CarolineNjorog3 / tanzania-water-wells

Code Pull requests Actions Projects Wiki Security Insights Settings

Modelling

0 stars 6 forks 0 watching 2 Branches 0 Tags Activity

Public repository · Forked from LynnsBaraka/tanzania-water-wells

master 2 Branches 0 Tags Go to file t Go to file Add file Code

This branch is up to date with LynnsBaraka/tanzania-water-wells:master . Contribute Sync fork

Amadig70 Merge pull request LynnsBaraka#13 from Amadig70/master 22 minutes ago

.ipynb_checkpoints	updated markdown	2 days ago
images	additional documents	28 minutes ago
Phase 3 Non technical presentatio...	additional documents	28 minutes ago
ReadMe.md	additional documents	28 minutes ago
index.ipynb	Final Notebook	9 hours ago
tanzania-water-wells.pdf	additional documents	28 minutes ago
test_set_values.csv	Adjusted the training set values	last week
training_set_labels.csv	Adjusted the training set values	last week
training_set_values.csv	Adjusted the training set values	last week

README

Predicting Water Pump Faults in Tanzania

Business Understanding

Ensuring access to clean and reliable water sources is critical for public health, economic development, and communities' overall well-being. However, in Tanzania, access to clean and reliable water remains a major socio-economic challenge. According to a World Bank report on universal access to water and sanitation in 2023, 39% (24 million) of the population in the country suffers from acute water shortage throughout the year. This leads to water-related preventable deaths and diseases, costing the Tanzanian economy more than $2.4 billion each year in excess medical costs and lost productivity. These are resources that could have been channeled towards much-needed development. It is estimated that providing adequate water through optimizing the current water supply will significantly reduce economic losses and generate savings that would facilitate more spending towards development.

Business Problem

Tanzania's water supply system is characterized by frequent water pump breakdowns resulting from lack of proper maintenance and inefficient management. This leads to disruptions in water supply, exacerbating the acute shortage of clean water and subsequent socio-economic losses.

The government of Tanzania, in collaboration with NGOs and partner organizations, aims to enhance access to clean water by improving the maintenance and functionality of water wells across the country. To achieve this, water point sustainability risk levels across the supply system need to be projected by learning from current point profiles to guide stakeholders' decisions through highlighting:

- Most dilapidated wells that should be prioritized for maintenance, repairs, or rehabilitation.
- Sites to be earmarked for future wells.
- Data-driven recommendations that are responsive to stakeholders' needs and actionable guide improve management practices and water accessibility.

Stakeholders

- Government of Tanzania: Interested in optimizing resources for maintaining and repairing water wells to ensure clean water access for citizens.

- NGOs focusing on clean water access: Seeking to identify and prioritize wells in need of repair to efficiently allocate resources and interventions.
- Private sector partners: Potentially interested in investing in water infrastructure projects aligned with corporate social responsibility initiatives or in collaboration with NGOs and governmental organizations.

Objective

The objective of this project is to predict the operational/functionality status of water pumps based on various features related to the pumps' installation. Specifically, the project will:

- Evaluate factors that affect the functionality of a pump.
- Identify and model combinations of features that best predict the functionality of a water pump.
- Test and validate the accuracy of the model.
- Draw conclusions and recommendations.

Data Understanding

The dataset for this project was obtained from www.drivendata.org titled "Pump it Up: Data Mining the Water Table" in 2015. It is divided into three CSV files: a training set containing 59,400 observations (80%) and a test set containing 14,850 observations (20%). The third dataset contains training set labels that detail status group information for each of the training set values indicating whether the pump is "functional", "non-functional", or "in need of repairs". Both the train and test datasets have 40 similar columns with information about water pumps in Tanzania, including various attributes such as pump location, construction details, management, payment details, and water quality.

Key Columns Include:

- id: Unique identifier for each water pump
- amount_tsh: Total static head (amount of water available to pump)
- date_recorded: Date the pump data was recorded
- funder: Organization or individual that funded the pump installation
- gps_height: Altitude of the pump location
- installer: Organization or individual that installed the pump
- longitude: Geographic longitude coordinate of the pump location
- latitude: Geographic latitude coordinate of the pump location
- wpt_name: Name of the waterpoint (e.g., name of the well)
- num_private: Number of private plots reserved for the waterpoint
- population: Population catchment served by the well
- construction_year: Year the pump was installed
- basin: Geographic basin of the pump location
- subvillage: Geographic location of the well within the village
- region: Geographic region of the pump location
- region_code: Code representing the geographic region
- district_code: Numeric code for the administrative district
- lga: Local Government Area
- ward: Administrative division within a district
- public_meeting: Indicator of whether there was a public meeting about the well
- recorded_by: Name of the person who entered the well into the database
- scheme_management: Management of the water scheme (e.g., water board)

- scheme_name: Name of the water scheme
- permit: Indicator of whether the waterpoint is permitted
- extraction_type: Method used to extract water from the well
- extraction_type_group: Grouped extraction type
- extraction_type_class: Class of extraction type
- management: Type of management for the pump
- management_group: Grouped management type
- payment: Payment type for water service
- payment_type: Type of payment
- water_quality: Quality of water provided by the pump

DATA PREPARATION.

Importing Necessary Libraries,Loading and Inspecting Datasets.

Importing the required libraries is the first step we undertook in performing data exploration, preprocessing, and predictive modeling. This involves loading the necessary packages that provide functions and tools for various data manipulation tasks. Additionally, it's essential to load and inspect the datasets to understand their structure, contents, and any initial preprocessing requirements. This initial inspection helps in planning the subsequent steps of data preprocessing and modeling.

Loading the dataset

Exploring CSV Datasets

This function is designed to import three CSV files (`training_set_values.csv` , `test_set_values.csv` , and `training_set_labels.csv`), explore their structure, and display the first few rows of each dataset.

Importing CSV Files

The function first imports the CSV files using the `pd.read_csv()` function from the Pandas library. Three DataFrames are created:

- `training_set_values` : Contains training data values.
- `test_set_values` : Contains test data values.
- `training_set_labels` : Contains labels for the training data.

Exploring Dataset Structure

The function defines a nested function `explore_structure()` to explore the structure of each dataset. This function utilizes the `.info()` method of Pandas DataFrame to display a concise summary of the dataset including column names, data types, and non-null counts.

Exploring First Few Rows

Another nested function `explore_first_few_rows()` is defined to display the first few rows of each dataset using the `.head()` method of Pandas DataFrame. This provides a preview of the data contained in each dataset.

Calling the Function

Finally, the function is called using `explore_datasets()` , which executes all the steps described above, effectively importing the CSV files, exploring their structure, and displaying the first few rows of each dataset.

Exploring Distribution of Water Well Conditions

After loading and exploring the datasets, it's essential to understand the distribution of different conditions of water wells in the training dataset. This information provides valuable insights into the dataset's characteristics and can influence subsequent analysis and decision-making.

Counting Instances by Condition

To understand the distribution of water well conditions, we utilize the `.value_counts()` method on the `status_group` column within the `training_set_labels` dataset. This allows us to count how many instances belong to each category in the `status_group` column, which represents the condition of the water wells.

Interpretation

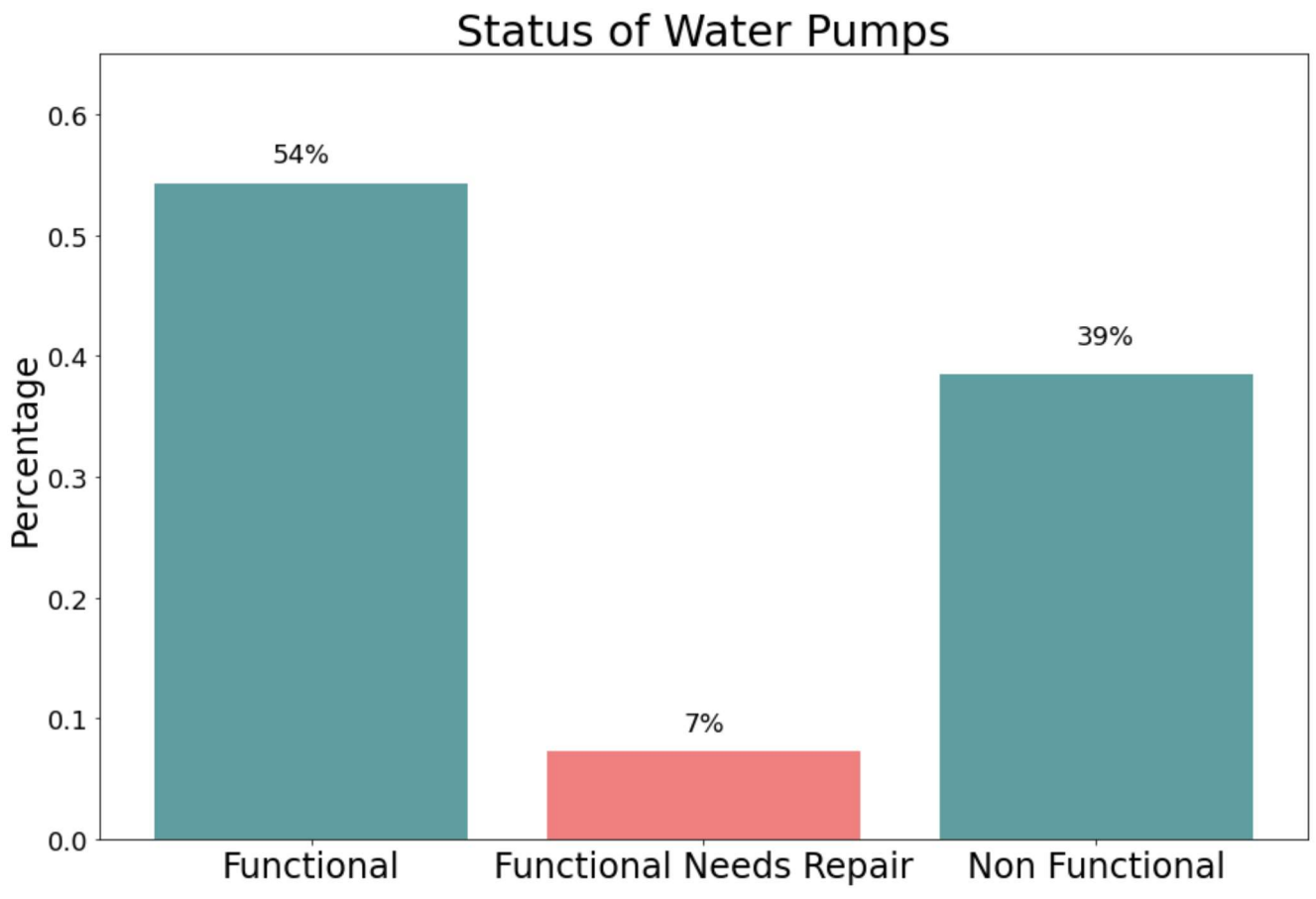
The output of this operation provides a breakdown of the number of instances for each condition category. By examining this distribution, we can gain insights into the prevalence of different water well conditions, such as functional, non-functional, or functional needs repair. This understanding is crucial for formulating strategies related to water well management, resource allocation, and decision-making.

