Pump Functionality Prediction
Using Naïve Bayes Classifier

Hayden McAlpin
hpmcalpin97@gmail.com
December 2021

ABSTRACT

In this paper, I describe the process and evaluation of using the Naïve Bayes Classifier to predict the functionality of various water pumps across Tanzania. This model will take into account several factors from the given dataset and output the predicted functionality class for the given pumps. With these predictions I will build a Graphical User Interface (GUI) in RStudio to show the locations via longitude and latitude of the water pumps and their associated status for users to see.

1. INTRODUCTION

Water is an essential resource to a quality and healthy lifestyle. However, it is highly important that the water people consume is clean. Unclean water can hold contaminants that are harmful to the human body. Contaminants in ground water may include natural substances like arsenic and manganese leaching from soil, run off from agricultural activities, discharge from sewage treatment, and leakage from landfills [8]. While clean water may be a commonality in the United States or other first world countries, many third world countries do not have this blessing. Millions of people across Tanzania, located in East Africa, don't have access to a safe source of water and even more don't have access to healthy sanitation sources [9]. One of the sources of water they have access to, come from various water pumps located across the country. With the little access the citizens have, it is important to maintain these water pumps and keep them functional as long as possible. When the pumps are no longer functional or need repair, it is important for them to be fixed or replaced according to their functionality status.

DrivenData.org holds many data science competitions that when successful, can be very beneficial to solving their associated problems. Some of these projects include Predicting Disease Spread, Richter's Predictor: Modeling Earthquake Damage, and (where the idea for the project came from) Data Mining the Water Table. DrivenData makes use of Taarifa, an open source platform that holds the reporting and triaging data of various infrastructure related issues, to gather data of the water pumps across Tanzania [2]. These competitions give individuals a chance to enhance their data science skills and potentially help make a difference in the world around them. This contest in particular, asks participants to build a model to predict whether a pump with various characteristics (detailed in the Data section) are functional, need repair, or nonfunctional. To attack this problem, I will be implementing a classification technique. It serves as the optimal data mining task type for class prediction [3]. The classification algorithm to be performed is the Naïve Bayes Classifier. With the results of this classifying model, I will build a GUI to display the location of the pumps in the user selected areas. This project aims to successfully predict the functionality of various pumps in the areas of Tanzania while also developing an application to aid individuals in visually identifying locations of pumps that need repair or replacement. This will allow those in charge of water pump maintenance to formulate the next steps in maintaining the pumps and visualize which areas of Tanzania need more repairs or replacements than others.

This project gives me the me the opportunity to gain quality data science experience in a classification problem context. All coding and project development will be done using R language and in the RStudio IDE. Beginning and completing a complete modeling process, starting with the problem understanding, through data analysis, model training and testing, and ending with the model's validation will add to my experience and knowledge of data science projects and classification concepts.

2. Related Work

The Naïve Bayes Classifier is a commonly used classification algorithm in the world of data science. As expected, there is a significant amount of previous projects that focus on using the Naïve Bayes classifier for various classification problems. This includes feature selection for text classification[1], sentiment analysis[8] (meaning the results or class labels can be positive, negative, or neutral), and identifying whether or not a patient may have heart disease [5]. This project uses a combination of aspects of the aforementioned projects. The classification is similar to that of the sentiment analysis as the pumps can belong to three different classes. The algorithm operates in similar fashion to all the projects as it identifies various probabilities of the train data to make classifications or determinations of the test data. The Naïve Classifier allows the calculations to be simplified and easily applied as it assumes that the features are unrelated [6]. Since it is widely used in various other classification problems and widely accepted it will be suitable to use in an exploration of solving this problem.

3. The Data

3.1 Ingestion

DrivenData.org supplies all data for the competition originating from the Taarifa database that holds data from the Tanzania Ministry of Water. The data is presented in 3 separate files. The first is the train.csv file (19442 KB, 59400 observations) which contains 40 variables:

