A REPORT

ON

PREDICTIVE MAINTENANCE

BY

Names of the Students	**ID. Numbers**
SHIVANG SINGH	2018A7PS0115H
BHAVYESH DESAI	2018A7PS0164H

AT

HAPPIEST MINDS (CoE Analytics), BANGALORE

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JUNE, 2020)

A REPORT

ON

PREDICTIVE MAINTENANCE

BY

Names of the Students	ID Numbers	Discipline
SHIVANG SINGH	2018A7PS0115H	B.E. COMPUTER SCIENCE
BHAVYESH DESAI	2018A7PS0164H	B.E. COMPUTER SCIENCE

Prepared in partial fulfillment of
thePractice School-I Course Nos.
BITSC221/BITSC231/BITSC241

AT

HAPPIEST MINDS (CoE Analytics), BANGALORE

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JUNE, 2020)

ACKNOWLEDGEMENT

All in all, PS-1 has been a great learning and enriching experience so far. Though we have put in our best efforts towards the Project, it wouldn't have been possible without our Instructor, Mentor, the Organization (Happiest Minds) and our College (B.I.T.S. Pilani).

We would like to express our sincerest gratitude to **Happiest Minds** and our **Industry Mentor, Mr Toram Suman** for not only providing us with an opportunity to work as interns in such an esteemed organization but also giving us their valuable time to provide us with the required guidance and help. We would like to thank our Mentor for providing us with such a wonderful opportunity to work on real life data and get hands on experience on such a fast-growing topic. We are indebted for all his help and guidance throughout the journey of Practice School-1.

We would also like to thank our **PS instructor, Prof. Aruna Malapati** for her constant support and mentorship. Her guidance and encouragement have kept us motivated so far and made us want to perform better.

We would also extend our gratitude to our college, **B.I.T.S. PILANI** for giving us the opportunity to work in the IT industry and gain important technical and soft skills. Especially during this unprecedented time where a lot of Companies are not looking to give Internships. We really appreciate the effort that was put in to give us this opportunity.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

(RAJASTHAN)

PRACTICE SCHOOL DIVISION

Station: Happiest Minds (CoE Analytics) **Centre:** Bangalore

Duration: 41 Days **Date of Start:** May 18, 2020

Date of Submission: June 23, 2020.

Title of the Project: Predictive Maintenance

Names of the Students	ID Numbers	Discipline
SHIVANG SINGH	2018A7PS0115H	B.E. COMPUTER SCIENCE
BHAVYESH DESAI	2018A7PS0164H	B.E. COMPUTER SCIENCE

Industry Expert:

Name: Toram Suman **Designation:** Lead Senior Data Scientist

Name of the PS Faculty: Prof. Aruna Malapati

Key Words:

Predictive Maintenance, Supervised Learning, Classification Problem in Machine Learning, Failure Behaviour, Data Exploration and Optimization, Feature Engineering, Collinearity Tests of Independent labels, Odds Ratio, Logistic Regression, Random Forest, Overfitting, UnderFitting, SMOTE, Confusion and Classification Matrix, ROC Curves.

Project Areas:

Tanzanian Water Pump Problem, Classification Problem in Machine Learning based on a set of Historical Data, Supervised Learning Neural Network, Statistical Analysis in Deep Learning, Feature Engineering in Python, Data Science, Binary Logistic Regression and Random Forest.

Abstract:

Predictive Maintenance provides various fields of Industries like Automotive, Production, Airline, Manufacturing, etc. an opportunity to reduce maintenance costs by predicting failures and its types beforehand. Since the cost incurred due to failure of a product is very high, Industries often end up repairing the product way before it even needs maintenance. The ability to predict the occurrence and type of failure reduces the cost of preemptive maintenance.

Our project focuses specifically on one such problem, the Tanzanian Water Pump Problem. We work on the data collected by the Tanzanian Government about various water pumps across Tanzania. Our aim is to classify the test set data into categories of Functional, Non-Functional and Functional but needs repair.

Shivang Singh

Bhavyesh Desai

Signatures of Students **Signature of PS Faculty**

Date: 23 June, 2020 **Date:** 23 June, 2020

TABLE OF CONTENTS

INTRODUCTION

Predictive maintenance refers to the group of techniques that help us in **estimating the condition** of various machinery/equipment.
Often in an industrial environment, the cost of replacement/maintenance of a machine is far less than the cost incurred due to its failure. That is why often machinery is **replaced/serviced prematurely**, resulting in wastage of money that could have been avoided.

Predictive maintenance aims at using various techniques of Machine Learning such as **Logistic Regression** and **Random Forest Classification** to predict the condition/status of the machinery.

The problem can be treated as a **classification problem** of identifying fault and non-faulty machinery from the data set or as a regression problem of predicting the estimated time left before the failure of the various machines such as engines.

As the aim of the problem is to predict values related to the condition of the machinery, these values may be used to **efficiently service** the machines only when required, hence **preventing extra costs** incurred due to either premature servicing of the machines or the cost that occurs due to the failure of the machines.

Depending on the problem statement and the data set, it may be a Classification Problem (Supervised or Unsupervised) or it may be a Regression Problem (Time Series Analysis).

MAIN TEXT

DOMAIN - Predictive Maintenance

- It involves a group of techniques that help in lowering Industrial costs for regular maintenance of products by estimating the condition of the current equipment.
- It helps in predicting when any equipment needs repair so as to reduce cost of **premature maintenance** while preventing any equipment Failure.
- Has a lot of applications in various industries like Airline, Manufacturing, Chemical Plants, Automobile, etc.
- Broadly covers three main types of solutions:
    - **Supervised Learning** - Use Historical Data to predict faults or time to failure for an equipment.
    - **Unsupervised Learning** - Finding the categories of faults itself based on the dataset. Using Segmentation and clustering.
    - **Time Series Analysis** - Studying the data collected by the sensors for each time cycle and using this to determine the time to failure.
- We decided to go ahead with the **Supervised Learning Classification Problem**.
- Some of the articles useful for the same are:
    - Quality Control
    - Tanzanian Water Pump

PROBLEM STATEMENT

- Once the domain was finalized, we began exploring multiple problem statements pertaining to it.
- On discussing some problem statements with our mentor, based on feasibility, time constraints,available dataset, we decided to go ahead with the **Tanzanian Water Pump Problem.**
- The relevance of the problem is justified due to the fact that it is organized as a competition on drivendata.
- It essentially requires us to classify the water pumps in Tanzania into 3 categories
    - **Functional,**
    - **Non-Functional,**
    - **Functional but needs repair**

  based on information about their installer, height, construction year,scheme,etc.
- Its relevance lies in the fact that if we are successful in predicting this, it can help improve maintenance operations and ensure that clean, potable water is available to communities across Tanzania.
- We consider ourselves lucky to have encountered an authentic dataset (provided by the Tanzanian Ministry of Water) with enormous amounts of information.
- The dataset available for this problem includes entries for around **59k water pumps** and for about **40 different labels**. It provides information like altitude, management group, quality group, population, scheme, location,etc.
- Due to the size of the dataset, our next task was to perform **Data Exploration and Optimization.** We had to make sure that the dataset we use for the model was not just large but clean as well.

DATA MANAGEMENT

DATA EXPLORATION

Available Dataset

- The **Classification label is the status_group label** with 3 possible entries as Functional, non functional and functional needs repair.
- The dataset consists of 40 labels of information that includes :
    - `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
    - `num_private` - private number of the pump
    - `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
    - `public_meeting` - True/False
    - `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
- There were **59,403** water pumps in Tanzania for which the data was provided.

Data Cleanup

- The takeaway observation from the dataset was that there were a lot of redundant labels as well as there were multiple missing entries for various labels.
- Our **First task was to eliminate labels** that were either redundant (had similar entries to another label) or were irrelevant (had no correlation to the classification label)
- **Next task was to remove unnecessary entries** which were either missing (incomplete information) or they were incorrect (eg 0 for categorical labels)
- **Label Cleanup**
    - **Redundant Labels** : Some of the label pairs like (installer,funder),(management,management_group),etc. had most of their entries to be the same. So, it made sense to go ahead with only one from each pair.
    - **Irrelevant labels**: Some of the labels like id, date_recorded,etc (called Assessment variables) were just used to enter data into the dataset. They have no correlation with the classification label (“status_group”) and hence were dropped.
    - **Unclear labels**: Labels like num_private had very little information in the description and had a lot of entries (one-third) as 0. So, it was dropped.
- **Row Cleanup**
    - **Missing entries**: The labels that had less(or none) missing entries, the missing entries were removed. If a label had most entries as missing, the label itself was dropped.

    - **Incorrect entries**: Entries with values like unknown, None, nan were dropped as well.
- Next, we saw the pictorial description of the location depicted using the latitude, longitude pairs and then decided to go ahead with using Region for describing the location aspect of the Pump. Reasons for choosing **Region** as the location aspect out of possible options like latitude,longitude,lga,ward,district,region_code and district_code:
    - Minimum Missing/Null entries.
    - 21 Unique values and the water pumps are well distributed between those 21 regions.
    - Though it seems region and region_code would be a 1-1 mapping but that was seen to be not true. region_code had less relevant data than region label.
- The focus now was to categorize the remaining labels into two types and perform Data Exploration on each one of them:
    - **Numerical Label**
        - Descriptive statistics (mean,min,max,median,std,etc.)
        - Relationship with status_group (classification label) using a boxplot.
    - **Categorical Label**
        - Various categories for the labels possible.
        - Relationship with status_group (classification label) using a stacked bar graph.

Updated Dataset

- After completing the cleanup of the data set, we are left with **26569 rows** and with the following **12 independent labels**:
    - `gps_height` - Altitude of the well
    - `region` - Geographic location
    - `population` - Population around the well
    - `public_meeting` - True/False
    - `permit` - If the waterpoint is permitted
    - `construction_year` - Year in which the pump was constructed.
    - `management` - How the waterpoint is managed
    - `payment_type` - What the water costs
    - `quality_group` - The quality of the water
    - `quantity_group` - The quantity of water
    - `source_type` - The source of the water
    - `waterpoint_type_group` - The kind of waterpoint

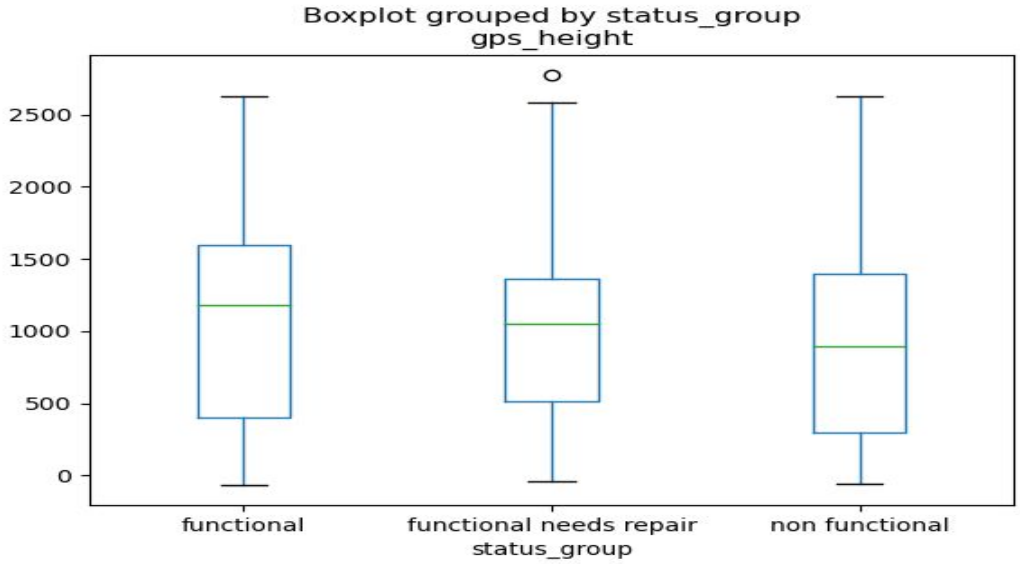
- Change in **% distribution** of **classification label (status_group)** after updating the dataset.
    - **Functional** - 54%-57%
    - **Non Functional** - 39%-36%
    - **Functional Needs Repair** - 7%-7%