The initial exploration of the dataset involves the following activities

Checking the shape of the dataset to understand its dimensions, i.e., the number of rows and columns.
Inspecting the properties of the dataset, such as data types, missing values, and basic statistics.
Understanding the description of the dataset, including the meaning of each column and its potential relevance to the analysis.
Sampling the first few rows of the dataset to get a glimpse of the data and understand its structure.

Visualizing Water Pump Status

Below is a bar chart visualizing the status of water pumps



Exploring Summary Statistics of Features

Understanding the summary statistics of both numerical and categorical features in the dataset is crucial for gaining insights into the characteristics and distribution of data. This step aids in comprehensively understanding the dataset and preparing it for analysis and modeling tasks related to water points or any other domain-specific dataset.

Summary Statistics of Numerical Features

The summary statistics of numerical features provide essential information such as:

- Count: The number of non-null values for each numerical feature.
- Mean: The average value of each numerical feature, giving an indication of the central tendency.
- Standard Deviation: The measure of dispersion or spread of numerical values around the mean.
- Minimum and Maximum: The minimum and maximum values observed in each numerical feature, indicating the range of values.
- Quartiles: The values that divide the data into four equal parts, providing insights into the distribution of numerical values.

These statistics help in understanding the central tendency, dispersion, and distribution of numerical features, which is valuable for identifying outliers, assessing data quality, and selecting appropriate modeling techniques.

Summary Statistics of Categorical Features

For categorical features, the summary statistics provide:

- Count: The number of non-null values for each categorical feature.
- Unique: The number of unique categories present in each categorical feature.
- Top: The most frequent category in each categorical feature.
- Frequency: The frequency of the top category, indicating its prevalence in the dataset.

These statistics help in understanding the distribution of categorical features, identifying dominant categories, and assessing data quality issues such as missing or inconsistent values.

Importance to Water Point Dataset

In the context of water point datasets, summary statistics of numerical features can reveal insights into various quantitative aspects such as water quantity, quality, and infrastructure characteristics. Understanding these numerical features' summary statistics can help in identifying patterns, trends, and potential issues related to water points, aiding decision-making processes regarding water resource management, infrastructure development, and service provision.

Similarly, summary statistics of categorical features provide insights into qualitative aspects such as water point functionality, management, and usage patterns. Analyzing these statistics can help in understanding the distribution of different water point conditions, types, and management practices, which are essential for informing policies, interventions, and resource allocation strategies related to water point management and service delivery.

Data Cleaning

Data cleaning is a crucial step in preparing datasets for analysis and modeling tasks. Below it explains a Python function that handles missing values and duplicates in datasets using Pandas.

Handling Missing Values.

Data Visualization

Exploring Top 30 Funders.

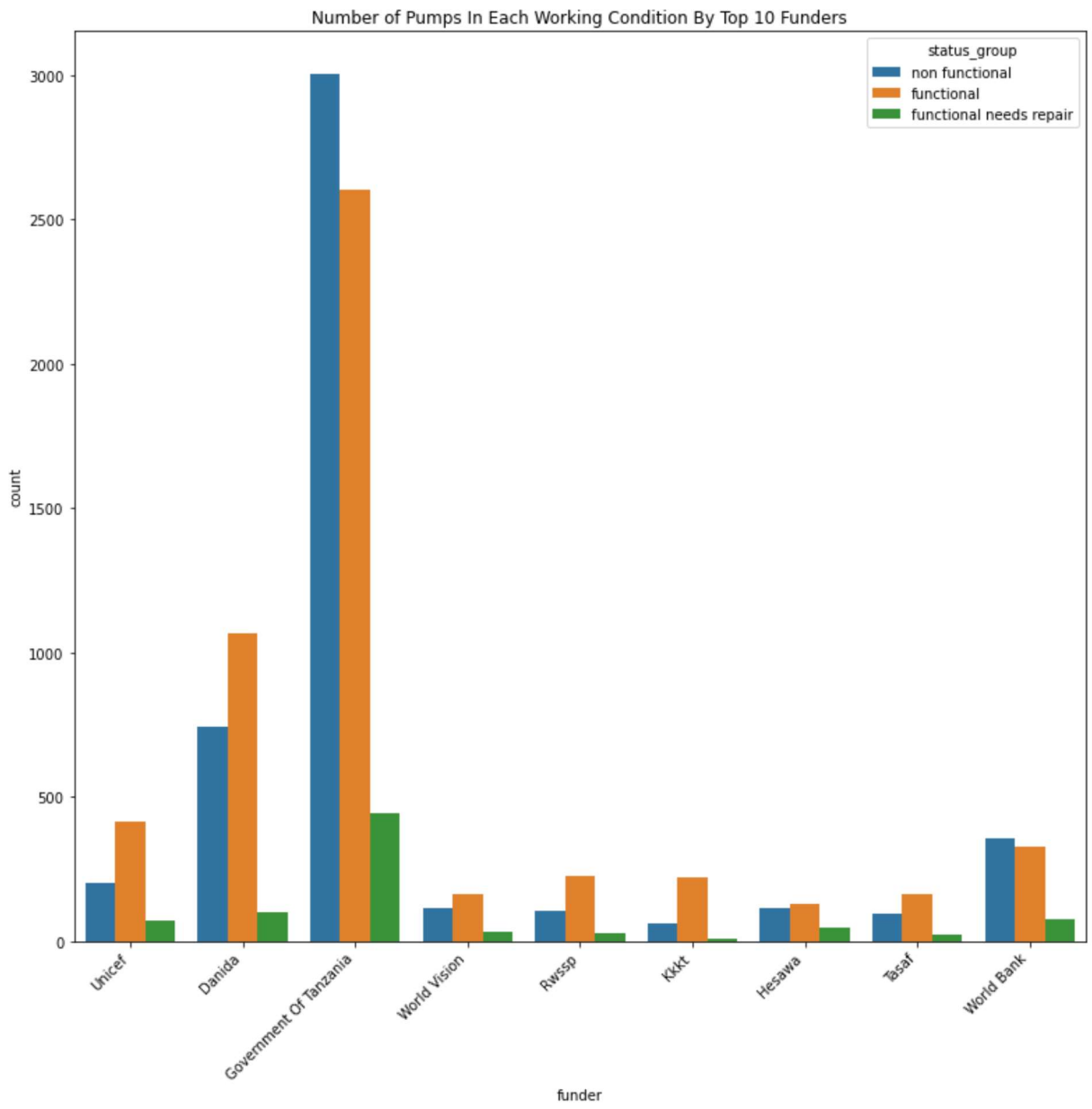
Visualizing the top funders in a dataset can provide insights into the distribution and importance of funding sources. This markdown demonstrates how to find and visualize the top 30 unique funders in a dataset using Python.

Interpreting Pump Conditions by Top 10 Funders

The count plot below visualizes the number of pumps in each working condition by the top 10 funders. Here's an interpretation of the results:

Government Of Tanzania: This funder appears to have the highest number of pumps across all working conditions, indicating a significant role in funding water projects and potentially having a larger infrastructure footprint. **Danida, Hesawa, Rwssp, World Bank, etc.**: These funders also show notable counts across different working conditions, suggesting significant contributions to water projects.

The count plot helps in understanding the distribution of pump conditions across different funders, enabling stakeholders to identify patterns, trends, and potential areas for intervention or improvement in water project management and service delivery.



