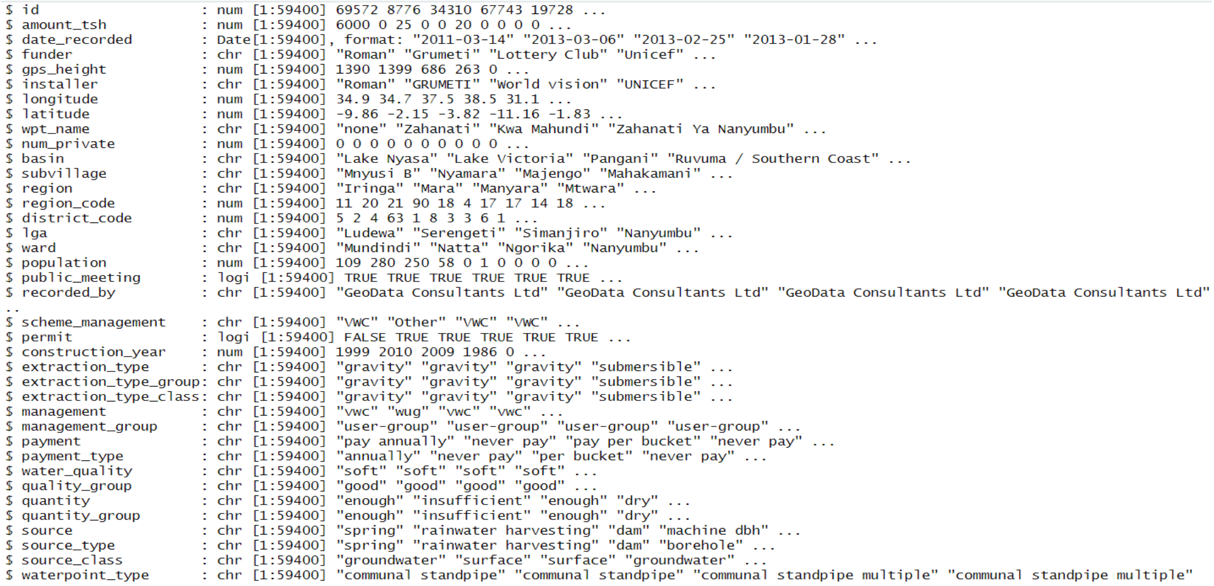
# OPIM390 Term Project Neat Report

# Barış Serhat Kaplan 30786

In many developing regions water sources and their maintenance is very crucial for these regions, rural areas of Tanzania is the given example for us for this project, the sustainability of water sources is a major concern due to infrastructure degradation and lack of maintenance. Identifying whether a water point is functional, needs repair, or is non-functional is important for effective resource allocation and planning. However, manual inspection of thousands of water points is costly and time-consuming. This project aims to build a predictive model using machine learning techniques to classify the functionality status of water pumps based on features such as geographic location, construction year, installer, water quality, and management scheme and other important features. The goal is to support governmental and non-governmental organizations in prioritizing maintenance and ensuring access to clean water with the labelling system which I did in the project.

