Exploring machine learning models to predict the functionality of water pumps

1st Rashed Alaleeli
School of Computer Science
University of Nottingham
efyra5@nottingham.ac.uk

2nd Toluwalase Soyebo
School of Computer Science
University of Nottingham
psyts12@nottingham.ac.uk

***Abstract*—Monitoring a water pump's functionality is, or at least should be, as important as building one. There is a relationship between the quality of water in a country and the quality of life there. Whilst more economically developed countries (MEDCs) have an effective way of monitoring water pumps' functionality and ensuring that water is consistently clean, developing countries such as Tanzania, do not yet have these same systems imposed and therefore may not regularly check their functionality. Our project focuses on analysing features that affect water pumps in Tanzania and utilising machine learning classification models and data provided by the Taarifa water points, to predict their functionality. The research results show that random forest is the best-performing classifier to predict the labels with an accuracy of 81.07%, those findings have significant implications for the management of water pumps and help to ensure efficient and effective access to water.**

***Index Terms*—Classification models, Data wrangling, Random Forest, Water pump's functionality**

I. Introduction

Water is an important resource for all living organisms and plays a vital role in economic activities such as agriculture. Throughout the years demand for water has been increasing, but not all countries have effective measures in place to ensure their supply is equilibrium to the demand. This is highlighted more in developing countries such as Tanzania, where there is an escalated demand for water due to rapid urbanization and climate change impacts [1]. Water pumps had been previously introduced in Tanzania, to increase the water supply, although there has been a lack of sufficient monitoring of these pumps to ensure they are functioning as they should be. As of 2015, 29% of the water pumps in Tanzania are not functional, this can be due to several characteristics such as funder, location of the water point, and its age [2]. This research aims to focus on identifying the important features, in predicting the status of a pump and investigate the best-performing classification models by training them on the data and features provided to predict the labels of unseen data. Predicting the functionality, in advance can offer a number of advantages, including lower emergency repair costs, more effective implementation of infrastructure development projects, and most importantly, assuring a steady supply of water.

Data wrangling is a critical step in ensuring that models offer predictions as effectively as possible, this can be done by handling outliers, incorrect and missing values. Because some features are repetitive and others do not provide useful information, it is vital to select features that play a substantial role in increasing the model's performance. After data wrangling, experimented with different feature engineering techniques, to ensure that the data set is in the right format, such as ensuring all input variables are numerical variables, and then using various classification models: Random forest, XGboost, logistic regression, and Gaussian Naïve Bayes (GNB) to predict the functionality of the water pump, and evaluate these models. In doing this we were able to investigate important research questions: (i) Which classification models perform best to predict functionality? (ii) How accurately can we predict the functioning of a pump in Tanzania based on its age?

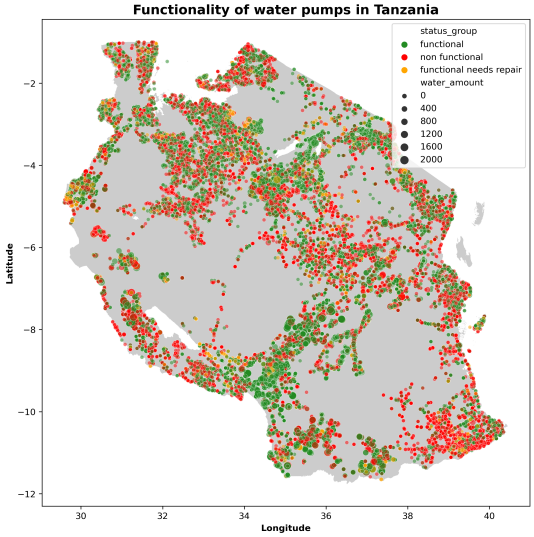

Fig. 1. Tanzania's water pumps and their condition.

II. Literature Review

Drawing from wider research, one hot encoding has been a successful method in converting categorical variables to numerical ones. Although some parts of the research were not relevant, the performance of one hot encoding has been evaluated against other feature engineering approaches like feature hashing. It was a common result for one hot encoding

to have the best performance regardless of the model with a PR-AUC of 0.730 and 0.728, over other feature engineering methods such as feature hashing with a PR-AUC score of 0.600 and 0.691 [3]. Due to the large sample size, feature selection methods were also important in [4] [5] [6], which explores the benefit of these methods, including, less chance of over-fitting, and reducing dimensionality reduction as well as the different approaches that one can take to handle this problem. In this paper, a key part of our feature selection approach stems from [7], where results show that a feature selection based on gini importance poses a huge benefit in identifying the optimal subset of features and dimensionality reduction of the data set as well as eliminating noise from the classification task. In the studies that are directly related to predicting a water pump's functionality, a common model that has consistently performed well is XGboost. The model has been the best performing, with over 80% accuracy score [8] [9]. In another study, the focus was on using decision trees, neural networks, and random forests, as the different predictive models, where the random forest was the best-performing model [10]. Our work builds on these different papers as we contrast different methods, for instance, one hot encoding was used to convert categorical variables to numerical variables, and control the input variables into our algorithm through feature selection. We have also utilised both XGboost and Random forest classifiers to predict the status group as well other machine learning models such as Gaussian Naïve Bayes.