- ID – Pump id [Nominal]
- amount_tsh - Total static head (amount water available to waterpoint) [Numeric]
- date_recorded - The date the row was entered [Ordinal]
- funder - Who funded the well [Nominal]
- gps_height - Altitude of the well [Ordinal]
- installer - Organization that installed the well [Nominal]
- longitude - GPS coordinate [Ordinal]
- latitude - GPS coordinate [Ordinal]
- wpt_name - Name of the waterpoint if there is one [Nominal]
- basin - Geographic water basin [Nominal]
- subvillage - Geographic location [Nominal]
- region - Geographic location [Nominal]
- region_code - Geographic location (coded) [Nominal]
- district_code - Geographic location (coded) [Nominal]
- lga - Geographic location [Nominal]
- ward - Geographic location [Nominal]
- population - Population around the well [Numeric]
- public_meeting - True/False [Binary]
- recorded_by - Group entering this row of data [Nominal]
- scheme_management - Who operates the waterpoint [Nominal]
- scheme_name - Who operates the waterpoint [Nominal]
- permit - If the waterpoint is permitted [Nominal]
- construction_year - Year the waterpoint was constructed [Ordinal]
- extraction_type - The kind of extraction the waterpoint uses [Nominal]
- extraction_type_group - The kind of extraction the waterpoint uses [Nominal]

- extraction_type_class - The kind of extraction the waterpoint uses [Nominal]
- management - How the waterpoint is managed [Nominal]
- management_group - How the waterpoint is managed [Nominal]
- payment - What the water costs [Numeric]
- payment_type - What the water costs [Numeric]
- water_quality - The quality of the water [Nominal]
- quality_group - The quality of the water [Nominal]
- quantity - The quantity of water [Numeric]
- quantity_group - The quantity of water [Nominal]
- source - The source of the water [Nominal]
- source_type - The source of the water [Nominal]
- source_class - The source of the water [Nominal]
- waterpoint_type - The kind of waterpoint [Nominal]
- waterpoint_type_group - The kind of waterpoint [Nominal]

This will be the data set used to train the model.

The second data set is in the trainresults.csv file (1180 KB, 59400 observations). It contains only two variables:

- ID – Same as Train set
- Status_Group – Labeled functionality of the pump [Nominal]

The third data set is in the test.csv file (4860 KB, 14850 observations). This contains the same 40 variables as the train.csv file.

3.2 Preparation

First the data is read into the R-script and put into data frames. In order to properly use and explore the train data set, the individual observation need to have their functionality labels assigned. To accomplish this, I used the *merge*() function in R to add the *status_group* labels to the train data frame by matching the observations by their id value. Once merged, the now complete Train data frame was tested for missing values using the following line of code:

```
Train[which(!complete.cases(Train)),]
```

This returns all rows in the data frame that have missing values. Only a small fraction of the data frame had missing values so these rows were removed as missing values can cause some difficulties and inaccuracies in the modeling process. Once the Train data contains only instances with complete information, the specific variables used for modeling were chosen. The largest assumption when using the Naïve Bayes algorithm is that the variables in the data are independent of one another. Therefore, I decided to choose variables that logically had little coordination and steered away from choosing multiple variables that contain similar information. The chosen variables were *ID, amount_tsh, longitude, latitude, basin, population, construction_year, extraction_type, water_quality, source_type,* and *waterpoint_type_group*. Many of the variables related to the location of the pumps so I only selected *longitude, latitude,* and *basin* to show their locations. The *longitude* and *latitude* values will be used to graphically depict their location while *basin* will be used in the modeling process.

3.3 Exploration

Various graphs are shown below to demonstrate who the different variables are distributed and how they vary by their classification group.

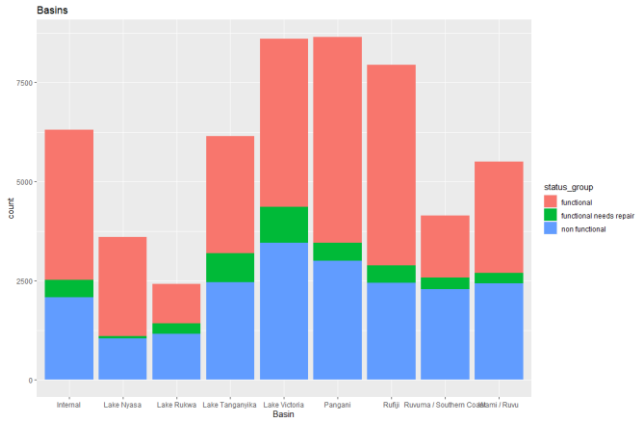

Figure 1. Basin by functionality status

Figure 1 shows the most common locations, relative to the basin, are Lake Victoria, Pangani, and Rufiji. All of the basins have a varying amount of each level of functionality. Most basins have a larger proportion of pumps that are nonfunctional.

