Estimation of Functionality of Water Pumps in Tanzania Using Machine Learning Techniques

Coauthor
Dursun Karaca Erdemir
Department of Computer Engineering
at TOBB ETU
dkerdemir@etu.edu.tr

Coauthor
Mert Kaya
Department of Computer Engineering
at TOBB ETU
mertkaya@etu.edu.tr

Coauthor
Utku Erden
Department of Computer Engineering
at TOBB ETU
uerden@etu.edu.tr

Coauthor
Mirac Uygur Ugurlu
Department of Computer Engineering
at TOBB ETU
m.ugurlu@etu.edu.tr

Abstract

Over 24 million people are impacted by the water crisis of the United Republic of Tanzania that's almost half of the population of Tanzania. Although they have access to a reasonable amount of water, it is still a serious issue since people do not know which water pumps are functional and that often leads to long distance travels resulting in disappointment. We are examining the dataset of water pumps provided by Drivendata.org in Tanzania with the goal of predicting the operating condition of a water pump. An important step in predicting a pumps functionality is analyzing, observing and manipulating features before applying machine learning algorithms to create models to achieve higher accuracy. We created models using KNN classifier, Decision Tree, XG Boost, Extra Trees Classifier, Random Forest and MLP classifiers, Naïve Bayes, analyzed their prediction accuracies and compared them for the dataset. Our code is available at https://colab.research.google.com/drive/1uqJieHZ4PUjDi_zz8eHFULk2ZQVoApo3.

1 Introduction

Water, especially in an underdeveloped country, is essential for agriculture and industrial development and almost every aspect of daily life. Unlike most of the other sub-Saharan countries Tanzania contains a lot of water but still faces the same problem where many areas of their country have no reliable access to water. Water pumps built all around the country would solve this problem however some of these pumps are non-functional or in need of repair. A data that contains information about these pumps would aid researchers and funders to decide the specifications about the upcoming pumps. Dataset shared by the Tanzanian Ministry of Water has 59400 samples in which has attributes such as GPS coordinates, funder of the well, source of the water and so on. Dataset has target values as "functional", "non-functional" and "needs-repair". We manipulated the data to eliminate the noise and miswritten entries. To enforce numerical values on categorical columns we used LabelEncoder which reduces the possibility of correlations being formed between the columns. After inspecting the linearity of data, we have trained some models. On the competition hosted by Drivendata we have achieved a final accuracy of 82.18% and this accuracy is the 470th among 9786 competing teams. Our team alias that is visible at Drivendata is WFTWaterForTansania.

2 Dataset Description

We are using the data provided by Taarifa and Tanzanian Ministry of Water to predict which water pumps are functional, functional needs repairs and non-functional.

2.1 Feature Observations

The dataset has 40 columns which means that there are 40 features. Out of the 40 features in the data, we have 26 categorical variables, 2 Boolean variables, 10 numerical variables and 2 date variables. Some features explained in below.

- **amount_tsh** - Total static head (amount water available to waterpoint)
- **date_recorded** - The date the row was entered
- **funder** - Who funded the well
- **gps_height** - Altitude of the well
- **installer** - Organization that installed the well
- **longitude** - GPS coordinate
- **latitude** - GPS coordinate
- **wpt_name** - Name of the waterpoint if there is one
- **basin** - Geographic water basin
- **subvillage** - Geographic location
- **region** - Geographic location
- **region_code** - Geographic location (coded)
- **district_code** - Geographic location (coded)
- **lga** - Geographic location
- **ward** - Geographic location
- **population** - Population around the well
- **recorded_by** - Group entering this row of data
- **scheme_management** - Who operates the waterpoint
- **scheme_name** - Who operates the waterpoint
- **permit** - If the waterpoint is permitted
- **construction_year** - Year the waterpoint was constructed
- **extraction_type** - The kind of extraction the waterpoint uses
- **extraction_type_group** - The kind of extraction the waterpoint uses
- **extraction_type_class** - The kind of extraction the waterpoint uses
- **management** - How the waterpoint is managed
- **management_group** - How the waterpoint is managed
- **payment** - What the water costs
- **payment_type** - What the water costs
- **water_quality** - The quality of the water