III. Methodology

The research workflow followed a precise path, as shown in Figure 2. Understanding the dataset was a critical step in exploring the data and detecting any trends or relationships between the attributes before proceeding with any analysis. Following that, data preprocessing consists of several phases, including handling missing values, handling outliers, and normalising. Then, data was visualised to investigate further correlations and which features influence the water pump's operation. These features will then be chosen based on their importance and potential impact on model performance. Before training the models, all categorical features must be transformed into numerical ones. Several classification models will then be produced in order to determine the best-performing one, and the condition of the water pumps from the test set will be predicted using that model.

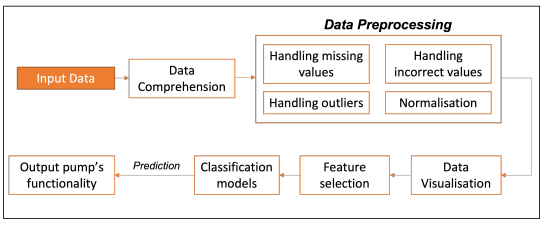

Fig. 2. Research workflow to predict pump's functionality.

A. Dataset

The dataset was collected by the Tanzanian Ministry of Water and Taarifa waterpoints dashboard. The dataset initially consisted of three different datasets, training labels, attributes, and test attributes. This led to two different ways to treat this dataset where one method concatenated both datasets and created a new column named 'indic' to distinguish them and guarantee that no samples were jumbled. Therefore had approximately 70000 samples from both training and testing, as well as 39 features that will help with predicting the labels. The other method focused on pre-processing these datasets as two separate datasets: training data with 59400 training data, and 40 attributes, whilst the test data had 14851 instances and 38 attributes as no target variable had been provided. The target variable was the status of the water pump, whether it could have been functional, non-functional, or needed repair.

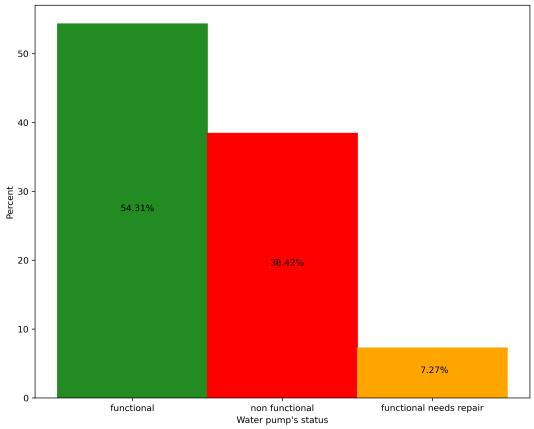

Fig. 3. Water pumps percentage at different status.

B. Data preprocessing

After going through the data to gather information about it, various columns were discarded as they were redundant features. Features were classified to be redundant if: they just had one unique value for all instances, the feature had no explanation, and features were duplicated. Because the duplicate columns contain the same unique value and the only variation between them is the spelling of these values, one of them will be dropped. A different approach was used for the numerical features where a correlation heat map was drawn to identify numerical variables which were similar, and therefore could also drop one of these variables to avoid overfitting such as 'region_code' and 'district_code', which had the highest correlation of 0.68.

To handle outliers one way was to proceed through all numerical features to count the number of outliers each feature has. Most of the values that are less than the quantile 5% or above the quantile 95% have been replaced with a null value so that they are replaced with an appropriate value when dealing with missing values in the dataset. But for some features, did not use the quantile percentage to remove the outliers, because the quantile percentage was close to the mean value

of the feature, so instead used a value that is higher than the percentage just to make sure that this does not affect the distribution of the feature.