Interpreting Pump Status Across Regions

The count plot below visualizes the distribution of pump status across different regions. Here's an interpretation of the results:

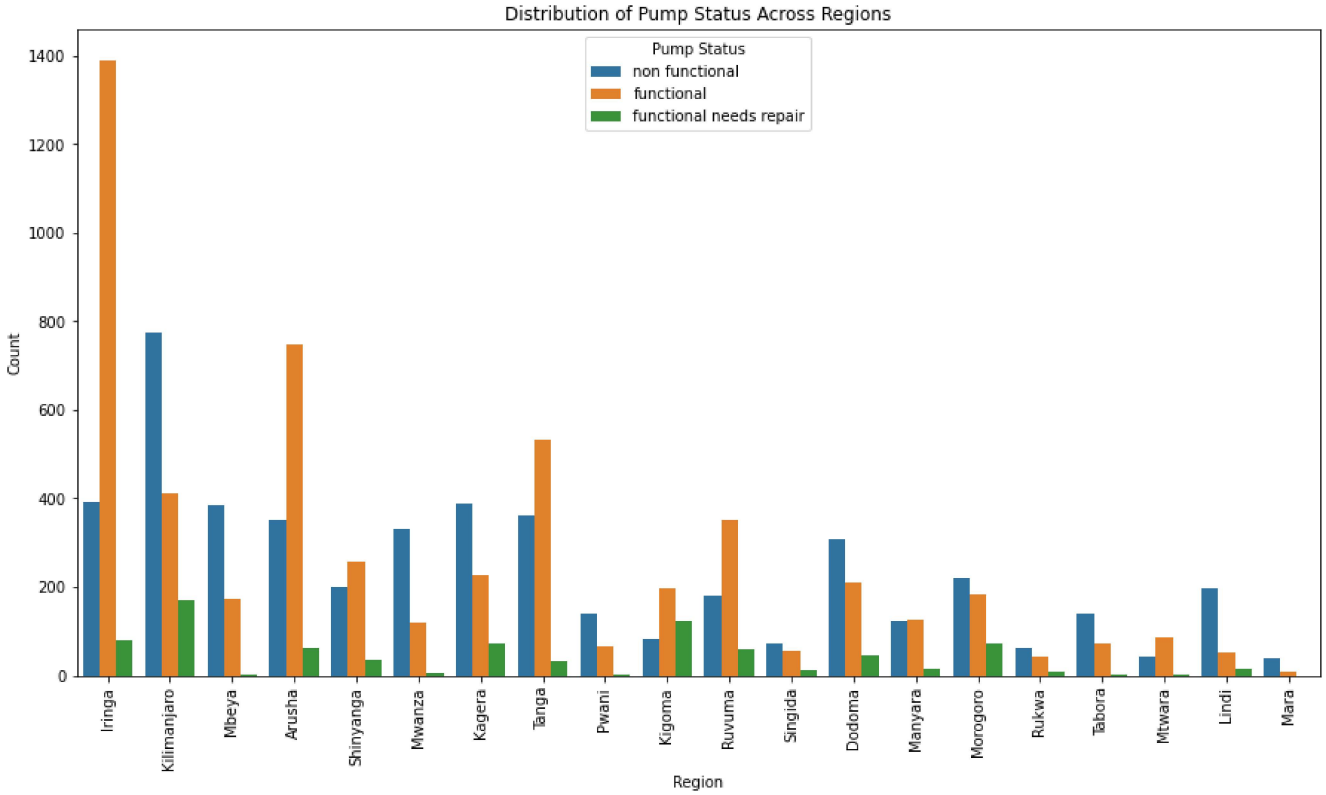
Distribution: The plot provides insights into how pump status varies across different regions. It helps in understanding the prevalence of different pump status categories such as functional, functional needs repair, and non-functional in each region.

Regional Disparities: Observing the distribution across regions can highlight regional disparities in pump functionality and maintenance. Some regions may have a higher proportion of non-functional pumps, indicating potential challenges in water access and infrastructure management.

Insights for Intervention: Identifying regions with a higher prevalence of non-functional pumps can inform targeted interventions and resource allocation strategies. It can guide decision-makers in prioritizing regions for repair and maintenance activities to ensure sustainable access to clean water.

Policy Implications: The visualization aids in formulating evidence-based policies and interventions aimed at improving water infrastructure and service delivery across different regions. It provides valuable insights for stakeholders involved in water resource management and development initiatives.

Overall, this visualization helps in understanding the spatial distribution of pump status across regions, enabling informed decision-making and targeted interventions to address water access challenges effectively.



Interpreting Pump Status Across Top 10 Installers

The count plot below visualizes the distribution of pump status across the top 10 installers. Here's an interpretation of the results:

Distribution: The plot provides insights into how pump status varies across different installers. It helps in understanding the prevalence of different pump status categories such as functional, functional needs repair, and non-functional among the top 10 installers.

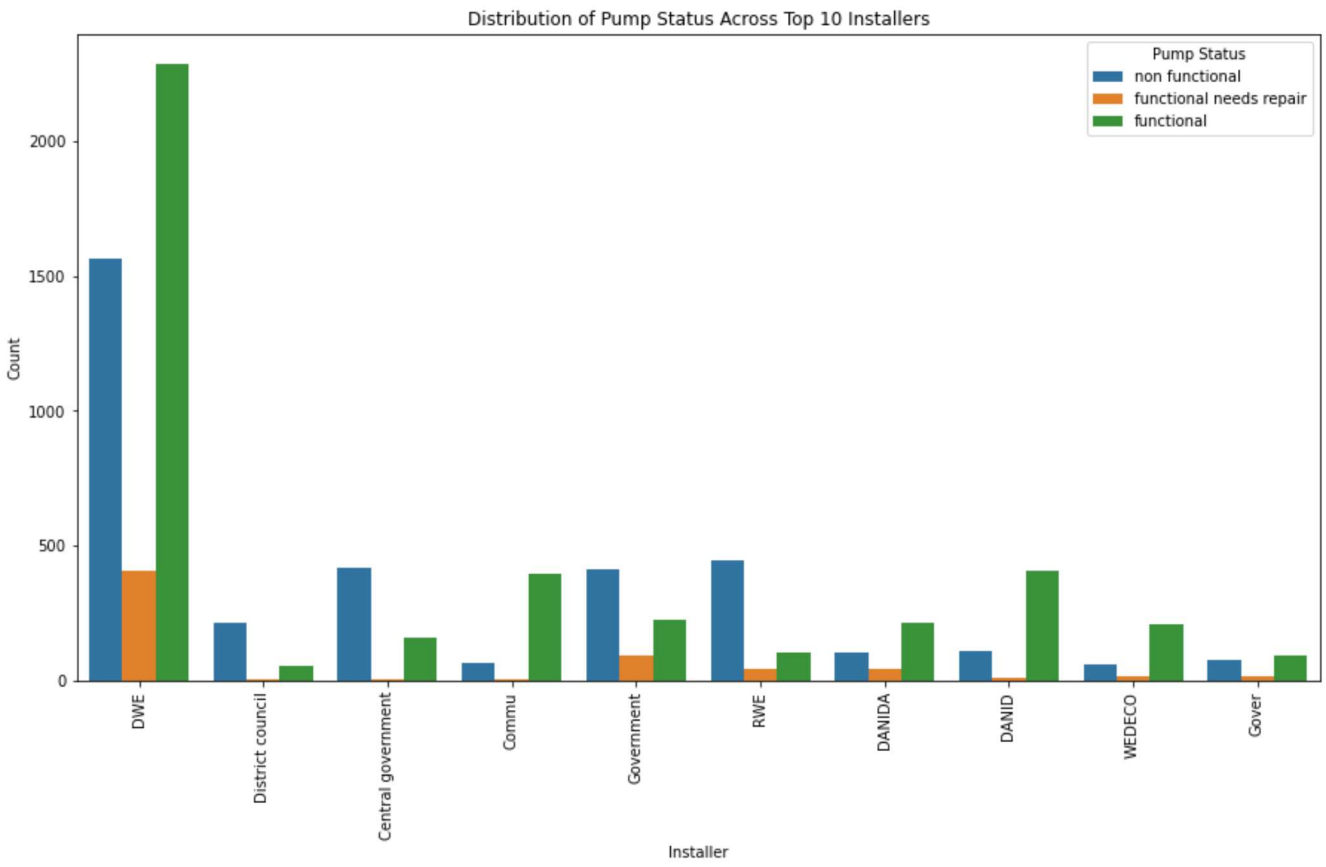
Installer Performance: Observing the distribution across installers can highlight variations in installer performance in terms of pump functionality and maintenance. Some installers may have a higher proportion of non-functional pumps, indicating potential challenges in installation quality or maintenance practices.

Quality Control: Identifying installers with a higher prevalence of non-functional pumps can inform quality control measures and training initiatives to improve installation standards and maintenance practices.

Intervention Strategies: The visualization aids in formulating targeted intervention strategies, such as capacity-building programs or performance incentives, to improve installer performance and ensure the reliability and sustainability of water infrastructure.

Stakeholder Engagement: It provides valuable insights for stakeholders involved in water infrastructure development, including governments, NGOs, and community organizations, to collaborate with installers and implement effective measures for enhancing water service delivery.

Overall, this visualization helps in understanding the distribution of pump status across top installers, enabling stakeholders to identify areas for improvement and implement targeted interventions to ensure the functionality and reliability of water pumps.



Interpreting Pump Status Across Basins

The count plot below visualizes the distribution of pump status across different basins. Here's an interpretation of the results:

Distribution: The plot provides insights into how pump status varies across different basins. It helps in understanding the prevalence of different pump status categories such as functional, functional needs repair, and non-functional in each basin.

