Comparing The Performance of Four Machine Learning Models to Predict the Functional Status of Water Pumps in Tanzania

1st Jennifer Brown
School of Computer Science
University of Nottingham
Nottingham, UK
pmyjb29@nottingham.ac.uk

2nd Abigail Kinnaird
School of Computer Science
University of Nottingham
Nottingham, UK
pmyak11@nottingham.ac.uk

***Abstract*—This paper compares the performance of different machine learning algorithms in their ability to predict the functional status of a water pumps in Tanzania, along with an investigation into which feature is most important in predicting this status. Two distinct methods are outlined, each of which includes separate data preprocessing steps which are then applied to two different machine learning models (random forest, decision tree for method 1, and multinominal logistic regression and k-nearest neighbours for method 2). Random forest was found to have the highest predictive power, with an accuracy and f1 score of 80.0% and 78.7% respectively, and was therefore concluded to be the most successful model in predicting functional status. Overall, the dry category of the feature quantity was found to be the most important predictor of functionality.**

I. Introduction

Our main research question is to compare the performance of different machine learning methods to predict the functional status of water pumps in Tanzania out of three classes: functional, nonfunctional, and functional needs repair. In addition to this main question, we wish to find which pump attribute in particular is the most important predictor. We will solve this multiclass classification problem using four different machine learning models. Our data set on which we will train our models has been taken from the platform Taarifa and includes 59,400 instances which represent 16% of the population of Tanzania [1]. Answering these two research questions will greatly aid in the improvement of water pump maintenance in Tanzania and potentially other locations around the world, ensuring safe drinking water is available to millions.

II. Literature review

For our literature review we looked at analyses of our data set, published on websites such as github and medium, by individuals attempting the Tanzanian Ministry of Water's data challenge. Our main area of focus was the methods of data preprocessing implemented by these authors, and the accuracies we might expect from different models.

The simplest data imputation approach adopted by many researchers on this project is to replace missing values with the median/mean for continuous data, and with the mode for the discrete data [2] [3] [4]. This value used for imputation is the average for the whole feature, or better, the average of each subgroup of that feature. For example, longitude may be imputed by region [5] [6]. A common, more complex technique used in machine learning research for effective imputation is k-nearest neighbours (KNN) [7]. Scaling of numerical features is required before applying models such as KNN and SVM, but not for logistic regression or tree-based algorithms [8]. Common scalers that appear in literature on this project are the StandardScaler, MinMaxScaler, and RobustScaler [9] [10] [11].

Most classification algorithms require all data to be numerical [12]. Categorical features therefore need to be converted into numerical features. For this data set we have observed that one-hot encoding was preferred to label encoding due to the lack of hierarchical order to the features [3] [4] [13]. Care needs to be taken with one-hot encoding when features have high cardinality, as it can lead to the curse of dimensionality [12] [14]. Frequency based binning has been used to address this problem for funders, installers, and lgas [3] [17]. This data set is imbalanced, with instances of functional needs repair being especially low. This creates issues during the training process when using predictive accuracy as a measure of model performance, as a high accuracy may only reflect the imbalanced nature of the data set rather than the model having a good predictive ability [15]. SMOTE is an algorithm that can balance the data set by oversampling the minority classes [16].

The most common methods of classification appear to be random forest and other tree-based algorithms. SVM, multinominal logistic regression (MLR), and KNN have also been used. In general, random forest had accuracies of around 80-85% were reported [3] [5] [17] [19]. SVM, MLR, and KNN performed less well, with accuracies of 71-77%, 70-75%, and 75-77% respectively [3] [4] [17] [13]. KNN is considered to be better than SVM when the amount of training data is much larger than the number of features [20].

III. Method 1

A. Data Exploration

As part of the data exploratory stage, the location of the water pumps was plotted using the longitude and latitude data, coloured by pump functionality (Figure 1). This shows that functionality may vary with location. Along with this the spread of the numerical data was investigated by plotting box plots (Figure 2). The number of NaN values were also found, which is vital for understanding the type of imputation techniques that are needed.

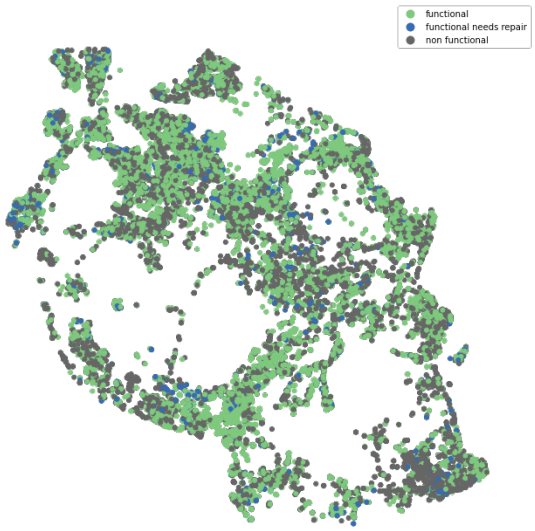

Fig. 1. Water pump location and functionality

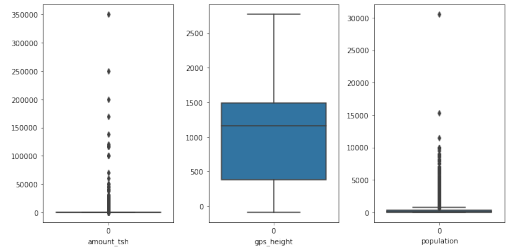

Fig. 2. Box plots for Amount_tsh, GPS height and Population

B. Preprocessing

Decision tree and random forest models were chosen for method 1 due to the high accuracy of tree-based algorithms from the literature. The preprocessing steps for method 1 will now be explained in detail. See Figure 3 for the summary of all the preprocessing steps and specifics for each variable.

Firstly, redundant features and features with missing values above 40% were dropped. Reducing the number of features is particularly important for tree-based algorithms as they suffer more from the curse of dimensionality [12]. For amount_tsh all 0 values were removed, as amount_tsh is just the distance between the free water you are extracting and the point you want it to be. Even though a 0 value is theoretically possible, it would not seem logical that a pump would be built where the free water is already at the level where the water needs to be accessed. Any outlier values were then replaced with NaN and imputed later. For the variables population and amount_tsh the top and bottom 5% of values were removed, for date recorded any value before 2010 was removed.