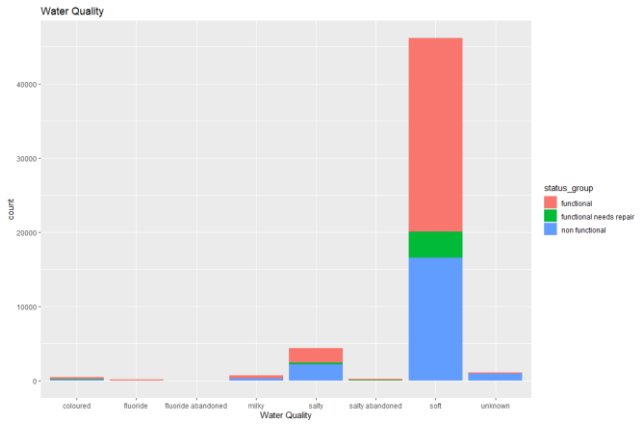

Figure 2. Basin by functionality status

As seen in Figure 2 a vast majority of the pump's water quality are soft. This may cause differing results in the model but it will be used in the initial evaluation to start.

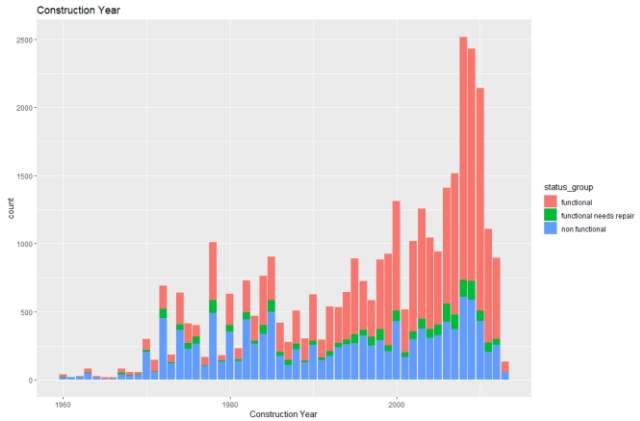

Figure 3. Construction Year by functionality status

Most of the pumps were built after the year 2000 and some date back to before 1960. As seen in Figure 3. A majority of the pumps are currently not functional.

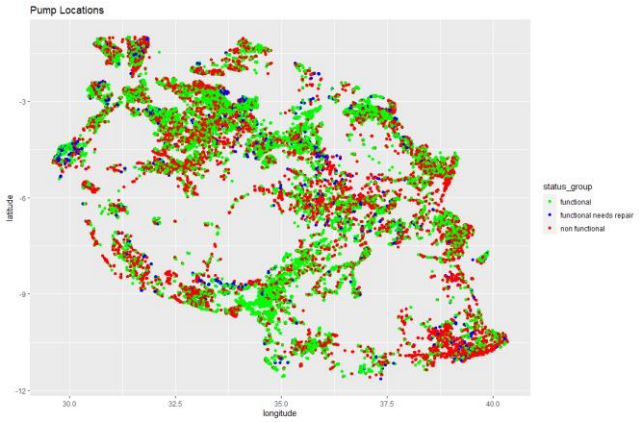

Figure 4. Pump locations by functionality status

Figure 4 shows how the different functionality of pumps is distributed across areas of Tanzania. Most of the country has some variability in each region. However, the south east region has a far more condense distribution of nonfunctional pumps and the area just south of the center has a large concentration of functional pumps.

4. Methodology

4.1 Naïve Bayes Classifier

The driving force of this classification model is the Naïve Bayes Classifier, as mentioned previously. The classifier is modeled after the Bayes formula:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Where $P(A|B)$ is the posterior, $P(B|A)$ is the likelihood, $P(A)$ is the prior, and $P(B)$ is the marginalization or evidence. However, for the Naïve Bayes Classifier the marginalization or evidence is not calculated [4]. In the context of this problem $A\ is\ functionality\ class$ and $B\ is\ the\ combination\ of\ data\ for\ each\ pump$. The likelihood is calculated by multiplying the probability of each variable given the functionality label, since we are assuming independence. The prior is calculated by determining the probability of each class label. All probabilities are calculated from the train or known data frame. The posterior is calculated for each class variable and then whichever class has the highest probably is assigned to the given instance.

4.1 Implementation

First the priors are calculates using the $prop.table()$ function. This returns the proportion or probability of each label in the given table. For the priors, the function is passed the $status\_group$ column from the Train data frame.