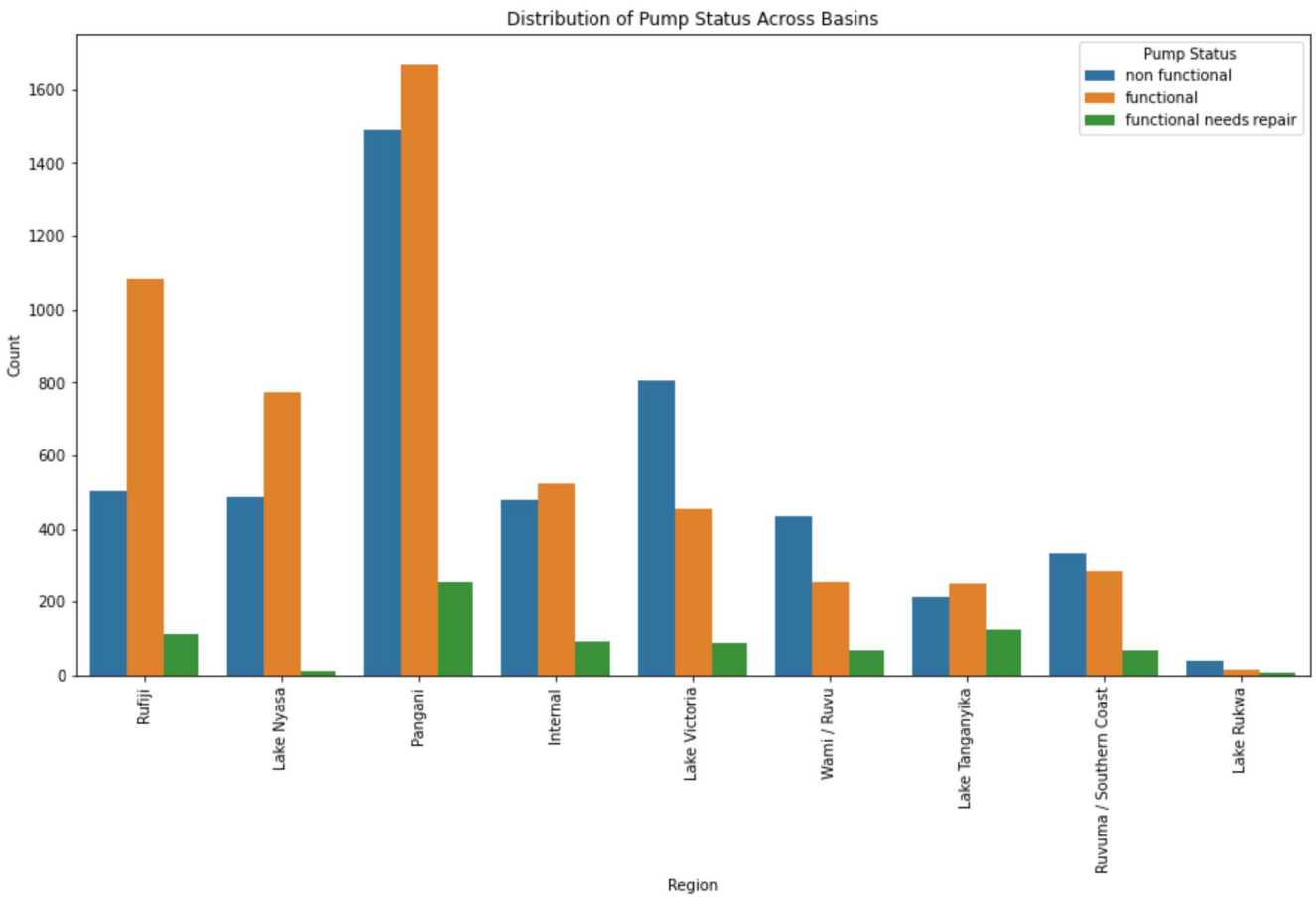
Regional Disparities: Observing the distribution across basins can highlight regional disparities in pump functionality and maintenance. Some basins may have a higher proportion of non-functional pumps, indicating potential challenges in water access and infrastructure management.

Resource Allocation: Identifying basins with a higher prevalence of non-functional pumps can inform resource allocation strategies and prioritization of interventions. It can guide decision-makers in directing resources and efforts towards regions with the greatest need for repair and maintenance activities.

Environmental Considerations: Basins with a high proportion of non-functional pumps may face environmental challenges or geological factors affecting water infrastructure. Understanding these factors is essential for implementing sustainable solutions and mitigating risks to water access and quality.

Policy Implications: The visualization provides valuable insights for policymakers and stakeholders involved in water resource management and development. It facilitates evidence-based decision-making and supports the design of targeted interventions to improve water infrastructure and service delivery across different basins.

Overall, this visualization helps in understanding the distribution of pump status across basins, enabling stakeholders to identify areas for intervention and prioritize resources effectively to ensure sustainable access to clean water.



Interpreting Distribution of Pump Status on Water Quality

The count plot below visualizes the distribution of pump status across different categories of water quality. Here's an interpretation of the results:

Water Quality Categories: The x-axis represents different categories of water quality, such as 'colored', 'salty', 'milky', etc. Each bar corresponds to a specific water quality category.

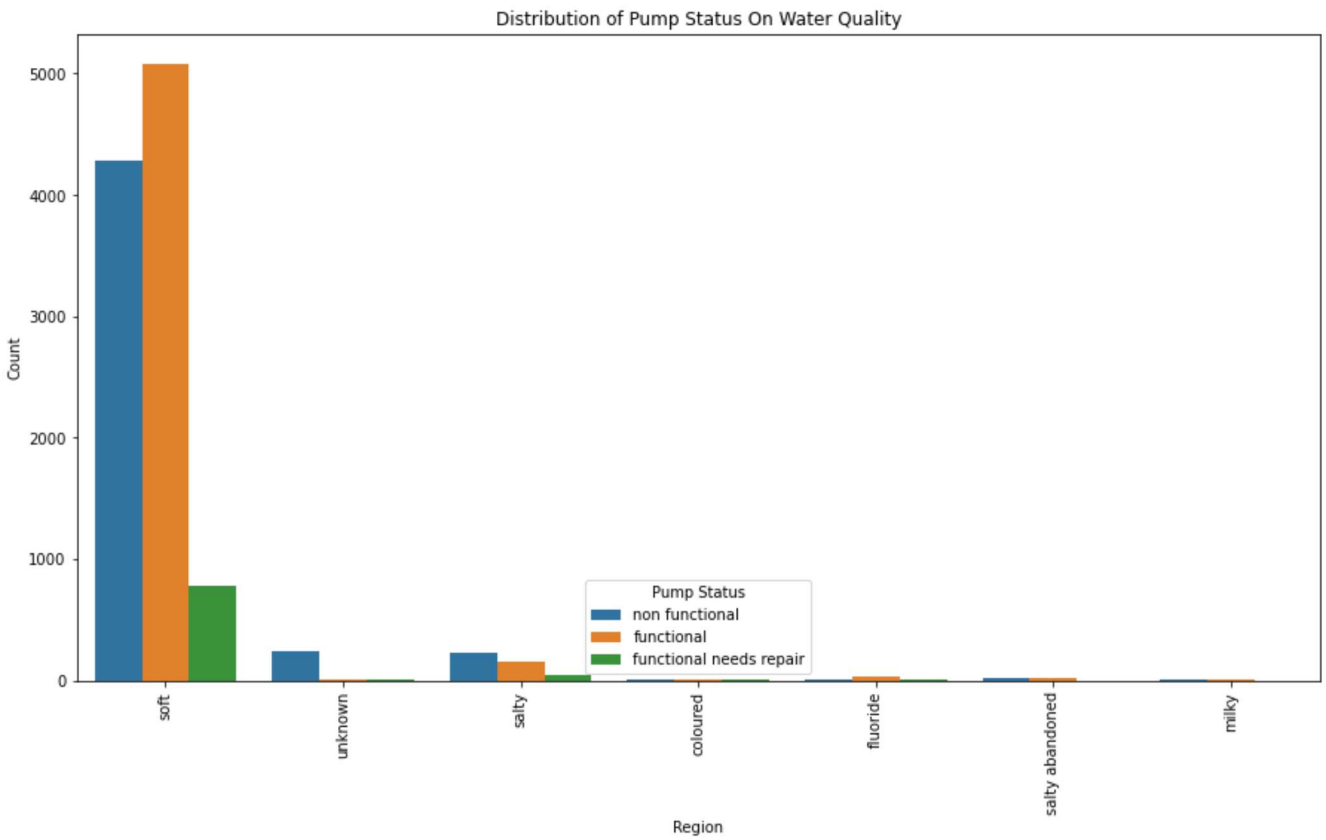
Pump Status: The height of each segment within the bars represents the count of pumps in a particular pump status category (functional, functional needs repair, non-functional). The colors indicate different pump status categories.

Comparison: The plot allows for a comparison between different water quality categories in terms of their distribution of pump statuses. For instance, it's evident whether certain water quality categories have a higher proportion of functional pumps compared to others.

Insights: Observing the distribution of pump statuses across water quality categories can provide insights into the relationship between water quality and pump functionality. It may highlight patterns or correlations between specific water quality issues and pump performance.

Potential Implications: Understanding how pump status varies across different water quality categories can have implications for water management strategies. It may inform decisions related to water treatment processes, infrastructure maintenance, or resource allocation to address specific water quality challenges.

Overall, this visualization helps in understanding how pump status is distributed across different categories of water quality, facilitating insights into the relationship between water quality issues and pump functionality.



Interpretating Distribution of Pump Status Across Various Extraction Types.