A KNN was adopted for imputation of most of the numerical variables as the number of NaNs was seen to be high for these columns. This gave distributions that maintained original peaks and features. Figure 4 shows an example of the imputation for GPS height. Table 1 summarises how the KNN was conducted. Longitude and latitude were then both imputed using the median grouped by region, these gave significantly better results than when a KNN imputer was used as it gave fewer impossible values i.e. values in the middle of a lake.

To reduce dimensionality further a new attribute age was created (difference between date recorded and construction

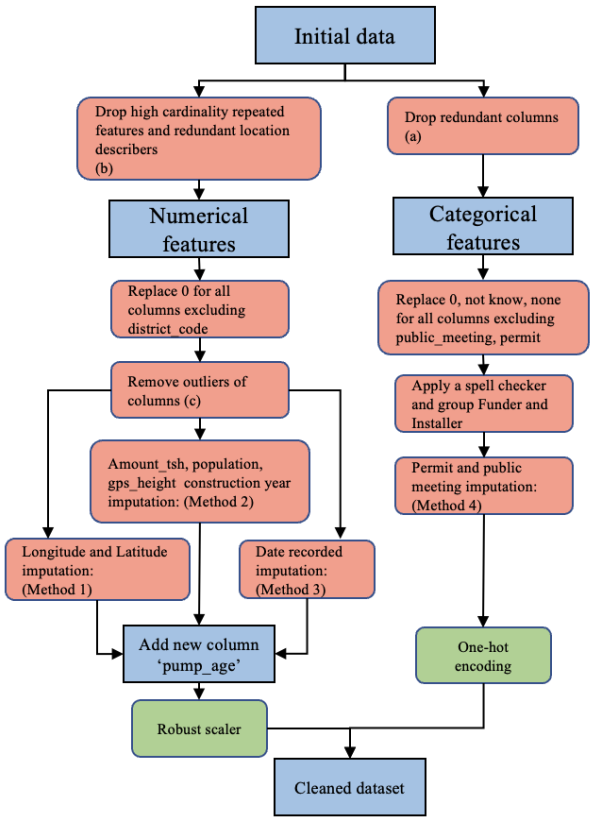

Fig. 3. Summary of preprocessing steps for Method 1. a) wpt_name, num_private, scheme_name, id b) payment, quantity group, recorded_by, waterpoint_type, source, water_quality, extraction_type_group, management_group, region_code, subvillage, ward c) population, amount_tsh, date recorded Method 1: Imputation using median grouped by region Method 2: KNN imputer Method 3: Mode imputation Method 4: SimpleImputer

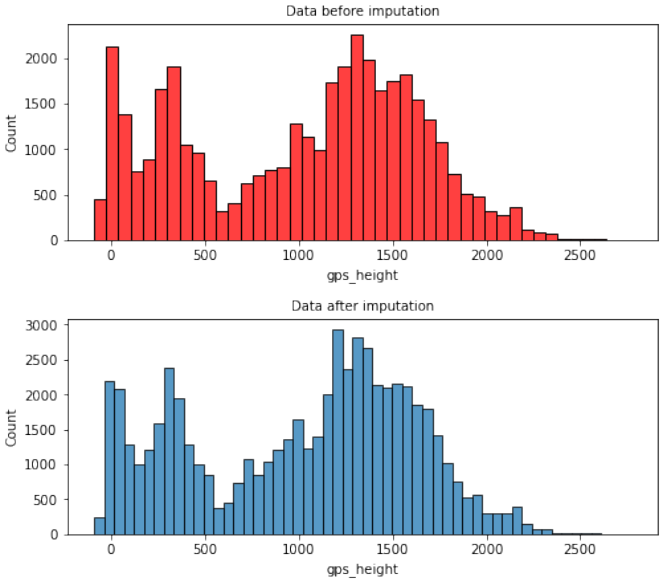

Fig. 4. Distribution of GPS height before and after KNN imputation

year) to encapsulate and allow the dropping of date recorded and construction year. All the numerical data columns where then scaled using a RobustScaler which was chosen as it is less affected by potential outliers.

For funder and installer, they were grouped into the top 20 categories. A spell checker was also implemented by

TABLE I
SUMMARY OF KNN FOR NUMERICAL DATA

Parameters	Population	Construction year	Amount_tsh	GPS height
K value	2	4	2	2
Other feature to impute over	Ward	Funder and installer	Source class, water-point type, basin	Ward

searching through for potential spelling errors or abbreviations. A dictionary of possible spelling errors and abbreviations were created for the top 20. All values that did not fit into one of these categories was then assigned Other.

To use the tree-based algorithms easily one-hot encoding was applied to all the categorical columns. This gave 271 columns in total including the numerical columns.There was a significant decrease in performance of the chosen models upon the application of SMOTE, and so the class imbalance will be considered through the choice of performance metric instead.

C. Data Modelling and Classification Methods

Tree-based methods work well for our research questions, as they can both easily predict multiclass classification problems and allow for the most important features to be easily identified. They also have the benefit of being more robust than other models regarding issues with preprocessing. This is particularly useful for this data set due to the large quantity of missing data that required imputation.

Decision trees work by splitting the data into lots of nodes based on the variables. The splitting in this model has been chosen to get the lowest gini impurity [21]. These leaf nodes then have associated probabilities corresponding to the different values of functional status. With a node higher up being more important in prediction. Model specific advantages of decision tree in this case is its transparency i.e., that we can easily tell how conclusions have been made. The major disadvantage that is common for decision trees is overfitting [22]. This will be reduced by investigation into setting a max depth.

Random forest, another tree-based algorithm, works by combining the results of many decision trees and is therefore an improvement on a singular decision tree. It is also much less prone to overfitting [22]. This model gave the best results in the literature, which suggests it could perform well. We again chose splitting of the nodes based on getting the smallest gini impurity value. Random forests are however less easy to interpret than decision trees, a decrease in our understanding of the model can lead to more skepticism of how effective it is.

