DSCI 5240 Section 004

Data Mining and Machine Learning for Business (Spring 2025 1)

Final Report

Group 5

Date: 3rd May 2025

Prepared By:

Varsha Oleti

Zuhayir Mustafa

Shaik Raiyan Mohsin

Kyle Porter

1. Objectives

1. Find a prediction model that can predict which water pumps are functional and which are not
2. Run at least 3 different models
3. Select model with highest possible accuracy in the validation dataset.

2. Abstract

This report defines different modeling solutions to attempt and classify whether a water pump is functional or nonfunctional. The database used for this report is 'DSCI 5240 Project Data' as defined by our professor. We ran 5 models to try and achieve the highest accuracy. These models are: Naïve Bayes, Logistic Regression, Ensemble methods, a Neural Network, and K-Nearest Neighbor. We also ran a Naïve model for a baseline comparison. We achieved highest validation accuracy of 0.7242 using naïve bayes. Business implications of this project include reducing repair costs, identifying trends that lead to non-functionality of water pumps, Reducing downtime, and overall increase in planning capabilities for both short term and long term.

3. Data Pre-Processing

3.1. Data Description

The dataset consisted of 5000 rows and 12 columns. It focused on various aspects of water pump installations such as the location, details of construction and the status of operations. The key variables in the dataset were:

- gps_height: The height altitude in meters
- longitude, latitude: The geographic coordinates
- population: The number of people served
- amount_tsh: The amount of water that was used for pumping
- construction_year: When the pump was built
- Categorical variables which indicated the origin: installer, funder, scheme_name
- Categorical attributes such as extraction_type, source_type, water_quality, quantity
- region_code, district code: Location codes of the area
- status_group: This was the target variable which tells us the status of whether it is functional, non-functional or needs repar

After reviewing the dataset, we felt that there were many columns which were redundant because they had too many missing values or did not provide any influence on the model we were about to create. We reduced the dataset to have 4,518 rows and 14 columns which were relevant based on the EDA and importance of variables.

3.2. Cleaning the Data

1. We dropped rows where either latitude or longitude had missing values and verified that there were no more missing values
2. For water pump IDs, we reset the data frame and filled out missing Water Pump ID values by incrementing the previous valid ID
3. For categorical values ('Water Source Type', 'Water Quality', 'Funder', 'Payment Type', 'Pump Type', 'Functioning Status'), we calculated the distribution of all values except the missing ones and randomly filled the missing values based on the observed proportions
4. For the population served, we replaced missing values with the median
5. For distance to nearest town, no imputation was done, and the rows were dropped
6. Installation year missing values were randomly filled based on the observed distribution

Original dataset – (5000,12)
After cleaningg. – (4518, 14)

We also didn't set a random seed when we imputed missing values in categorical variables. So, every time we ran the dataset, it differed slightly and ultimately changed the results of confusion matrix. We have set random seed = 0