- **quality_group** - The quality of the water
- **quantity** - The quantity of water
- **quantity_group** - The quantity of water
- **source** - The source of the water
- **source_type** - The source of the water
- **source_class** - The source of the water
- **waterpoint_type** - The kind of waterpoint
- **waterpoint_type_group** - The kind of waterpoint

The training set has 59400 rows and test set has 18450 rows. The target part of the training dataset's ratio can be seen in the *Figure1*. Also state of water pumps by the distribution by their region can be seen in the *Figure2*. It is expected that after training our models with the training dataset, the distribution of test data's predicted values will be similar.

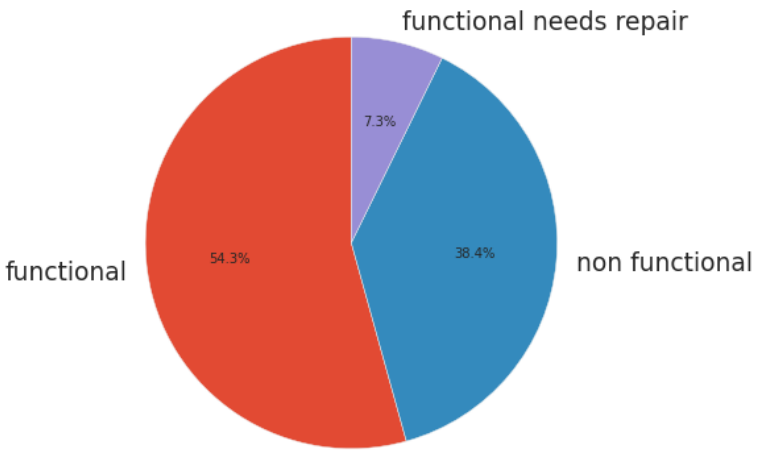

Figure 1

The target values in the data provided for the project, is neither linearly separable nor binary, therefore not linear.

This prevents the usage of linear approaches (Linear Regression, Rainbow Test, etc.) while constructing a classifier. We will be using non-linear techniques since they are more feasible.

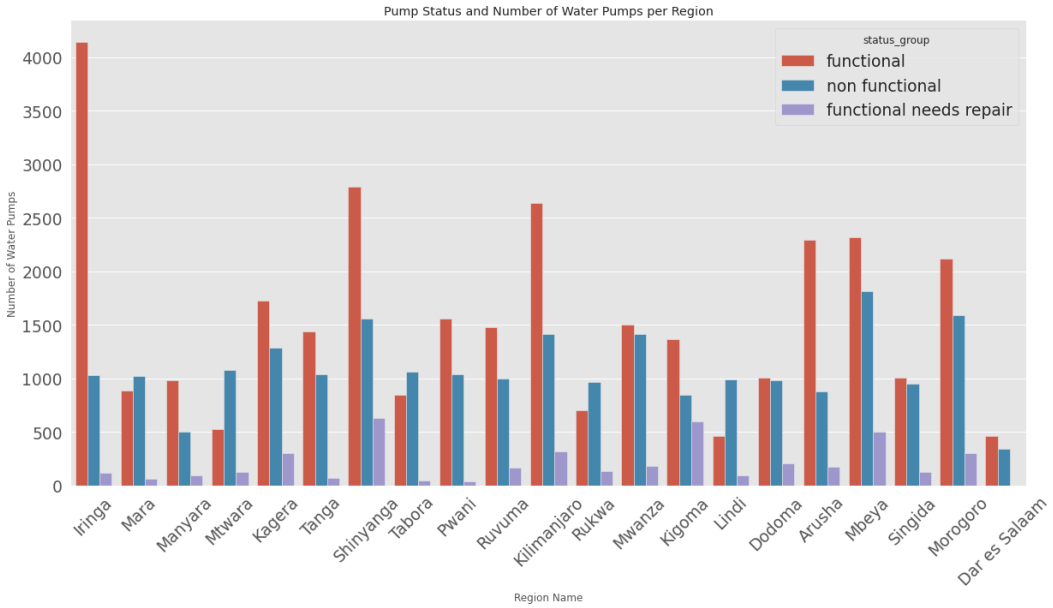

Figure 2

2.2 Missing Values

The missing values in training dataset should be dealt with. The features that has unassigned values and their quantity are shown at *Figure 3.* The percentage of missing values in whole training data is 1.939%.