D. Model training and evaluation

For both models a 75/25 training to test split was used. Accuracy, precision, f1 and AUC-ROC were the chosen metrics to evaluate the performance. Hyperparameters were evaluated based on scores on testing data. The models in both cases were created using the training data and then evaluated using the testing data.

Firstly, for decision trees we need to find the optimum max depth to reduce overfitting. This was found by looking at the performance for predicting the validation test set over a range of different max depths. For all the metrics 24 was the best value see Figure 5. We can see it decline after this value as overfitting has occurred.

The same process was then followed for random forest (see Figure 6). In this case however we want to find the optimum values for max depth and number of estimators. For max depth 25 was seen as the optimum value and for number of estimators it was 250.

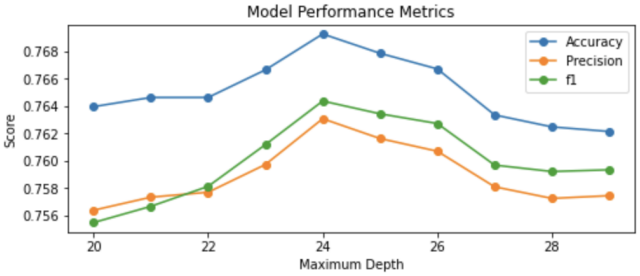

Fig. 5. Hyperparameter training for decision tree

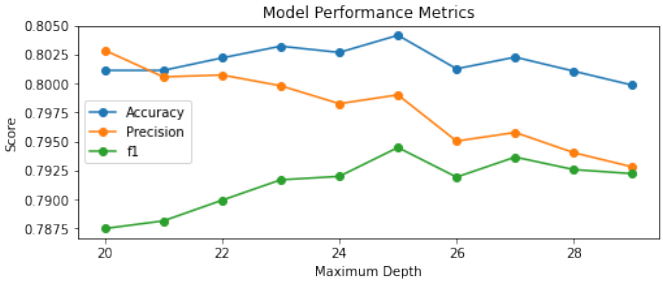

Fig. 6. Hyperparameter training for random forest

E. Results

Overall, both these models performed well. With weighted f1 scores of 76% for decision tree and 79% for random forest. The main issue with both these models was the performance in predicting the minority group. With f1 scores of 0.37 and 0.38 respectively. These are both much lower than the overall score, showing this as an area of weakness. This weakness can be shown clearly in the confusion matrices (see Figures 7 and 8).

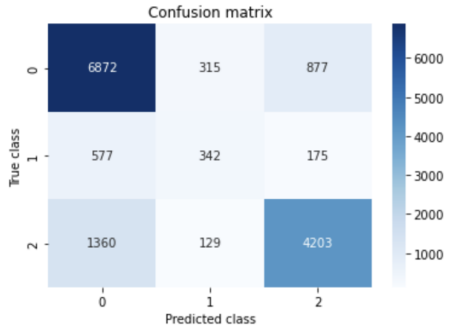

Fig. 7. Confusion matrix for decision tree

There is much more of a tendency of both models to incorrectly predict 0 and 2 (functional and nonfunctional) compared to the minority group of 1. With 0 being the most likely, which correlates to it being the largest data group. Random forest preforms better at not over-predicting nonfunctional, however it over predicts functional more.

Finally, we evaluate the performance by looking at the AUC-ROC plot. Let us consider functional as being a positive result and nonfunctional and functional but needs repair as a negative result. Grouping them by a negative value meaning 'some action is required'. Running the model again with this grouping, we can produce AUC-ROC plots. A straight-line plot is also plotted for convenience of comparison to prediction for a random process (AUC score of 0.5). Random forest is shown

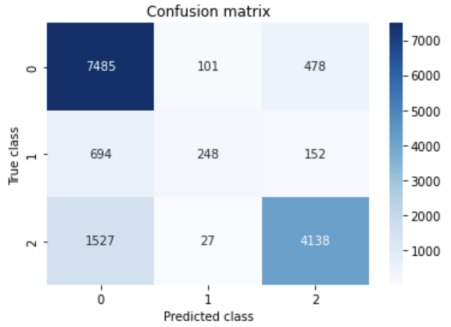

Fig. 8. Confusion matrix for random forest

to follow this straight line less and has an AUC value of 0.896 compared to 0.793 for decision tree. Therefore, in the context of this project random forest again performs better.

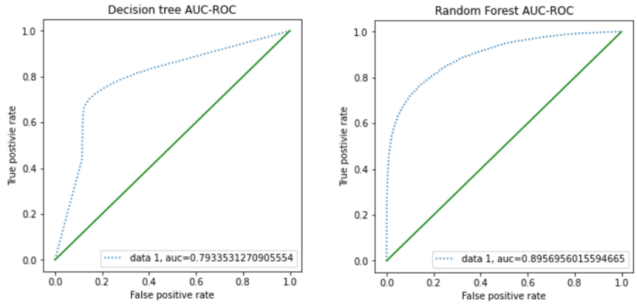

Fig. 9. AUC-ROC plots for decision tree and random forest

F. Feature importance

For our second research question we now consider which features are the most important. This was carried out using the built in scikit feature importance. We see that for both models the numerical features are of large importance, particularly longitude, latitude and gps height along with the categorical feature quantity dry (see Figures 10 and 11)

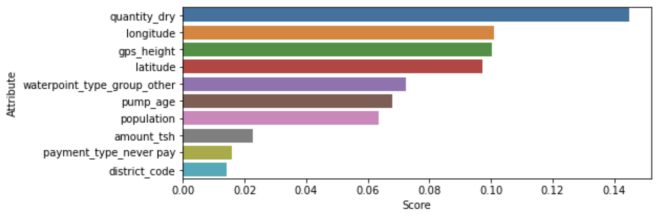

Fig. 10. Feature importance for decision tree

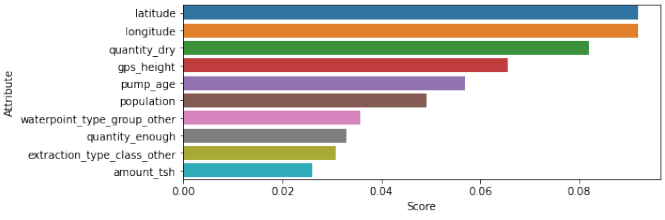

Fig. 11. Feature importance for random forest

Overall we see, random forest performed better than decision tree in all the metrics that were used to compare.