An alternative approach to handle outliers in each numerical column. Where calculating the kurtosis and skewness of the data elucidated the outliers in the 'amount_tsh' and 'population' columns. The 5% outliers for both features were treated by setting them to be missing values. A major difference in this approach was using the less than or equal to signal for these columns as the 5% quantile was equal to the minimum value and therefore just using the less than approach would not change the minimum value, and would still be 0. For features where kurtosis was not high, the instances below the 5% and above 95% quantile were replaced with these values respectively rather than missing value, this adapts the winsorization approach, aiming to replace these outliers with the nearest non-outlying outlier and therefore not affecting the data distribution as much. As part of handling outliers the 'region _code' had outliers, upon further investigation, it was clear that these values were imputed wrongly, such as using 60 instead of 6, these errors were handled, by ensuring values with a code above 31 would be corrected, as the maximum for Tanzania is 31. These region codes above 31 had other codes which were a more viable option, as they had a greater mode, and were also less than 31, and therefore assigned these as the unique values for the region, ensuring each region had exactly one unique code. From plotting the longitude and latitude on the Tanzanian map, it appeared that some points were plotted far from Tanzania; and these were the same points that were shown to be an outlier in the boxplot. After investigating those values, it was discovered that more than 2200 samples have a longitude of 0.0 and a latitude of -2.0e^{-8}, and these coordinates were in the Gulf of Guinea; therefore, these values were dropped as they were incorrect values.

An approach to dealing with missing values differed based on the feature type. There were not many missing values in numerical features, therefore replaced with their mean value, and that approach wouldn't impact their distribution greatly. Replaced missing values in boolean features like with the most frequent value in that feature. Also, 'scheme_name' was dropped since most of it was missing.

In contrast, for categorical features, introduced a different technique, defining a function that would preserve the top five most common values while replacing the rest unique values with 'other' and missing values with 'unknown'. There were features with a value of none or None; based on the feature, it was determined to eliminate the instances or the column if it contains numerous none values and would not impact the model's performance.

The construction year feature has many samples with zeros, which is an anomaly for the years and was considered as missing values. To handle those values, Defined a new column named 'age_since_recorded' that will calculate the age using the information given. Following that, defined a function that will compute the age and return a missing value if the construction year equals zero, otherwise, it will return the subtraction of the recorded year and construction year. When looking at the unique values of the new feature, found that there are ages below zero, which is unusual. After investigating these ages, discovered that the recorded year is older than the construction year, which does not make sense, therefore eliminated these instances. And the missing values were replaced by an age that is far from the unique values to distinguish them from the others, as it would be inappropriate to replace them with their mean values because more than 30% of the values in the feature are missing, and that approach would change the distribution of the ages.

Another approach applied to deal with missing values, was first to handle categorical variables with missing data. A major difference in handling categorical data was the number to be used as the most common. The k value to be used for the most common varied with each feature, where it was decided based on where there was a maximum difference in mode, between k and k-1, and therefore some features would keep the 9 most important values, whilst others would keep the 6 most important values and set the rest to other. In handling numerical data, different conditions were made in how to handle them. Where handling the 'amount_tsh' varied on several conditions, as contextually these are factors that could affect the feature. The same principle applied for filling in other features although different conditions were used.

In dealing with construction years, an alternative approach was implemented, by using the KNNImputer to replace the missing values with the median value of their k-nearest neighbor. To optimise K, GridSearchCV was used and provided a value of 3. After KNNImputer was used to calculate the missing years, a new feature column was created 'pump's_age', which was calculated through the difference in the construction year and the year recorded. Several formatting and prepossessing were done in this stage to ensure, the year could be extracted and also for any age below 0 to be dropped, as this would not be possible in this context, as it would elucidate that the pump had not been built.

C. Scaling data

The usage of normalisation was based on the model's performance, min-max normalisation was employed using equation (1), to normalise both the water amount and the population. Just these columns were considered for normalisation since the other numerical features are location-related and normalising them might modify the process of visualising them later.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{1}$$

Standard scaling using equation (2) was another approach to transform the data. The dataset consisted of numerical and categorical columns, that have instance values between 0-3500, so standard scaling was used to ensure all features had similar weighting when used for feature selection.

$$X^i_{std} = \frac{X^i - \mu_x}{\sigma_x}, \tag{2}$$

D. Data visualisation

The data was visualised to investigate the link between the age of the water pump and its functioning. As shown in Figure 4, ages were classified into different groups, and the percentage of functionality for each group was calculated. It is clear that as the pumps get older, the percentage of them being non-functional increases; for example, 26.1% of pumps between the ages of 0 and 10 are non-functional, while 66.7% of pumps between the ages of 41 to 50 are non-functional. Unknown ages that were substituted with the age of 60, have almost the same distribution as years 11 to 20.