Second, the likelihood's of the instances are calculated. The Train data frame is first split into three data frames for each functionality type (functional, functional needs repair, and non functional). The process of finding the probabilities for each individual attribute varies by the data types. The $amount\_tsh$ and $population$ attributes are numerical and therefore are calculated using the following formula:

$$P(B) = 1/\sqrt{2\pi\sigma^2}e^{-\left(\frac{(B-\mu)^2}{2\sigma^2}\right)}$$

While the other attributes are categorical and so their probabilities are found in a similar fashion to that of the priors, using a probability table from $prop.table()$. These probabilities all come from self built functions. An example for each type is shown below:

```
pPopulation <- function(x, y) {

  mu <- mean(as.numeric(y))
  sig2 <- var(as.numeric(y))

  p <- vector(mode = "list", length = length(x))
  p <- 1/sqrt(2*pi*sig2)*exp(-(x-mu)^2/(2*sig2))

  return(p)
}
```

This is the type of function used to find the associated probabilities for the $amount\_tsh$ and $population$ attributes. The categorical attributes' probabilities are found in a similar fashion to the function given below:

```
pWaterpoint <- function(x, y) {

prob <- prop.table(table(y))
p <- vector(mode = "list", length = length(x))

p[which(x == "cattle trough")] <- prob["cattle trough"]
p[which(x     == "communal standpipe")] <- prob["communal
standpipe"]
p[which(x == "dam")] <- prob["dam"]
p[which(x == "hand pump")] <- prob["hand pump"]
p[which(x == "improved spring")] <- prob["improved spring"]
p[which(x == "other")] <- prob["other"]

return(p)
}
```

Once these functions are created the probabilities of the instances can be found. The probabilities for each class type are then calculated by storing the probabilities in 3 separate data frames (one for each functionality class). Note: The probabilities found in for the categorical variables must be converted to numerical to be useful in the data frames. To do this I used the $as.numerical()$ function. The likelihood for each instance is then calculated by multiplying the probabilities for each attribute used in the model together. The Probability for the instance as a whole is then calculated by multiplying the likelihood by the associated prior probability. These values are stored in the $Probability$ column of the data frames.

4.1 Model Evaluation

To evaluate the developed model, the F1 score is used. This is calculated by the following formula:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Where,

$$precision = \frac{TruePositives}{True\ Positives + FalsePositives}$$

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

This values was calculated for various combinations of attributes by testing the model against the Test data. The following table shows some of these calculations:

Table 2. F1 Score

Attributes	F1 Score
All 8	0.4906
No extraction	0.4954
No waterpoint	0.4962
No waterpoint No extraction	0.4987
No waterpoint No year	0.49939
No waterpoint No year No population	0.4986
No waterpoint No year No quality	0.4998
No waterpoint No year No quality No population	0.4993
No waterpoint No year No quality No source	0.4930

To begin, all attributes were considered and the F1 score calculated. From that point on the combination of attributes was evaluated in stages. In each stage, one attribute was removed (top 2 from each stage) being displayed in the table. The attribute that resulted in the maximum improvement of F1 score was permanently removed from the combinations. This continued until the F1 score no longer saw an improvement from attribute removal. Through this process I was able to determine that the use of attributes $amoun\_tsh$, $population, basin, extraction\_type$, and $source\_type$ was the optimal combination for the model. While the F1 scores that were found were not optimal they still resulted in some significant predictions and therefore can be useful for this project.

5. Results

The combination of attributes found above was then used on the model using the Test data. These results were then stored in a data frame $Results$ which had columns $ID, longitude, latitude,$ and $Functionality$. The following plot shows the distribution of the predicted functionality of the pumps.

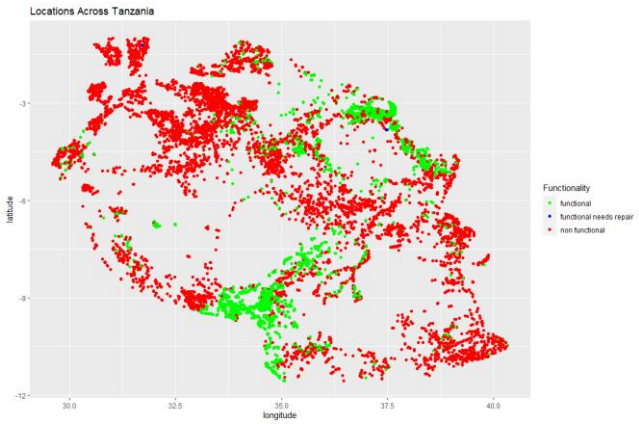

Figure 5. Pump locations by predicted functionality status

The model predicted that a vast majority of the pumps would be nonfunctional. There are however some common areas that are a majority of functional pumps. The area south of the center of Tanzania is mostly classified as functional just as it had been in the Train data. These results were then used in the GUI described next.