Feature	Number of Unassigned Values	Column Type
funder	3635	Categorical
installer	3655	Categorical
subvillage	371	Categorical
public_meeting	3334	Boolean
scheme-management	3877	Categorical
scheme-name	28166	Categorical
permit	3056	Boolean

Figure 3

These values show that quantity of missing and Nan values, however it is observed that some features have values implying that their value is unassigned. Such as “Unknown”, “-”, “not-known”, “none”. It is necessary that handling these values properly.

It is observed that in training data the features named “construction_year”, “amount_tsh”, “num_private” have many values that are equal to 0. The percentage of 0 values can be seen in *Figure 4*.

Feature	Percentage of 0 Values (%)
amount_tsh	70.099
num_private	98.719
construction_year	34.975

Figure 4

It is observed that in training data, “recorded_by” feature’s variance is equals to 0, that means for all samples its value is always the same.

2.3 Correlations of Features

We use Pearson Correlation to show features correlation with each other. After encoding the Categorical features using Label Encoder, we create a table that shows each features correlation with each as shown in Figure 5. Label encoding is explained further at 3.1

The sample correlation coefficient r produced by Pearson Correlation measures the strength and direction of linear relationships between pairs of continuous variables. Pearson Correlation evaluates if there exists a linear correlation at the same pairs of variables in the population, represented by a population correlation coefficient, ρ (“rho”). The Pearson Correlation is a parametric measure [1].

Highly correlated features are observed. The features that has more than 90 percent correlation are classified as highly correlated features. Which are:

extraction_type and extraction_type_group: 0.949524
quantity and quantity_group: 1.000000
source and source_type: 0.943818
waterpoint_type and waterpoint_type_group: 0.982154

The features that has a correlation rate that between 60% and 90%:

funder and installer: 0.607049
region_code and district_code: 0.678602
scheme_management and management: 0.795872
extraction_type and extraction_type_class: 0.695052
extraction_type_group and extraction_type_class: 0.784300
payment and payment_type: 0.685673