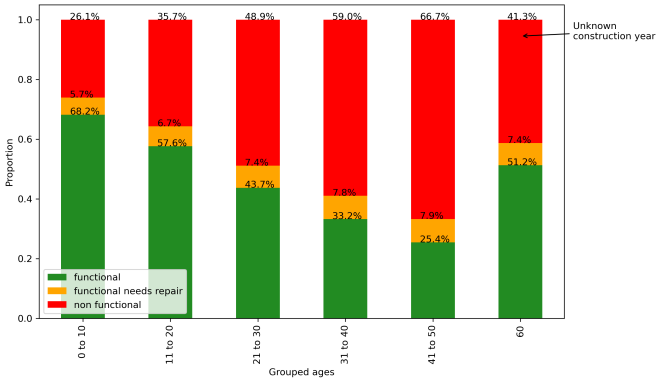

Fig. 4. Pumps functionality by the age since the last recorded year.

E. Feature engineering and selection

Categorical features must be translated into numerical values before training classification models. This was achieved by first identifying features with a large number of unique values like 'subvillage' which has 18567; these were dropped, while attributes with a lower number of unique values were used. To convert those features, used ordinal encoding where each unique value in an attribute will be mapped to a numerical value. This approach was used on both the training and testing datasets. Following that, used the f_regression feature selection method, which ranked the features in the same order as if they were positively correlated, and then chose the top ten. Data were split into training and test sets, using a ratio of 70-30 based on research provided by the University of Texas which states that the model performs best when a 70-80% split is used for training and 30-20% of the data is used for testing [11].

Another approach to converting categorical variables was to use one-hot encoding for nominal features. Using one-hot encoding over ordinal mapping for nominal features was because the value for these instances had no rank, and so using one-hot encoding ensures that there is no implied ordinal relationship. Using this method increased the feature space for this data set, and therefore to reduce this, feature selection was applied to narrow the feature space to only using features above a certain threshold of 0.027. The importance of each feature was calculated using random forest and is measured as the averaged impurity decrease computed from all decision trees in the forest, without making any assumptions about whether the data is linearly separable or not. The binary classification was converted through a binary mapping function, and ordinal data were converted to numerical data, through ordinal mapping, where the values that were set as other were always 0, and then the mode of each feature would be allocated the highest number.

F. Classification models

Several classification models were used in this paper: from ensemble algorithms to probabilistic algorithms, neural networks, and supervised learning algorithms, each with its own set of parameters and trained on various attributes. Began with a decision tree but before training, performed hyperparameter tuning via grid search to determine the ideal parameters for the model and discovered that the model performs best on training with a maximum depth of 20 and minimum sample leaf of 10. As in Figure 5, plotted the scores at different depths by calculating the accuracy scores on the train and test data, at depth 20 the training score reached nearly 95% and the test score around 75%, indicating that the model performs exceptionally well on the training set but not on the test set, indicating overfitting. And settled on a depth of 8, where both results are practically identical, for optimum precision.

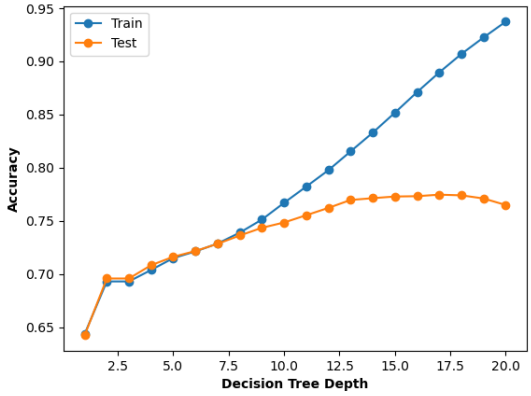

Fig. 5. Comparing the performance of the decision tree on different depths using the accuracy of both the train and test sets.

While for the random forest classifier, performed parameter tuning via random search instead of grid search since it is more computationally expensive and takes longer to find the ideal parameters. Random search selects a random collection of hyperparameters, including maximum depth, minimum samples leaf, minimum samples split, and the number of estimators, then calculates the score, returning the best set of hyperparameters with the best score as an output. According to the random search, the model works best with 200 estimators, minimum samples split of 4, minimum samples leaf of 2, and max depth of 20. However, after experimenting with parameters manually, found that the model performs better when using 300 estimators.

Multi-layer Perceptron (MLP) was implemented as another classification model, which is an example of Artificial Neural

Networks (ANNs) making the model's neurons fully connected. Prior to training, parameter tuning was a key step to determine the optimal set of parameters that maximises the model's performance. To achieve that, experimented the model with different numbers of neurons, and it performs best with 7 neurons. Next, used a random search to look for the optimal set of parameters, and it was found that the model performs best with logistic as the activation function, maximum iterations of 1000, and alpha of 0.0001.