DATA VISUALIZATION

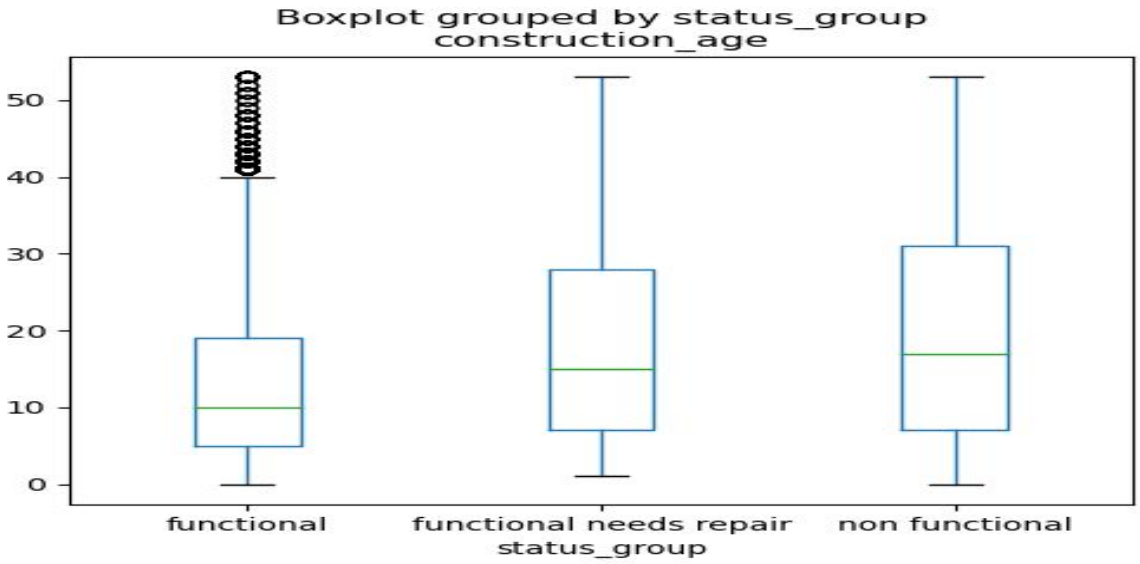
- **Numerical features** such as **'gps_height', 'population' and 'construction_year'** can be represented by real values. Instead of directly using construction_year, it has been converted into the more meaningful data feature **'age'** by subtracting the construction year from the current year.

    - **gps_height**

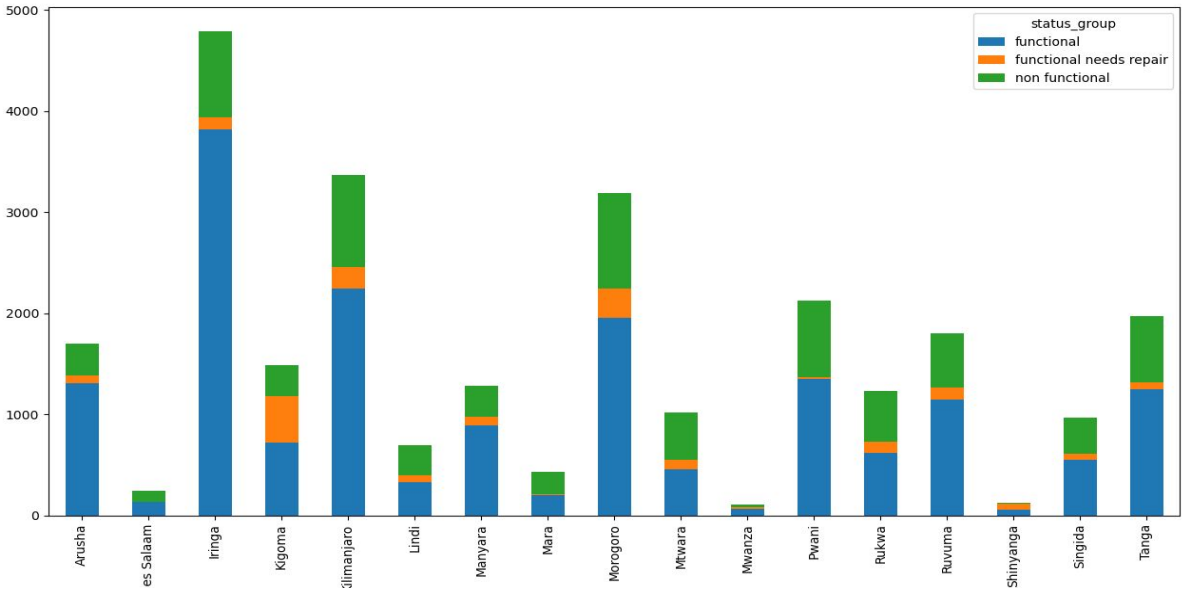
        - count 26569.000000
        - mean 1007.794083
        - std 638.923582
        - min -63.000000
        - 25% 360.000000
        - 50% 1098.000000
        - 75% 1531.000000
        - max 2770.000000
        - Name: gps_height, dtype: float64



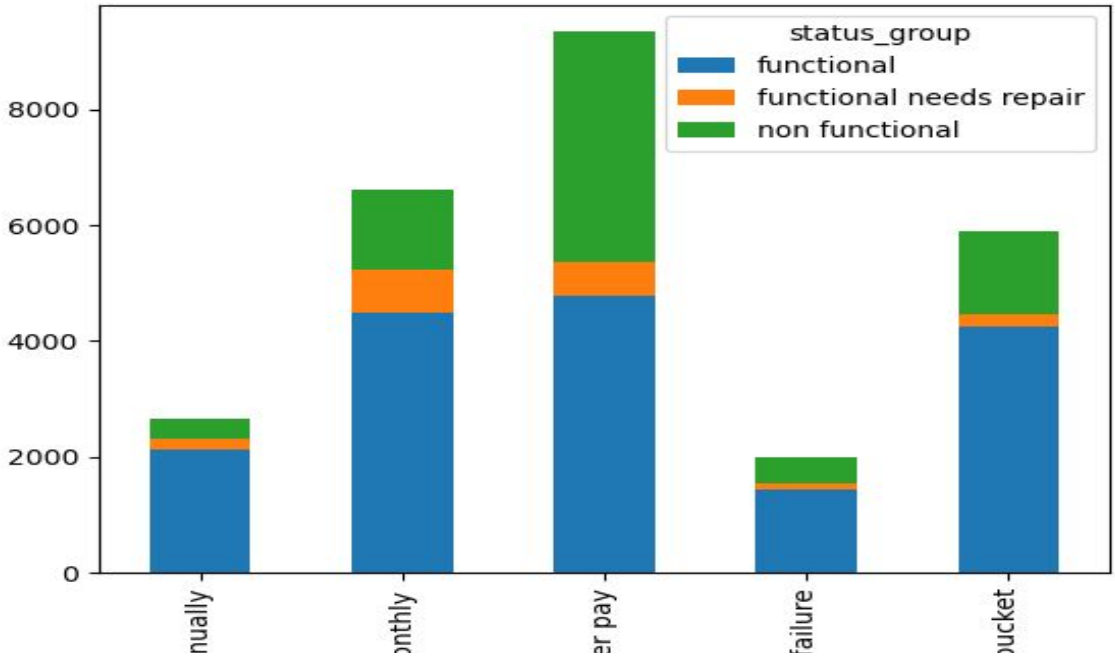
- **Population**
  - count 26569.000000
  - mean 250.571531
  - std 521.655559
  - min 0.000000
  - 25% 22.000000
  - 50% 130.000000
  - 75% 300.000000
  - max 30500.000000
  - Name: population, dtype: float64
- **Age**
  - count 14850.000000
  - mean 15.766309
  - std 9.894426
  - min 0.000000
  - 25% 9.000000
  - 50% 15.518762
  - 75% 17.000000
  - max 53.000000
  - Name: age, dtype: float64



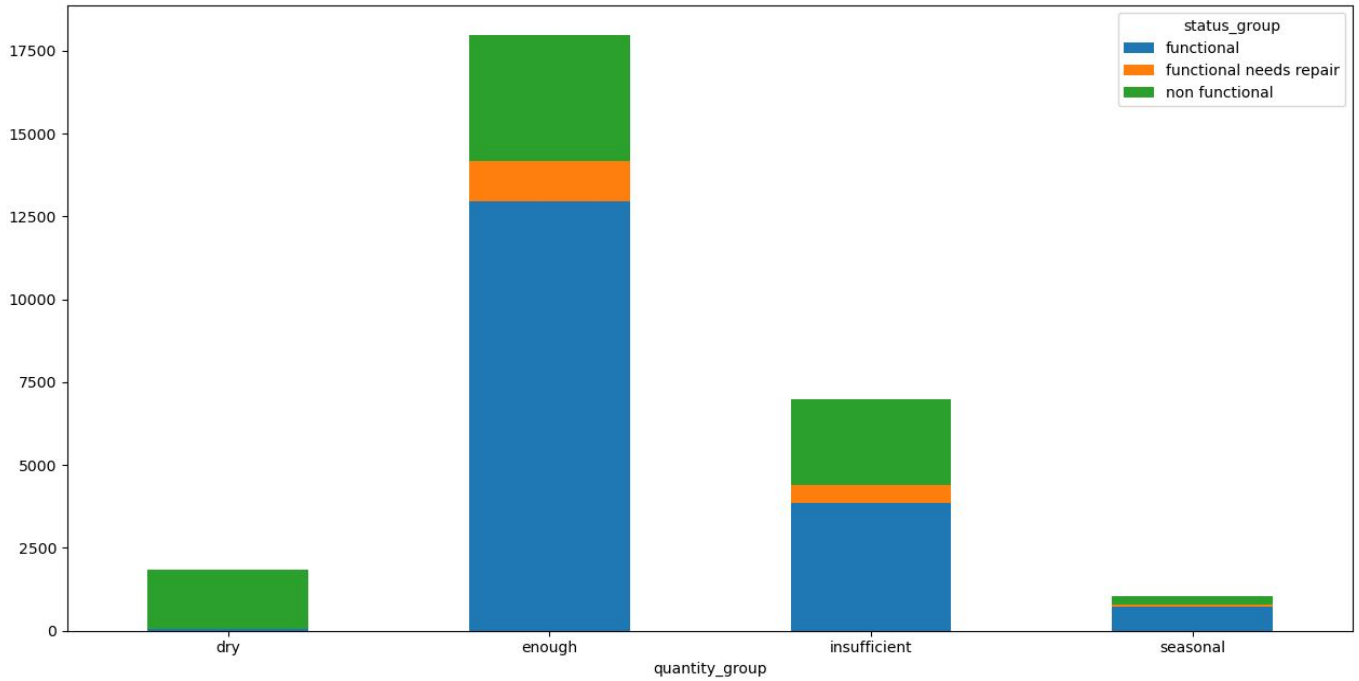
- **Categorical features** that include **'region', 'public_meeting', 'permit', 'quantity_group', 'management', 'payment_type', 'quality_group', 'source_type', 'waterpoint_type_group'** in our final data set can only take on a set of discrete values (**Categories**).
    - Region -