IV. Method 2

A. Data Exploration

Through exploratory data analysis, features with a high number of missing values were identified (see Figure 12). The feature cardinality was also investigated, in addition to features with potential outliers, and features that contained similar or identical information. The distribution of variables by functional status was also investigated (see Figure 13

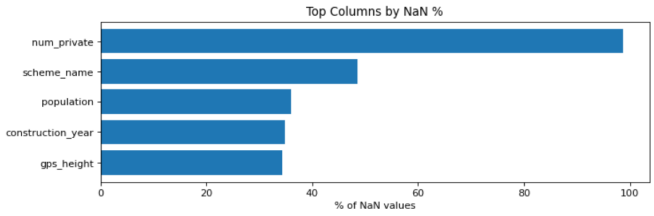

Fig. 12. Top 5 columns with the most NaN values

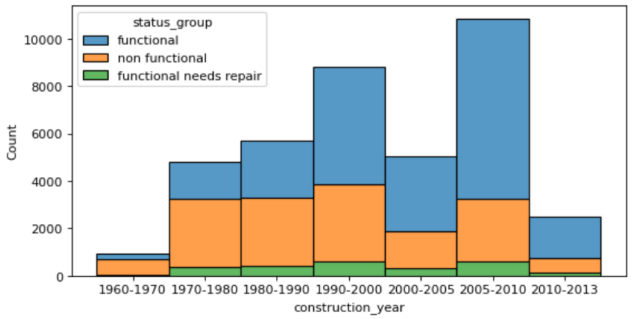

Fig. 13. Functional status by construction year (binned)

B. Preprocessing

KNN is impacted by dimensionality, as it works best when the dataset is more densely packed in the feature space [23]. MLR is also impacted by dimensionality, as more dimensions lead to overfitting [24]. MLR is also affected by multi-collinearity. If features contain similar information, they will be correlated with each other and negatively impact the model. Therefore sort to reduce the number of features, especially ones correlated with other features, by as much as possible without losing information which may be important.

As with method 1, the columns that contained a high number of unique values, missing values, or contained identical information to another column were dropped. Out of the features which had similar categories those with higher cardinality were chosen, as opposed to method 1. While this led to an increase in dimensionality after feature encoding compared to using lower cardinality features, however dimensionality was reduced in other ways by categorical feature binning of the top 50 lgas, a category of 125 unique values. Categorical feature binning was applied to the top 20 funders and installers. There were many spelling errors and synonyms that would need to be corrected to increase successful categorisation. To find the most common of these errors and synonyms, SimilarityEncoder from the Dirty Cat package to create a correlation matrix between the top funder and installer names to more easily find similarities [25].

Outliers affect the performance of machine learning models by skewing the data [26]. In addition, they may be erroneous.

Logistic regression is fairly robust to outliers [27], however KNN less so [28]. For population, there were many instances of the value 0 and 1 that were considered to be missing data and removed, as shown in Figure 14 The top 0.5% of population was also removed, and the top 5% of amount_tsh. Values of 0 for gps_height, and -2e-08 for latitude, and strings in categorical columns such as 'not known', 'none', and '0' were also classed as missing values. For the date_recorded column, the value of 2002 was taken to mean 2012, but the value of 2004 was classed as missing as there were no records taken in 2014.

After outlier removal, imputation was performed. Construction year and date recorded were imputed based on median and mode respectively. Public meeting and permit were imputed by the mode for each lga, as these seemed to be affected by local government authority. The imputation of population, longitude, and latitude, and gps_height was carried out using the median of each region, as other location features contained more missing data. Median was used as it was more robust to outliers than the mean. There were four regions where only missing values for both population and gps_height, so the total median of these features were used instead. Similarily, amount_tsh was imputed by the median of each 'waterpoint_type', with total median being used for 'dam'. Figure Y shows population before and after imputation. It can be noted that this imputation is not ideal and may increase multicollinearity, which will be considered in the discussion. As with method 1, missing scheme_management values were class as 'unknown'.

MLR and KNN affected by the magnitude of the features and therefore scaling was applied. Out of the three scalers described in the literature, RobustScaler gave the best model performance. MLR and KNN required data to be numerical, and therefore one-hot encoding was applied to the categorical features. After one-hot encoding, the total number of columns was 203. Pump status was then Label encoded to complete the preprocessing process.

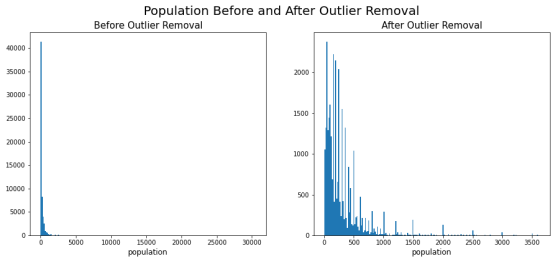

Fig. 14. Outlier removal for population, by removing the values 0, 1, and the top 0.5%.

C. Data modelling and classification methods

1) Multinomial Logistic Regression: Logistic regression classifies data by fitting a sigmoid function to the independent variables in order to predict the probability of the dependent variable, i.e. the class. Multinomial logistic regression (MLR) natively supports multiclass classification and can easily be implemented in python [29]. After fitting an MLR model, we obtain three curves. We can make inferences about the relative importance of a feature or each class based on relative size and sign of the intercepts and coefficients.

MLR was chosen because both the predictive value of a feature and individual categories of that feature could be assessed, which would allow us to answer our second research question of what the most important predictor of functionality was. To investigate the predictive value of the whole feature rather than individual categories, separate MLR models were fitted with each feature removed and then the change in testing accuracy from the original model was calculated. This process is known as backward elimination [3]. The bigger the decrease in testing accuracy upon the removal of a certain feature signifies that that feature is more important to our model. Variables that increase model peformance upon removal could be removed from the original model to increase prediction accuracy.