Also, tried linear support vector machine (SVM) and logistic regression, but they struggle to perform well on large datasets for a variety of reasons, one of which is that they are computationally expensive because both are linear algorithms that involve the dot product of input features and a weight vector. Another problem is that both models require a linear relationship between the input features and the output variable, which is not always the case in big datasets. It is preferable to implement decision trees, random forest classifiers, or neural networks for these relationships.

Another classification approach was gradient boosting, which works by merging numerous models to generate a stronger model that reduces the percentage error. Before training the model, parameter tuning was performed via random search, which randomly selects parameters such as the number of estimators, learning rate, maximum features, and maximum depth and returns the parameters that perform the best. From the parameter tuning, 30 estimators and a learning rate of 0.5 will be used for the model. After evaluating the model on both the train and test sets, it appears to perform the same, indicating that there is no overfitting.

XGBoost is a more regularized form of gradient boosting and uses advanced regularization (L1 & L2), to improve the generalisation capabilities. Typically it is expected that XGBoosting performs better than gradient boosting. In addition as the number of features (39) is significantly less than the number of training samples XGBoost seems to be an appropriate model to use. A classification probabilistic algorithm was also used to compare the performance using ensemble methods and other types of classification models. Gaussian Naive Bayes (probabilistic model), is a rather simple and efficient model as it does not require any parameters.

IV. Results

As this paper focuses on different ways to carry out a predictive model task, dataset A, and B indicate the different approaches were investigated.

In evaluating steps to achieve results, data preprocessing was essential to remove any outliers, missing and incorrect values. Each dataset implemented different methods for preprocessing the data. And, to evaluate that stage, both approaches were trained on the same models with the same parameters, and the accuracies will be used then to evaluate which preprocessing approach performs better. Table I summarises the results obtained, dataset A performs better when trained with most models including, decision tree, random forest, MLP, and logistic regression. However, dataset B outperforms dataset A on linear SVM with an accuracy of 61.2%, while dataset A performs much lower with an accuracy of 35.43%.

TABLE I
Model's accuracy when trained on different datasets and approaches.

Approach	Accuracies (%)				
	DTC	RFC	MLP	LRC	SVM
Dataset A	73.48	81.07	72.53	61.18	35.43
Dataset A + normalisation	73.48	81.07	70.74	59.91	41.52
Dataset A + feature selection	73.24	79.66	70.62	58.89	53.12
Dataset B	70.82	78.46	54.43	55.37	61.2

To evaluate the performance of the models on dataset A, they were trained on different preprocessing approaches which include, using normalisation, and feature selection methods, as in Table I, in order to find the optimal method that maximises the accuracy of models when predicting the labels. After training the models on dataset A, found that random forest classifier have the highest performance on unseen data. The model achieved an accuracy of 91.79% on the training set and 81.07% on the validation set, even though there is a huge difference between the two accuracies which indicated overfitting, but the model's accuracy is the best among all other models. Also, found that tree-based algorithms such as decision tree and random forest does not need data to be normalised since the models perform the same with or without feature scaling, this is due to partitions being made based on relative feature values. On the other hand, MLP performs better when using data with no normalisation and achieved an accuracy of 72.53% on unseen data, as shown in Figure 6, the model can predict functional pumps nearly correctly, and less correctly for non-functional labels as the model predicted 2574 pumps to be functional while they are non-functional. However, the model was not able to predict any labels indicating the pump needs repair which makes it less performing comparing it with the others.

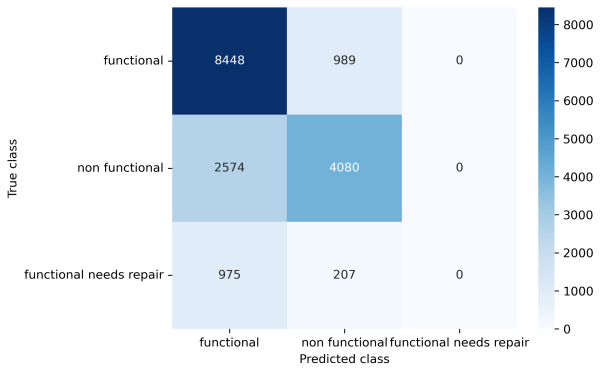

Fig. 6. Confusion matrix of predicted labels by Multi-layer Perceptron classifies.

Linear SVM and logistic regression are less reliable since

their performance is so low when predicting labels since they are hard to train on large datasets and require a linear relationship between the input and output. The performance of linear SVM is different each time it was run, and it performs better when normalising the data with an accuracy of 41.52%, and 35.43% when data is not standardised. While logistic regression had a higher accuracy when trained on normal data with an accuracy of 61.18%. Subsequently, using the best-performing model, predicted the labels of instances from the test dataset, but it was not possible to calculate the accuracy since the original labels were not provided previously. However, to evaluate the result produced a bar plot, as shown in Figure 7, the age groups had the same distribution as the training dataset, which is shown in Figure 4. This answers the first question, making the Random forest classifier the best model to predict the labels.