Final counts for categorical after setting seed
Final plots after setting seeds

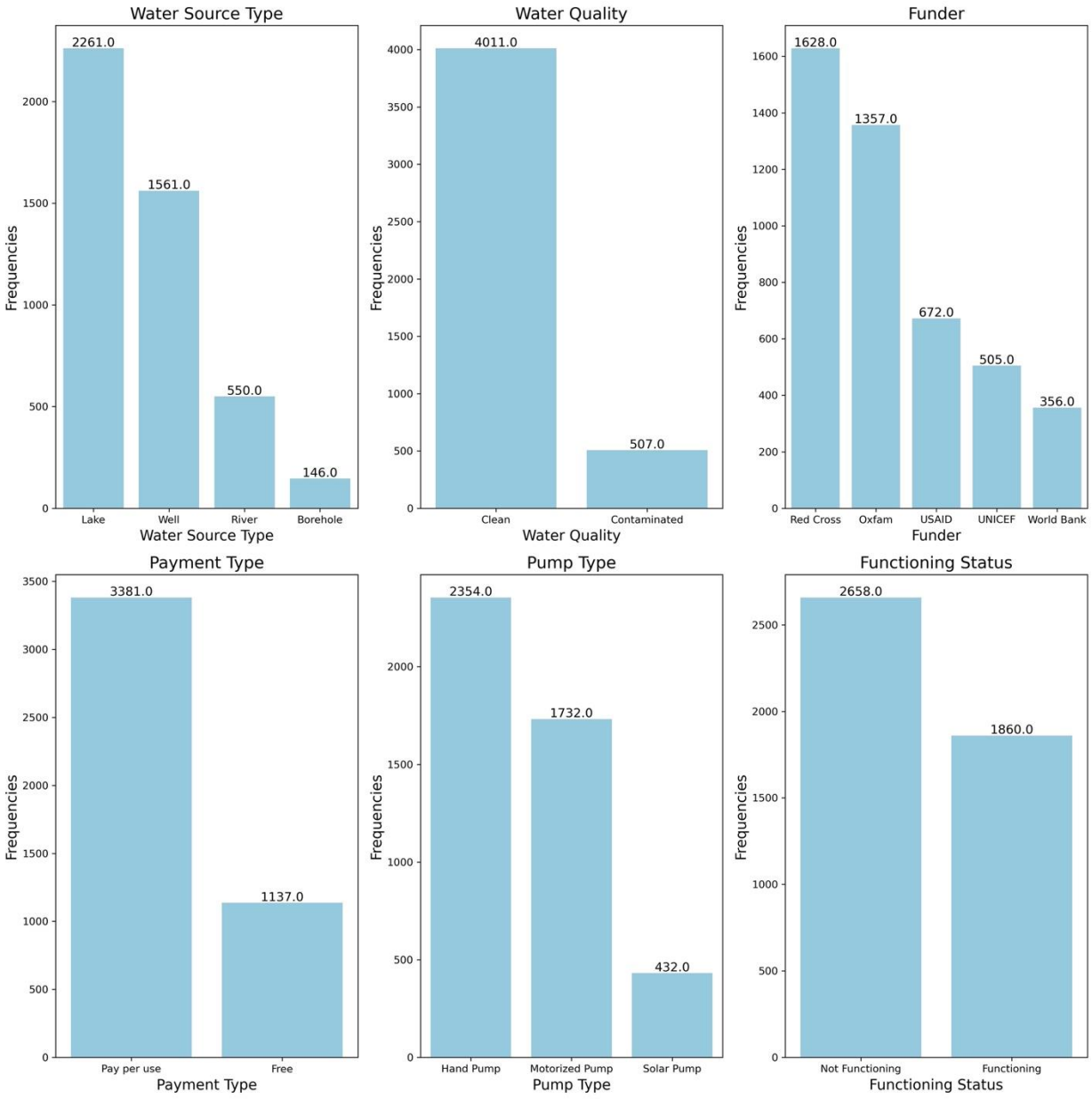

Figure 1: Final counts for categorical after setting seed

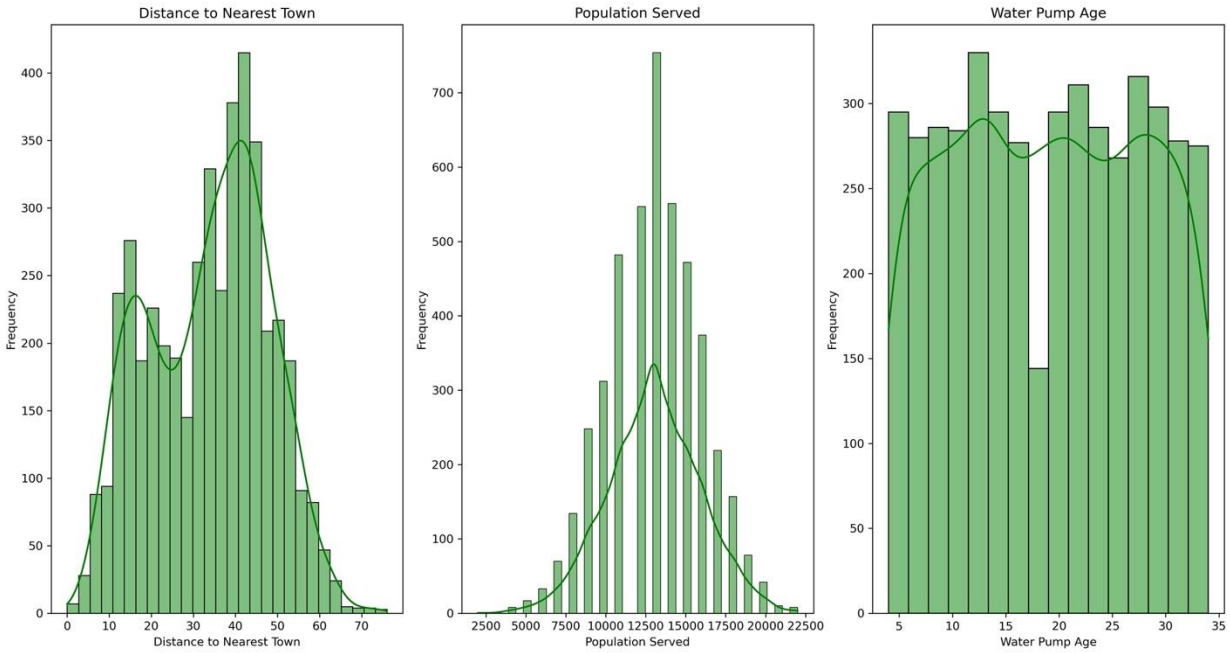

Figure 2: Final plots after setting seeds

4. Problem Identification

The aim is to predict whether a pump is functioning or not functioning, suggesting that the *target variable is 'Functioning Status'* and the class of interest is 'Non functioning'. Since the *results are binary* (functioning [1] and non-functioning [2]), this is a classification problem.

Selecting appropriate algorithms from known set –

Sr.No	Algorithm	Type of Learning	Data Scaling	Remarks
1	K Nearest Neighbors	Lazy	Normalization	Uses proximity (distance) to other entities to assign majority class
2	Logistic Regression	Linear	Standardization	Most appropriate algorithm for this dataset as this algorithm is ideal for binary classification. We can also find the most important features to reduce number of variables used for neural networks.
3	Random Forest Decision Trees	Ensemble	Standardization	Can handle both numerical and categorical data.
4	Neural Networks	Deep learning	Standardization	Can capture complex relationships
5	Support Vector Machines	Linear/Non-Linear	Categorization	Requires all attributes to be categorical data. Requires Extensive Data-preprocessing.
6	Naïve Bayes	Probability	Categorization	Numerical Variables must be binned into categories, like in SVM.

Table 1: Selecting appropriate algorithms from known set

All the algorithms listed above are supervised training models for classification.

5. Performance Metrics to Evaluate Performance

5.1. Distribution of 'Functioning Status' –

Functioning Status	Frequency	Proportion
Not functioning	2658	0.5883
Functioning	1860	0.4116
Total	4518	0.9999

Table 2: Distribution of 'Functioning Status'

The distribution of the 'Functioning Status' is skewed towards 'Not Functioning' as proportion is close to 60%. Therefore, this is a somewhat balanced dataset.

5.2 Confusion Matrix –

Actual↓/Predicted →	Non functioning	Functioning
Non-Functioning	True Positive	False Negative
Functioning	False Positive	True Negative

Table 3: Confusion Matrix

For such a dataset, **Accuracy is a good measure to evaluate the model performance alone. However, since it is somewhat balanced (60-40%)** we will also evaluate and compare performances on the model based on Specificity, Precision, Recall, F1Score and AUC-ROC Score. Their formulas and definitions are as follows -

1. **Accuracy –**
   Tells us the percentage of accurate predictions by the model. Formula is **-**
   (True Positive + True Negative) / Total

2. **Specificity –**
   It tells us the percentage of accurate 'Functioning' predictions by the model. Formula is -
   (True Negative)/(False Positive + True Negative)

3. **Precision -**
   A high precision implies that the percentage of non-functioning pumps identified by the model is high. Formula is -
   (True Positive)/(True Positive + False Positive)

4. **Recall –**
   If precision is high, recall is low. It tells us the percentage of actual non-functioning pumps. Formula is -
   (True Positive)/(True Positive + False Negative)

5. **F1-Score -**
   It helps us to understand the trade-off between precision and recall. Formula is -
   2* [(Precision*Recall)/(Precision + Recall)]
6. **AUC-Roc Score –**
   Stands for area under the ROC curve. It quantifies the model's ability to distinguish between the classes. A higher AUC value (closer to 1) indicates a better performing model. The ROC curve plots:
   a. True Positive Rate (TPR), also known as Recall, on the Y-axis.
   b. False Positive Rate (FPR) calculated as 1−Specificity1−Specificity, on the X-axis.

This model is useful to allocate maintenance resources for non-functioning pipes. Looking at the business implications, **False negatives** (when a pump that is actually **Not Functioning** is predicted as **Functioning**) would be more problematic here, as this means a pump might go unrepaired, leading to more water supply issues or service interruptions. Therefore, we will aim to also maximize recall.

Therefore, while our objective is to **identify a model that maximizes accuracy, we will also aim to optimize AUC-ROC score and recall.**

6. Data Splitting into Training and Validation Set

We are using sklearn's 'train_test_split' function to split the entire dataset into training set and validation set. The split 70-30, that is, 70% of the data will be used to build the model (training set) and 30% of data will be used for evaluating performance. All metrics are calculated using the validation set.

7. Benchmark Naïve Model

First, we will set a benchmark model that **guesses that all pumps are non-functioning in the validation set.** This helps us understand if the other models-built work better than the naïve guess. We will capture the results of the validation dataset in the following table.

7.1 Data Scaling.

We must convert the y_train and y_test into binary variables of 1 and 0 to run the model and generate the confusion matrix. For the following algorithms -when we dummy encode the Functioning Status column, we will drop the first label, such that it will result in a variable called 'Functioning Status_Not Functioning' where Non Functioning will become 1 and Functioning will become 0. To follow the same convention, we will map the same numbers here.