Here's a breakdown of the interpretation:

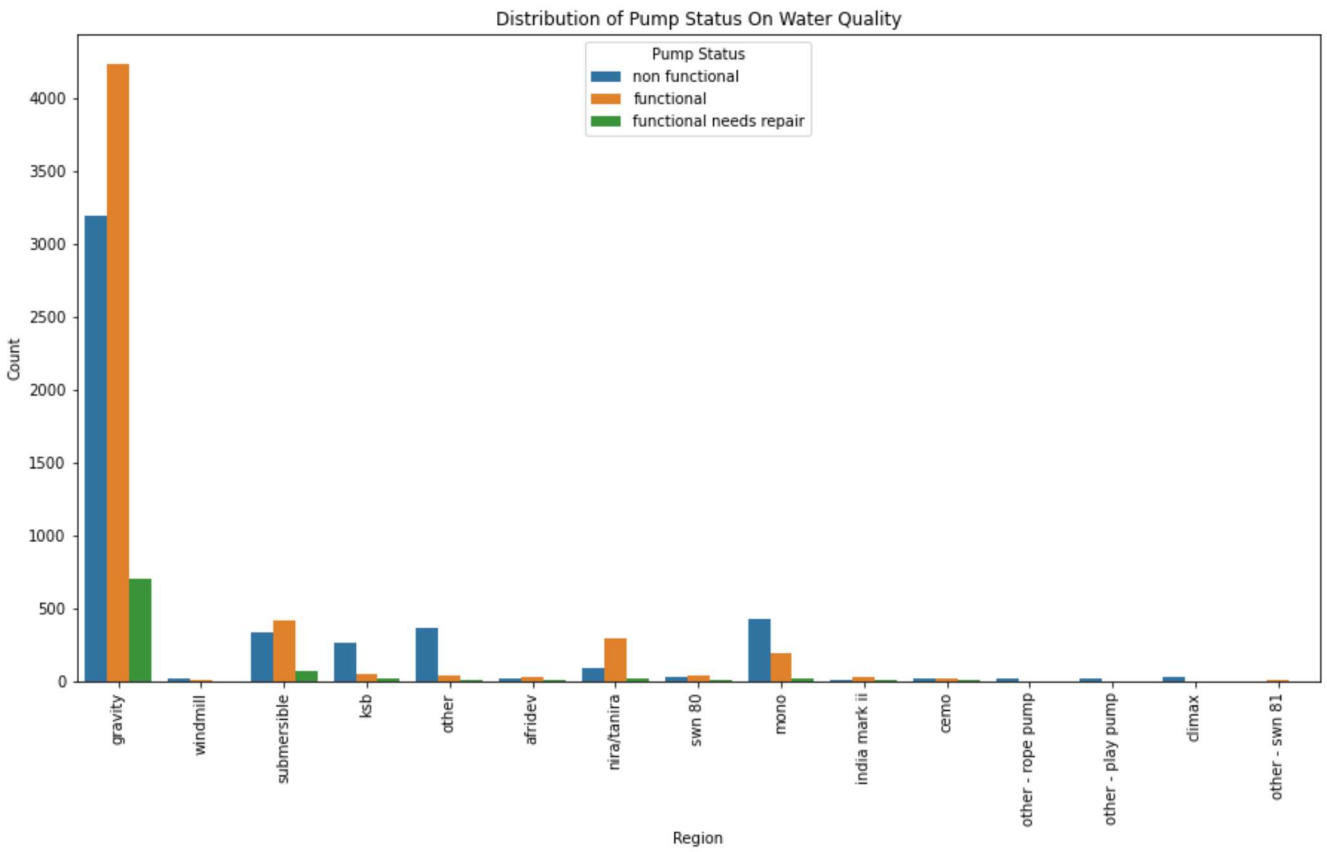
Extraction Types: Extraction types represent the methods employed for extracting water from wells. Each extraction type is labeled on the x-axis.

Count of Pumps: The y-axis displays the count of pumps corresponding to each extraction type.

Plot Representation: Each bar in the plot represents the count of pumps for a specific extraction type. The segments within each bar represent the distribution of pump status categories.

Overall, this visualization facilitates a comparative analysis of pump status across different extraction types, enabling stakeholders to make informed decisions regarding water resource management and maintenance.

The Government of Tanzania is the biggest funder of water wells in Tanzania. Most of the wells did not have information on who funded them. However, among those with funding information available, Danida, Hesawa, RWSSP, and the World Bank were the second, third, fourth, and fifth largest funders of the wells, respectively.



Distribution of Pump Status across different Water Point Types.

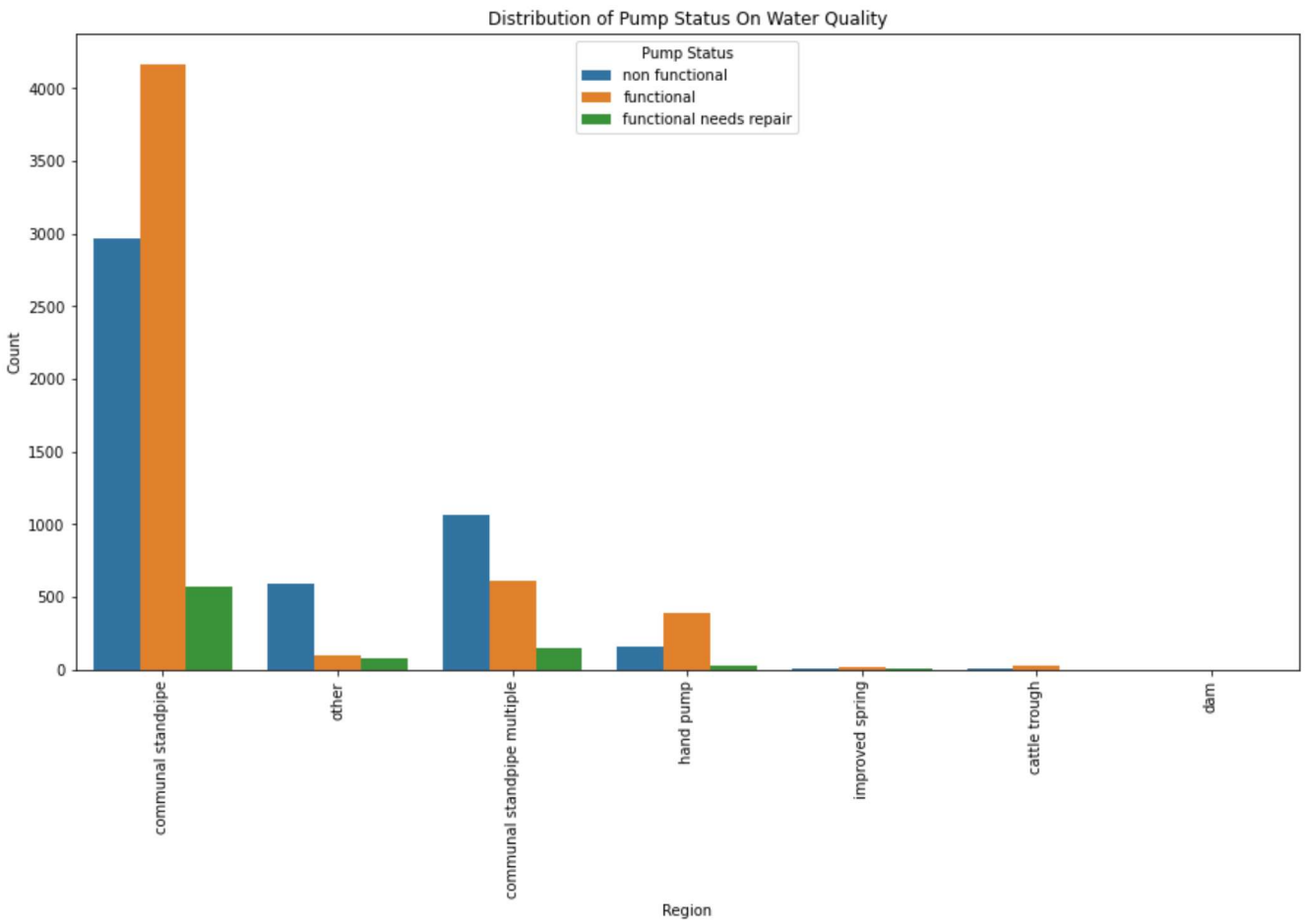
Water Point Types: Water point types represent the various types of water sources, such as hand pumps, communal standpipes, or wells. Each water point type is labeled on the x-axis.

Count of Pumps: The y-axis displays the count of pumps corresponding to each water point type.

Pump Status Categories: Pump status categories are distinguished by different colors in the legend. These categories indicate the operational status of the pumps, such as functional, non-functional, or functional but needs repair.

Insights: By examining the plot, one can discern how the distribution of pump status varies across different water point types. This insight can provide valuable information for assessing the performance and maintenance needs of different types of water sources.

Overall, this visualization aids in comparing the pump status across various water point types, enabling stakeholders to identify patterns and prioritize interventions for improving water infrastructure.



Analyzing Distribution of Amount in TSH Across Pump Status

This visualization presents a box plot depicting the distribution of the amount in Tanzanian Shilling (TSH) across different pump status categories. Here's an explanation of the results:

Box Plot: The box plot visually represents the distribution of the amount in TSH for each pump status category. Each box represents the interquartile range (IQR) of the data, with the median indicated by a horizontal line inside the box.