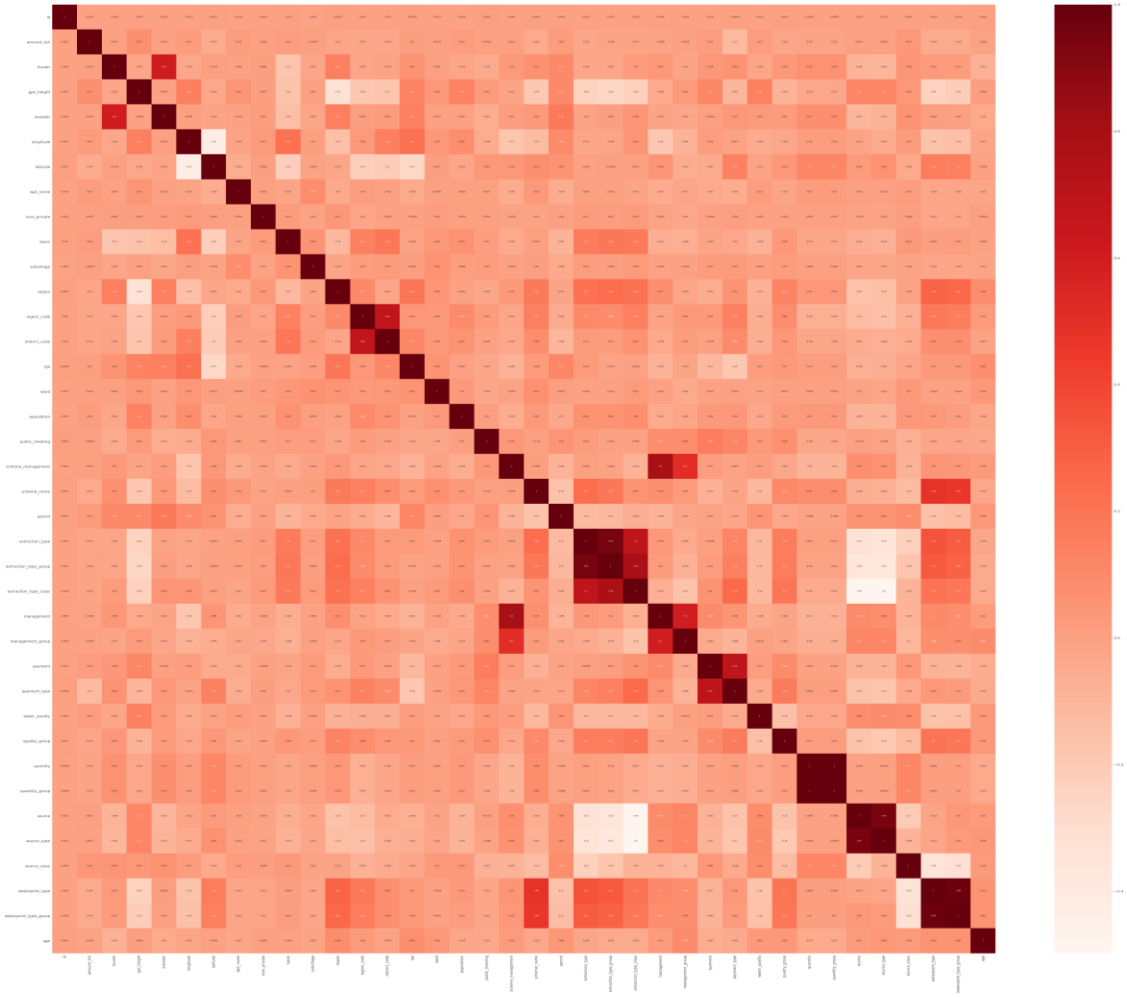

Figure 5

3 Methodology

In this part, the information gathered about the data is inspected, we discuss our methodology of feature extraction, feature selection, data transformation methods and finally we check different classifiers, create models and calculate their accuracy using test dataset.

3.1 Data Manipulation

We assigned the keyword “unknown” for all the Categorical and Boolean feature’s missing values, including the keywords that are implying that they are unassigned. This will be useful and necessary since we cannot use samples that includes unassigned variables. Also assigning non-assigned variables different values could cause a noise sample.

This exchange causes an anomaly in Boolean features that is there are no longer 2 distinct values but 3. We allowed this because the fact that a Boolean value is assigned could give an information on predicting the label. It is the same reason why we do not delete the samples that has an unassigned value.

It is necessary that we alter some values in longitude and latitude features since these columns consist some values that are not in the longitude, latitude range of Tanzania. We assumed that these values, which are 0 for latitude and 0.0002 for longitude, entered by the purpose of explaining that it is a missing value. We have replaced these values with the means of those columns calculated without these missing values.

We are extracting “num_private” and “amount_tsh” features, because they are including high percentage of values that are equal to 0. For 0’s in “construction_year”, we calculated the mean of non-0 values and assigned it to the values that are equal to 0 with this mean, since extracting the “construction_year” column would cause data loss.

It is observed that in training data, “recorded_by” feature had 0 variance thus it has been removed.

Feature named “id” is used to identify samples and it has been removed hence it has no information relevant to our machine learning models.

It can be observed from Figure 5 that some of the attributes has more than 0.9 Pearson correlation thus the values that has high Pearson correlation consists similar information for us. Further reasoning for removals:

Feature named “region” has 0.678 correlation with “district_code” and represents a similar info with region_code thus it has been removed.

Feature named “quantity” and “quantity_group” has 1.0 correlation which means they have similar information for us thus quantity has been removed.

Feature named “source” and “source_type” has 0.943 correlation which means they have similar information for us thus quantity has been removed.

Feature named “payment” has 0.685 correlation with “payment_type” thus it has been removed.