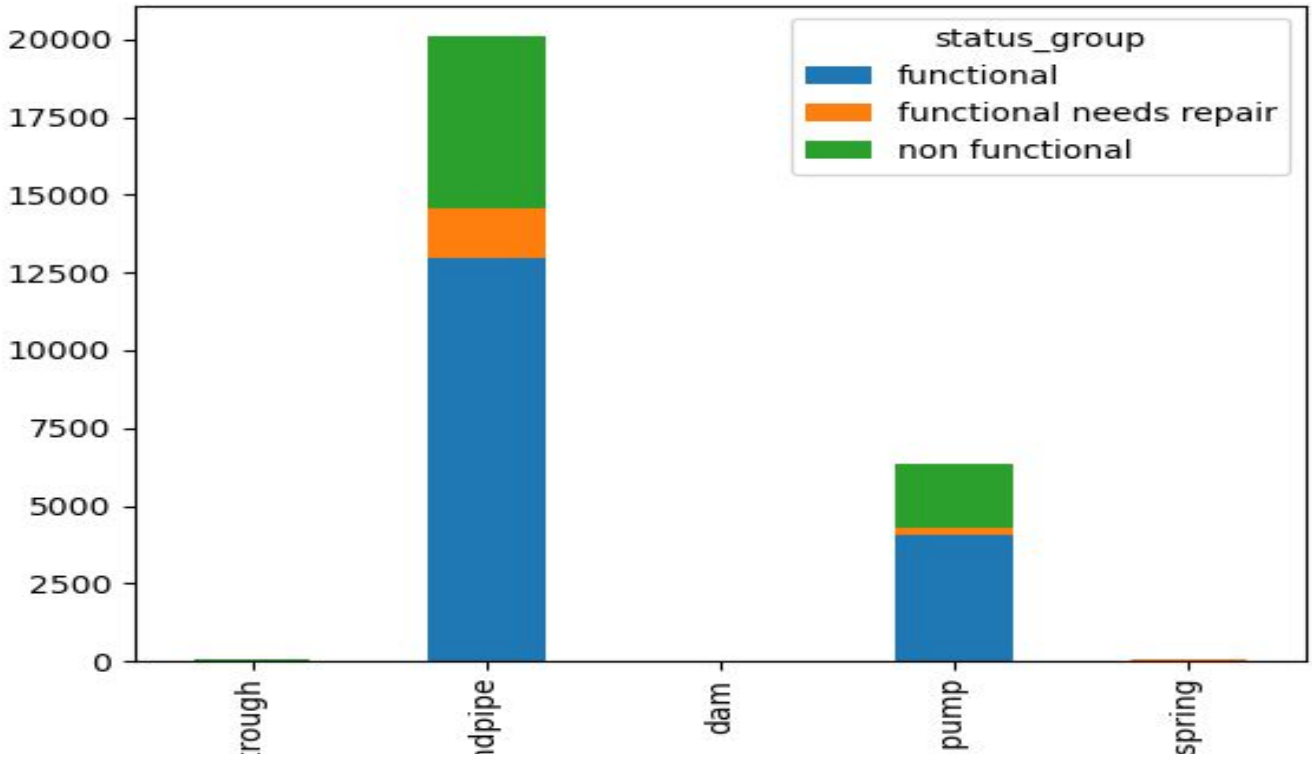
    - payment_type -



- quantity_group



- waterpoint_type_group



FEATURE ENGINEERING

INTRODUCTION

- A **Feature** refers to the input data to an algorithm in the form of Structured columns.
- **Feature Engineering** refers to the act of infusing Domain Knowledge to extract important features from raw data and transforming them into formats that are suitable for machine learning models.
- It's application achieves the following goals:
    - Preparing the proper input dataset so that it is compatible with the Machine Learning Algorithms
    - Improving the performance of the Machine Learning Models
- **Supervised Learning** requires the use of previously known input-output pairs to train the model and so it becomes essential that the dependence between the classification label and the set of independent labels is accurate and precise.
- Some of the essential Feature Engineering Techniques include:
    - Imputation
    - Dropping Entries
    - Dropping Labels
    - Extracting Dates
    - Handling Outliers
    - Grouping Operations
    - One Hot Encoding
    - Scaling

RELEVANT FEATURE ENGINEERING TECHNIQUES

1. Imputation

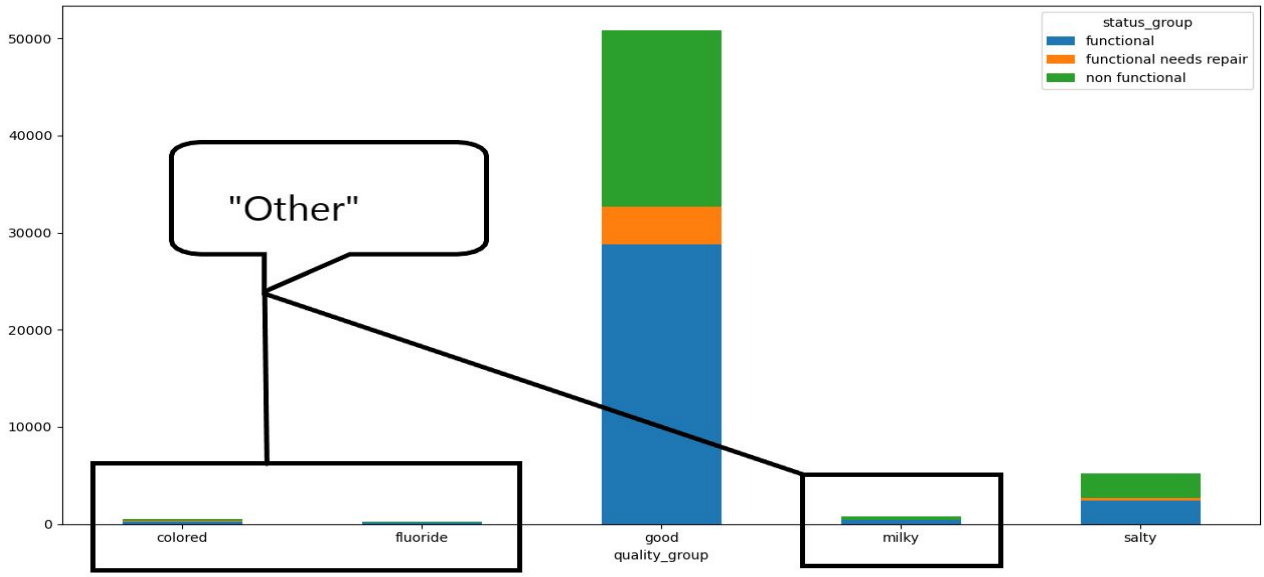
➢ Refers to the process of replacing Missing Entries with meaningful data (based on Domain, Label description and Label entries).

➢ Used when the number of Missing entries is large and dropping them could lead to a change in the overall distribution of the classification label.

➢ Some of the most common values used to replace missing data are:

- **Numerical Labels** - Mean, Median, 0, etc.
- **Categorical Labels** - Other, Maximum Occurred Value, etc.

➢ The **quality_group** label had 5 categories namely colored, fluoride, milky, salty and good. Out of these, we saw that there were negligible entries for categories colored, fluoride and milky but removing them changed the overall distribution of the classification label (status_group). So they were replaced by the imputed group “other”.



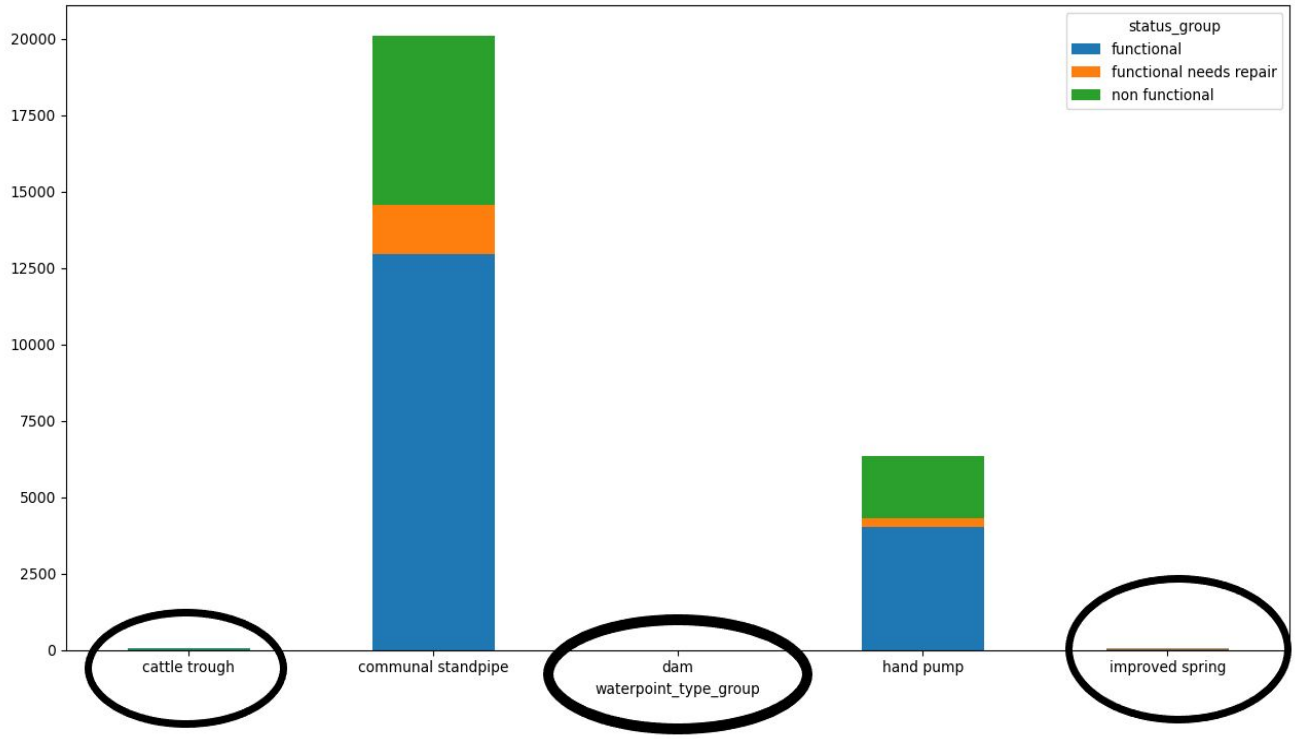
2. Dropping Entries

➢ Refers to the process of dropping the row entries for some label entries to eliminate a category value.