6. Gui

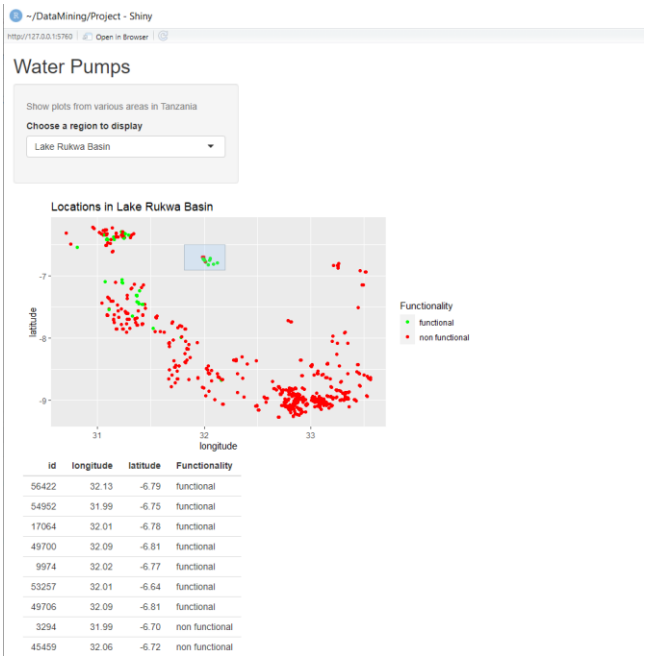

Figure 6. Pump locations by predicted functionality status

Figure 6 shows a screenshot of the GUI. The GUI was developed in R using the shiny library. This library allows someone to build their own GUI's using various tools. For this project I was able to develop a GUI that allows the user to select which region they would like to view. The various options are the entire country of Tanzania or the various Basins observed in the data. The user can also identify the individual data for each point by capturing it in the brush region. This is done by clocking and dragging the mouse over the desired area of the graph. The GUI will then show a table for the data of the selected points.

7. Conclusion

The results of the project allow users to get an idea of which areas need the most attention and most work on the water pumps. This project was a perfect opportunity to see the Naïve Bayes Classifier in practice and in a greater detail than I had previously known. While the model developed was not optimal it is still suitable for understanding how the classifier works and in the development of the predictions and resulting GUI. The get more accurate results, other algorithms and other classifiers may be considered.

I gained a lot of knowledge and practical data science experience through the entire process of the project. Completing a project from all stages is very valuable experience. I also gained a lot of knowledge in coding in R that I hadn't previously had.

8. Future work

Some potential ideas to build on this project include:

- Exploring all combinations of the attributes in the Test data
- Observing how the Bayes Classifier works and not assume independence of attributes

- Look at various other classification algorithms and do a study on which performs better in this context.

9. REFERENCES

[1] Chen, Jingnian, et al. "Feature selection for text classification with Naïve Bayes." Expert Systems with Applications 36.3 (2009): 5432-5435.

[2] DrivenData.org. Date accessed: September 15, 2021.

[3] Kabakchieva, Dorina. Predicting Student Performance by Using Data Mining Methods for Classification. March 2013.

[4] Kumar Reddy, Suman. “Categorical Naïve Bayes Classifier implementation in Python”. October 31, 2020..

[5] Pattekari, Shadab Adam, and Asma Parveen. "Prediction system for heart disease using Naïve Bayes." International Journal of Advanced Computer and Mathematical Sciences 3.3 (2012): 290-294.

[6] Saxena, Rahul. “How the Naïve Bayes Classifier Works in Machine Learning”. February 6, 2017..

[7] Thwe, Phyu, Yi Yi Aung, and Cho Cho Lwin. "NAÏVE BAYES CLASSIFIER FOR SENTIMENT ANALYSIS." International Journal Of All Research Writings 3.7 (2021): 32-35.

[8] Van Leeuwen, F. X. R. "Safe drinking water: the toxicologist's approach." Food and Chemical Toxicology 38 (2000): S51-S58.

[9] Water.org. “Tanzania’s water and sanitation crisis”. Date accessed September 15, 2021.