2) K-Nearest Neighbours: KNN predicts class labels, first by calculating the distances between a test point and all training points, and then finding the most frequent class label out of the k closest training data points to that test point. This is repeated for each test point in order to classify the data [30]. As distances need to be calculating between all testing and training data points, KNN is computationally expensive. KNN was chosen to answer the first research question as it was easy to implement (only requires one hyperparameter to tune), and makes no assumptions about the distribution of the data [28], whereas MLR makes assumptions such as there is a linear relationship between features and the outcome [31].

D. Model training and evaluation

For both models a train test split of 75/25 was used and then SMOTE was applied. The performance metrics considered for both models were accuracy, f1 score, and balanced accuracy. F1 score and balanced accuracy are more informative than accuracy on imbalanced data sets like ours. MLR did not require cross-validation, and the best model was chosen by dropping the least important features instead. Through backward elimination the least important features in the MLR model were identified, and the best model accuracy on all three evaluation metrics was achieved with the combined removal of public_meeting, amount_tsh, latitude. 10-fold cross validation was used for KNN to find the optimal value of k. To prevent the cross-validation algorithm from applying SMOTE to the validation sets, SMOTE was implemented in the cross-validation pipeline rather than applying it during the preprocessing stage, such that the cross-validation score would be more similar to testing accuracy of the same parameters [32]. The best model for accuracy and f1 score was k = 2, and for balanced accuracy it was k = 15, as shown in Figure 15. KNN is affected more by outliers and overfitting when k is small [33], however if k is large the decision boundaries become over simplified and bias increases. There is a large difference between training and testing accuracy for low k, as shown in Figure 15, which implies overfitting at k=2 [34]. After k = 4, the decrease in accuracy difference slows, so the model does not improve greatly as k is increased further, and the bias may also increase. Since the balanced accuracy for k = 4 is higher than for k = 2, and the accuracy and f1 score are similar, the best model seems to be k = 4.

E. Results

1) Multinomial Logistic Regression: The breakdown of the scores for MLR can be seen in the classification report See Figure 16. This shows a weighted f1 score of 0.72 which is consider a good score. The score for the functional but need repair was considerably lower than the other two. This issue can be further explored in confusion matrix.

This model can be seen to over predict functional i.e 1471 incorrect predictions of function when the true value was nonfunctional. There is also and under prediction of functional but needs repair. See Figure 17 for more detail. Now looking at the feature importance, the five most important features were found to be quantity, payment_type, waterpoint_type, lga, and extraction_type_group. These features reduced model accuracy

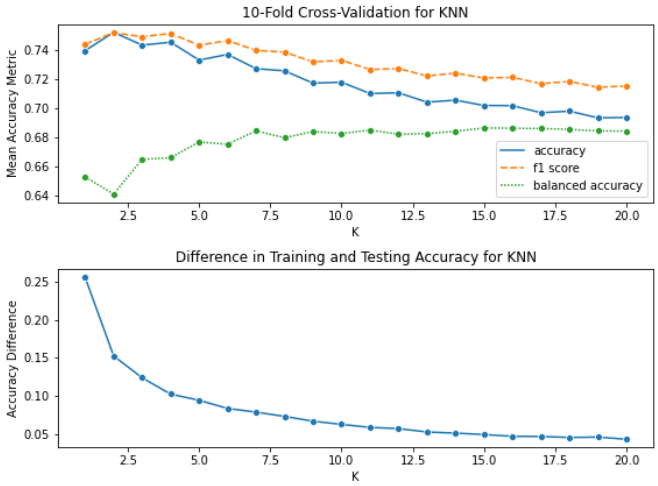

Fig. 15. Plot of 10-fold cross validation scores for K = 1 to 20, and of the difference in training and testing accuracies for the accuracy score metric

	precision	recall	f1-score	support
0	0.77	0.77	0.77	7963
1	0.26	0.49	0.34	1076
2	0.79	0.66	0.72	5811
accuracy			0.70	14850
macro avg	0.61	0.64	0.61	14850
weighted avg	0.74	0.70	0.72	14850

Fig. 16. Classification report for MLR

by a large percentage when removed individually, as shown in Figure 18. The top features were investigated more in depth, and the feature categories affected water pump functionality differently were noted and seemed to make intuitive sense. For example, dry was most likely to be classed as nonfunctional, enough to be functional, and insufficient to be functional but needing repair. After public_meeting, amount_tsh, and latitude had been removed from the model, the remaining least important predictors of functionality were longitude and district_code.

2) K-Nearest Neighbours: From the classification report for KNN we see a balanced f1 score of 0.76. It also has issues with much lower scores for the minority class. With it over predicting functional and under predicts functional but needs repair. However this problem has been improved upon from KNN.

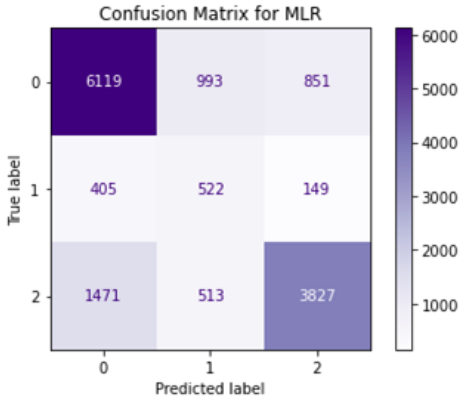

Fig. 17. Confusion matrix for MLR

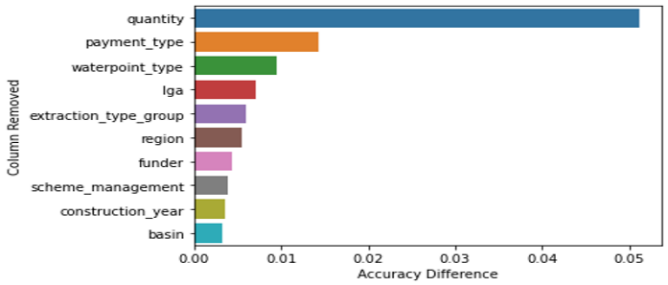

Fig. 18. Feature importance for MLR, determined by accuracy difference after backward elimination

	precision	recall	f1-score	support
0	0.79	0.82	0.80	7963
1	0.34	0.51	0.41	1076
2	0.83	0.70	0.76	5811
accuracy			0.75	14850
macro avg	0.65	0.68	0.66	14850
weighted avg	0.77	0.75	0.76	14850

