DSCI 5240

Data Mining and Machine Learning for Business

FINAL REPORT

**Title: Using Data to Optimize Water Pump Infrastructure**

**Group – 3**

Ujjaini Kaviraj - 11763515

Mihir Girish Sali - 11815717

Sindhuja Jupudi - 11724251

Richard Freeman - 11717263

**ACKNOWLEDGMENT**

We would like to sincerely thank Professor Javier Rubio-Herrero for his inspiring and dynamic approach to teaching. His guidance and insights have greatly contributed to our learning journey. We are also grateful to his teaching assistant for their timely and helpful support throughout the course. Working on this project using Python has been both a rewarding and educational experience, reinforcing our technical skills and deepening our understanding of data mining. This project has been a collaborative achievement, and we truly value the opportunity to apply what we've learned to a real-world problem. It has enriched our academic growth and fostered meaningful teamwork.

Thank you, Group 3

**EXECUTIVE SUMMARY**

The primary objective of this project is to develop a machine learning model capable of predicting the operational status of rural water pumps based on various physical, geographical, and socio-economic attributes. This includes determining whether a pump is functional (coded as 1) or non-functional (coded as 0). By analyzing features such as the pump’s age, installation year, source type, payment model, and more, the aim is to identify patterns that influence functionality. This prediction model can then be used to optimize maintenance schedules and improve resource allocation.

This project focuses on predicting the operational status of water pumps using data mining techniques. By leveraging real-world water infrastructure data, we apply classification models to predict whether a pump is functional or not. The project involved cleaning, transforming, and exploring the data followed by building several machine learning models including Logistic Regression, Random Forest, XGBoost, and Voting Ensemble

**DATASET DESCRIPTION:** The dataset used in this project contains 5,000 records, each representing a unique water pump installation. It includes 11 features along with a target variable indicating the operational status of the pump. These features cover a wide range of information, from technical specifications and geographic data to funding sources and community impact.

The dataset has the following features:

**Water Pump ID:** A unique code assigned to each pump for identification purposes.

**Water Source Type**: Describes where the water comes from, such as a well, borehole, river, or lake

**Water Quality:** Indicates the cleanliness of the water provided by the pump (e.g., clean, contaminated).

**Distance to Nearest Town:** The distance from the water pump to the nearest town or community (in kilometers).

**Population Served:** Estimates the number of people who rely on the pump for their water needs.

**Installation Year:** The year the water pump was installed

**Funder:** Identifies the organization or entity that financed the installation of the pump.

**Payment Type:** The payment type for water usage (e.g., free, pay per use).

**Water Pump Age**: The age of the water pump in years

**Pump Type:** The type of pump used (e.g., hand pump, motorized pump, solar pump).

**GPS Coordinates**: Provides the latitude and longitude of the pump's location, enabling spatial analysis

**Functioning Status**: The status of the water pump, indicating whether it is functioning or not functioning

A black and white screen with white text

AI-generated content may be incorrect.

Figure 1. Displaying a few rows and columns of dataset

A screenshot of a computer

AI-generated content may be incorrect.

*Figure 2. Data types in dataset*

Figure 2 shows that most characteristics are object data type, meaning they comprise strings, characters or mixed data. This includes information such as water pump ID, source type, water quality, and GPS coordinates. The installation year, Distance to Nearest Town, Population Served, and water pump age are provided as float64 data types, indicating numerical values.

A screenshot of a computer program

AI-generated content may be incorrect.

*Figure 3. Missing values*

An initial review of the dataset revealed that several features had missing entries—around 250 in each affected column. Key variables such as *Installation Year*, *Water Pump ID*, and *Population Served* all had 250 missing values. Importantly, the target variable *Functioning Status* also had 250 missing entries. Since this variable is essential for training any classification model, addressing its missing values was a top priority.

To ensure data quality and maintain the integrity of our analysis, we also needed to address missing values in other critical features, including *Water Quality*, *Funder*, and *Pump Type*. These were handled through appropriate imputation techniques, such as filling in missing numerical values with the mean and categorical values with the most frequent category (mode).

# DATA PREPARATION

## Handling Missing values:

* + **Numerical Columns:** The columns Populated served, water pump age, distance to nearest town will be imputed to maintain the central tendency. The missing values for the numerical columns in the dataset were imputed using KNNImputer model

A computer code with text

AI-generated content may be incorrect.