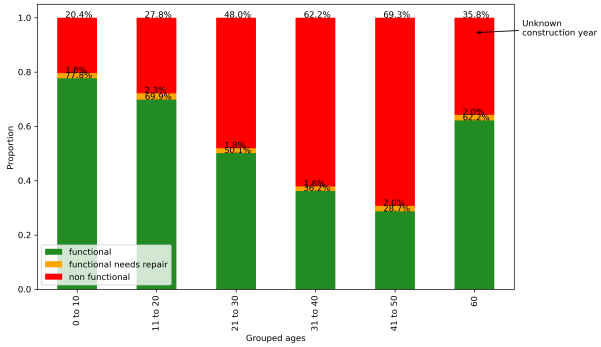

Fig. 7. Bar plot of the predicted labels across different age groups.

In evaluating the performance of the models on dataset B, they were trained using different processing steps including experimenting with different ways to convert nominal categorical features either through mapping or one hot encoding, how to handle the construction missing value, through using KNNImputer or dropping these values, which then had an effect on the feature importance, as well as experimenting with the different threshold value in feature selection. After training the model, gradient boosting seems to be the best classifier, although the results were very close, with a 0.05% difference between the gradient boosting and random forest classifier. Throughout training all models, the training test consistently performed better than the test accuracy, this highlights overfitting in the model. The models initially there was a difference of more than 5% constant for all models, which also represents overfitting in this model, and further hyperparameter techniques can be used like GridSearch.

To answer the first research question, our model was trained using a different number of features, which varied by adjusting the threshold. Through experiments, increasing the features used increased the accuracy as shown in II. All models performed were the highest when setting the threshold value to 0.008. Evidently increasing the features improves the model performance as all model's accuracy was highest using 0.008 threshold value, although increasing the number of features used to train the model, can lead to overfitting and also introduces the problem of the data points becoming sparse and the curse of dimensionality problem. The ensemble methods were the best-performing models, with the random forest classifier having the highest accuracy of 78.78% on dataset B, whereas 81.07% on dataset A. Gaussian Naïve Bayes (GNB) had the lowest accuracy, so whilst it can be argued that differentiating between the best models to use is not easy, it is clear that GNB would not be used as a predictive model, as it assumes all features are independent of each other, and this leads to inaccurate predictions.

TABLE II
ACCURACY SCORE ON THE VALIDATION SET WITH A DIFFERENT NUMBER OF FEATURES.

Threshold value	Accuracy (%)			
	XGBoost	RFC	GB	GNB
0.05 (5 features)	67.99	69.97	69.2	58.74
0.027 (9 features)	74.61	76.69	76.52	65.02
0.01 (16 features)	76.33	77.96	78.12	64.59
0.008 (23 features)	77.2	78.78	78.61	65.08

In answering the second research question, only the pump's age since the last recorded year was used as the input variable, and this led to a significant decrease in the accuracy score. However, all models seemed to perform well, considering that the input feature was only age. From the results shown in Figure 8 and Table III, the accuracy of these models ranges from 51.51-58.61%. The model performance also did not follow the trend, as when other features were used, and not just the pump's age. Random forest and decision tree had the highest performance with an accuracy of 58.59% and 58.61% respectively, those models were trained using database A. While Gaussian Naive Bayes Naive Bayes was the best-performing algorithm when trained on database B, as opposed to being one of lowest performing models in the initial research question across all that were trained. Due to the close range in accuracy score, the results elucidate that the pump's age is not a strong enough feature to predict the functionality of the pump. This is evident, also in the confusion matrix, which was not able to accurately predict the non-functional pumps. However, the age feature can have a high influence on the model's performance making it an important one.

TABLE III
ACCURACY SCORE USING ONLY PUMP'S AGE.

Model (Classifier)	Accuracy score (%)
Gaussian Naïve Bayes	58.25
Random Forest	58.06
XGB	58.06
Gradient Boost	58.06

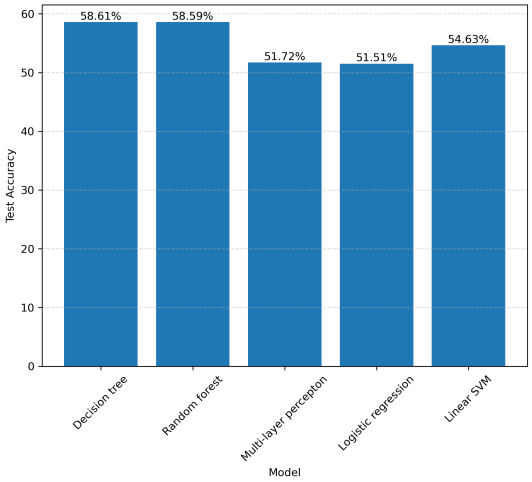

Fig. 8. Histogram of model's accuracy on the validation set when trained on age feature.