Fig. 19. Classification report for KNN

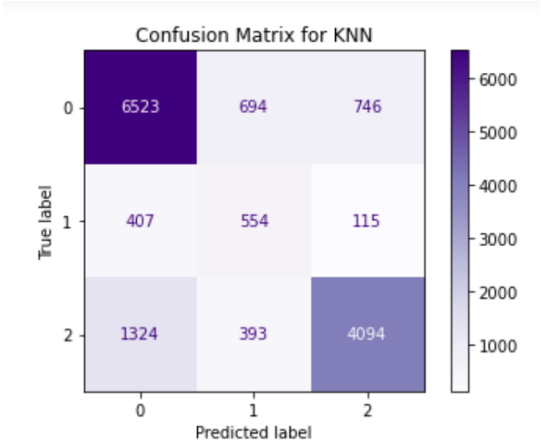

Fig. 20. Confusion matrix for KNN

Overall when we considering our first research question for this method, KNN led to a greater predictive accuracy than MLR. The recall in particular was the most improved between the two models. Our second question can however only be answered by MLR as KNN cannot compute feature importance.

V. Discussion

A. Interpretation of Results

TABLE II
Summary of Results of Each Model

Metric	Decision tree	Random forest	MLR	KNN
F1 score	0.759	0.787	0.717	0.759
Accuracy	0.769	0.804	0.705	0.752
Most important feature	Quantity dry	Longitude and latitude	Quantity dry	N/A

Table II summarises our results in relation to our two research questions. Random forest performed the best out of the four models, both in terms of accuracy and f1 score, with f1 score accounting for the imbalanced nature of the data set.

For feature importance, the biggest predictor of functionality for the decision tree and MLR models was the category dry for the feature quantity. In the MLR model we could see it was the nonfunctional class in particular that was being predicted by quantity_dry. This is unsurprising, as 97% of water pumps recorded as having quantity dry are nonfunctional. Longitude and latitude, however, were more important than quantity_dry to the random forest model. While longitude and latitude are also important predictors for the decision tree, they were among the least important features in the MLR model. The same preprocessing technqiues were implemented for these location features, so it must be due to the different models used and the effect of other features on these variables. While the random forest model had the highest prediction accuracy, we will consider the feature that is important to all the models as the biggest predictor of functional status i.e. quantity_dry. As well as model choice and preprocessing techniques, results may have also been affected by the methods of data collection and the biases introduced by them. For example, missing values were not missing at random. Looking at the NaN values it is noticed that lots of data points have missing values for population, construction year and gps_height. There could be an unexplored relationship between missing values and functional status. If values are missing it may imply they are harder to record, which may also correlated to how well maintained the pump is.

B. Comparing and Critiquing Methodologies and Results

Since random forests perform highly on this data set, as seen from our literature review, it was considered a good choice of model. However, a lower accuracy was achieved for random forest than in the literature review which may be due to a number of reasons. Firstly, random forests are immune to the curse of dimensionality [35], so when preprocessing the data for this model, features with a higher cardinality could have been chosen in order to provide more information for the model. Secondly, while one-hot encoding may be better than label encoding for nominal data, it may negatively impact the performance of tree models in method 1 by increasing the importance of categorical features, as trees are split for each dummy value of a feature rather than considering the feature as a whole, which introduces sparsity [36]. One-hot encoding especially of high cardinality features such as lga may also lead to overfitting of the decision tree model. While random forests do not experience the curse of dimensionality, KNN suffers heavily from it, and therefore the choice of higher cardinality features in method 2 may have negatively affected the performance of this model. In addition, the variables date recorded and construction year could have been combined into a new variable pump age as in method 1 to again reduce the number of variables. The use of label encoding rather than one-hot encoding could have led us to find the most important features overall as well as the most important feature categories, and so could have corroborated results found by MLR, which is a lower accuracy model. Conversely, this could have also been done by implementing a backward elimination method like in method 2. A different approach to outlier removal was performed in method 1 compared to method 2. In method 2, the value of 0 for amount_tsh was not removed, however considering that a value of 0 is not logical, it may have represented missing data. In method 1, the amount_tsh values of 0 were removed, and then the bottom 5% of the remaining values was removed. This caused all amount_tsh values below 10 to be removed, however the values above 0 may have been realistic values. In addition for the different approaches to amount_tsh, the removal of population outliers differed, with more data being removed in method 1 despite these additional outliers not dramatically affecting the skew. The removal of more data may be detrimental to model performance. These could be reasons why the importance of amount_tsh and population differed greatly between methods. In method 1, the construction year of 2002 was considered an outlier and was removed for later imputation, however in method 2 it was considered to be an input error of the more realistic value 2012. It is unknown however if the differing importance of pump_age and construction year to methods 1 and 2 respectively more due to the column merging or the different outlier removal and imputation methods .

Data imputation approaches also differed between methods. In method 1, the boolean columns permit and public meeting were simply imputed with True, however in method 2 they were imputed with the mode by lga. While True was the most common value for most lgas, some lgas had most or all values as False, especially for permit, so would have likely been false instead. The simpler approach in method 1 therefore will have led to different conclusions about public meeting especially. The imputation approach in method 2 for population, gps_height, amount_tsh, and construction year also differed. This method lead to large spikes in these distributions due to missing values being imputed with the total feature median when there was no data for particular subgroups. If population, for example, used other variables such as lga and ward in addition to imputing by region, there would be less missing data to be imputed with total median, and multicollinearity would be decreased, improving the performance of the MLR model in particular. In method 1 however, care was taken not to affect the distribution of these variables and an KNN imputer was used. This will have had a positive impact on model performance compared to method 2. However, the low value of k used in the imputation of gps_height, amount_tsh , and construction year may have lead to overfitting and produced less realistic values for the data than a larger value of K.