## Exploratory Data Analysis

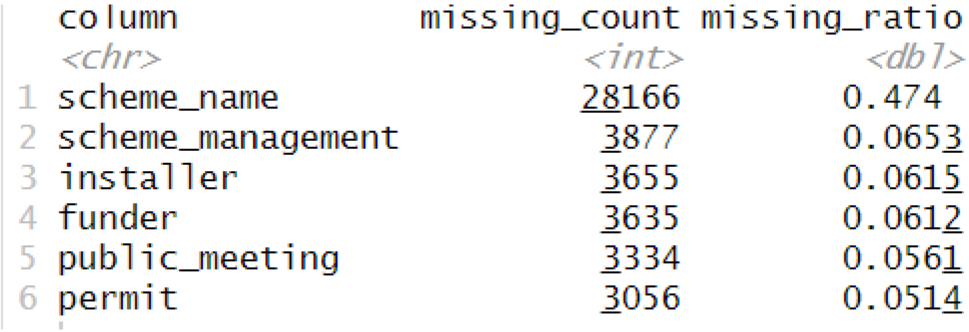
Our data has 40 columns and 59400 observations, so I  need to do analysis for data because data analysis gives us further information for data preprocessing and feature selection. I checked and controlled the features, data types of features, dimensions of our files and na/null values in our data because I need to fill them appropriately for our model.



```
$ id                   : num [1:59400] 69572 8776 34310 67743 19728 ...
$ amount_tsh           : num [1:59400] 6000 0 25 0 0 20 0 0 0 0 ...
$ date_recorded        : Date[1:59400], format: "2011-03-14" "2013-03-06" "2013-02-25" "2013-01-28" ...
$ funder               : chr [1:59400] "Roman" "Grumeti" "Lottery Club" "Unicef" ...
$ gps_height           : num [1:59400] 1390 1399 686 263 0 ...
$ installer            : chr [1:59400] "Roman" "GRUMETI" "World vision" "UNICEF" ...
$ longitude            : num [1:59400] 34.9 34.7 37.5 38.5 31.1 ...
$ latitude             : num [1:59400] -9.86 -2.15 -3.82 -11.16 -1.83 ...
$ wpt_name             : chr [1:59400] "none" "Zahanati" "Kwa Mahundi" "Zahanati Ya Nanyumbu" ...
$ num_private          : num [1:59400] 0 0 0 0 0 0 0 0 0 0 ...
$ basin                : chr [1:59400] "Lake Nyasa" "Lake Victoria" "Pangani" "Ruvuma / Southern Coast" ...
$ subvillage           : chr [1:59400] "Mnyusi B" "Nyamara" "Majengo" "Mahakamani" ...
$ region               : chr [1:59400] "Iringa" "Mara" "Manyara" "Mtwara" ...
$ region_code          : num [1:59400] 11 20 21 90 18 4 17 17 14 18 ...
$ district_code        : num [1:59400] 5 2 4 63 1 8 3 3 6 1 ...
$ lga                  : chr [1:59400] "Ludewa" "Serengeti" "Simanjiro" "Nanyumbu" ...
$ ward                 : chr [1:59400] "Mundindi" "Natta" "Ngorika" "Nanyumbu" ...
$ population           : num [1:59400] 109 280 250 58 0 1 0 0 0 0 ...
$ public_meeting       : logi [1:59400] TRUE TRUE TRUE TRUE TRUE TRUE ...
$ recorded_by          : chr [1:59400] "GeoData Consultants Ltd" "GeoData Consultants Ltd" "GeoData Consultants Ltd" "GeoData Consultants Ltd"
..
$ scheme_management    : chr [1:59400] "VWC" "Other" "VWC" "VWC" ...
$ permit               : logi [1:59400] FALSE TRUE TRUE TRUE TRUE TRUE ...
$ construction_year    : num [1:59400] 1999 2010 2009 1986 0 ...
$ extraction_type      : chr [1:59400] "gravity" "gravity" "gravity" "submersible" ...
$ extraction_type_group: chr [1:59400] "gravity" "gravity" "gravity" "submersible" ...
$ extraction_type_class: chr [1:59400] "gravity" "gravity" "gravity" "submersible" ...
$ management           : chr [1:59400] "vwc" "wug" "vwc" "vwc" ...
$ management_group     : chr [1:59400] "user-group" "user-group" "user-group" "user-group" ...
$ payment              : chr [1:59400] "pay annually" "never pay" "pay per bucket" "never pay" ...
$ payment_type         : chr [1:59400] "annually" "never pay" "per bucket" "never pay" ...
$ water_quality        : chr [1:59400] "soft" "soft" "soft" "soft" ...
$ quality_group        : chr [1:59400] "good" "good" "good" "good" ...
$ quantity             : chr [1:59400] "enough" "insufficient" "enough" "dry" ...
$ quantity_group       : chr [1:59400] "enough" "insufficient" "enough" "dry" ...
$ source               : chr [1:59400] "spring" "rainwater harvesting" "dam" "machine dbh" ...
$ source_type          : chr [1:59400] "spring" "rainwater harvesting" "dam" "borehole" ...
$ source_class         : chr [1:59400] "groundwater" "surface" "surface" "groundwater" ...
$ waterpoint_type      : chr [1:59400] "communal standpipe" "communal standpipe" "communal standpipe multiple" "communal standpipe multiple"
```

*Fig01. Data types of our Train Data*

Data types are crucial to analysis categorical, numeric, and integer values because I  convert categorical values to factors.