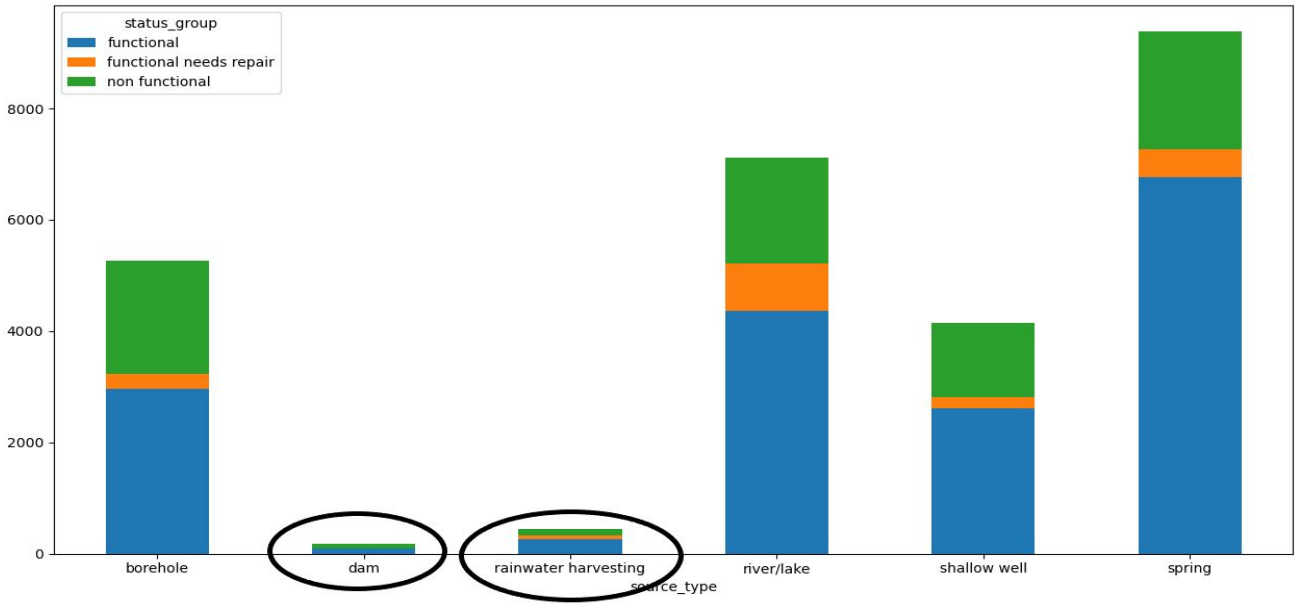
➢ Used when the number of missing Entries/data entries for a label value is negligible and its removal preserves the dataset size and classification label distribution.

➢ It is essential to check that the other labels (their relationship with the classification label) are unaffected by dropping those entries.

➢ The **Waterpoint_type_group** label has three out of its five categories with negligible data. The data entries pertaining to these three categories (cattle trough, dam, improved spring) were dropped resulting in just two categories (communal standpipe and hand pump).



➢ In **source_type** label, two out of six columns have negligible data when compared to its neighbouring category values. While ensuring the conditions for dropping entries pertaining to these two categories(dam and rainwater harvesting), the label was left with four categories (borehole, river/lake, shallow well, spring).



3. Dropping Labels

➢ On Analysis of Domain Information and their effect on the classification label, labels like public_meeting and management were found to be insignificant and hence dropped.

➢ The **public_meeting** label had two categories namely true and false, there was nearly no information about the label categories and it preserved the classification i.e. its presence didn't affect the functionality of the water pump.

➢ The **management** label had multiple categories with no possible grouping parameter and very little effect on the classification label. Going ahead with such large categories would have diluted our model, affecting its accuracy.

4. Extracting Dates

➢ Incompatibility with the model input format was the main reason that a label with date data type had to be converted to float while preserving its true meaning and essence.

➢ Thus the **construction_year** label was converted to a new label named **construction_age** (years) which depicted the relative age of the water pump (0 being the youngest and 30,500 being the oldest)

➢ The transformation applied was:

construction_age=max(construction_year)-construction_year

Eg. Max year - 2013

construction_year		construction_age
2013	→	0
2012		1
2000		13

5. Handling Outliers

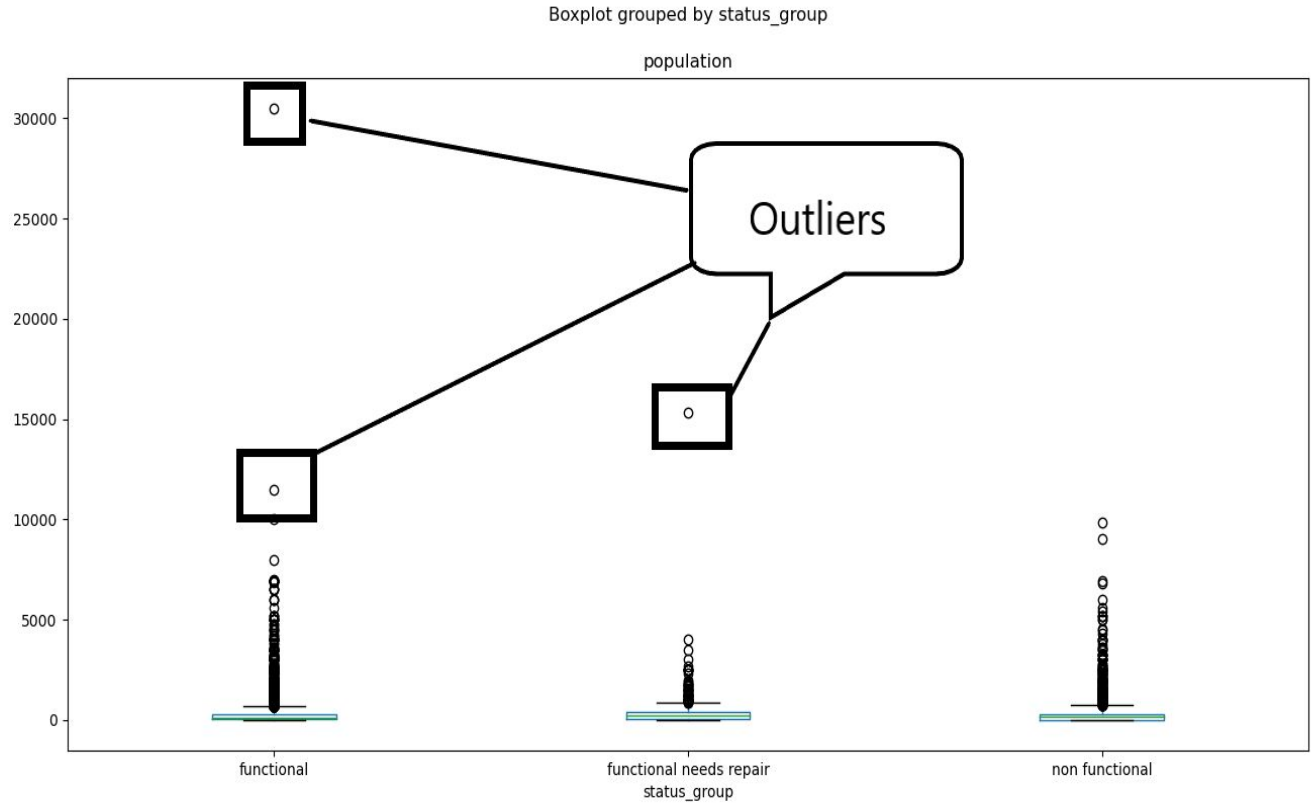
➢ **Outliers** refers to the data entities for a particular numerical label that affect the Measure of central tendency of the label.

➢ They can be identified by multiple measures like standard deviation, percentiles. The most frequently used is the percentile measure.

➢ They can be handled by either

- **Capping** the entry limit. (or)
- **Dropping** the outlier entries.

➢ The **population** label was visualized using box plots (based on percentile distribution). It was seen to have three distinct outlier entries (>10,000) . Since they were less in number, they were dropped.



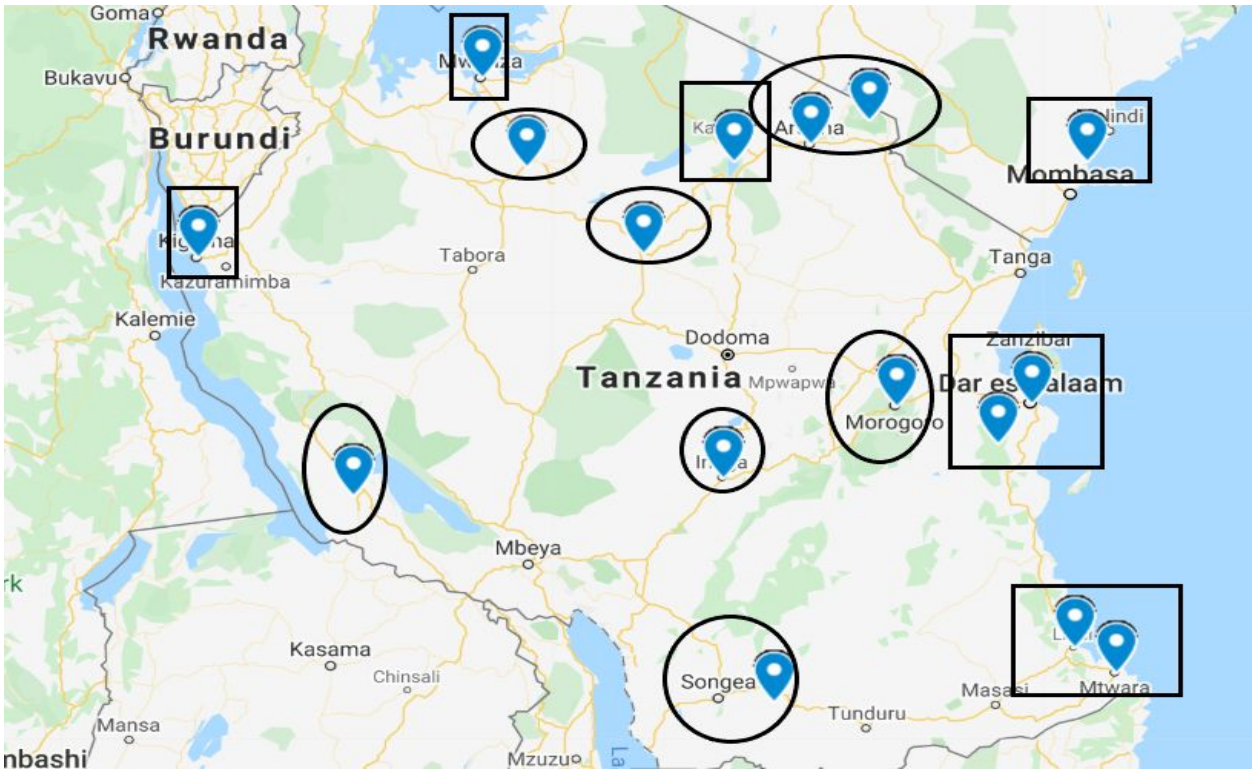
6. Grouping Operations

➢ Labels with multiple categories dilute the model parameters and hence should be grouped as much as possible to ensure minimum categorization while ensuring that the distribution of the classification label originally is still relevant after the change.

➢ The **region label had 21 categories** so some grouping was necessary. On investigating the regions using their (latitude,longitude) pairs, we saw another parameter to categorize the water pumps keeping the location aspect intact.

➢ A new label **closeness_to_waterbody** replaced the original region label. It denoted the closeness of the region to a natural waterbody (Ocean,lake,rivers,etc).