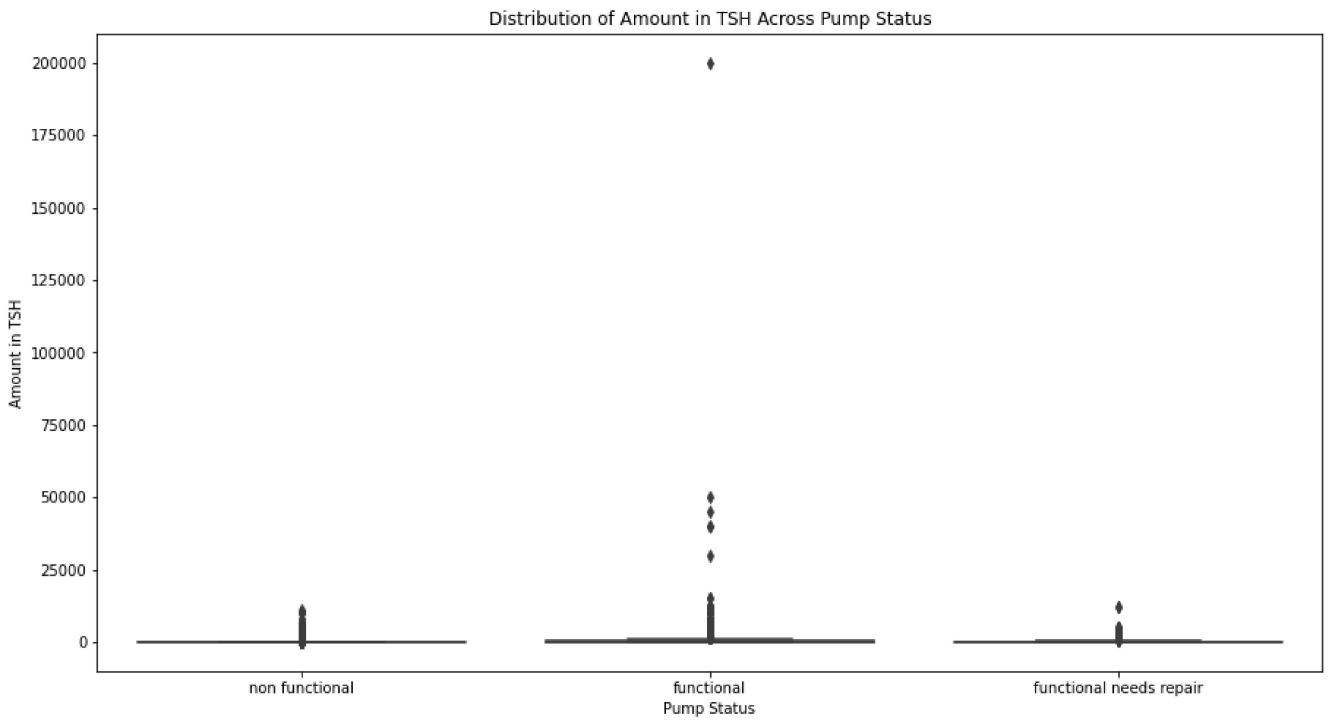
Pump Status Categories: Pump status categories are displayed on the x-axis. These categories indicate the operational status of the pumps, such as functional, non-functional, or functional but needs repair.

Amount in TSH: The y-axis represents the amount in Tanzanian Shilling (TSH) associated with each pump status. The box plot provides insights into the variability and central tendency of the amount for different pump status categories.

Box Plot Elements:

- The box represents the IQR, with the lower and upper hinges marking the first and third quartiles, respectively.
- The whiskers extend from the edges of the box to the minimum and maximum values within 1.5 times the IQR from the first and third quartiles, respectively.
- Outliers beyond the whiskers are indicated as individual data points.

This visualization facilitates understanding of how the amount in TSH varies across different pump status categories, enabling stakeholders to assess the financial implications associated with pump functionality and maintenance.



Interpretation of Pump Status Distribution by Management

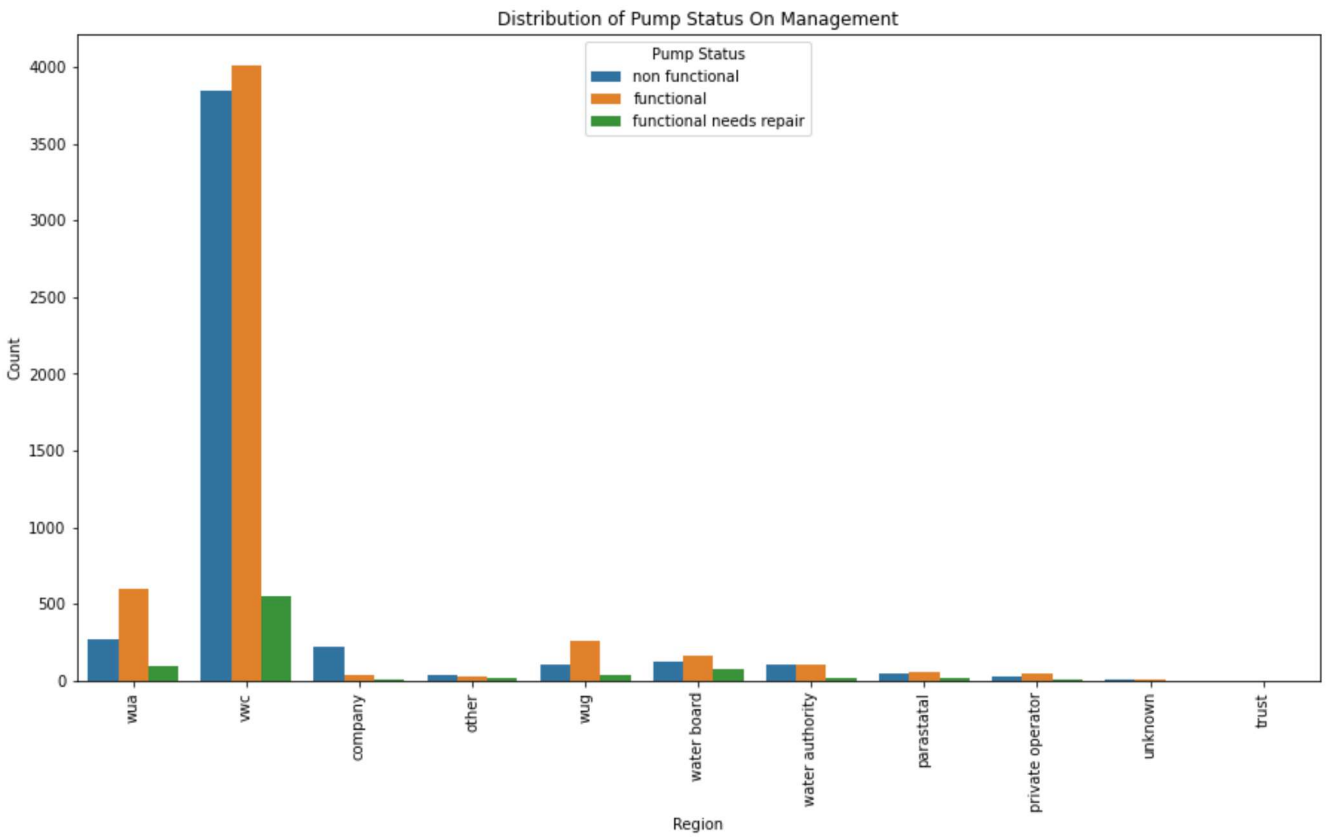
The countplot below visualizes the distribution of pump status across different management categories. Here's an interpretation of the results:

x-axis (Management): Represents different management categories associated with the waterpoints. **y-axis (Count)**: Indicates the count of waterpoints belonging to each management category. **Hue (Pump Status)**: Different colors represent the status of the pumps, categorized as functional, functional needs repair, or non-functional.

Key Observations:

Pump Status Distribution: Across different management categories, we observe variations in the distribution of pump statuses. This indicates that the management approach may have an influence on the functionality of waterpoints. **Functional Pumps Dominance**: In several management categories, the majority of pumps are functional, indicating successful management practices leading to operational waterpoints. **Non-Functional Pumps**: Some management categories exhibit a higher proportion of non-functional pumps, suggesting challenges or deficiencies in management practices in those areas. **Functional Needs Repair**: In certain management categories, there's a notable presence of pumps needing repair, highlighting the importance of proactive maintenance and management strategies.

This visualization provides insights into how the management approach correlates with the functionality status of waterpoints, which can inform decision-making and resource allocation for improving waterpoint management and maintenance practices.



Analyzing Pump Status Geographical Distribution

This visualization presents a scatter plot using Plotly Express, mapping the geographical distribution of pumps based on their latitude and longitude coordinates. Here's an explanation of the results:

Geographical Distribution: The scatter plot displays the location of pumps on a map, with latitude and longitude coordinates defining each pump's position.

Pump Status Colorization: Pump status is represented by different colors on the map. Each color corresponds to a specific pump status category, such as functional, non-functional, or functional but needs repair. The color legend provides clarity on the mapping of pump status.

Hover Information: Hovering over each pump marker provides additional information, such as the pump status. This interactive feature enhances the visualization by enabling users to explore detailed data points on the map.

Map Layout: The map layout is set to "open-street-map" style, providing a clean and easily interpretable backdrop for visualizing the pump distribution.

This visualization facilitates geographical analysis of pump status distribution, enabling stakeholders to identify regions with clusters of functional or non-functional pumps. It can aid in decision-making processes related to resource allocation, maintenance prioritization, and infrastructure planning.

Dropping Unnecessary Columns

The following columns were identified as unnecessary for our objective and have been dropped from the dataset:

- **wpt_name**: Name of the waterpoint (not relevant for our analysis).
- **num_private**: Number of private pumps (not relevant for our analysis).
- **subvillage**: Subvillage location (not relevant for our analysis).
- **district_code**: District code (not relevant for our analysis).
- **lga**: Local government authority (not relevant for our analysis).