Feature named “waterpoint_type” has 0.982 correlation with “waterpoint_type_group” thus it has been removed.

Feature named “extraction_type_class” has 0.784 correlation with “extraction_type_group” thus it has been removed.

3.1.1 Label Encoding

We assumed that values that have the same letters but has upper or lower cases are same and we mapped the values that has same letters in same numerical values. By this reasoning we transform all the data into lowercase letters first. In addition, some values have whitespaces in their end, we trim all of the values for same reason.

Label Encoding is used to transform Categorical and Boolean values to numerical values. It is necessary for the making of classification models and making predictions. We label-encoded our data by a method named “Label Encoder” created by python library “sklearn” which labels each distinct categorical value from 0 to n-1 for a dataset with n features.

3.1.2 Date Parser

We have 2 date columns, which are named “construction_year” and “year_recorder”. Since we could not handle the data that way, we needed to find a way to manipulate them. For this reason, we constructed a date parser method. This method basically finds the difference between the construction year and year recorder, and creates a new column named “age”. After creating the “age” column, we extract “construction_year” and “year_recorder” columns from the data.

3.2 Feature Selection

The reason why Feature Selection is one of the core concepts of machine learning is that it hugely impacts the performance caused by irrelevant features in our data. These features can decrease the accuracy hugely by making it learn based on irrelevant features.
The benefits of feature selection:

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.

- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.[2]
- Feature Selection methods are used to determine the features which contribute most to the prediction variable. We will use feature selection methods to reduce the possibility of overfitting by removing the features that are unnecessary such as 0 variance features or the ones that consist noise. We also can improve accuracy by noise removal.

3.2.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique which extracts some features, creates some features that are correlated with the removed features and adds them to back into the data in a way retaining as much information as possible. LDA uses the information from all the features selected to create a new axis and projects the data on to the new axis in such a way as to minimizes the variance and maximizes the distance between the means of the two classes. [3]

We will use linear discriminant analysis to features that we think that are related. The features "gps_height", "latitude", "longitude" are all numeric values that gives us an information about the location of the water pump. By the intuition that these features are related, we used linear discriminant analysis to these columns and create 2 new features that are related to these 3 columns and extracted the original 3 columns.

3.2.2 Forward Selection with KBest

Forward feature selection method creates models with one-to-n features and returns the feature count for highest accuracy model among these.
K-Best algorithm created by python library "sklearn" which uses models for each combination of k features on a given dataset and returns the feature combination that has highest points. Using the combination of these methods would return the best combination of features with the optimal feature count.
K-Best calculates the point of a given combination of features by using a preselected function which is f-classif in our case. F-classif calculates Anova F value for each given sample [4]. F value, more commonly known as F1 value is used to predict test accuracy.

3.2.3 Variance Threshold

Variance Threshold is a feature selection approach which removes all features whose variance is below some threshold. First, it removes all zero-variance features (features that has the same value at all samples). The point of importance is how do we choose this threshold. [5]

$$Var[X] = p(1 - p)$$

We use Bernoulli distribution formula for choosing the threshold value. The distribution can be summarized by a single variable p that defines the probability of an outcome 1. Given this parameter, the probability for each event can be calculated as the formula on above. In the case of flipping a fair coin, the value of p would be 0.5, giving a 50% probability of each outcome. [6][7][8]

3.3 Machine Learning Classifiers

The classifiers that we use are:

- KNN
- Naïve Bayes Classifier
- Decision Tree
- MLP

- XG Boost
- Extra Trees Classifier
- Random Forest Classifier

3.3.1 KNN

The reason why KNN is a non-parametric algorithm is it does not make assumptions on data distribution. KNN also a lazy algorithm since rather than learning any model, it does a generalization to it. (KNN does not train some parameters of some function where input X gives output y).[9]

While training our model, our neighbor size is selected as 20, since we observed that our training data has a lot of noisy samples. Thus, we decreased the possibility of overfitting to noisy data. Moreover, we select the weight attribute as distance and the power parameter as Minkowski.