*Figure 4. Numerical columns imputation technique*

* + **Categorical columns:** water source type and payment type will be filled with mode to ensure consistency in dataset. The categorical columns are imputed using SimpleImputer with the strategy as ‘mode’

A computer screen shot of a program

AI-generated content may be incorrect.

*Figure 5. Categorical columns imputation technique*

* + The missing GPS coordinates and water pump ID will not be imputed as there will be meaningfulness for the analysis regarding incorrectly logged data points.\

1. **Consistency checks**: The water source type and Pump type were reviewed for consistency and missing values will be imputed.
2. The GPS coordinates transformed into separate columns like Latitudes and Longitudes.
3. Irrelevant columns like GPS and Water Pump ID were removed
4. Outliers were retained as they may carry real-world meaning

# EXPLORATORY DATA ANALYSIS FINDINGS

Exploratory Data Analysis (EDA) was conducted to uncover the underlying patterns in the dataset and provide context for the modeling phase. Histogram and KDE plots revealed a slightly right-skewed distribution for 'Water Pump Age', suggesting that most pumps are relatively new, but a significant minority are 30+ years old and still operational. This has important implications for infrastructure longevity planning. 'Water Quality' was heavily skewed towards the 'Contaminated' category, signaling widespread issues in water safety that must be addressed.  
  
The class distribution of the target variable 'Functioning Status' was initially imbalanced, prompting the need for class balancing techniques. A correlation heatmap was generated to study linear relationships between features. The strongest finding was a 0.14 correlation between Functioning Status and Distance to Nearest Town, confirming expectations, while most other variables showed weak linear associations. Importantly, this indicated the suitability of nonlinear models such as Random Forest and Gradient Boosting which do not rely on linear assumptions. The EDA also surfaced the need to prioritize features like 'Source Type' and 'Water Quality', which appeared to correlate with functional status based on visual inspection of class-wise distributions.

A graph with blue and orange bars

AI-generated content may be incorrect.

*Figure 6. Water Source Type Distribution*

This bar chart presents the count of water pumps categorized by their source type (e.g., Well, Borehole, Lake, River). The majority of the pumps in the dataset source their water from wells and lakes. This finding is important because the source type can affect the likelihood of contamination and mechanical wear. For example, surface water sources like lakes and rivers may introduce more debris into the pump system, leading to more frequent breakdowns compared to underground sources.

A diagram of water pump age and functioning status

AI-generated content may be incorrect.

*Figure 7. Water Pump Age Distribution*

This graph illustrates the distribution of pump ages in the dataset. A majority of the pumps are less than 30 years old, but there is a noticeable tail of older pumps. The presence of operational pumps older than 30 years underscores the importance of maintenance and the durability of certain pump types. Older pumps were also more likely to be non-functional, reinforcing the need to include age as a key predictive feature.

A chart of a diagram

AI-generated content may be incorrect.

*Figure 8. Distance to Nearest Town vs Functioning Status*

This distribution plot reveals how remote the water pumps are relative to the nearest towns or settlements and their functioning status. While many pumps are located within accessible distances, there are notable outliers representing very remote installations. Such remoteness likely correlates with reduced maintenance access and could be predictive of non-functional pumps.

**Correlation Heatmap**

A screenshot of a computer generated image

AI-generated content may be incorrect.

*Figure 9. Correlation Matrix*

This heatmap visualizes pairwise Pearson correlations between numeric features. The strongest finding was a 0.14 correlation between Functioning Status and Distance to Nearest Town, which is intuitive. Most other feature correlations were weak, indicating limited linear dependencies supporting the use of non-linear models like Random Forests and XGBoost.

# METHODOLOGY

The primary objective of this project was to predict the operational status of water pumps using supervised machine learning techniques. To achieve this, the dataset was divided into two subsets: 70% for training and 30% for testing. This split allowed the models to learn patterns from a substantial portion of the data while preserving a meaningful set for evaluating their performance on unseen samples.

Before modeling, the dataset underwent thorough preprocessing. Since missing data can significantly compromise the reliability of predictive models, appropriate imputation strategies were applied to ensure completeness and consistency. Features were preprocessed using a column transformer that scaled numeric fields and encoded categorical ones. Five models were trained using the processed data: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and a Neural Network. A Voting Classifier ensemble of XGBoost and Gradient Boosting was also implemented. Model performance was evaluated using test accuracy and classification metrics such as precision, recall, and F1-score. Cross validation is done as there is slight over-fitting in the model. It reduces the overfitting by ensuing generalization and provides a more reliable performance estimate.