- **ward**: Administrative ward (not relevant for our analysis).
- **public_meeting**: Whether there was a public meeting related to the waterpoint (not relevant for our analysis).
- **recorded_by**: Entity recording the data (not relevant for our analysis).
- **scheme_name**: Name of the waterpoint scheme (not relevant for our analysis).
- **extraction_type_group**: Grouped extraction type (redundant with 'extraction_type').
- **extraction_type_class**: Classification of extraction type (redundant with 'extraction_type').
- **management_group**: Grouped management type (redundant with 'management').
- **payment_type**: Payment method (redundant with 'payment').
- **quality_group**: Grouped water quality (redundant with 'water_quality').
- **quantity_group**: Grouped water quantity (redundant with 'quantity').
- **source_type**: Source type (redundant with 'source').
- **source_class**: Source class (redundant with 'source').
- **waterpoint_type_group**: Grouped waterpoint type (redundant with 'waterpoint_type').

These columns were deemed unnecessary as they provided redundant information and were not relevant for our analysis objectives.

Remaining Columns

After dropping the unnecessary columns, the following columns remain in the dataset:

- **id**: Identifier for each waterpoint.
- **amount_tsh**: Amount of water available in Tanzanian Shilling.
- **date_recorded**: Date of data recording.
- **funder**: Organization or individual who funded the waterpoint.
- **gps_height**: GPS height of the waterpoint.
- **installer**: Organization or individual who installed the waterpoint.
- **longitude**: Longitude coordinate of the waterpoint location.
- **latitude**: Latitude coordinate of the waterpoint location.
- **basin**: Geographic water basin.
- **region**: Geographic region.
- **region_code**: Region code.
- **population**: Population around the waterpoint.
- **scheme_management**: Management of the waterpoint scheme.
- **permit**: Whether the waterpoint is permitted.
- **construction_year**: Year of construction of the waterpoint.
- **extraction_type**: Type of extraction used for the waterpoint.
- **management**: Management of the waterpoint.
- **payment**: Payment method for the waterpoint.
- **water_quality**: Quality of water from the waterpoint.
- **quantity**: Quantity of water from the waterpoint.
- **source**: Source of water for the waterpoint.
- **waterpoint_type**: Type of waterpoint.

These remaining columns are considered relevant for answering various questions related to waterpoint functionality, management, and quality.

Feature Engineering

Calculating Waterpoint Age

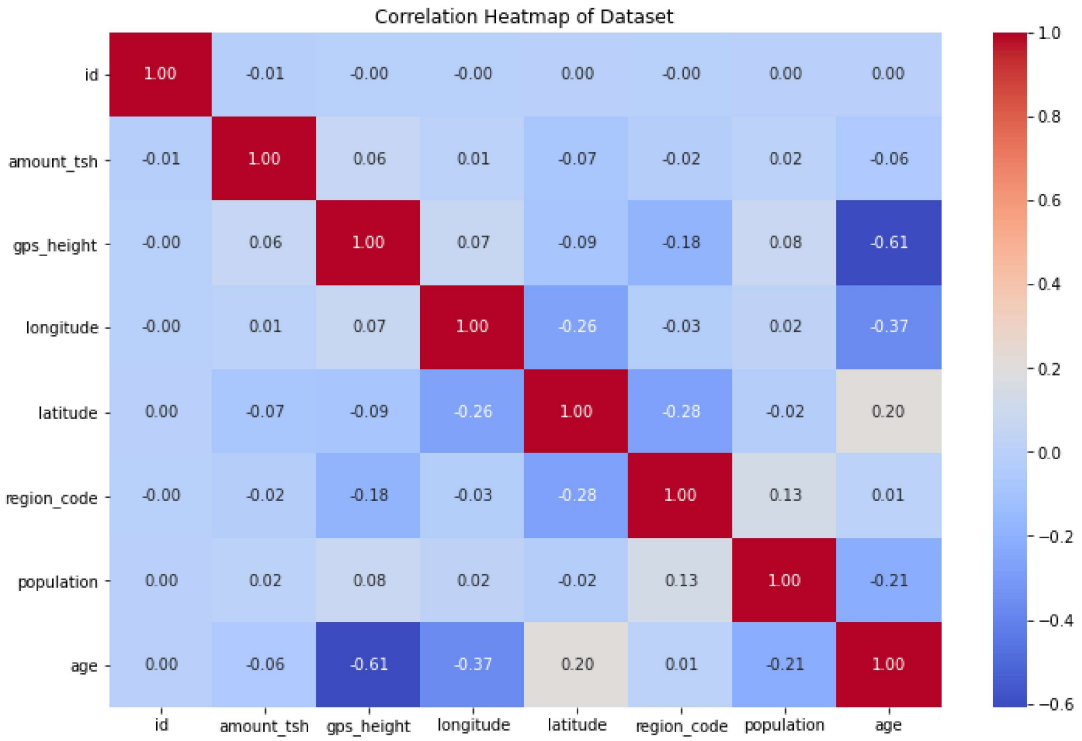
In this feature engineering step, we calculate the age of the waterpoint by subtracting the construction year from the current year.

Correlation Heatmap Analysis

To understand the relationships between different features in the dataset, we visualize the correlation heatmap.

Relevance:

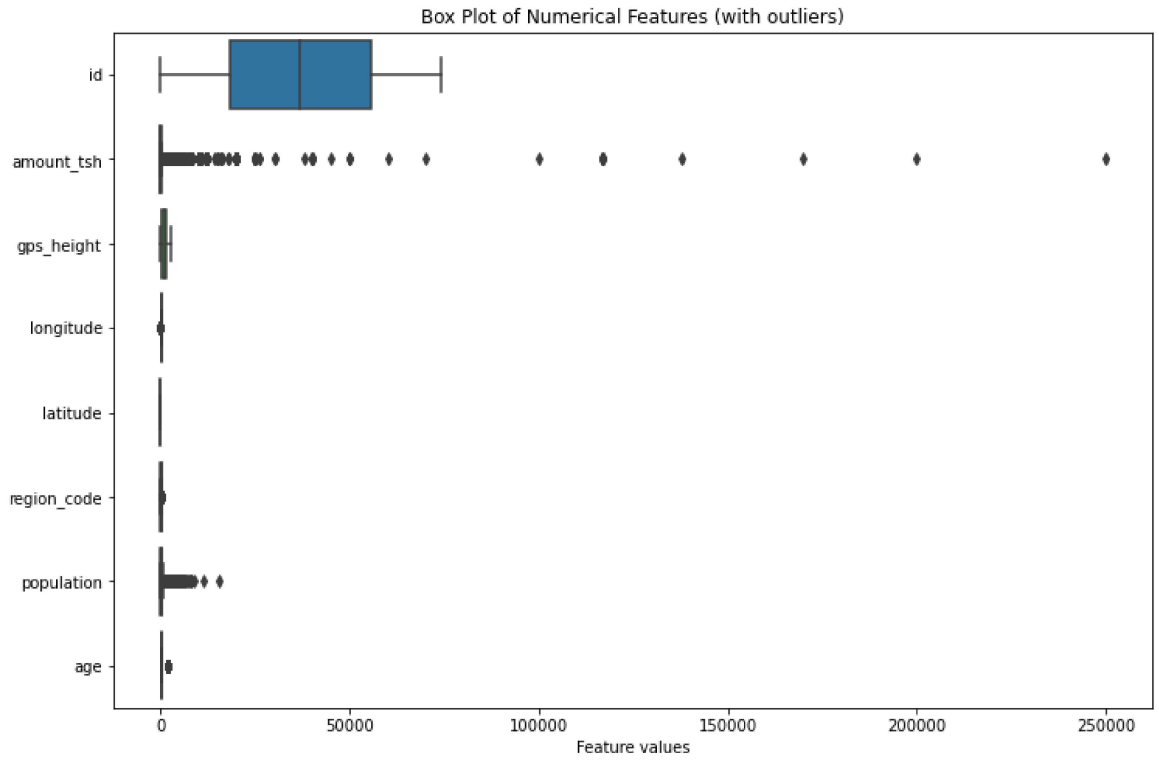
Correlation Analysis: The heatmap visualizes the correlation coefficients between all pairs of numerical features in the dataset. It helps identify patterns and relationships among variables. **Feature Selection**: High correlations (positive or negative) between features may indicate redundancy or multicollinearity, guiding feature selection or engineering efforts. This analysis aids in building more accurate predictive models by identifying relevant features. **Insight Generation**: Correlation analysis provides insights into potential relationships between features, enabling better understanding of the dataset's structure and underlying patterns.



Explaining Box Plot of Numerical Features

The resulting box plots provide insights into the distribution of each numerical feature, including central tendency, dispersion, and potential outliers.

Box plots are effective tools for visualizing the distribution of numerical data and identifying outliers, providing valuable insights into the dataset's characteristics.



Handling Categorical Columns (e.g., funder, installer, basin, etc.):

For categorical columns, we'll encode them using one-hot encoding to convert them into a format suitable for machine learning algorithms. In the provided code we'll use OneHotEncoder from scikit-learn to encode categorical columns into binary vectors.This will create binary columns for each category within each categorical feature.

Mapping Boolean Values to Numeric

This code snippet aims to replace boolean values (`True` and `False`) with numeric equivalents (`1` and `0` , respectively) for the 'permit' column in the training dataset. Here's the explanation:

Mapping boolean values to numeric representations is often useful for tasks such as machine learning, where algorithms may require numerical inputs rather than boolean ones.