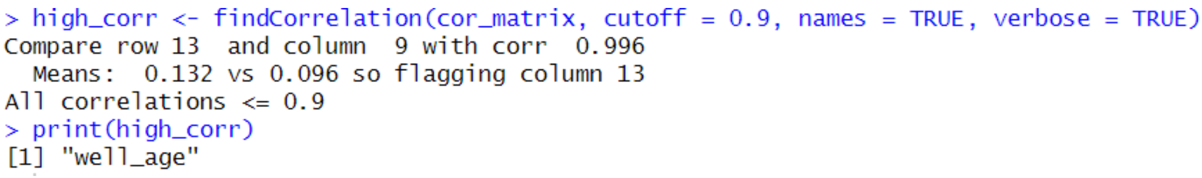


```
  column              missing_count missing_ratio
  <chr>                       <int>         <dbl>
1 scheme_name                 28166        0.474
2 scheme_management            3877        0.0653
3 installer                    3655        0.0615
4 funder                       3635        0.0612
5 public_meeting               3334        0.0561
6 permit                       3056        0.0514
```

*Fig02. Na values of Train Data*

## Data Preprocessing and Feature Selection

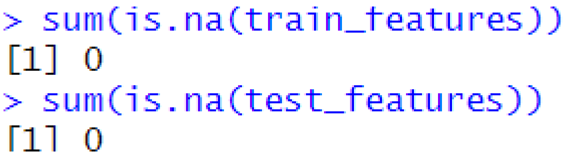
Models cannot predict the NA/null values I need to fill them or removed. According to weights of features. I decided to remove scheme_name, filled scheme_management, installer and funder missing values with unknown and permit and public_meeting's are binary decisions ı filled with mode of other observations. Next, I merged the training features with labels using a left join on the id column and performed feature engineering. From the date_recorded column, I extracted year, month, and day, and calculated well_age by subtracting the construction_year from the year of recording. To handle zero or missing construction years, I treated them as NA and imputed the missing values with the median well age. I then converted all character columns (excluding the target label) into categorical (factor) types, a critical step for classification models like Random Forest and logistic regression.



```
> high_corr <- findCorrelation(cor_matrix, cutoff = 0.9, names = TRUE, verbose = TRUE)
Compare row 13  and column  9 with corr  0.996 
  Means:  0.132 vs 0.096 so flagging column 13 
All correlations <= 0.9 
> print(high_corr)
[1] "well_age"
```

*Fig04. High Correlation Control*

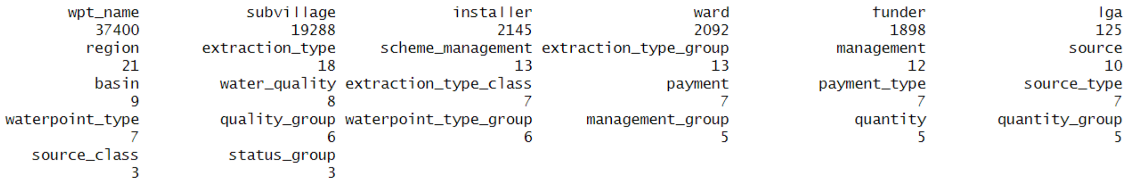
Additionally, i reduced the feature space by removing near-zero variance variables using the nearZeroVar() function and eliminated highly correlated numeric features based on a correlation matrix.



```
> sum(is.na(train_features))
[1] 0
> sum(is.na(test_features))
[1] 0
```

*Fig04. Final NA values*

Lastly, to prevent overfitting and reduce complexity, i dropped categorical variables with high cardinality (too many unique values), such as subvillage, funder, lga, wpt_name, installer, and ward. These steps helped me build models.