3.3.2 Naïve Bayes Classifier

Naïve Bayes is a probabilistic classifier that utilizes Bayes Theorem. It does an assumption which is the attributes are conditionally independent. Naïve Bayes is easy to be scaled to larger datasets since it takes linear time.[9]

While training our model, we use default parameters of Naïve Bayes since we do not have any data about features prior probabilities.

3.3.3 Decision Tree

Decision Tree makes decisions by a tree generated using features. In our case, it splits our samples into 3 homogeneous sets based on output of features Gain functions. [9]

While training our model, we do not give a max depth to our tree because performance metric is not significant to us.

3.3.4 MLP

Multilayer Perceptrons, or MLPs for short are most commonly used Neural Network model. MLPs are suitable for our dataset which is a classification problem. They are very flexible thus can be used generally to learn a mapping from inputs to outputs. [10]

In our model, we used MLP with maximum iterations number of 1000 which is the number of how many iterations the solver makes until convergence. Because of our datasets size, even though at 1000 iterations our model does not converge, training the model takes hours when we select a higher rate.

3.3.5 XG Boost

The names that XG Boost goes for are gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines Gradient boosting performs well where new models are created that predict the accuracies of prior models and then added together to make the final prediction. It uses a gradient descent algorithm to minimize the loss when adding new models. [11]

While we do not worry about the performance metric, we selected our n_estimators parameter a considerably high number to begin with. Then we changed its value to find and optimal point.

3.3.6 Extra Trees Classifier

Extremely Randomized Trees Classifier (Extra Trees Classifier) is an ensemble learning technique which combines multiple decision trees that are not correlated in a forest, like random forests. Its only difference is the way decision trees is constructed. [12] Like XGBoost, we select n_estimastors high, then we changed its value to find and optimal point.

3.3.7 Random Forest Classifier

Random Forest Classifier is an ensemble learning technique which combines multiple decision trees that are not correlated in a forest, based on the "votes" of all the trees. Random forest classifier fits a number of decision trees on various sub-samples of datasets and in order to improve accuracy, it uses average of the model and controls over-fitting. The size that sub-samples have are always the same as the original input sample size, but the samples are shuffled each iteration. [9]

Like XGBoost, we select n_estimastors high, then we changed its value to find and optimal point.

4 Results

In order to test our model's accuracies locally, we used K-fold cross validation with cv parameter being equal to 10. *Figure 6* shows some of our classifier's accuracies with some combinations of feature selection methods that we used. In the feature selection column, we used some abbreviations to point out which feature selection methods were used at that row.

Removal – No Removal: Pearson correlation has given us some results that can be interpreted differently. We checked if our data fits better without removing columns that correlate between 0.6-0.9 range or not. If "Removal" is indicated, then the features named "id", "amount_tsh", "num_private", "region", "quantity_group", "source_type", "payment", "waterpoint_type_group", "extraction_type_group" are extracted from data.

If "No removal" is indicated, then these features that are written above is contained in our data while fitting the classifiers.

LDA – No LDA: These keywords indicate whether Linear Discriminant Analysis is used before the feature selection methods or not.

Figure 7 shows a simple graph of the results that we have shown at *Figure 6*

Classifiers	Feature Selection	K-fold Accuracy (cv = 10)
KNN with 20 neighbours	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal*, No LDA	0.5398
Decision Tree	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.7440
Naïve Bayes	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.5198
MLP with max_iter = 1000	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.5431
XG Boost	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.7436
Extra Trees Classifier	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.8129
Random Forest Classifier	VarianceThreshold(threshold=(.6 * (1 - .6)), No Removal, No LDA	0.8153
KNN with 20 neighbours	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.5839
Decision Tree	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.7444
Naïve Bayes	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.6349
MLP with max_iter = 1000	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.5433