V. Discussion

Our research has explored different machine learning algorithms in predicting Tanzania water pump functionality, and has found that random forest classifier is the best predictive model. This result held true regardless of the different data pre-processing techniques and feature selection methods explored. However, the difference in accuracy results indicates the significance of the methods prior to using the predictive models.

Being provided with both training and test dataset, there were two different approaches, to preprocessing the dataset as a whole, or separately. By combining the datasets as one, ensures that during the preprocessing step, it looks at the entire dataset holistically, and therefore ensures all preprocessing steps are consistent across the whole dataset, but treating the training and test data separately provides a more realistic evaluation of the model's performance for the unseen data. In treating the dataset separately it also means that the statistical analysis values would be different, like the mean mode and quantile percentage. In this dataset, the difference in this statistical analysis was not a lot, although the combined dataset together tends to have higher values, the reason for the lack of difference may be a result of the missing values specifically in the numerical column, of the training data and also the training dataset consisting of the majority of the outliers.

In handling outliers, an approach applied to some features in both datasets was the values less than the quantile 5% or above the quantile 95% have been replaced with a null value so that they are replaced with an appropriate value when dealing with missing values. Removing these numerical outliers aims to remove the extreme values so that they do not have a significant effect on the models, and improve the data distribution for the dataset. Although for some columns such as the 'amount_tsh', this approach could not be applied as the quantile percentage was close to the mean value of the feature, and used a threshold value that was higher than the percentage, to ensure the distribution of the feature was not affected too much. Due to the mean being different for each dataset, other threshold values were used, and the values above the threshold were handled differently. One way was to set those values to be null, and they could be handled with missing values, this was the best method to preserve the integrity and distribution of the data, although this would significantly reduce the instances in the dataset. As a result another approach aiming to keep the sample size the same as the initial dataset was to divide the values above the threshold by 100. In doing so this still kept the general distribution of the data, as the maximum value, was still from the same instance, although changed the central tendency of the data, as the values had decreased.

Techniques used to handle missing values varied depending on the feature type. For categorical features, values were handled using a function that would preserve the most common values, and doing that rather than dropping these instances, preserves the sample size and therefore reduces the chance of overfitting. The function mode varied depending on the dataset to avoid generalising each categorical feature, as in some features, the difference between how common a value was close, and as a result would still be kept, and therefore the x for the mode was determined based on the feature. Also, features where the majority had missing values were dropped such as 'scheme_name'. In handling the missing values in numerical features, an approach using the mean and median was applied for features with a low number of missing values since it will not affect the distribution, however, this method can have a negative impact on features with a high number of missing values as it would change the distribution. Another approach looked more at generalising this statistical analysis, which focused more on implementing these missing values within their subgroups, for example, the 'amount_tsh' missing value imputed based on the 'region_code', 'waterpoint_type, 'source', and 'quality_group', as contextually these are factors that could affect the 'amount_tsh'. The advantage of using this method is to ensure values that are being imputed are likely to be close to the actual data rather than a generic approach. This was also applied through KNNimputer to fill in the construction year, as dropping these values would have resulted in the sample size reducing significantly as there were over 18000 missing instances. Although a benefit of dropping these instances is it provides a more accurate answer when looking at the research question focusing on the age, as we are using the actual dataset.

In converting the categorical variables, similar approaches were used for binary and ordinal data, where a mapping function was implemented to convert these features as doing so preserved the ordered relationship from the categorical feature. For nominal features, whilst mapping was also used, one hot encoding was another approach applied in converting these categorical variables, to mitigate the chance of creating an ordinal relationship that was not initially in the original dataset. However, a downside of one hot encoding is that it increases in dimensionality space. Applying the function to only selects the most common values in each category was one of the

ways to reduce the dimensionality, and reduce the chance of a sparse matrix, nonetheless, the mapping used for these nominal features was an effective approach to ensure the dimensionality of the space did not increase.