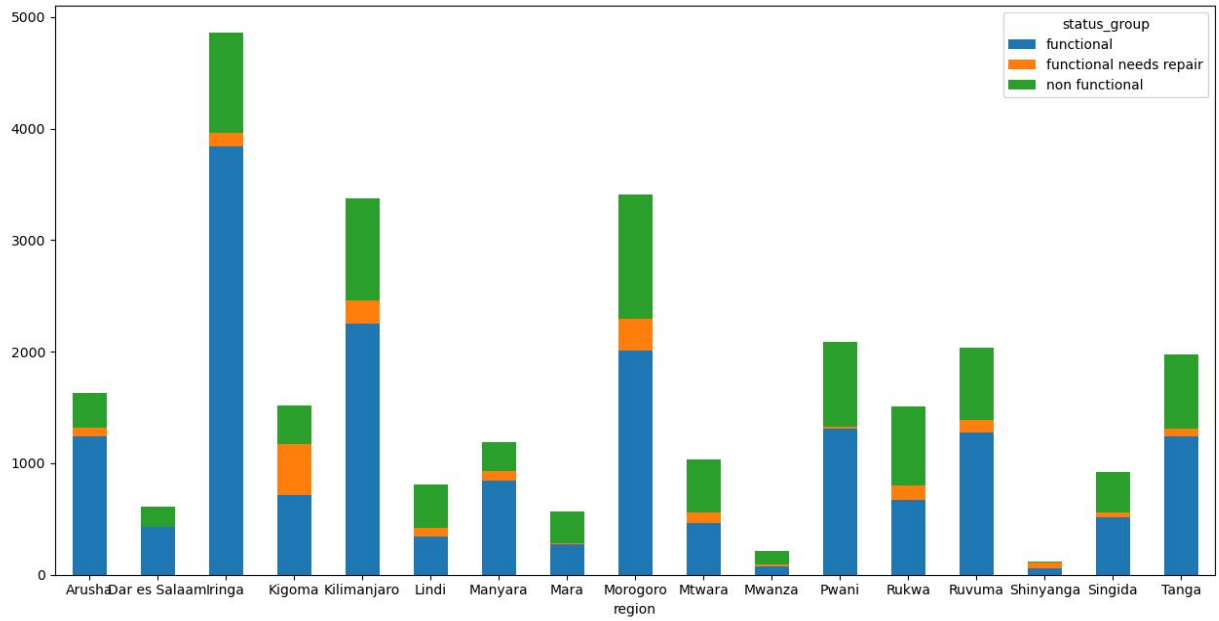
Circle - Far, Square - Near [closeness_to_waterbody]



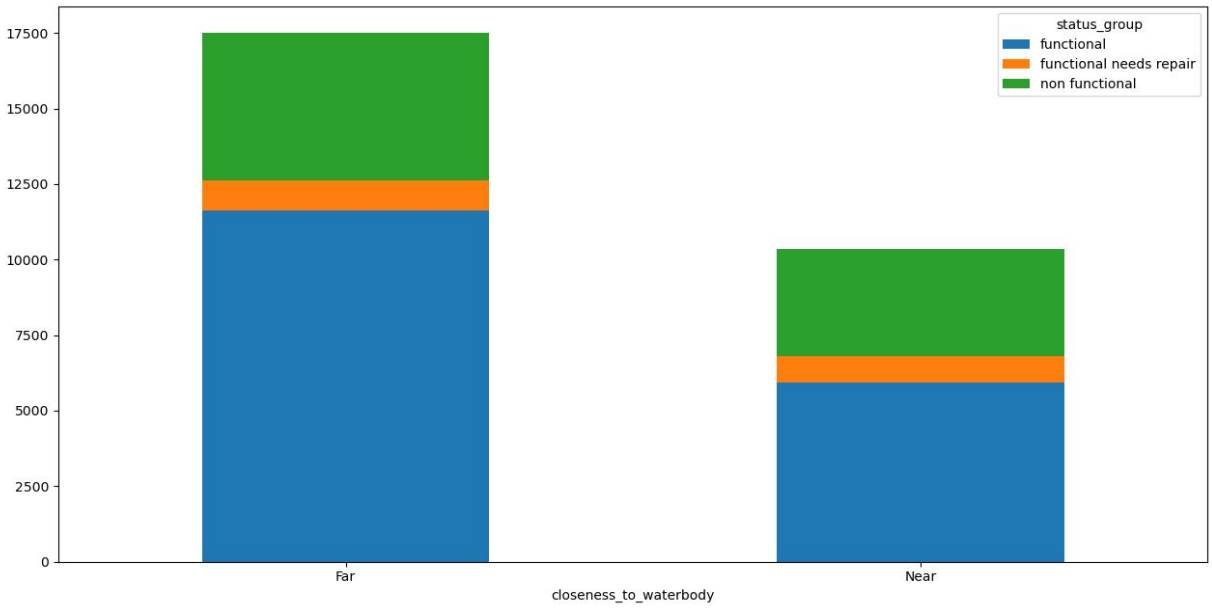
➢ It has just two categories near and far. It was mapped with regions:

- **Near** - Pwani, Dar es Salaam, Lindi, Tanga, Mtwara, Mwanza, Kigoma, Manyara, Singida.
- **Far** - Remaining 11 Regions (Kilimanjaro, Mara, Rukwa, Arush, Iringa, etc.)

FROM



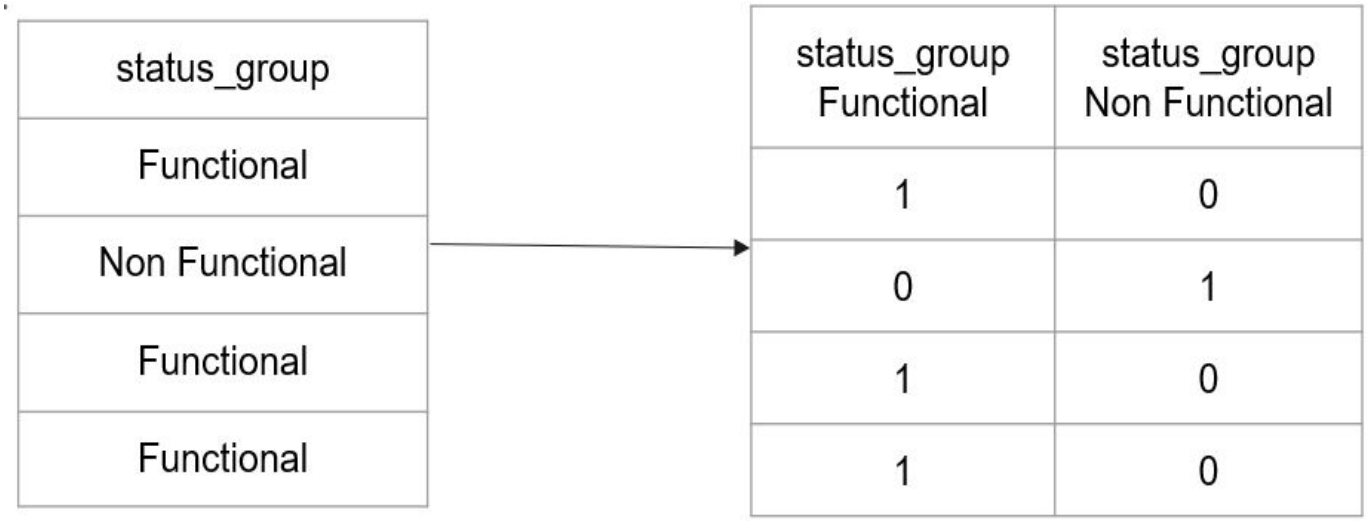
To



7. One-Hot Encoding

- ➢ To be compatible with the model, all Numerical labels had to be converted to “float” data type whereas all Categorical labels to “categorical” data type.
- ➢ But for comparisons to take place between categorical and numerical labels, the categorical labels had to be encoded to a numerical equivalent.
- ➢ **One-Hot Encoding** splits a categorical label into multiple labels, each representing a specific category of that label. So, for a particular entry, only the label which satisfies has 1 and all else have 0.
- ➢ This is the reason why **minimum categorization** had to be ensured as all these encoded labels along with the numerical labels, form the data inputs to the model.

Eg. One-Hot Encoding for a label with 2 Categories



status_group
Functional
Non Functional
Functional
Functional

status_group Functional	status_group Non Functional
1	0
0	1
1	0
1	0

DEALING WITH CLASS IMBALANCE

A great imbalance was present in the data set in the form of frequencies of the two categories (0: 19413, 1: 8432). To deal with this, the **SMOTE** technique was applied to generate data based on the existing data.

```
#SMOTE for balancing data
counter = Counter(train_result['is_non_functional'])
print(counter)
oversample = SMOTE()
x, y = oversample.fit_resample(train_set, train_result['is_non_functional'])
counter = Counter(y)
print(counter)
```

```
Counter({0: 19413, 1: 8432})
Counter({0: 19413, 1: 19413})
```

The SMOTE class present in the **imbalanced-learn library** was used for this purpose. It can be seen that the final data set has equal frequencies for both categories, hence eliminating any possible bias that could happen due to the imbalance.

DATA SET SPLIT

- The data set was split into two parts, the training set (X_train, Y_train) and the testing set (X_test, Y_test). The split was done in a 7:3 ratio. **train_test_split()** from the **scikit-learn library** was used to randomly distribute the data into the two sets.

```
# Split the samples into 0.7 for training and 0.3 for testing
# Using train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(
    x, y, test_size = 0.3)
# print(type(X_train))
# print(type(Y_train))
```

- The split was done so that the model can be **checked for overfitting** by **cross validating** the model with the testing set data.

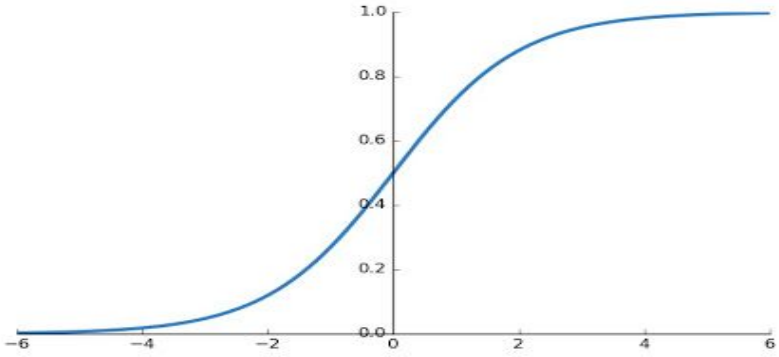
- (X_train, Y_train) was used to train the models while (X_test, Y_test) was used to check how well the models generalise to unseen data.

CLASSIFICATION MODELS

For preliminary classification, the output variable was transformed into either **0 (Functional or Functional but needs Maintenance)** or **1 (Non-Functional)**. This was done as the aim of the project was to **classify water pumps as functional vs non-functional.** Hence the functional but need maintenance pumps were grouped with functional ones.

LOGISTIC REGRESSION

Logistic Regression is very similar to Linear Regression in the aspect that it tries to find a linear relationship between the input features and the output variable. The real value obtained is mapped to the interval [0, 1] using the **sigmoid function**. The final value hence belongs to the range [0, 1] and is considered to be the probability that the given input belongs to category 1.



LogisticRegression() classifier from the scikit-learn library was used to perform the analysis.

