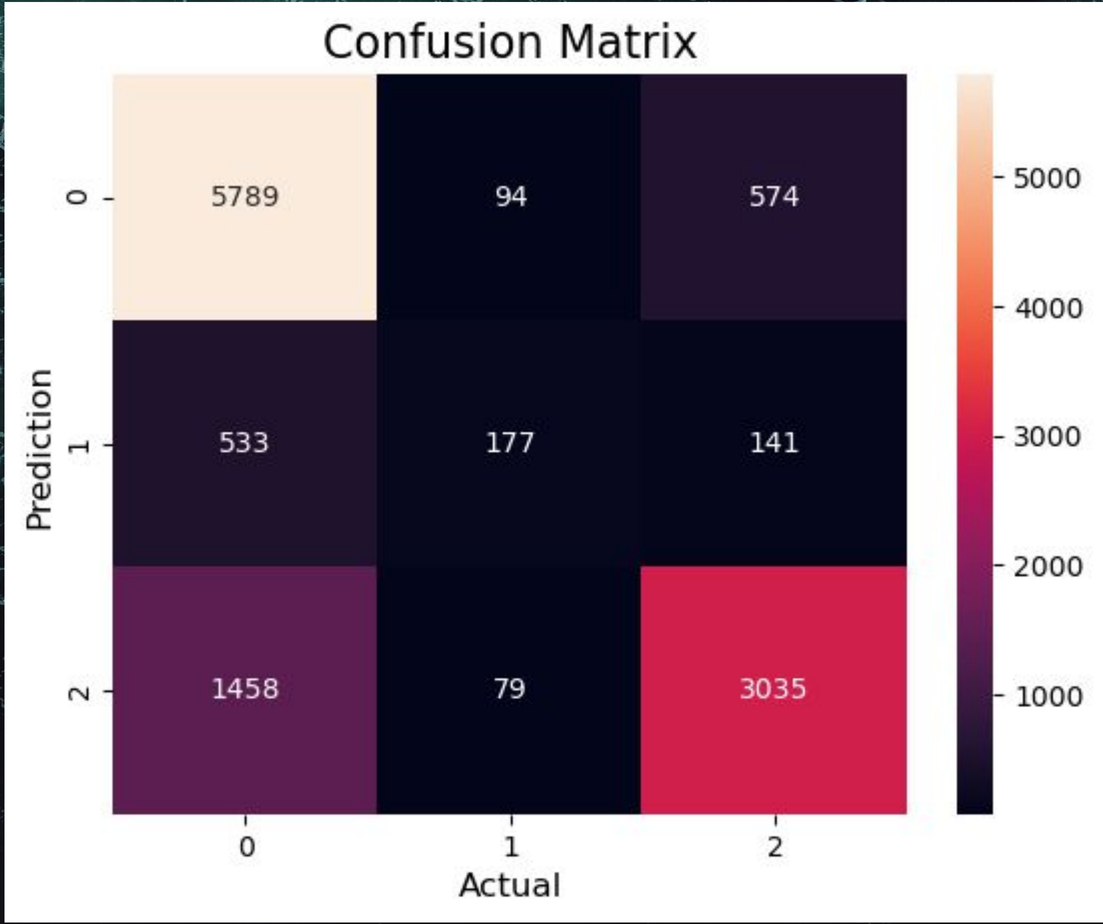
- **Hyperparameter tuning:**
  Changed from Gini to Entropy Criterion
  Used a depth of 14 levels
- **Best score:**
  Entropy criterion, max depth = 14
- **Accuracy: 75.8%**
- **Precision: 75%**
- **Recall: 76%**

Results:
Confusion Matrix

0: Functional

1: functional needs repair

2: non-functional



Recommendations

- Consider engineering the ternary classification problem into a binary classification problem and see if this improves the model parameters: Compare ternary versus binary classification models.

- More feature engineering to identify other features that could be useful for the model.

- Try other models like the Random Forest Classifier, KNN, and AdaBoost to see if they have better metrics