XG Boost	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.7432
Extra Trees Classifier	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.8120
Random Forest Classifier	VarianceThreshold(threshold=(.6 * (1 - .6)), Removal, LDA	0.8106
KNN with 20 neighbours	Forward Selection with k = 36*, No Removal, no LDA	0.5397
Decision Tree	Forward Selection with k = 36, No Removal, no LDA	0.7422
Naïve Bayes	Forward Selection with k = 36, No Removal, no LDA	0.5259
MLP with max_iter = 1000	Forward Selection with k = 36, No Removal, no LDA	0.5431
XG Boost	Forward Selection with k = 36, No Removal, no LDA	0.7026
Extra Trees Classifier	Forward Selection with k = 36, No Removal, no LDA	0.8128
Random Forest Classifier	Forward Selection with k = 36, No Removal, no LDA	0.8145
KNN with 20 neighbours	Forward Selection with k = 26*, Removal, LDA	0.5839
Decision Tree	Forward Selection with k = 26, Removal, LDA	0.7444
Naïve Bayes	Forward Selection with k = 26, Removal, LDA	0.6349
MLP with max_iter = 1000	Forward Selection with k = 26, Removal, LDA	0.5434
XG Boost	Forward Selection with k = 26, Removal, LDA	0.7440
Extra Trees Classifier	Forward Selection with k = 26, Removal, LDA	0.8120
Random Forest Classifier	Forward Selection with k = 26, Removal, LDA	0.8106
KNN with 20 neighbours	Removal, LDA	0.5836
Decision Tree	Removal, LDA	0.7450
Naïve Bayes	Removal, LDA	0.6210
MLP with max_iter = 1000	Removal, LDA	0.5432
XG Boost	Removal, LDA	0.7421
Extra Trees Classifier	Removal, LDA	0.8126
Random Forest Classifier	Removal, LDA	0.8101
KNN with 20 neighbours	Removal, LDA	0.5396
Decision Tree	No Removal, No LDA	0.7433
Naïve Bayes	No Removal, No LDA	0.4801
MLP with max_iter = 1000	No Removal, No LDA	0.5532
XG Boost	No Removal, No LDA	0.7451
Extra Trees Classifier	No Removal, No LDA	0.8125
Random Forest Classifier	No Removal, No LDA	0.8158

372 Figure 6

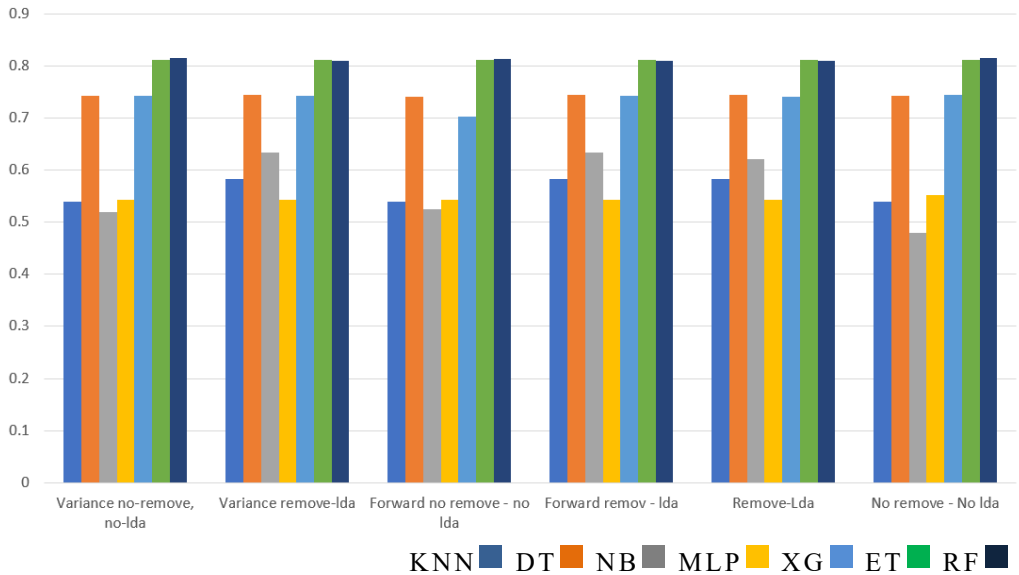

Figure 7

Our initial approach as mentioned before, was extracting the "recorded_by" feature and merging "date_recorded" and "construction_year" thus we have removed these columns and calculated a new column named "age". After dealing with missing values, our best performing model was Random Forests with an accuracy rate of 0.8158 followed by Extra Trees Classifier with 0.8125.