For the cleaning of the funder and install columns, different methods were used to address the issue of spelling errors and synonyms. A similar number of spelling errors and synonyms were corrected, however the ones found were different. For example, method 1 did not consider RWSSP and DWSSP to mean rural and district water supply and sanitation program respectively, and missed the common misspelling of Adra as Acra. In method 1, a brute force approach was taken which was time consuming and relied on guessing where possible errors may be. Method 2 found the most common misspelling by total number instead, with misspellings being easy to spot through a correlation matrix. Overall, it is unknown which method would have been better data cleaning due to the similarity in the number of values in the top 20 funder and installers. A combining of the two sets of spelling error and synonym dictionaries could be implemented in future work to detect an ever greater number and increase the accuracy of predictions by detecting the errors missed by the individual dictionaries.

Scaling was applied in both methods, however it is not needed for tree-based models so there was no need to scale the data in method 1. While the model performance would not have been negatively impacted by the scaling, it does make the results of tree algorithms less interpretable. SMOTE was chosen in method 2 to account for the fact that there was an imbalance in the data set. However, SMOTE has issues when applied to high dimensional data, and also creates synthetic data which may not be reliable and which decreases the variance of the data. This may lead to more incorrect predictions.

For decision tree and random forest, more exploration of the hyperparameters could have been carried out. In decision tree,

only max depth was considered. Another good technique for reducing the size of the tree and additionally choosing which branches are most important is pruning. For random forest, the min values for leaf node and split were not optimised.

Now looking at the model fitting for method 2. KNN, k = 4 was chosen but there could have been more of an investigation between the bias variance trade off as a larger value of k may have given better results. KNN also only used the Euclidean distance. A good comparison would have been to use Manhattan as this is suggested to be better for higher dimensional data. [37] It could have also been considered that, for the methods affected by dimensionality, PCA could have been used for feature reduction, however this would prevent the interpretation of feature importance so should only be used for increasing prediction accuracy. For MLR a better method for reducing the number of variables may have been ridge regression, as this would reduce the impact of multicollinearity on model performance.

VI. Conclusion and Future Work

To conclude, the main research question to compare multiple machine learning methods has been answered through the comparison of four of our own models with each other and with those in the literature. To complete this comparison effectively, the processes of data exploration, preprocessing and hyperparameter tuning were carried out, including justification of each step. Through these methods, the most important feature for prediction was concluded to be quantity dry, and the model with the highest prediction accuracy was found to be random forest. This model will help to save lives by allowing for the dispersion of resources at the time and place they are required. The importance of quantity to functional status allows for higher risk pumps to be targeted and repaired. Future work should build upon the two methods in this paper, implementing the most successful steps, as considered in the discussion, to achieve a more robust and better performing model. The importance of location should also be investigated as it was identified to be important to some models but not others. These considerations will allow continued support for communities in all locations of Tanzania to allow them to keep vital water pumps running.

References

[1] "Country/Tza"country/TZA Place Explorer- Data Commons, https://datacommons.org/place/country/TZAutm

[2] B. Loznik, "Pump it up-how to deal with missing data?," Medium, 27-Dec-2021. [Online]. Available: https://medium.com/@brendaloznik-48450/pump-it-up-how-to-deal-with-missing-data-ac60178f1ae5. [Accessed: 19-Apr-2023].

[3] J. Topor, S. Mekala, S. Hong, and S. Dunn, Predicting Tanzanian Water Pump Maintenance Needs. [Online]. Available: https://rstudio-pubs-static.s3.amazonaws.com/339668-006f4906390e41cea23b3b786cc0230a.html. [Accessed: 19-Apr-2023].

[4] A. A. Awan, "Pump it up data mining the water table," Deepnote, 05-Apr-2021. [Online]. Available: https://deepnote.com/@abid/Pump-it-up-data-mining-the-water-table-ff691bdb-84c7-4969-8146-c730900d0efd. [Accessed: 19-Apr-2023].

[5] K. ShreeBalaji, V. Rahul, R. Rafri devi, and M. Bhuvaneshwari, "Pump it up: Mining the water points using XGBoost classifier," Medium, 18-Aug-2019. [Online]. Available: https://medium.com/@shreebalajirj04/pump-it-up-mining-the-water-points-using-xgboost-classifier-47cd1b5d3507. [Accessed: 19-Apr-2023].

[6] J. dills, "Tanzania-water-table/clean-data.ipynb at master · JDILLS26/Tanzania-water-table," GitHub. [Online]. Available: https://github.com/jdills26/Tanzania-water-table/blob/master/clean-data.ipynb. [Accessed: 22-Apr-2023].

[7] J. Brownlee, "KNN imputation for missing values in machine learning," MachineLearningMastery.com, 17-Aug-2020. [Online]. Available: https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/. [Accessed: 22-Apr-2023].

[8] Z. Jaadi, "When and why to standardize your data," Built In. [Online]. Available: https://builtin.com/data-science/when-and-why-standardize-your-data. [Accessed: 22-Apr-2023].

[9] V. D. Studio, "Data Transformation and feature engineering in Python," Visual Design, 27-Jul-2021. [Online]. Available: https://www.visual-design.net/post/data-transformation-and-feature-engineering-in-python. [Accessed: 22-Apr-2023].

[10] ashwinsharmap, "StandardScaler, MinMaxScaler and Robustscaler techniques - ml," GeeksforGeeks, 21-Feb-2023. [Online]. Available: https://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/. [Accessed: 22-Apr-2023].

[11] J. Brownlee, "How to scale data with outliers for machine learning," MachineLearningMastery.com, 27-Aug-2020. [Online]. Available: https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/. [Accessed: 21-Apr-2023].