7.2 Confusion Matrix for Naive

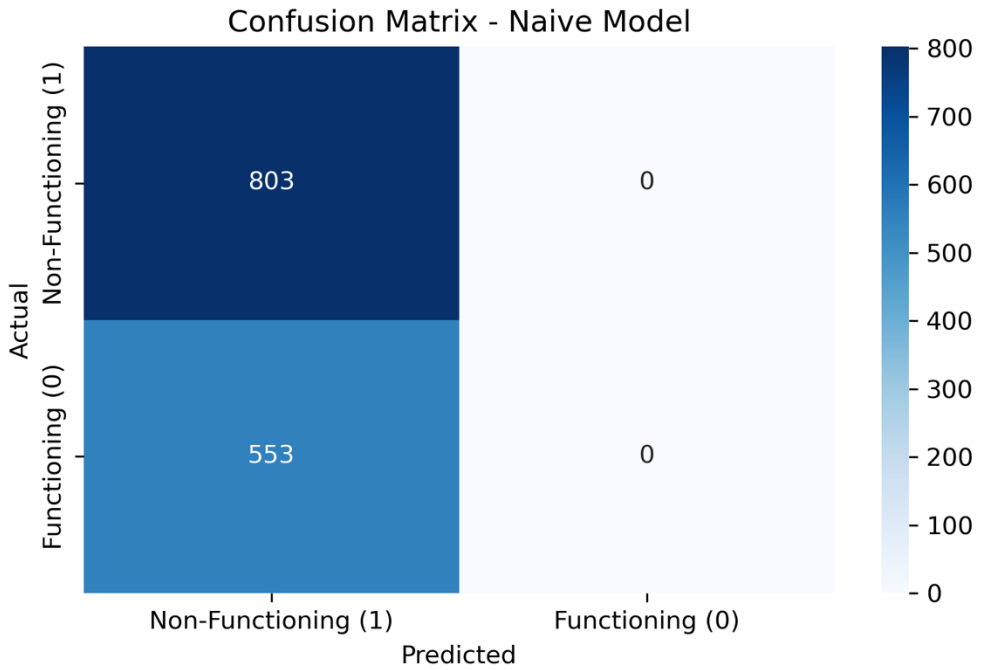

Figure 3: Confusion Matrix for Naive Model

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC

Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000

Table 4: Naive model confusion matrix

As we can clearly see from the above example, **by guessing all as non-functioning, we are able to achieve close to 60% accuracy**, *thereby confirming our earlier fact that accuracy is not a good metric for this dataset*. The *specificity for this model is 0*, **meaning that it fails to correctly identify functioning pumps**, which is not ideal. *The ROC score of 0.5 suggests that there is great room for improvement*.

8. K Nearest Neighbours

It is a non-parametric method that classifies or predicts a new record based on similar records in the training data. Similarity is calculated through distance formulas.

8.1. Data Scaling.

This algorithm uses distance calculations and identifies k neighbors based on entities that are close to it. It calculates the majority class and assigns the predicted value. Since it uses distance-based calculation, it is very sensitive to variables that have greater magnitude, and it is important to scale the data.

Normalization is often preferred for KNN because it ensures all features contribute equally to the distance calculation by transforming them into the same range (usually [0,1]). On the other hand, standardization (which scales data to have a mean of 0 and a standard deviation of 1) does not limit the range of values. As a result, even after standardization, a feature with larger values might still have more influence on the distance than other features.

8.2 Identifying Appropriate K

We must be careful while selecting the appropriate k values as –

1. Low values of K can capture noise (over-fitting)
2. Very high values of k can provide more smoothing (underfitting)

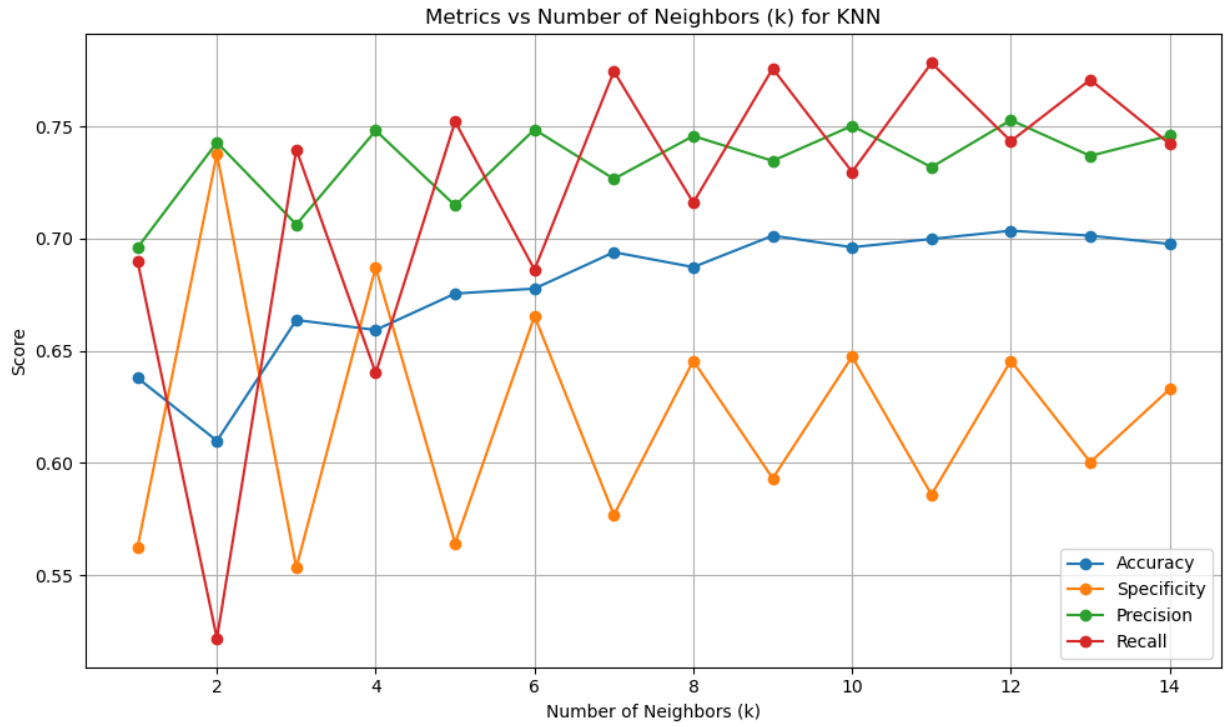

Figure 4: Metrics vs Number of Neighbors for KNN

8.3. Observations from the plot –

1. Specificity is very volatile between k = 1-6. Therefore, we can discard them
2. Accuracy is highest at k=12
3. Difference between precision and recall is lowest at k = 12

k	Accuracy	Specificity	Precision	Recall	F1 Score	AUC-ROC
1	0.6379	0.5624	0.6960	0.6899	0.6929	0.6262
2	0.6099	**0.7378**	0.7429	0.5218	0.6130	0.6668
3	0.6637	0.5533	0.7063	0.7397	0.7226	0.6863
4	0.6593	0.6872	0.7482	0.6401	0.6899	0.7028
5	0.6755	0.5642	0.7148	0.7522	0.7330	0.7160
6	0.6777	0.6655	0.7486	0.6862	0.7160	0.7261
7	0.6940	0.5769	0.7266	0.7746	0.7498	0.7350
8	0.6873	0.6456	0.7458	0.7161	0.7306	0.7404
9	0.7013	0.5931	0.7347	0.7758	**0.7547**	0.7448
10	0.6962	0.6474	0.7503	0.7298	0.7399	**0.7469**
11	0.6999	0.5859	0.7319	**0.7783**	0.7544	0.7439
12	**0.7035**	0.6456	**0.7528**	0.7435	0.7481	0.7460
13	0.7013	0.6004	0.7369	0.7709	0.7535	0.7460
14	0.6976	0.6329	0.7459	0.7422	0.7441	0.7452

Table 5: K Confusion Matrix specifications

Our aim is to maximize accuracy, while aiming to achieve optimum Recall and AUC-ROC scores. In lieu of this, k = 12 seems to be the ideal number of neighbor selection.