To reduce the dimensionality of this dataset, two different approaches were considered. One method focused on using the 'f_regression' feature selection method, which ranked the features in the same order as if they were positively correlated, and then chose the top ten. The usage of this approach was dependent on the model's performance, and as illustrated in Table I, models perform lower when using those methods, therefore did not consider using them. Another approach was to use the random forest classifier to assess the feature importance, and therefore could select features based on a certain threshold value, this allows one to measure the feature importance computed from all decision trees, without making assumptions about whether the data is linearly separable or not. Although with using the random forest classifier, detecting the best coefficient value to use was not as easy, due to their close nature once past a certain value.

When analysing the models we have aimed to cover a range of algorithms to explore findings from other research. To be able to compare ensemble algorithms to other models. The random forest classifier algorithm was the best-performing algorithm across all datasets with the highest predictive accuracy on the validation set. Whilst evident, that the differences in the classifier approaches were due to how the dataset had been preprocessed and the different feature selections applied. There was also a clear difference in how the best performance of the other ensemble methods such as gradient boost performed better than other types of models, as ensemble methods combine multiple models and therefore lead to better predictions. Although the difference in models did not improve in the overfitting of the data, where most of the models seemed to overfit the data, with a big discrepancy between the training and test dataset. Furthermore, the computational time it took for ensemble methods, such as gradient boosting may be an opportunity cost, when predicting the models.

VI. Conclusion

To conclude, this research provides insight into predicting the functionality of water pumps in Tanzania, using machine learning algorithms such as random forest classifier and how these algorithms can be used to monitor these water pump's based on their features. Additionally, research shows that the water pump's age alone is an inadequate feature as an indicator of the water pump's functionality. Moreover, this paper highlights the potential of alternative machine learning algorithms such as gradient boosting and XGBoost, although not only for predicting the target variables but also for other machine learning tasks such as feature selection, rather than using a Random Forest Classifier. To further expand on this project, one could explore the different approaches to feature selection like using machine learning algorithms, and other approaches such as different search algorithms. This poses numerous advantages in the future; collecting data for features that are necessary. Additionally, other data preprocessing techniques can be investigated, like dropping columns with a significant number of missing values (e.g.amount_tsh), and this could offer valuable insight and improve the performance of the model. Lastly, more time can be spent on hyperparameter tuning to obtain the optimal sets of parameters for each model and increase their performance, and use grid search instead to look for every possible set.

VII. Contribution

Both RA and TS worked on all parts of the research, writing about the methods they have done and reviewing each other's work. RA concatenated both datasets, and implemented decision tree, random forest, MLP, logistic regression, and linear SVM classification models. TS preprocessed datasets separately and developed Gaussian Naïve Bayes, Random Forest, XGBoost, and Gradient Boosting .

References

[1] J. Malleo, "Tanzania: Water crisis in Tanzania - high time for embracing technology and innovation over Nature dependence solution," allAfrica.com, https://allafrica.com/stories/202211210054.html (accessed May 13, 2023).

[2] G. Joseph, L. A. Andres, G. Chellaraj, J. Grabinsky Zabludovsky, S. C. Ayling, and Y. R. Hoo, "Why do so many water points fail in Tanzania? an empirical analysis of contributing factors," Feb. 2019.

[3] C. Seger, An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing, 2018.

[4] J. Li, K. CEHNG, S. WANG, F. MORSTATTER, and R. TREVINO, Feature Selection: A Data Perspective, 2017. - citation for why feature selection is important V. Kumar, "Feature selection: A literature review," The Smart Computing Review, vol. 4, no. 3, 2014. doi:10.6029/smartcr.2014.03.007

[5] S. Visalakshi and V. Radha, "A literature review of feature selection techniques and applications: Review of feature selection in Data Mining," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014. doi:10.1109/iccic.2014.7238499

[6] [1] G. Chandrashekar and F. Sahin, A survey on feature selection methods, 2013.

[7] B. H. Menze et al., "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of Spectral Data," BMC Bioinformatics, vol. 10, no. 1, 2009. doi:10.1186/1471-2105-10-213 -

[8] Darmatasia and A. M. Arymurthy, "Predicting the status of water pumps using data mining approach," 2016 International Workshop on Big Data and Information Security (IWBIS), 2016. doi:10.1109/iwbis.2016.7872890

[9] G. Bejarano, M. Jain, A. Ramesh, A. Seetharam, and A. Mishra, "Predictive analytics for smart water management in developing regions," 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 2018. doi:10.1109/smartcomp.2018.00047 Enterprise Miner, 2016.

[10] I. K. Chowdavarapu, and V. D. Manikandan, Data Mining the Water Pumps: Determining the functionality of Water Pumps in Tanzania using SAS® Enterprise Miner, 2016.

[11] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation" (2018). Departmental Technical Reports (CS). 1209.