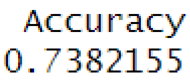


```
             wpt_name            subvillage             installer                  ward                funder                   lga
                37400                 19288                  2145                  2092                  1898                   125
               region       extraction_type     scheme_management extraction_type_group            management                source
                   21                    18                    13                    13                    12                    10
                basin         water_quality extraction_type_class               payment          payment_type           source_type
                    9                     8                     7                     7                     7                     7
      waterpoint_type         quality_group waterpoint_type_group      management_group              quantity        quantity_group
                    7                     6                     6                     5                     5                     5
         source_class          status_group
                    3                     3
```

*Fig05. Cardinalities*

## Model Assessment

Training data which is combined with train set and train label datasets, is ready after analysis, feature selection and preprocessing. Combined Train Data has 0 NA values and just have crucial features for prediction. Firstly, I tried logistic regression model. However, the logistic regression model yielded a significantly lower accuracy (~73.8%). This is mainly due to its inability to capture complex and nonlinear relationships in the data. Given the presence of many categorical variables and the multiclass nature of the target variable, logistic regression underperforms.



```
 Accuracy 
0.7382155 
```

*Fig06. Accuracy of Logistic regression model*

In contrast, Random Forest handles categorical features and interactions more effectively, making it a more suitable choice for this classification task, so I tried random forest model. I analyzed importance of every features in random forest model with Gini Index.

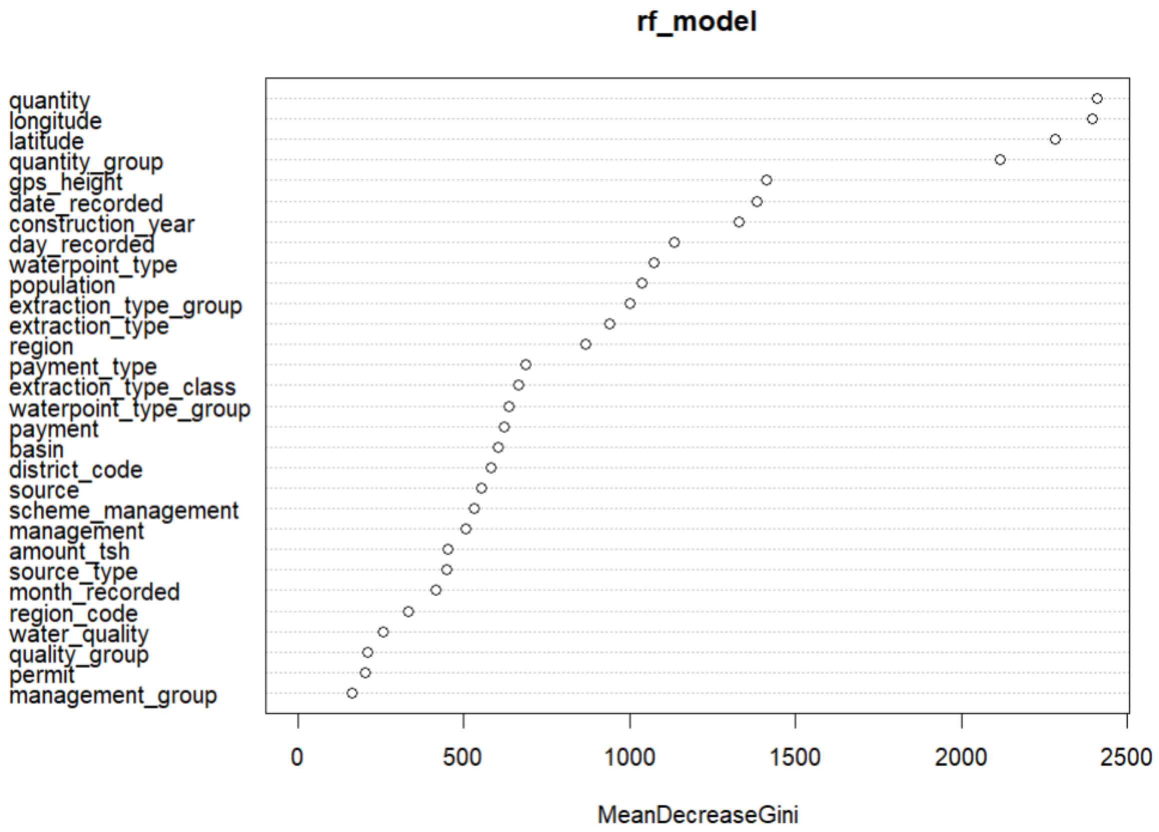
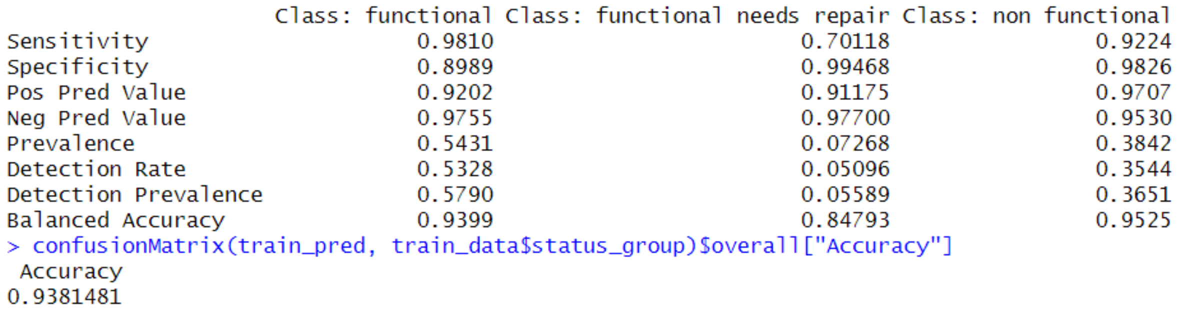