1. It has the highest accuracy of 70.35% - suggesting that 70% are correctly identified under non-functioning and functioning.
2. It has a recall value of 74.35%, slightly lower than the highest achieved at k =11 of 77%, therefore this number is satisfactory. It suggests that from all Non-functioning pumps, 74% are identified correctly. (optimizes maintenance from all other options)
3. It has the highest precision value at 75.28%. It suggests that 75.28% of all pumps identified by the model as non-functioning are non-functioning.
4. The F1 Score tells us about the balance between precision and recall. For k =12 it is 74.81% which is slightly less than the highest achived (75.47%). Therefore, this is also satisfactory.
5. It has the second highest AUC-ROC score of 74.60%

a. Confusion Matrix for k =12

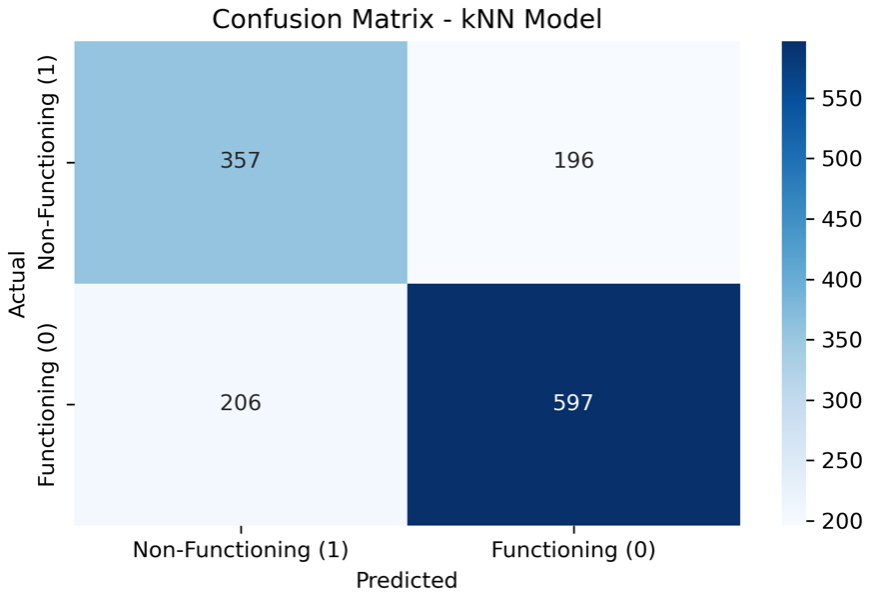

Figure 5: Confusion Matrix for k=12

8.4. Building on previous table

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000
kNN	0.7035	0.6456	0.7528	0.7435	0.7481	0.7460

Table 6: Naive vs kNN

The kNN , while being a simple algorithm, is already performing better than our naïve benchmark model. **It shows an improvement in Accuracy of 18.79%** from 59% in naïve to 70% in kNN. Even the AUC-ROC score has improved from 50% to 74.6%

9. Logistic Regression

Considered to be the most appropriate predictive model for this dataset because it requires a binary output, can handle both categorical and numerical attributes to predict the outcome and it is simple to understand and run.

9.1. Data Scaling

MinMax Scaler was used to normalize the dataset used for kNN model. The same dataset can be used to run the logistic regression model. Therefore, we will use the same training and test set used for previous algorithm.

9.2. Checking for Multicollinearity

Dummy encoding already removes the possibility of multicollinearity problem. Therefore, VIF will be calculated only on numerical variables of the training dataset.

Features	VIF
Distance to Nearest Town	6.325584

Population Served	**9.668008**
Installation Year	6.379985
Water Pump Age	6.316716
Latitude	5.128617
Longitude	6.279949

Table 7: Checking multicollinearity

Any Feature with VIF >= 10 must be removed. Population Served has a high multicollinearity of 10. Therefore, we will remove that from our training and test dataset.

9.3. Results From our model

```
Optimization terminated successfully.
         Current function value: 0.545269
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:     Functioning Status   No. Observations:                 3162
Model:                          Logit   Df Residuals:                     3145
Method:                           MLE   Df Model:                           16
Date:                Tue, 29 Apr 2025   Pseudo R-squ.:                 0.1958
Time:                        15:00:53   Log-Likelihood:                -1724.1
converged:                       True   LL-Null:                       -2144.0
Covariance Type:            nonrobust   LLR p-value:                2.102e-168
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                          3.2779      0.514      6.375      0.000       2.270       4.286
Water Source Type_Lake        -1.2133      0.360     -3.367      0.001      -1.920      -0.507
Water Source Type_River       -2.3678      0.372     -6.359      0.000      -3.098      -1.638
Water Source Type_Well        -3.3193      0.363     -9.150      0.000      -4.030      -2.608
Water Quality_Contaminated     0.9692      0.150      6.448      0.000       0.675       1.264
Funder_Red Cross               0.2479      0.101      2.449      0.014       0.049       0.446
Funder_UNICEF                  0.7731      0.152      5.094      0.000       0.476       1.071
Funder_USAID                   0.7490      0.135      5.560      0.000       0.485       1.013
Funder_World Bank              0.9247      0.171      5.411      0.000       0.590       1.260
Payment Type_Pay per use      -0.4237      0.097     -4.348      0.000      -0.615      -0.233
Pump Type_Motorized Pump       0.1017      0.089      1.138      0.255      -0.073       0.277
Pump Type_Solar Pump          -0.7730      0.145     -5.348      0.000      -1.056      -0.490
Distance to Nearest Town      -1.7629      0.223     -7.903      0.000      -2.200      -1.326
Installation Year             -0.1762      0.333     -0.529      0.597      -0.829       0.476
Water Pump Age                 0.0854      0.335      0.255      0.799      -0.571       0.742
Latitude                      -0.0163      0.172     -0.094      0.925      -0.354       0.322
Longitude                     -0.0732      0.178     -0.411      0.681      -0.422       0.276
==============================================================================================
```

Figure 6: Results from model

9.4. Observations –

1. Pump Type_Motorized Pump is notsignificant but Pump Type_Solar Pump is significant
2. Installation Year, Water Pump Age, Latitude and Longitude are not significant variables

9.5. Explanation Of Results –

We have used the same normalized dataset as in knn. For kNN, we mapped the y_train and y_test with the following code –

y_train_knn = y_train.map({'Functioning': 0, 'Not Functioning': 1})
y_test_knn = y_test.map({'Functioning': 0, 'Not Functioning': 1})

and then converted to integer to allow the algorithm to run successfully using -

y_train_logi = y_train_knn.astype(int)
y_test_logi = y_test_knn.astype(int)

Therefore, to maintain consistency 1 is Non -Functional and 0 is Functional. In lieu of this, the explanation of results are as follows –

1. Water Source Type

Borehole	Lake	River	Well
Reference category for dummy encoding. **More likely to be non-functional**	Since the coefficient is negative, it **has a lower likelihood of the pump being non functional**	Since the coefficient is negative, it **has a lower likelihood of the pump being non functional**	Since the coefficient is negative, it **has a lower likelihood of the pump being non functional**

Table 8: Water Source Type

2. Water Quality

Clean	Contaminated
Reference category for dummy encoding. **Lower likelihood of being non-functional**	Positive coefficient suggests that there is a **higher likelihood of pump being non functiona**l. This is expected behavior

Table 9: Water Quality

3. Funder

Oxfam	Red Cross	UNICEF	USAID	World Bank
Reference category for dummy encoding. Cannot say anything specific for this variable.	likely to be non functional due to positive coefficient	High likelihood of pump being non functional	High likelihood of pump being non functional	Very high likelihood of the pump being non functional

Table 10: Funder

4. Payment Type

Free	Pay Per Use
Reference category for dummy encoding. **Free use pumps are more likely to be non functional**	**Less likely to be non functional.** This is expected behavior.