[12] VanderPlas, Jake. Python Data Science Handbook : Essential Tools for Working with Data, O'Reilly Media, Incorporated, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/nottingham/detail.action?docID= 4746657. [Accessed: 15-Apr-2023]

[13] B. T. Duong, "Data Science vs. pump it up competition," Medium, 25-Oct-2021. [Online]. Available: https://medium.com/geekculture/data-science-vs-pump-it-up-competition-cccc8d58bb64. [Accessed: 22-Apr-2023].

[14] B. Loznik, "Pump it up - A comprehensive guide to EDA," Medium, 27-Dec-2021. [Online]. Available: https://medium.com/@brendaloznik-48450/pump-it-up-a-comprehensive-guide-to-eda-c7cdcf0480f8. [Accessed: 22-Apr-2023].

[15] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002 Jun 1;16:321–57.

[16] Hu. Li, Peng. Zou, Weihong. Han and Rongze. Xia, "A Combination Method for Multi-class Imbalanced Data Classification," 2013 10th Web Information System and Application Conference, Yangzhou, China, 2013, pp. 365-368, doi: 10.1109/WISA.2013.75.

[17] Domptail, "Tanzania-water-wells-ongoing/Tanzania waterpoints.ipynb at master · Domptail/Tanzania-water-wells-ongoing," GitHub. [Online]. Available: https://github.com/domptail/Tanzania-Water-Wells-ongoing/blob/master/Tanzania%20Waterpoints.ipynb. [Accessed: 22-Apr-2023].

[18] K. Kalluri, "Predicting the functional status of pumps in Tanzania," Medium, 07-Jul-2017. [Online]. Available: https://towardsdatascience.com/predicting-the-functional-status-of-pumps-in-tanzania-355c9269d0c2. [Accessed: 22-Apr-2023].

[19] A. Carl, "Andrew-Carl/pump-IT-up-data-mining-the-tanzania-water-table: Predicting functionality of groundwater pumps throughout Tanzania.," GitHub. [Online]. Available: https://github.com/Andrew-Carl/Pump-it-Up-Data-Mining-the-Tanzania-Water-Table. [Accessed: 22-Apr-2023].

[20] D. Varghese, "Comparative study on classic machine learning algorithms," Medium, 10-May-2019. [Online]. Available: https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222. [Accessed: 22-Apr-2023].

[21] A. Geron, Hands-on Machine Learning with Scikit-Learn, Keras and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastapol , CA: O'Reilly, 2023.

[22] S. Talari, "Random Forest vs decision tree: Key differences," KDnuggets, https://www.kdnuggets.com/2022/02/random-forest-decision-tree-key-differences.html: :text=Random

[23] P. Grant, "k-Nearest Neighbors and the Curse of Dimensionality," Medium, Jul. 24, 2019. [Online]. Available: https://towardsdatascience.com/k-nearest-neighbors-and-the-curse-of-dimensionality-e39d10a6105d

[24] E. Travers, "Why does logistic regression overfit in high-dimensions?," Eoin Travers, Aug. 25, 2020. [Online]. Available: http://eointravers.com/post/logistic-overfit/. [Accessed: May 08, 2023]

[25] BrendaLoznik, "waterpumps/2. Data cleaning Feature engineering.ipynb at main · BrendaLoznik/waterpumps," GitHub, Dec. 27, 2021. [Online]. Available: https://github.com/BrendaLoznik/waterpumps/blob/main/2.

[26] Nair, "Outliers: Keep Or Drop?," Medium, Jul. 18, 2022. [Online]. Available: https://towardsdatascience.com/outliers-keep-or-drop-892b599b8ab6. [Accessed: May 08, 2023]

[27] J. Lever, M. Krzywinski, and N. Altman, "Logistic regression," Nature Methods, vol. 13, no. 7, pp. 541–542, Jun. 2016, doi: https://doi.org/10.1038/nmeth.3904.

[28] Genesis, "Pros and Cons of K-Nearest Neighbors - From The GENESIS," From The GENESIS, Sep. 25, 2018. [Online]. Available: https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/. [Accessed: May 08, 2023]

[29] J. Brownlee, "Multinomial Logistic Regression With Python," Machine Learning Mastery, Dec. 31, 2020. [Online]. Available: https://machinelearningmastery.com/multinomial-logistic-regression-with-python/. [Accessed: May 08, 2023]

[30] A. Christopher, "K-Nearest Neighbor," Medium, Feb. 03, 2021. [Online]. Available: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4. [Accessed: May 08, 2023]

[31] R. Daines, "LibGuides: Statistics Resources: Multinomial Logistic Regression," resources.nu.edu, Apr. 18, 2023. [Online]. Available: https://resources.nu.edu/statsresources/Multinomiallogistic

[32] K. S. V. Muralidhar, "The right way of using SMOTE with Cross-validation," Medium, Mar. 31, 2021. [Online]. Available: https://towardsdatascience.com/the-right-way-of-using-smote-with-cross-validation-92a8d09d00c7. [Accessed: May 08, 2023]

[33] N. Xue, "Basic Understanding of KNN Algorithm," Medium, Apr. 30, 2021. [Online]. Available: https://ningyixue.medium.com/basic-understanding-of-knn-algorithm-7acfead6f12e. [Accessed: May 08, 2023]

[34] J. Brownlee, "How to Identify Overfitting Machine Learning Models in Scikit-Learn," MachineLearningMastery.com, 11-Nov-2020. [Online]. Available: https://machinelearningmastery.com/overfitting-machine-learning-models.

[35] P. P. Pathak, "Decision Trees and Random Forests — explained with Python implementation.," Medium, Jul. 18, 2021. [Online]. Available: https://towardsdatascience.com/decision-trees-and-random-forests-explained-with-python-implementation-e5ede021a000. [Accessed: May 08, 2023]

[36] R. Ravi, "One-Hot Encoding is making your Tree-Based Ensembles worse, here's why?," Medium, Jan. 12, 2019. [Online]. Available: https://towardsdatascience.com/one-hot-encoding-is-making-your-tree-based-ensembles-worse-heres-why-d64b282b5769. [Accessed: May 08, 2023]

[37] Gohrani, K. (2019) Different types of distance metrics used in machine learning, Medium. Available at: https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7 (Accessed: 09 May 2023).