If we observe Naïve Bayes accuracies from *Figure 7* it is seen that it performs better whenever we reduce the features using any feature extraction technique. The reason is that Naïve Bayes assumes that features are independent as we mentioned in 3.3.2, thus this could mean that the features we are extracting are, in fact dependent to other features Even though as we extract some features Naïve Bayes performs better, Random Forest and Extra trees classifiers accuracies seems to be slightly decreased. This difference is caused by extracted features for sure, thus this tells us the features we extracted still has more effect on the target than the noise caused by them for random forests. In another perspective, the features we have deleted because of their unassigned value rates might be giving information about the target value with the fact that they are unassigned.

Extracting correlated features using LDA, Variance threshold, Forward Selection also has a slight negative affect on Random Forest's and Extra Tree's, DT's and XG Boost's accuracies, however KNN, MLP, NB accuracies are increased considerably. Our feature selection methods were useful for these models. Even eliminating "id" feature reduces accuracy for RF's so that means there is a connection of water pumps id's and their functionality or it is just a coincidence for this data.

Even though feature selection methods used improved mentioned model's accuracies, our best performing model was still Random Forest's with our initial approach.

5 Conclusion

In conclusion, we observed our dataset's features and labels, experienced with all the feature selection methods that we came across, tried Variance threshold, Forward Selection, Feature extraction with correlations and extraction of features with mostly 0.0 values, experienced label encoding methods, observed their effects on different classifiers, experimented with different classifier parameters and tried to explain the reasoning behind their resulting accuracies.

We have also learned how misinformation and null values in the dataset can affect overall accuracy, and how delicate programmers should be while manipulating data. Furthermore, we studied,

experimented with and tested over 7 different classifiers to improve total accuracy. This has given us considerable insight about how there is often multiple solutions to a problem and the importance of even the slightest improvement.

Our results show that for the dataset provided for us, after experiencing with different classifiers, the best accuracy we got was Random Forest model with our initial approach that is filling empty values and removing "recorded_by" feature and creating "age" column.

References

[1] University, K. (2020, July 13). SPSS Tutorials: Pearson Correlation. Retrieved July 23, 2020, from https://libguides.library.kent.edu/SPSS/PearsonCorr

[2] Shaikh, R. (2018, October 28). Feature Selection Techniques in Machine Learning with Python. Retrieved July 23, 2020, from https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

[3] Maklin, C. (2019, August 04). Linear Discriminant Analysis In Python. Retrieved July 23, 2020, from https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2

[4] Feature_selection, S. (2020). Sklearn.feature_selection.f_classif¶. Retrieved July 23, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html

[5] Machine Learning Library, P. (2020). Variance Threshold. Retrieved July 23, 2020, from https://php-ml.readthedocs.io/en/latest/machine-learning/feature-selection/variance-threshold/

[6] VarianceThreshold, S. (2020). Sklearn.feature_selection.VarianceThreshold¶. Retrieved July 23, 2020, from https://scikitlearn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

[7] Distribution, B. (2020, May 12). Bernoulli distribution. Retrieved July 23, 2020, from https://en.wikipedia.org/wiki/Bernoulli_distribution

[8] Brownlee, J. (2020, February 09). Discrete Probability Distributions for Machine Learning. Retrieved July 23, 2020, from https://machinelearningmastery.com/discrete-probability-distributions-for-machine-learning/

[9] Gahukar, G. (2019, January 25). Classification Algorithms in Machine Learning... Retrieved July 24, 2020, from https://medium.com/datadriveninvestor/classification-algorithms-in-machine-learning-85c0ab65ff4

[10] Brownlee, J. (2019, August 19). When to Use MLP, CNN, and RNN Neural Networks. Retrieved July 24, 2020, from https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/

[11] Brownlee, J. (2020, April 21). A Gentle Introduction to XGBoost for Applied Machine Learning. Retrieved July 24, 2020, from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

[12] Gupta, A. (2020, July 01). ML: Extra Tree Classifier for Feature Selection. Retrieved July 24, 2020, from https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/