Table 11: Payment type

5. Pump Type

Hand Pump	Motorized Pump	Solar Pump
*Reference category for dummy encoding.**Likely to be non functional.***	No strong evidence to get likelihood.	**Not statistically significant**

Table 12: Pump type

6. Distance to Nearest Town – The negative coefficient indicates that remote pumps have lower chance of being functional. That is, further away a pump from the nearest town, it is more likely to be nonfunctional, which is expected as accessibility of maintenance workers reduces.
7. Solar Pump Type, Installation Year, Water Pump age, Latitude and Longitude are not statistically significant in identifying Non Functioning Pumps.

We can run another Logistic regression model by removing the insignificant variables, but if we are removing insignificant variables, we will have to remove other pump types from consideration, but motorized pump type is statistically significant, and hand pump is a reference variable whose values are added to the constant. Therefore, we will consider this model only.

9.6. Confusion Matrix for Logistic Regression

By experimenting with different threshold probabilities, we computed the following matrices –

Threshold	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
0.50	**0.7235**	0.6148	0.7506	**0.7983**	**0.7737**	0.7065
0.60	0.7190	0.7034	**0.7813**	0.7298	0.7547	**0.7166**
0.70	0.6903	**0.8336**	0.8377	0.5915	0.6934	0.7125

Table 13: Different threshold probabilities

Therefore, looking at the above metrics, **our choice of model has with a probability threshold of 0.5**, as it gives us the **best accuracy score and also gives the best recall** and **F1Score.** Although the AUC-ROC is not the highest, it is comparable to the highest and is therefore satisfactory.

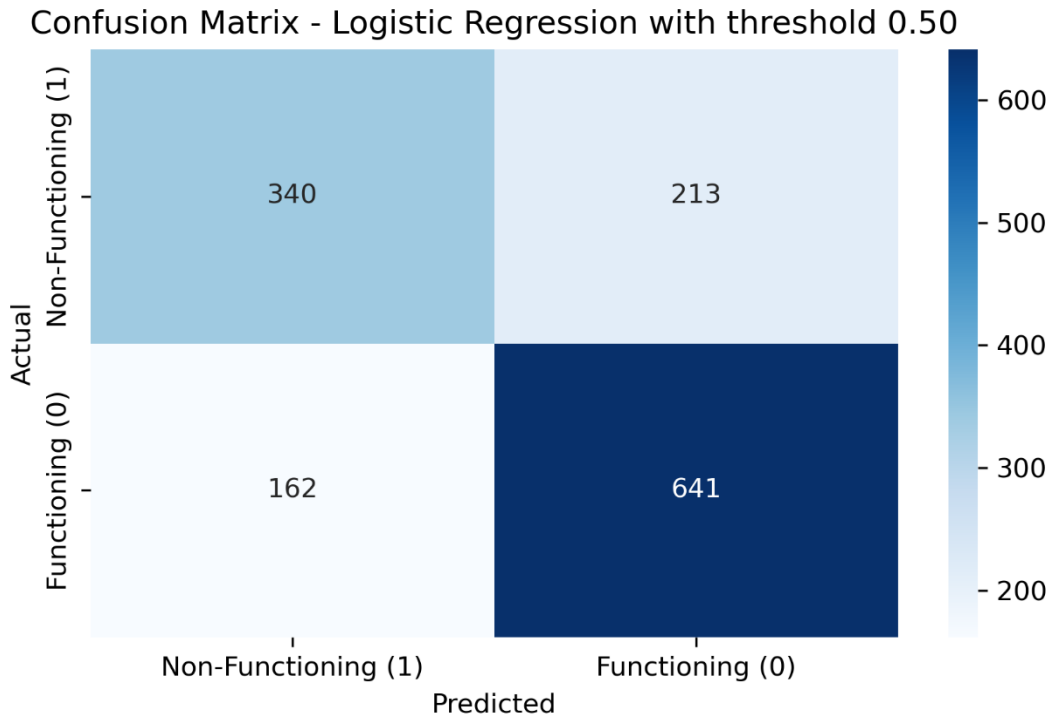

Figure 7: Confusion Matrix of Logistic Regression with 0.5

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000
kNN	0.7035	**0.6456**	**0.7528**	0.7435	0.7481	**0.7460**
Logistic	**0.7235**	0.6148	0.7506	**0.7983**	**0.7737**	0.7065

Table 14: Naive vs kNN vs Logistic Regression

We have managed to further increase our accuracy scores in the Logistic Model from kNN. The logistic Regression model now gives us the best accuracy, recall and F1Scores, while its AUC-ROC score is comparable to kNN.

10. Ensemble Methods with Random Forest DecisionTrees

A decision tree is a set of nested tests that use a divide and conquer method to predict or classify. These algorithms are considered greedy because they consider locally optimal decisions and thus overfit the training data. A random forest algorithm uses many decision trees.

We will use the bagging (replace with substitution) variant to conduct this algorithm and Entropy as the criteria for attribute selection within each decision tree. Entropy refers to the amount of information required to predict an event with certainty. The lesser the entropy, lesser the information required to decide confidently.

10.1. Data Scaling

While Decision Trees do not need numerical columns to be scaled, it is good practice to scale it. We **will standardize the numerical columns (using z scores)**. We also need to **one-hot encode the categorical variables**.

10.2. Selecting Parameters

There are 4 important parameters to experiment through to get better accuracy (and other performance metrics)

1. **max_depth –**
   This refers to **depth** or levels in a Decision Tree. More depth allows more splits and higher chances of getting purity at leaf nodes. But more depth can also result in overfitting. Therefore, a tradeoff needs to be made to get better generalization. In bagging, we can use shallow trees to improve generalization as ensemble bagging compensates for individual weakness.

   We will experiment with depths from 4,6,7,8
2. **n_estimators -**
   This refers to number of trees in the ensemble. More trees result in better averaging, but there are diminishing returns after a certain point, that is, the average doesn't change much after a point.

   We will experiment with 200,400,500

3. **max_samples -**
   This refers to the percentage of sample drawn with replacement to train each tree. In the assignment, we saw that experimenting with this did not result in changing the performance ceteris paribus.

   We will experiment with 0.25,0.50
4. **max_features** –
   This refers to how many features considered to train per tree.