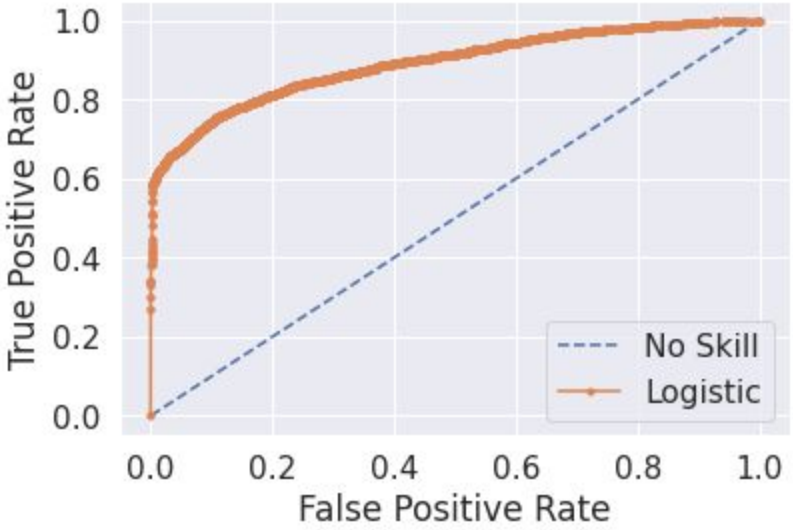
```
# Logistic Regression
lr = LogisticRegression(max_iter = 4000)
clf = lr.fit(X_train, Y_train)

#score the model on training and testing data
print(lr.score(X_train, Y_train))
print(lr.score(X_test, Y_test))

array = confusion_matrix(lr.predict(X_test), Y_test)
array = pd.DataFrame(array, range(2), range(2))
array = array/array.sum()*100
sns.set(font_scale=1.4) # for label size
sns.heatmap(array, annot=True, annot_kws={"size": 16}) # font size
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Logistic Regression - Testing Data Set')
plt.show()

array = confusion_matrix(lr.predict(X_train), Y_train)
array = pd.DataFrame(array, range(2), range(2))
array = array/array.sum()*100
sns.set(font_scale=1.4) # for label size
sns.heatmap(array, annot=True, annot_kws={"size": 16}) # font size
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Logistic Regression - Training Data Set')
plt.show()

ns_probs = [0 for _ in range(len(Y_test))]
Y_probs = lr.predict_proba(X_test)
Y_probs = Y_probs[:, 1]

ns_auc = roc_auc_score(Y_test, ns_probs)
print('No Skill: ROC AUC=%.3f' % (ns_auc))
lr_auc = roc_auc_score(Y_test, Y_probs)
print('Logistic: ROC AUC=%.3f' % (lr_auc))

ns_fpr, ns_tpr, _ = roc_curve(Y_test, ns_probs)
lr_fpr, lr_tpr, _ = roc_curve(Y_test, Y_probs)

plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
plt.plot(lr_fpr, lr_tpr, marker='.', label='Logistic')

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.show()
```

To measure the performance of the model, accuracy was calculated and the confusion matrix was generated.

The model had an **accuracy of 82.04%** when X_test was used to calculate the accuracy.

The model displayed an **accuracy of 82.16%** when X_train was used to calculate the accuracy. This shows us that the difference between the accuracies obtained on the training and testing sets is negligible. Hence

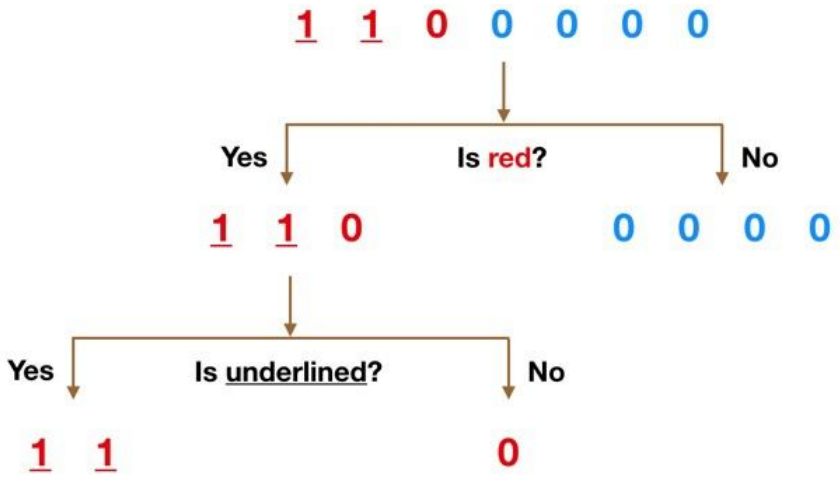
we can conclude that the Logistic Regression model does not overfit our data.



The **ROC Curve** also gives us an idea about the relationship between the True Positive Rate and the False Positive Rate of the model. It has an AUC of 0.891.

RANDOM FOREST

Random Forest is a classification technique that makes use of multiple decision trees to predict the output.



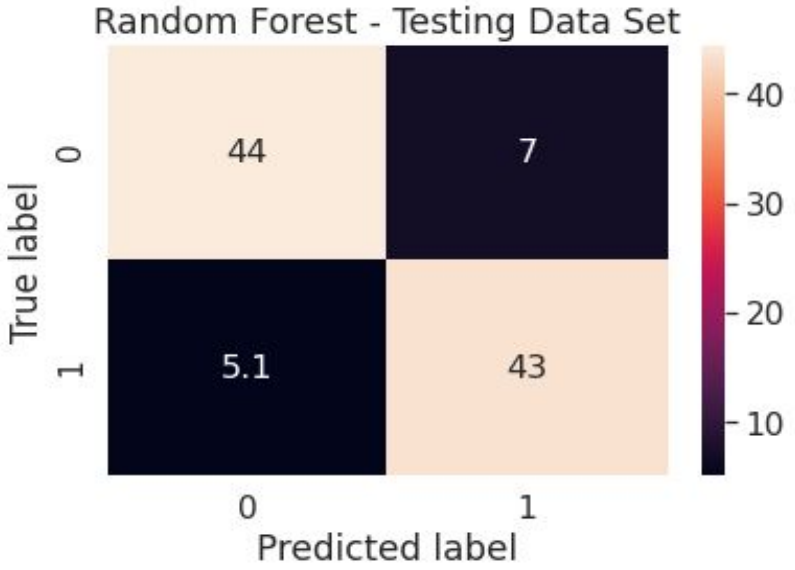
A **Decision Tree** is a kind of classification algorithm where a decision is made at every edge of the tree and the data is split based on that decision to its children nodes. The leaf nodes signify the category the data belongs to.

Random Forest makes use of **Ensemble learning** and uses multiple decision trees that are generated based on the data set (treated with random replacement). As a decision tree is very sensitive to changes in data, the trees generated are different from each other.

Random Forest was implemented using the RandomForestClassifier() classifier.

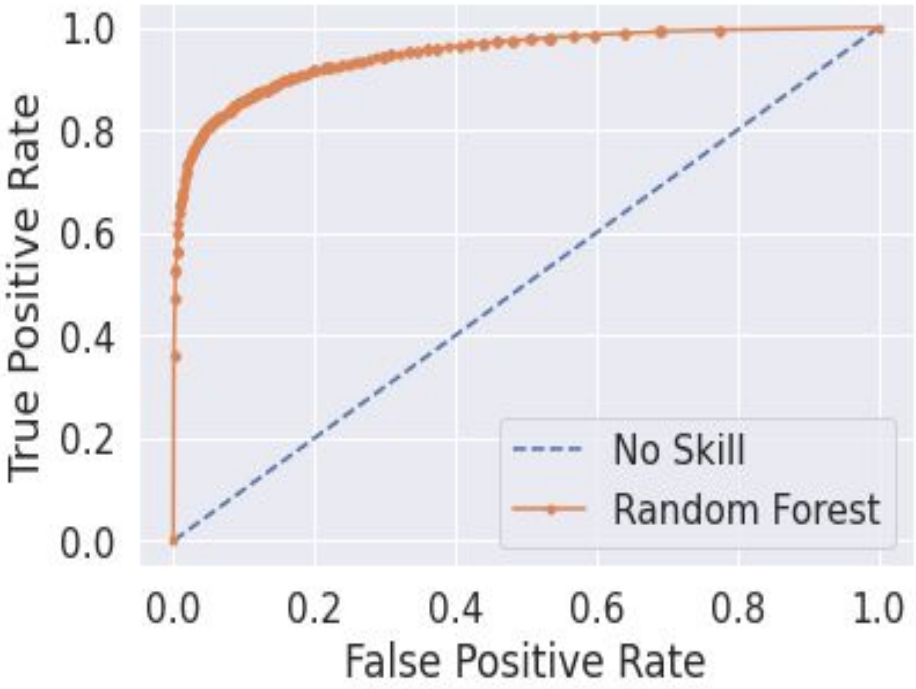
```
# Random Forest
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)

#score the model on training and testing data
# print(rf.score(X_train, Y_train))
# print(rf.score(X_test, Y_test))

#print confusion matrix
array = confusion_matrix(rf.predict(X_test), Y_test)
array = pd.DataFrame(array, range(2), range(2))
array = array/array.sum().sum()*100
sns.set(font_scale=1.4) # for label size
sns.heatmap(array, annot=True, annot_kws={"size": 16}) # font size
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Random Forest - Testing Data Set')
plt.show()

array = confusion_matrix(rf.predict(X_train), Y_train)
array = pd.DataFrame(array, range(2), range(2))
array = array/array.sum().sum()*100
sns.set(font_scale=1.4) # for label size
sns.heatmap(array, annot=True, annot_kws={"size": 16}) # font size
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Random Forest - Training Data Set')
plt.show()

ns_probs = [0 for _ in range(len(Y_test))]
Y_probs = rf.predict_proba(X_test)
Y_probs = Y_probs[:, 1]

ns_auc = roc_auc_score(Y_test, ns_probs)
print('No Skill: ROC AUC=%.3f' % (ns_auc))
rf_auc = roc_auc_score(Y_test, Y_probs)
print('Random Forest: ROC AUC=%.3f' % (rf_auc))

ns_fpr, ns_tpr, _ = roc_curve(Y_test, ns_probs)
rf_fpr, rf_tpr, _ = roc_curve(Y_test, Y_probs)

plt.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
plt.plot(rf_fpr, rf_tpr, marker='.', label='Random Forest')

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.show()
```

To measure the performance of the model, accuracy was calculated and the confusion matrix was generated.



An **accuracy of 87.85%** was obtained after feeding the model X_test as input and comparing the results with Y_test. At a glance, it seems that Random Forest overforms the Logistic Regression model.

On the training data set, Random Forest has a **staggering accuracy of over 99%**. This clearly shows that the **Random Forest model overfits** our data set.



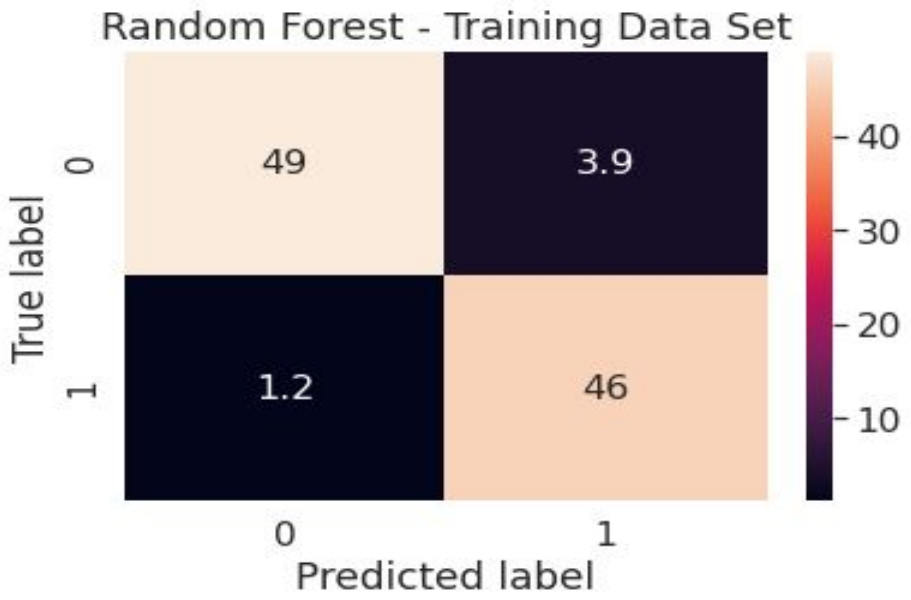
The **ROC Curve** also gives us an idea about the relationship between the True Positive Rate and the False Positive Rate of the model. It has an AUC of 0.948.