**MODELS USED**

1. **Logistic Regression**:  
   This is a simple, linear model used primarily as a baseline. While highly interpretable, it assumes linear relationships between features and the target. Due to the non-linear nature of many relationships in the dataset, it performed relatively poorly.

Test Accuracy: Approximately 73.5%

1. **Random Forest**:  
   An ensemble method that constructs multiple decision trees on random subsets of the data and averages their predictions. It is robust to outliers and handles categorical variables well. It performed competitively in this problem.  
   Test Accuracy: Approximately 74.2%
2. **Gradient Boosting**:  
   This model builds decision trees sequentially, each correcting the mistakes of the previous one. It often achieves higher accuracy than Random Forest but can be more sensitive to overfitting. It provided one of the top performances.  
   Test Accuracy: Approximately 74.5%
3. **XGBoost**:  
   An optimized and scalable variant of Gradient Boosting. It includes built-in handling of missing values and regularization, making it well-suited for large, noisy datasets. Its performance matched Gradient Boosting.  
   Test Accuracy: Approximately 74.2%
4. **Neural Network (MLPClassifier):**  
   A simple feed-forward multi-layer perceptron used to capture more complex patterns in the data. While powerful in theory, this implementation was outperformed by tree-based methods due to limited tuning.  
   Test Accuracy: Approximately 72.8%
5. **Voting Ensemble (XGBoost + Gradient Boosting)**:  
   This ensemble aggregates predictions from XGBoost and Gradient Boosting using soft voting, leading to improved and stabilized performance. It achieved the highest accuracy among all models.  
   Test Accuracy: **74.60%**

## Results:

Every model has been fitted and run the model to get the Accuracy Score as follows:

**Classification Models Results:**

|  |  |
| --- | --- |
| **Classifiers** | **Accuracy (%)** |
| LogisticRegression | 73.5% |
| Random Forest | 74.2% |
| XG Boosting | 74.2% |
| Gradient Boosting | 74.5% |
| NeuralNetwork | 72.7% |
| Ensemble (XGBoost + Gradient Boosting) | 74.60% |

**INFERENCE ON RESULTS**

Among the individual models, Gradient Boosting achieved the highest standalone accuracy at 74.5%, closely followed by Random Forest and XGBoost, both at 74.2%. These results highlight the strength of tree-based ensemble methods in handling complex, structured data with mixed feature types. The Neural Network, while capable of modeling non-linear relationships, underperformed slightly with test accuracy of 72.7%, possibly due to limited architecture complexity or sensitivity to data scaling.

To further improve predictive performance, an ensemble model combining XGBoost and Gradient Boosting was implemented using a soft voting strategy. This approach aggregated the strengths of both base learners and ultimately yielded the best performance overall with a test accuracy of **74.60**%. The ensemble's success demonstrates the power of combining diverse but complementary models to enhance generalization and stability.

These results validate the effectiveness of machine learning in supporting data-driven decision-making for infrastructure maintenance. The modeling process not only achieved high predictive accuracy but also provided insights into key factors influencing pump performance such as water source type, distance to towns, and installation history. These insights can guide policymakers and engineers in prioritizing repairs, planning new installations, and allocating resources more efficiently.

**Managerial Implications**

The Voting Ensemble (XGBoost + Gradient Boosting) achieved the best performance with a test accuracy of 74.60%.

The classification report shows:  
- Precision: 76% for functional pumps, 72% for non-functional pumps  
- Recall: 82% for functional pumps, 64% for non-functional pumps  
- Macro average F1-score: 0.73  
These results indicate a balanced trade-off between identifying working and non-working pumps.

**CONCLUSION**

Working on this project has been a valuable and eye-opening experience. It allowed us to apply what we’ve learned in class to a practical, real-world problem: predicting whether a water pump is functional or not. Throughout the project, we explored different machine learning models ranging from simpler algorithms like Logistic Regression to more advanced ensemble methods like Random Forest, Gradient Boosting, and XGBoost.

Beyond just building models, we learned how critical data preparation is. Handling missing values, converting data types, and understanding outliers were all essential steps that directly impacted our results. We also realized how important it is to understand the story behind the data, things like how the water source or pump age might influence whether a pump is working.

Overall, this project helped us connect technical skills with real-world problem solving. It showed us how data science can make a real difference especially when it comes to improving infrastructure and supporting communities. Most importantly, it gave us the confidence to take on future projects with a more thoughtful and structured approach.