   We will experiment with 0.25,0.50

a. Plotting the Results –

We generate models based on the above-mentioned parameters and test them on validation set to get the following plots for accuracy vs depth -

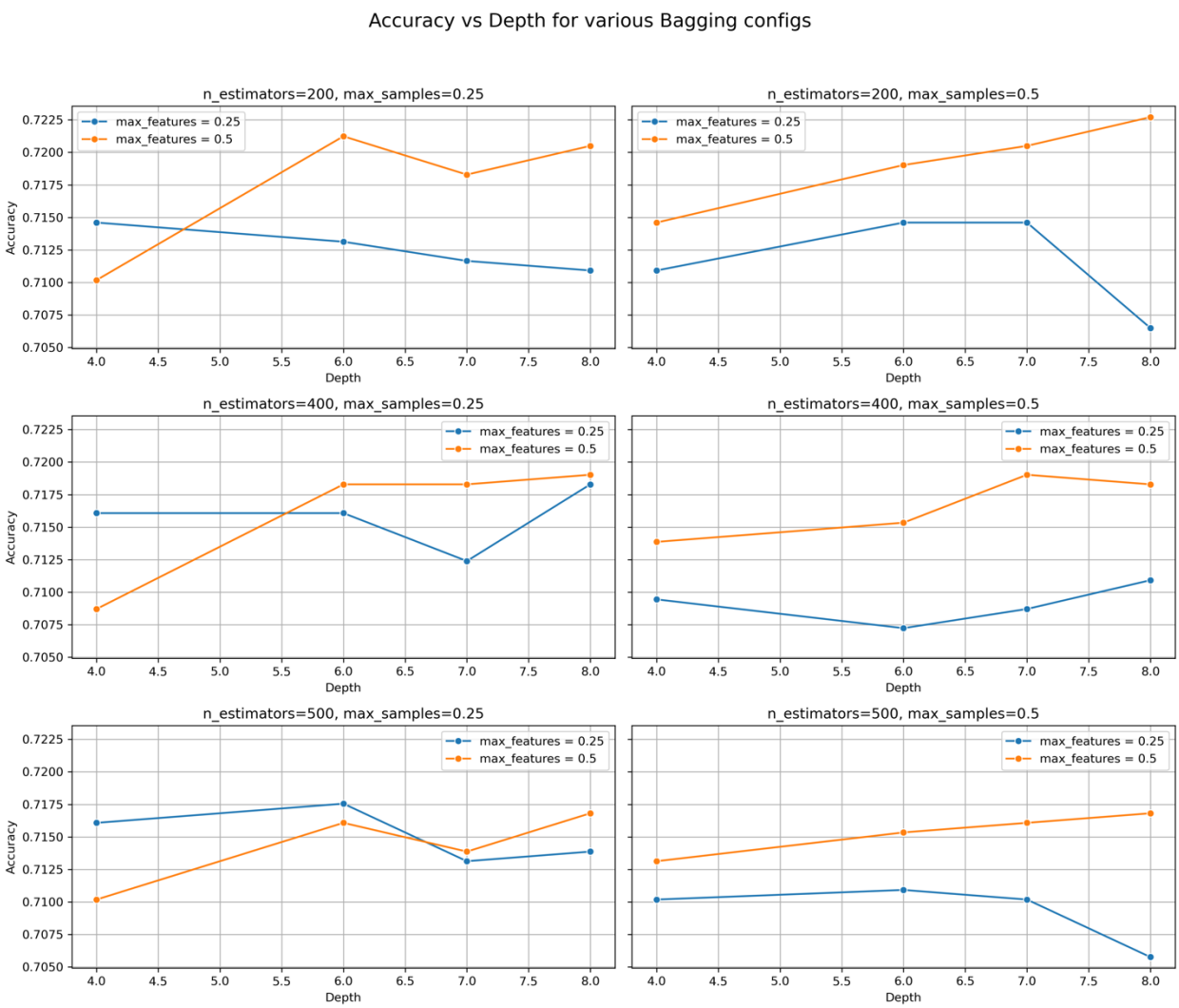

Figure 8: Accuracy vs Depth for various Bagging Configs

Observations –

1. Greater number of features results in better accuracy in general
2. The accuracy decreases for max_features 0.25, with n_estimators =200 and max_samples = 0.25.
3. For plot n_estimators = 200 , max_samples = 0.5 , the plot diverges for max_features = 0.25 and 0.5 at depth 7, suggesting overfitting with increasing depth. Similar behaviour can be experienced in plot with n_estimators =500, max_samples = 0.5.
4. Just by visually assessing the plots, depth = 6 looks to be a good choice. Depth = 7 and 8 show large divergence in some plots.

b. Results of the exercise –

The models with the highest accuracies are –

Depth	n_estimators	max_samples	max_features	Accuracy	Precision	Recall	F1 Score	Specificity	ROC-AUC Score
8	200	0.5	0.5	0.7227	0.7429	0.8132	0.7765	0.5913	0.7607
6	200	0.25	0.5	0.7212	0.7396	0.8169	0.7763	0.5823	0.7681
8	200	0.25	0.5	0.7205	0.7398	0.8144	0.7753	0.5841	0.7643

Figure 9: Models with highest accuracy

The highest accuracy obtained is 0.7227, which is less than logistic regression model.

c. Confusion Matrix for Decision Tree

The confusion matrix for model with

1. Depth = 8
2. N_estimators = 200
3. Max_samples = 0.5
4. Max_features = 0.5

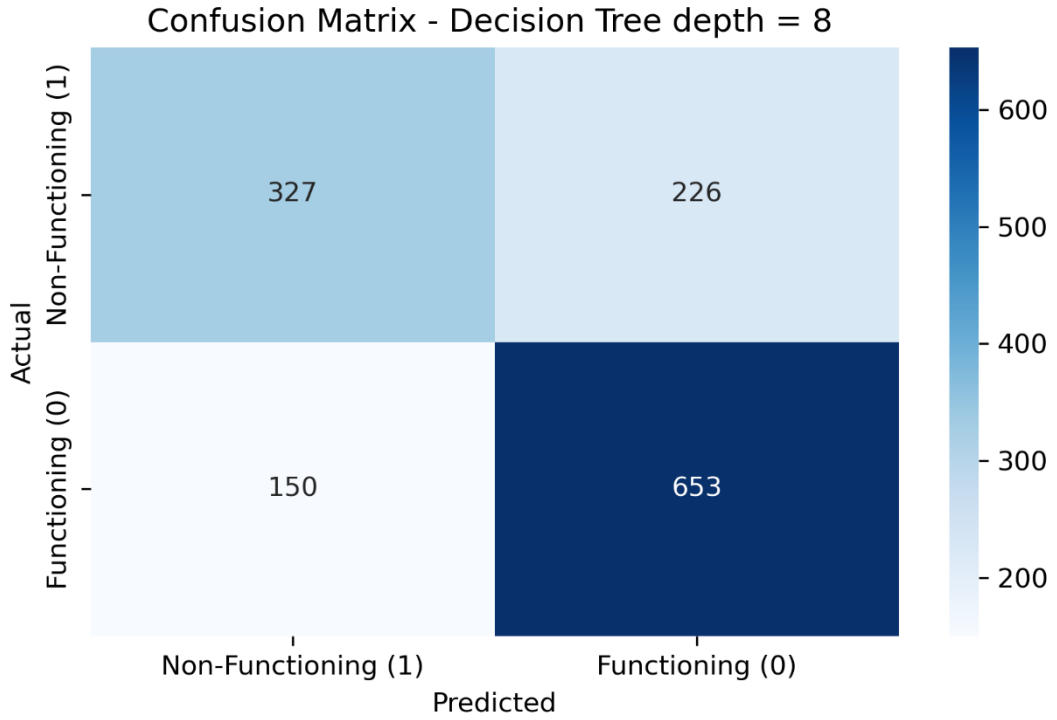

Figure 10: Confusion Matrix for Decision Tree with depth=8

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000
kNN	0.7035	**0.6456**	**0.7528**	0.7435	0.7481	0.7460
Logistic	**0.7235**	0.6148	0.7506	0.7983	**0.7793**	0.7065
Ensemble	0.7227	0.5913	0.7429	**0.8132**	0.7765	**0.7607**