RANDOM FOREST - HYPERPARAMETER TWEAKING

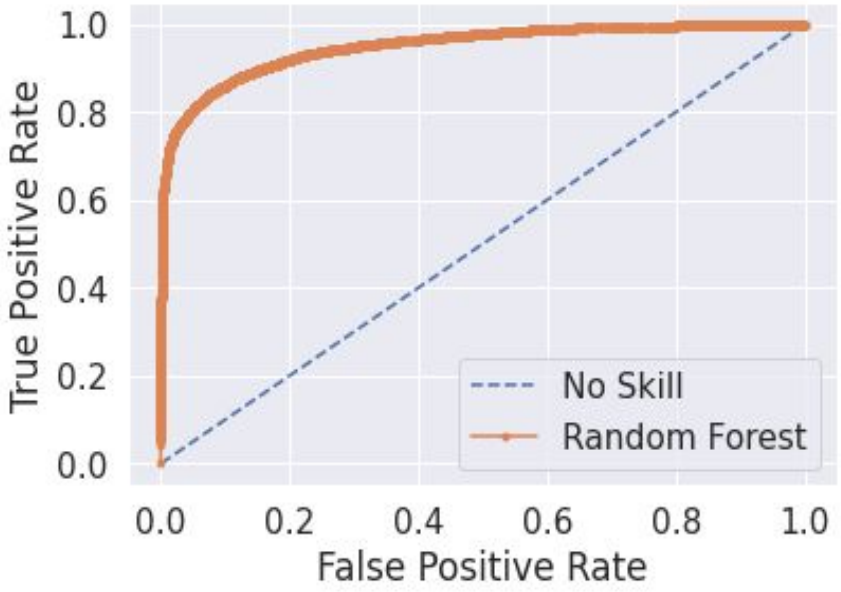
To deal with the problem of overfitting, we performed tweaking of the hyperparameters pertaining to the **RandomForestClassifier()** classifier.

Some changes made were:

- min_samples_leaf = 2 (default: 1)
- min_samples_split = 3 (default: 2)
- max_features = "log2" (default: “sqrt”)



The final a**ccuracy on testing data was found to be 88.22% (increased)** and the **final accuracy on training data was found to be 94.88% (decreased).** We can see that the **extent of overfitting has reduced**.



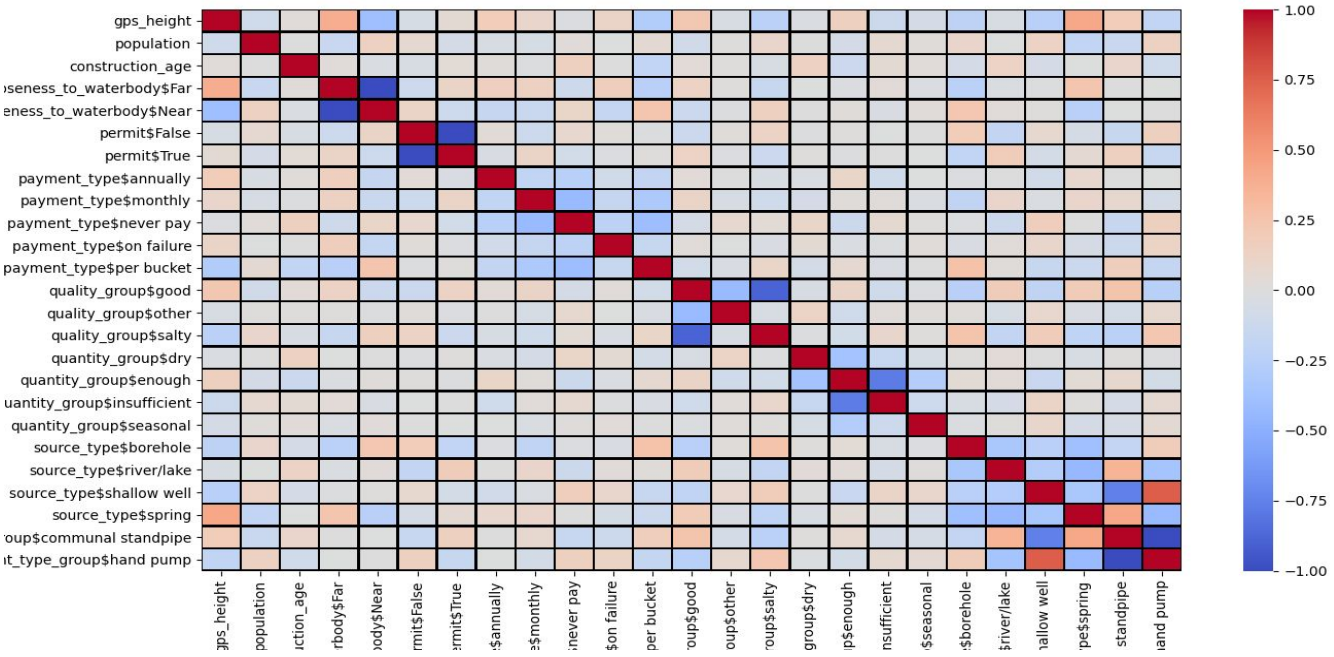
The new AUC for the ROC Curve was 0.951.

RESULTS

- ➢ It was finally decided that **Logistic Regression** is a more suitable model for our problem as our aim was not only to come up with a model to predict the category a pump belongs to but also analyse **how the various features of the data set affect the functionality of a Tanzanian Water Pump**.
- ➢ **Interpretation of the model results** and **Relationships between labels** can be easily understood using Logistic Regression by analysing the various statistical aspects such as the feature coefficients, the p-values of various features, the Odds Ratio, marginal effects, collinearity between the independent labels etc.

➢ Relationship between Independent Labels

- **<u>Collinearity Tests between Independent Labels</u>**
  - This is checked to ensure that no correlated labels are used as parameters to the model as they would decrease the accuracy of the model.
  - A **heatmap** (seaborn library) provides the best visual aid to check for correlations and the correlation method used is **Pearson Correlation**.
  - As can be seen from the heatmap below, there are negligible label categories that are found to be correlated and this number lies well within the threshold.
  - The diagonal elements show that a label is strongly correlated to itself.



➢ Model Summary

Logit Regression Results

Dep. Variable:	status_group	No. Observations:	27178
Model:	Logit	Df Residuals:	27159
Method:	MLE	Df Model:	18
Date:	Mon, 22 Jun 2020	Pseudo R-squ.:	0.3826
Time:	21:13:23	Log-Likelihood:	-11630.
converged:	True	LL-Null:	-18838.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	6.4951	0.188	34.504	0.000	6.126	6.864
closeness_to_waterbody[T.Near]	-0.0789	0.038	-2.077	0.038	-0.153	-0.004
permit[T.True]	-0.3293	0.036	-9.069	0.000	-0.400	-0.258
payment_type[T.monthly]	-0.7933	0.055	-14.362	0.000	-0.902	-0.685
payment_type[T.never pay]	0.1129	0.049	2.320	0.020	0.018	0.208
payment_type[T.on failure]	-1.2416	0.087	-14.262	0.000	-1.412	-1.071
payment_type[T.per bucket]	-1.1551	0.059	-19.428	0.000	-1.272	-1.039
quality_group[T.other]	-0.4295	0.141	-3.050	0.002	-0.706	-0.154
quality_group[T.salty]	-0.3220	0.064	-5.034	0.000	-0.447	-0.197
quantity_group[T.enough]	-5.5698	0.177	-31.431	0.000	-5.917	-5.223
quantity_group[T.insufficient]	-4.8629	0.179	-27.237	0.000	-5.213	-4.513
quantity_group[T.seasonal]	-5.9757	0.198	-30.139	0.000	-6.364	-5.587
source_type[T.river/lake]	-1.5667	0.054	-28.800	0.000	-1.673	-1.460
source_type[T.shallow well]	-0.5960	0.066	-9.077	0.000	-0.725	-0.467
source_type[T.spring]	-1.7646	0.055	-31.877	0.000	-1.873	-1.656
waterpoint_type_group[T.hand pump]	-0.8875	0.065	-13.707	0.000	-1.014	-0.761
gps_height	-0.0004	3.11e-05	-11.550	0.000	-0.000	-0.000
population	-0.0002	3.86e-05	-6.071	0.000	-0.000	-0.000
construction_age	0.0407	0.001	29.776	0.000	0.038	0.043

- **Coefficient of Variables (coef):**
    - Positive coefficients depict a positive relation between an independent label and the classification label while negative coefficients depict opposite behaviour.
    - The interpretation of the estimated coefficients for categorical predictors is relative to the reference level of the predictor.
    - The reference level for a numeric categorical predictor is the level with the lowest value or for a text categorical predictor, is the level that is first in alphabetical order
- **Standard Error (std err):**
    - It estimates the variability between the coef estimates if we repeat the process of collecting samples multiple times.
    - It helps in measuring the precision of coef. Smaller the error, more precise the coefficient estimate.
- **P-values (P>|Z|)**
    - It is the probability that the measure provides us with evidence against the null hypothesis.
    - Usually, the threshold is considered to be 0.05**.**
    - If greater than 0.05, the association is not statistically significant.
    - If less than or equal 0.05, the association is statistically significant.
- **95% Confidence Interval ([0.025 0.975])**
    - Refers to the range of values where the variable's value is likely to lie.
    - It provides a strongly confident estimate of the value.

- **Odds Ratio**
  - It uses the concept of odds of an event happening rather than true values and is hence considered a better estimate for variable interpretation.
  - For **Numerical data**, values greater than 1indicate increasing odds of the event happening as predictor value increases.
  - For **Categorical data**, values are in reference to category value and provide comparative response to the predictor value.

Intercept	661.891345
closeness_to_waterbody[T.Near]	0.924108
permit[T.True]	0.719461
payment_type[T.monthly]	0.452354
payment_type[T.never pay]	1.119546
payment_type[T.on failure]	0.288918
payment_type[T.per bucket]	0.315027
quality_group[T.other]	0.650805
quality_group[T.salty]	0.724663
quantity_group[T.enough]	0.003811
quantity_group[T.insufficient]	0.007728
quantity_group[T.seasonal]	0.002540
source_type[T.river/lake]	0.208739
source_type[T.shallow well]	0.550998
source_type[T.spring]	0.171250
waterpoint_type_group[T.hand pump]	0.411704
gps_height	0.999641
population	0.999766
construction_age	1.041554

➢ Model Results

- **Accuracy** -
    - 82.16% on Train Data and 82.04% on Test Data. The negligible difference indicates that there was no overfitting while training the model.
    - It gives the ratio of predicting True Positives and False Negatives in the sample.