*Fig07. Variance importance of Features*

Also, I built the model and test it for every label. The Random Forest model demonstrated strong overall performance with an accuracy of **93.8%**, indicating high predictive power on the training dataset. Sensitivity (recall) was particularly high for the “functional” (0.9810) and “non functional” (0.9224) classes, showing the model's effectiveness in correctly identifying those categories. Although the sensitivity for “functional needs repair” was lower at 0.7012, its precision (Positive Predictive Value) was still strong at 0.9118, meaning that when the model predicts this class, it is often correct. The balanced accuracy values—especially 0.9525 for "non functional"—also reflect the model’s reliability across classes with varying prevalence rates. These metrics confirm that the Random Forest model is both accurate and robust, particularly in distinguishing between working and non-working water points.



```
                     Class: functional Class: functional needs repair Class: non functional
Sensitivity                     0.9810                        0.70118                0.9224
Specificity                     0.8989                        0.99468                0.9826
Pos Pred Value                  0.9202                        0.91175                0.9707
Neg Pred Value                  0.9755                        0.97700                0.9530
Prevalence                      0.5431                        0.07268                0.3842
Detection Rate                  0.5328                        0.05096                0.3544
Detection Prevalence            0.5790                        0.05589                0.3651
Balanced Accuracy               0.9399                        0.84793                0.9525
> confusionMatrix(train_pred, train_data$status_group)$overall["Accuracy"]
 Accuracy 
0.9381481 
```

*Fig08. Accuracies of Random Forest model*

I evaluated the performance of the Random Forest model using **5-fold cross-validation** to ensure that the model generalizes well to unseen data. Cross-validation involves splitting the training data into multiple folds, training the model on a subset, and validating it on the remaining part in rotation. This process helps assess the model's robustness and prevents overfitting. The consistent performance observed across the folds confirmed that the Random Forest model is not only accurate on the training set but also reliable when applied to new data. Given its high accuracy, strong class-wise performance metrics, and stability under cross-validation, the Random Forest model proves to be a highly suitable and applicable solution for predicting the operational status of water pumps.

## Conclusion

In this project, I developed a machine learning model to predict the operational status of water pumps in Tanzania using the provided dataset. After extensive data preprocessing and feature engineering—including handling missing values, removing high-cardinality and highly correlated variables, and deriving new features such as well age,  I built and evaluated multiple classification models.

Among the models tested, the Random Forest classifier achieved the best results. It demonstrated high predictive performance with a training accuracy of 93.8%, strong sensitivity and precision across all classes.

To validate the generalizability of the Random Forest model, I also performed 5-fold cross-validation, which confirmed its stability and reliability on unseen data. Given its robustness, accuracy, and consistent results across cross-validation folds, the Random Forest model was selected as the final model for predicting the status of water pumps. It was then applied to the test dataset to generate predictions for submission. Overall, the model demonstrates strong potential for practical application in real-world resource planning and maintenance prioritization efforts.