Table 15: Naive vs kNN vs Logistic vs Ensemble

The **Ensemble model has the highest Recall and AUC-ROC Scores**; However, *logistic model has the highest accuracy metrics.*

11. Neural Networks

It is a supervised classification algorithm that resembles human learning. Its nature of computation is a black- box. We will experiment with models containing –

1. Activation Models –
   3 activation functions sigmoid (logistic), TanH and RelU.
2. Hidden Layers –
   We will experiment with 1 hidden layer (shallow learning) and 2 hidden layers (deep learning)
3. Number of Neurons -
   The formula for sigmoid activation function is like the formula of logistic regression. We would like to check the neural network with one hidden layer with sigmoid function gives similar accuracy to logistic regression with how many neurons. That neuron number will be the basis for increasing and decreasing for other models.

11.1. Data Scaling-

For the input layer, the numerical columns need to be scaled. We will use the StandardScaler (which standardizes using z score calculation). We will one hot encode all categorical columns. From the logistic Regression model, we saw that *Population Served* had a high VIF (close to 10) and numerical variables *Installation Year, Water Pump Age. Latitude a*nd *Longitude* are not statistically significant. We will drop these columns to make the model simpler and improve performance.

11.2. Building a model –

We conducted several neural networks with constant parameters –

1. 8 Neurons in Hidden Layer
2. 1 Neuron in Outer Layer
3. Epochs = 50
4. Batch Size = 32
5. Loss = 'Binary Cross Entropy"

Model	Layer 1 Neuron	Activation Function	Output Layer Neuron	Activation Function	Accuracy
1	8	Sigmoid	1	Sigmoid	0.7198
2	8	Sigmoid	1	relU	0.7146
3	8	Sigmoid	1	tanH	**0.7220**
4	8	relU	1	Sigmoid	0.7176
5	8	relU	1	relU	**0.7227**
6	8	relU	1	tanH	0.7176
7	8	tanH	1	Sigmoid	0.7168
8	8	tanH	1	relU	0.7176
9	8	tanH	1	tanH	**0.7212**

Table 16: Model Comparison

From the above table it is evident that having relU in both hidden and output layer gives the best accuracy of 0.7227. But all other accuracy is also very similar.

1. The sigmoid function gives an output [0,1] which is most appropriate for this dataset and problem
2. tanH is equally convenient for classification problems but gives an output [-1,1] which is not the ideal output, therefore it is better suited for hidden layers.

3. relU gives an output [0, infinity) which is also not appropriate for the output layer.

Therefore, the appropriate models for consideration condense into –

Model	Layer 1 Neuron	Activation Function	Output Layer Nueron	Activation Function	Accuracy
1	8	Sigmoid	1	Sigmoid	**0.7198**
4	8	relU	1	Sigmoid	0.7176
7	8	tanH	1	Sigmoid	0.7168

Table 17: Appropriate models consideration

Now we will experiment with numbers on neurons in the hidden layer to see if it improves accuracy

Model	Layer 1 Neuron	Activation Function	Output Layer Nueron	Activation Function	Accuracy
1	8	Sigmoid	1	Sigmoid	0.7198
2	4	Sigmoid	1	Sigmoid	0.7035
3	16	Sigmoid	1	Sigmoid	0.7035

Table 18: Models with changed neurons

Increasing or decreasing the neurons decreased accuracy. Therefore, the optimum number of neurons is 8.

Now we will experiment by adding another hidden layer of relU or tanH with 16 ,8 and.4 Neurons

Model	Layer 1 Neuron	Activation Function	Layer 2 Nueron	Activation Function	Output Layer Nueron	Activation Function	Accuracy
1	8	Sigmoid	16	Sigmoid	1	Sigmoid	0.5922
2	8	Sigmoid	16	relU	1	Sigmoid	0.7065
3	8	Sigmoid	16	tanH	1	Sigmoid	0.7065
4	8	Sigmoid	4	relU	1	Sigmoid	0.708
5	8	Sigmoid	8	relU	1	Sigmoid	0.7146
6	8	Sigmoid	8	tanh	1	Sigmoid	0.7124
7	8	Sigmoid	4	tanh	1	Sigmoid	0.708

Table 19: Models with 16, 8 and 4 neurons

Therefore, only one hidden layer gave better results than adding 2 layers.

Therefore, the model chosen is –

Model	Layer 1 Neuron	Activation Function	Output Layer Nueron	Activation Function	Accuracy
1	8	Sigmoid	1	Sigmoid	0.7198

Table 20: Chosen model

11.3. Confusion Matrix for Neural Networks –

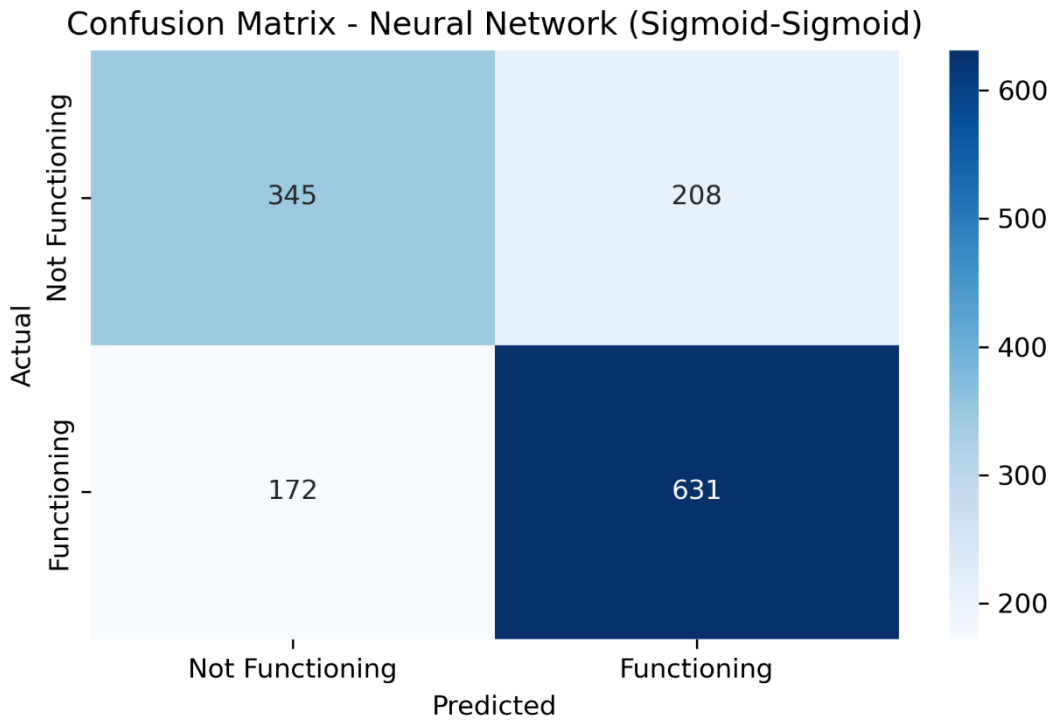

Figure 11: Confusion Matrix - Neural Network

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000
kNN	0.7035	**0.6456**	**0.7528**	0.7435	0.7481	0.7460
Logistic	**0.7235**	0.6148	0.7506	0.7983	**0.7793**	0.7065
Ensemble	0.7227	0.5913	0.7429	**0.8132**	0.7765	0.7607
Neural Networks	0.7198	0.6239	0.7521	0.7858	0.7686	**0.7658**