- **Precision** -
    - It is a measure of result relevancy. High precision indicates balanced classes and low false positive rate.
- **Recall** -
    - It is a measure of whether the correct values were returned. High Recall indicates low false negative rate.
- **F1 Measure** -
    - Since Recall and Precision are important measures but there is a trade off between them. F1 Measure gives the relative strength based on recall and precision values.
    - It's a weighted average between Precision and Recall.Should be as high as possible.
- **Confusion Matrix** -
    - It Describes the performance of a classification model. It provides information about the four types of classifications True Positives, False Positives, True Negatives and False Negatives.
    - Can be used to calculate Precision, Recall and F1 Measure for a model.

- **<u>ROC Curve</u>** -
    - Information about the performance of the model when compared to a Random Model.

```
Confusion / Classification Matrix

Predicted     0     1    All
Actual
0          5133   762   5895
1          1598  4155   5753
All        6731  4917  11648

1- Non Functional
0- Functional (Functional + functional needs repair)

Functional% = Actual Functional Correctly Predicted/ Actual Functional = 87%
Non Functional% = Actual Non Functional Correctly Predicted/ Actual Non Functional =1407/3090 = 73%

================================================================================
Accuracy is 0.7973901098901099 ~ 80%

Classification Report
              precision    recall  f1-score   support

           0       0.76      0.87      0.81      5895
           1       0.85      0.72      0.78      5753

    accuracy                           0.80     11648
   macro avg       0.80      0.80      0.80     11648
weighted avg       0.80      0.80      0.80     11648


--------------------------------------------------------------------------------
```

INFERENCES

- For **Each Independent label** , we see the following attributes:
    - Category(s) with most Non Functional Character (using coefficients and odds ratio).
    - Observation and Justification of the Trend inferred based on Common Intuition and Domain Knowledge .
    - Conclusion about the Non Functional behavior of Water Pumps of a certain characteristics.

- **Interpreting Coefficients** from Model Summary:
    - It is defined with respect to a **Reference level** (usually first alphabetic category for Categorical Labels and 0 for Numerical Labels)
    - **Negative Coefficients** means that the event (Non Functional character) is more likely at the **Reference level**.
    - **Positive Coefficients** means that the event (Non Functional character) is more likely at the **Predictor level**.

- **Interpreting Odds Ratio**
    - It depicts the odds of an event happening. It is the ratio of the probability of an event happening to it not happening.
    - Also defined with respect to a **Reference level** (usually first alphabetic category for Categorical Labels and 0 for Numerical Labels)
    - **Less than 1** - Odds of the event happening at the **Reference level** is higher than that at the Predictor level.
    - **Greater than 1 -** Odds of the event happening at the Predictor level is higher than that at the Reference level.

Categorical Labels

1. Closeness_to_waterbody

- Non functional more likely when the pump is far away from a natural source of water.
- **As Expected** - Since water pump requires water input, closer it is to a waterbody, more functional it is
- **Conclusion** - More non functional Pumps further from Water Bodies.

2. Permit

- Non functional more likely when the Pump has no Permit(official allowance to do something).
- **As Expected** - Since a water pump requires regular maintenance and permits are required for official maintenance work. In order to cut costs, lack of permit usually leads to lack of maintenance of the Machinery.
- **Conclusion** - People should issue Permits to avoid Failure of Pumps.

3. Payment Type

- Non functional more likely when the payment frequency is low.
- **As Expected** - Paid pumps are more likely to be maintained than other pumps that are free to use or follow the 'payment on failure' model.
- **Conclusion -** Regular payments reduce the non functional character of a water pump.

4. Quality Group

- Non functional more likely when the water quality is good.
- **As Expected** - If the quality of water is good, then the pump will be used more regularly and hence it becomes prone to regular failure.

- **Conclusion** - More non functional Pumps are present where quality of water is good.

5. Quantity Group

- Non functional more likely when the water pump is dry.
- **As Expected** - Since a water pump's utility is to provide water, if it goes dry, its usage decreases and so it gets neglected. Next is (insufficient,enough, seasonal) but even their combined odds of failures(1.4%) is not remotely comparable to that for dry pumps(98.6%)..
- **Conclusion** - Most non functional Pumps lie in dry regions.

6. Source Type

- Non functional when the source is a borehole and to some extent for shallow wells as well.
- **As Expected** - Since a water pump requires water input, Drilling boreholes and maintaining them is costly and since groundwater is used, the process of extraction is slow, saline intrusion degrades the quality of water as well. Shallow wells have limited water input and so they are not a long term, reliable source of water.
- **Conclusion** - More non functional Pumps are found where sources for water are boreholes and shallow wells.

7. Waterpoint type group

- Non functional more likely for a communal standpipe.
- **As Expected** - A communal standpipe is subject to overuse, could be easily vandalized and mishandled due to disputes over sharing whereas handpumps are restricted to a smaller group and hence can be more easily maintained.

- **Conclusion** - More non functional Pumps when the waterpoint is communal.

Numerical Labels

1. GPS Height

- Non functional character increases with decrease in altitude of the well.
- **As Expected** - The deeper the water pump, more stable the supply of water and lesser is the chance of going dry. Deep Water Pumps are less prone to surface bacterial degradation as well.
- **Conclusion** - More non functional Pumps for shallow water pumps.

2. Construction Age

- Non functional character increases with age of the Pump.
- **As Expected** - As is applicable to all types of machineries, they become rusty and more susceptible to failure as they grow old.
- **Conclusion** - Older Water Pumps show a strong Non Functional Character.

CONCLUSION

- From all results obtained from reading various research papers and articles, discussions held to analyse the findings, plotting various relationship graphs between labels and the classification variable, we can conclude that using Supervised Learning techniques is proving to be an adequate approach for Predictive Maintenance to reduce costs incurred due to Maintenance of machinery.

- It can also be seen that cleaning and analysing the data set is as important as configuring the Machine Learning model itself.

- The data set may contain many **duplicated features.** It is very important to clean up the data set before feeding it to the model to prevent issues due to overfitting on the training set. **Overfitting** results when the **model fits the training set too well** and hence **fails to generalise** to points not present in the training set.

- Another issue that the data set may have is that it may contain many **missing values.** It is very important to appropriately deal with these missing values either by **removing the entries** that contain them or by **replacing those missing values** with some default value such as the **mean of the column** (in case of numerical features) or using some other special methods (in case of categorical features). Care must be taken while removing entries so as to **not distort the distribution of the output variable** by a large amount.

- When used directly, Logistic Regression can only be used to predict the YES/NO outcomes, however, in our case, we need to classify the machinery into three categories. Hence here we can apply the technique of **One vs All Classification** for the Multi-Class Classification problem.

- Choosing the statistical model is also very important. The most suitable model highly depends on the particular use case. While complex algorithms like Random Forest Classification might seem like they would work better than simpler ones like Logistic Regression, it might not be the case.

- **Feature Engineering** is a very useful technique to boost the performance of the category. While it may seem that individual features do not contribute significantly to the model, **new features** can be derived from these existing features which can then significantly contribute to the model. Dealing with **outlier values** is also important to make an efficient model.

- While at a glance it may seem that Random Forest Classifier outperforms Logistic Regression, we can see that the Random Forest model overfits the training data set for our model. Various methods such as **SMOTE technique for imbalanced data set** may be used to **reduce the extent of overfitting** so that the model can **generalise to unseen data**.

- Many statistical measures such as the **Odds Ratio, ROC Curves and the p-values** pertaining to the features are very useful in estimating how the feature affects the model. In case of Logistic

Regression, the **sign (+ve or -ve) or the coefficient** of the feature can be used to determine whether it **impacts the output** positively or negatively.

- A **low value of coefficient** can be equated to the feature having **minimal impact** on the output. A positive coefficient implies that the output tends to category 1 as the feature increases while a negative coefficient implies that the output tends to category 0 as the feature decreases.

REFERENCES

1. https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/
2. https://www.kaggle.com/danielmartinalarcon/tanzanian-water-pumps
3. https://joomik.github.io/waterpumps/
4. https://itnext.io/predicting-functional-water-pumps-in-tanzania-using-random-forests-and-logistic-regression-in-ffa04b0617f2
5. https://www.kaggle.com/c/ds1-predictive-modeling-challenge
6. https://towardsdatascience.com/predicting-the-functional-status-of-pumps-in-tanzania-355c9269d0c2
7. https://rstudio-pubs-static.s3.amazonaws.com/339668_006f4906390e41cea23b3b786cc0230a.html
8. http://courseprojects.souravsengupta.com/cds2016/tanzanias-water-problem-pump-it-up/
9. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.boxplot.html
   https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
   https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
   https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

10. https://www.statsmodels.org/stable/api.html
11. https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/how-to/binary-logistic-regression/
12. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
13. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
14. https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76
15. https://towardsdatascience.com/binary-logistic-regression-using-python-research-oriented-modelling-and-interpretation-49b025f1b510
16. https://towardsdatascience.com/better-heatmaps-and-correlation-matrix-plots-in-python-41445d0f2bec
17. https://www.researchgate.net/publication/335390126_Advantages_and_disadvantages_for_water_distribution_systems
18. Coursera Courses:
    a. Introduction to Data Science in Python
    b. Data Collection and Processing in Python.
    c. Neural Networks and Deep Learning

GLOSSARY

- **numpy** - a python library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **pandas** - a python library used for data manipulation and analysis
- **matplotlib** - a python library used to visualize data in the form of various types of graphs. Works well in conjunction with numpy and pandas.
- **seaborn** - a python library based on matplotlib. Provides a higher level interface for plotting statistical data.
- **scikit-learn** - a python library that has various basic functions and machine learning models in built which can be used directly
- **Jupyter Notebook** - an open source web browser based application that allows easy execution of Python code. Widely used in the Data Science domain.
- **LibreOffice** - an open source office suite that can be used to access CSV files. Alternatively, Excel was used -Windows 10
- **Kaggle** - an online community that regularly hosts Data Science competitions
- **DrivenData** - another online community that regularly hosts Data Science competitions
- **Model** - represents what was learned by an Machine Learning algorithm
- **Classification** - refers to the Machine Learning problem of classifying objects into various categories based on either known categories (Supervised Learning) or when the categories are not known beforehand (Unsupervised Learning)

- **Supervised Learning** - refers to problems where the “correct” output is known beforehand and we try to predict values based on those “correct” outputs
- **Unsupervised Learning** - refers to problems where the “correct” output is not known beforehand
- **Feature Engineering** - generation/modification of features using existing features.
- **imbalanced-learn** - a python library useful for dealing with imbalanced data sets.
- **Confusion Matrix** - matrix showing relation between predicted values and actual values
- **ROC Curve** - curve showing relationship between true positive rate and false positive rate.
- **p-values** - measure of usefulness of features.
- **SMOTE** - advanced data science technique to generate data for minority classes based on existing data such that the resulting data set has equal amounts of data in each category.
- **statsmodels** - a python library useful for finding the statistical features of the data set.
- **Odds Ratio** - statistical measure of the strength of correlation between two features.
- **Precision -** measure of result relevancy. Higher values indicate low false positive rates.
- **Recall -** measure of whether correct values were returned. Higher values indicate low false negative rates.
- **F1 Measure -** measure based on precision and recall values. Higher values indicate a more accurate model.
- **Pearson Correlation -** statistic measure of linear correlation between two variables.