Table 21: Naive vs kNN vs Logistic vs Ensemble vs Neural Networks

The logistic regression still holds the position for highest accuracy, but the Nueral network model has a much higher AUC-ROC score, suggesting that Neural Networks may be a better model. The accuracy scores of logistic, ensemble and neural networks are comparable (almost similar)

12. Naïve Bayes

It is a non-parametric method that classifies or predicts a new record based on similar records in the training data. Similarity is calculated through distance formulas.

12.1. Data Preparation –

We will **one hot encode the categorical variables**.
The remaining numerical variables are –

1. **Distance to Nearest Town -**
   We will bin this into categories. The descriptive stats are -
   min 0
   25% 21
   50% 35

75% 44
max 76
We will bin them as follows based on real world estimates and then one hot encode them too–

Category	Range (km)	Remarks
Urban	0-10	Near on in a town
Semi – Urban	11-25	Reasonably close to a town
Rural	26-45	Far from a town but accessible
Remote	>45	Very Far away from any town

Table 22: Distance to Nearest Town

2. **Population Served –**
   We will bin this into categories, and one hot encode them. The stats are –
   min 2000
   25% 11000
   50% 13000
   75% 15000
   max 22000

   We will bin them as –

Category	Range
Low	>= 11000
Medium	11001 - 13000
High	13001 - 15000
Very High	>15000

Table 23: Population Served

3. **Water Pump Age** –
   This is left untouched – no standardization or normalization. The age itself is ordinal (ordered by magnitude)
4. **Installation Year -**
   This is left untouched – no standardization or normalization. The Year itself is ordinal (ordered by magnitude)
5. **Latitude –**
   We will drop this variable as this cannot be categorized or binned successfully. Distance to Nearest Town will also give the same information.
6. **Longitude -**
   We will drop this variable as this cannot be categorized or binned successfully. Distance to Nearest Town will also give the same information.

12.2. Confusion Matrix for Naïve Bayes –

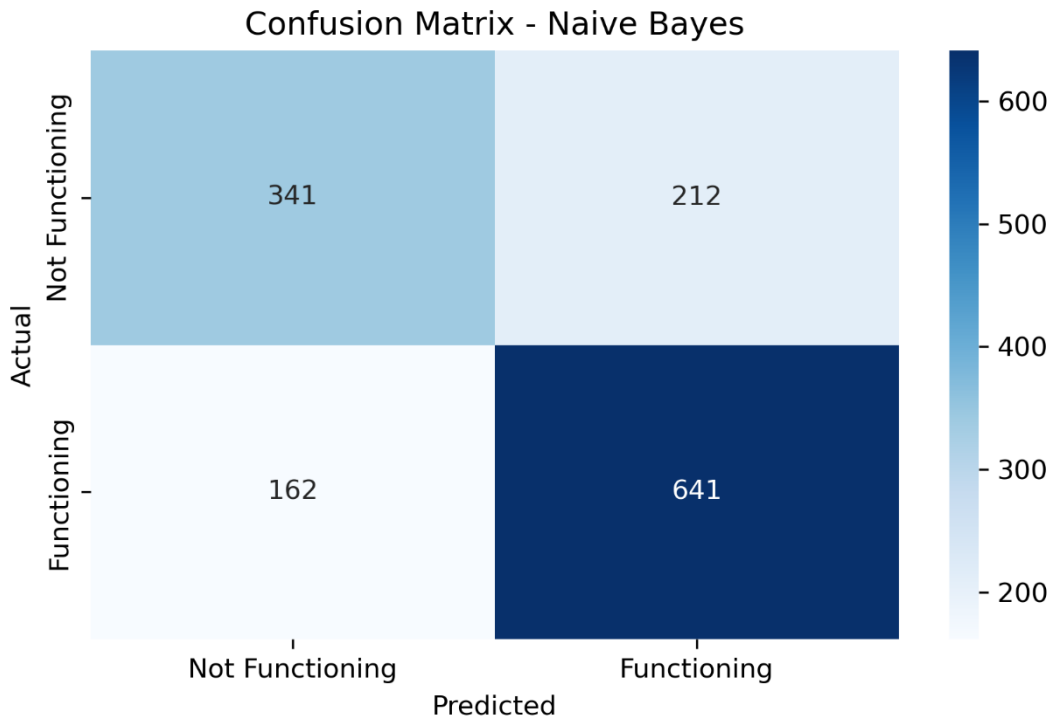

Figure 12: Confusion Matrix - Naive Bayes

Building on our table –

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve	0.5922	0.000	0.5922	1	0.7439	0.5000
kNN	0.7035	**0.6456**	**0.7528**	0.7435	0.7481	0.7460
Logistic	0.7235	0.6148	0.7506	0.7983	**0.7793**	0.7065
Ensemble	0.7227	0.5913	0.7429	**0.8132**	0.7765	0.7607
Neural Networks	0.7198	0.6239	0.7521	0.7858	0.7686	**0.7658**
Naïve Bayes	**0.7242**	0.6166	0.7515	0.7983	0.7742	**0.7807**

Table 24: Naive vs kNN vs Logistic vs Ensemble vs Neural Networks vs Naive Bayes

Therefore, Naïve Bayes predictive model has managed to give us the best accuracy and the best AUC-ROC Score. Scoring highest in 2 of the 3 most important metrics identified for this model. The recall at 79% is comparable to the highest value of 81%.

13. Conclusion

Following table in descending order of accuracy –

Model	Accuracy	Specificity	Precision	Recall	F1Score	AUC-ROC
Naïve Bayes	0.7242	0.6166	0.7515	0.7983	0.7742	0.7807
Logistic	0.7235	0.6148	0.7506	0.7983	0.7793	0.7065
Ensemble	0.7227	0.5913	0.7429	0.8132	0.7765	0.7607
Neural Networks	0.7198	0.6239	0.7521	0.7858	0.7686	0.7658
kNN	0.7035	0.6456	0.7528	0.7435	0.7481	0.746
Naïve	0.5922	0	0.5922	1	0.7439	0.5

Table 25: Table in descending order of accuracy for models

The result of this project finds that Naïve Bayes to be the most accurate way to predict whether a water pump is functional or nonfunctional. Although it doesn't have the best recall (Ensemble) the combination of having the highest accuracy and AUC-ROC along with negligible difference in precision and recall gives us confidence in our Naïve Bayes.