Dropping 'id' Column from Training Set Labels

Dropping the 'id' column is a common preprocessing step, especially since the column doesn't contribute meaningful information to the analysis or model training process. In many datasets, 'id' columns are simply unique identifiers assigned to each data entry and are not relevant to the analysis tasks at hand. Removing such columns can simplify the dataset and improve computational efficiency.

Modelling: Train-Test Split

The purpose of this step is to prepare the data for model training and evaluation.

Splitting the dataset into training and testing subsets is essential for evaluating the performance of machine learning models. The training set is used to train the model, while the testing set is used to assess its performance on unseen data. This step helps to prevent overfitting and provides a more accurate estimation of the model's generalization ability.

Data Preprocessing: Handling Missing Values and Feature Scaling

This code snippet illustrates two essential preprocessing steps: handling missing values and scaling features. These steps are crucial for preparing the dataset before training machine learning models.

Handling Missing Values with SimpleImputer:

- The `SimpleImputer` from the `sklearn.impute` module is used to handle missing values in the dataset.
- In this example, missing values are replaced with the mean of the respective columns.

Feature Scaling with StandardScaler:

- Feature scaling ensures that all features have the same scale, preventing certain features from dominating others during model training.

Encoding Target Variable with LabelEncoder:

- For classification tasks with categorical target variables, the `LabelEncoder` from the `sklearn.preprocessing` module is used to encode class labels into numerical values.

These preprocessing steps are essential for improving the performance and interpretability of machine learning models. Handling missing values ensures that all data points are utilized in model training, while feature scaling helps in achieving better convergence and performance, especially for algorithms sensitive to feature magnitudes. Encoding the target variable enables the use of classification algorithms that require numerical labels.

Shape of Feature Matrix and Target Variable

This code snippet prints the shapes of the feature matrix `X` and the target variable `y`. Understanding the shape of these arrays is essential for ensuring the compatibility of data with machine learning models.

- **Shape of X**:
  - Indicates the dimensions of the feature matrix `X`.
  - The shape is represented as `(number of samples, number of features)`.
- **Shape of y**:
  - Indicates the dimensions of the target variable `y`.
  - The shape is represented as `(number of samples,)`.

Checking the shape of `X` and `y` is crucial for verifying that the data is properly loaded and processed, and it ensures that the dimensions align correctly for model training and evaluation.

Splitting Data into Training and Validation Sets

In machine learning, splitting the dataset into training and validation sets is crucial for model development and evaluation. Here's why it's important:

Model Training:

- The training set is used to train the machine learning model by fitting it to the training data. This involves learning the underlying patterns and relationships between features and target labels.

Model Evaluation:

- The validation set is used to evaluate the performance of the trained model. By assessing the model's performance on unseen data (validation set), we can estimate how well the model generalizes to new, unseen examples.

Preventing Overfitting:

- Splitting the data helps in detecting and preventing overfitting, where the model learns to memorize the training data rather than capturing underlying patterns. By evaluating the model on a separate validation set, we can detect overfitting and fine-tune model hyperparameters to improve generalization.

Hyperparameter Tuning:

- The validation set is also used for hyperparameter tuning, such as optimizing regularization parameters or adjusting the model complexity. This process helps improve the model's performance and generalization ability.

Assessing Performance:

- Splitting the data allows us to assess the model's performance metrics, such as accuracy, precision, recall, or F1-score, on the validation set. These metrics provide insights into how well the model performs on different aspects of classification or regression tasks.

By splitting the dataset into training and validation sets, we ensure that the machine learning model is trained effectively, evaluated accurately, and optimized for better performance and generalization on unseen data.

First Model: Logistics Regression Model

Interpretation of Results

Accuracy: The overall accuracy of the model is 0.765, indicating that it correctly predicts the class labels for approximately 76.5% of the instances in the validation set.

Classification Report:

- **Precision**:
  - Class 0: Precision of 0.78 means that among instances predicted as class 0, 78% are actually class 0.
  - Class 1: Precision of 0.37 suggests a lower precision for class 1 predictions.
  - Class 2: Precision of 0.78 indicates a relatively high precision for class 2 predictions.
- **Recall**:
  - Class 0: Recall of 0.88 implies that 88% of actual class 0 instances were correctly classified.
  - Class 1: Recall of 0.18 indicates a lower recall for class 1 predictions.
  - Class 2: Recall of 0.69 suggests a moderate recall for class 2 predictions.
- **F1-score**:
  - F1-score is the harmonic mean of precision and recall, providing a balanced metric.
  - Class 0: F1-score of 0.83 indicates a good balance between precision and recall.
  - Class 1: F1-score of 0.25 reflects a lower harmonic mean due to lower precision and recall.
  - Class 2: F1-score of 0.73 suggests a relatively balanced performance for class 2 predictions.
- **Support**:
  - The support refers to the number of instances of each class in the validation set.

Confusion Matrix:

- The confusion matrix provides a detailed breakdown of the model's predictions.
- It indicates the number of true positive, true negative, false positive, and false negative predictions for each class.
- For example, [2836 87 286] in row 1 suggests that for class 0, 2836 instances were correctly predicted, 87 instances were falsely classified as class 1, and 286 instances were falsely classified as class 2.

Convergence Warning:

- The warning indicates that the logistic regression model failed to converge, which may affect its performance.
- It suggests increasing the maximum number of iterations or scaling the data to address the issue.

Overall, these results provide insights into the performance of the model across different classes and highlight areas for potential improvement.

Interpreting these results helps in understanding the performance of the logistic regression model and identifying areas for improvement or further optimization.

Interpretation of Results

Confusion Matrix:

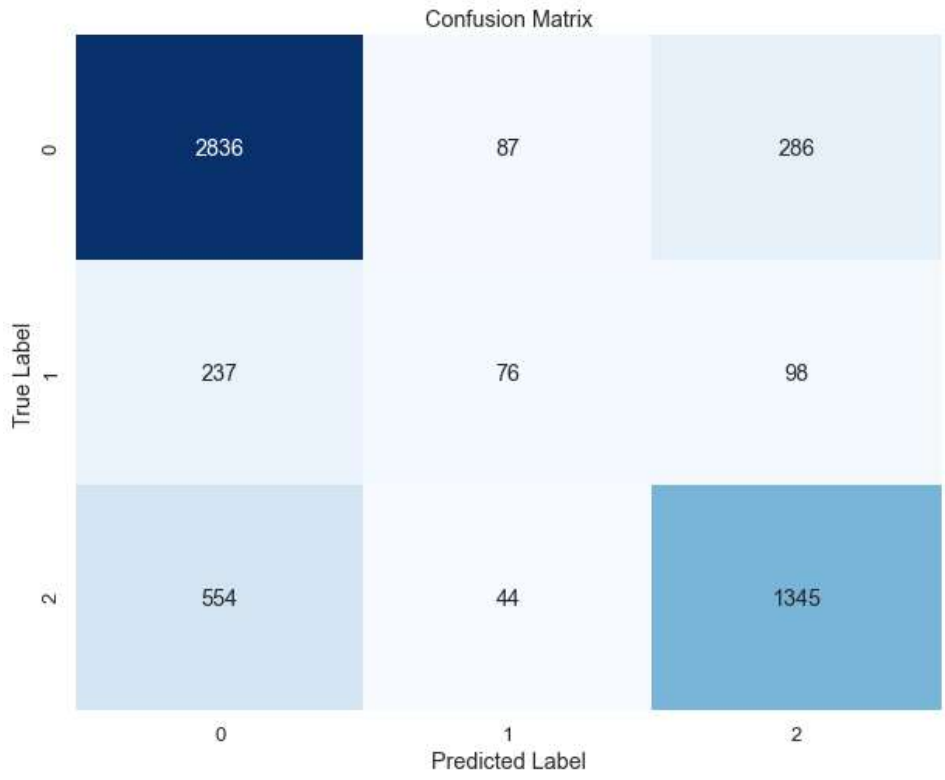
- The confusion matrix provides a visual representation of the model's performance by comparing predicted labels with true labels.
- It consists of a grid where each row represents the true class, and each column represents the predicted class.
- The values in the cells of the matrix indicate the number of instances that fall into each combination of true and predicted classes.

Heatmap Visualization:

- The heatmap visualizes the confusion matrix using color gradients to highlight different levels of performance.
- Darker shades represent higher values, indicating more instances classified into a particular combination of true and predicted classes.
- Lighter shades indicate lower values, suggesting fewer instances in those combinations.
- Annotated values within each cell provide the actual counts of instances for each combination.

Interpretation:

- By examining the confusion matrix, you can assess the model's performance across different classes.
- It allows you to identify which classes the model performs well on and which ones it struggles with.
- For instance, diagonal elements with higher values indicate correct predictions, while off-diagonal elements suggest misclassifications.
- Overall, the heatmap provides valuable insights into the strengths and weaknesses of the model's predictions.



Second Model: Logistics Regression with SMOTE

Adressing Imbalanced datasets

SMOTE (Synthetic Minority Over-sampling Technique): - Class imbalance occurs when one class has significantly fewer instances than another class, which can lead to biased models.

- SMOTE works by generating synthetic samples for the minority class to balance the class distribution in the dataset.

Resampling Training Data:

- This resampling technique helps alleviate the class imbalance issue by increasing the representation of the minority class in the training data.

Training Logistic Regression Model:

- By training on the resampled data, the model learns from a more balanced dataset, which can improve its ability to generalize to unseen data and make accurate predictions.

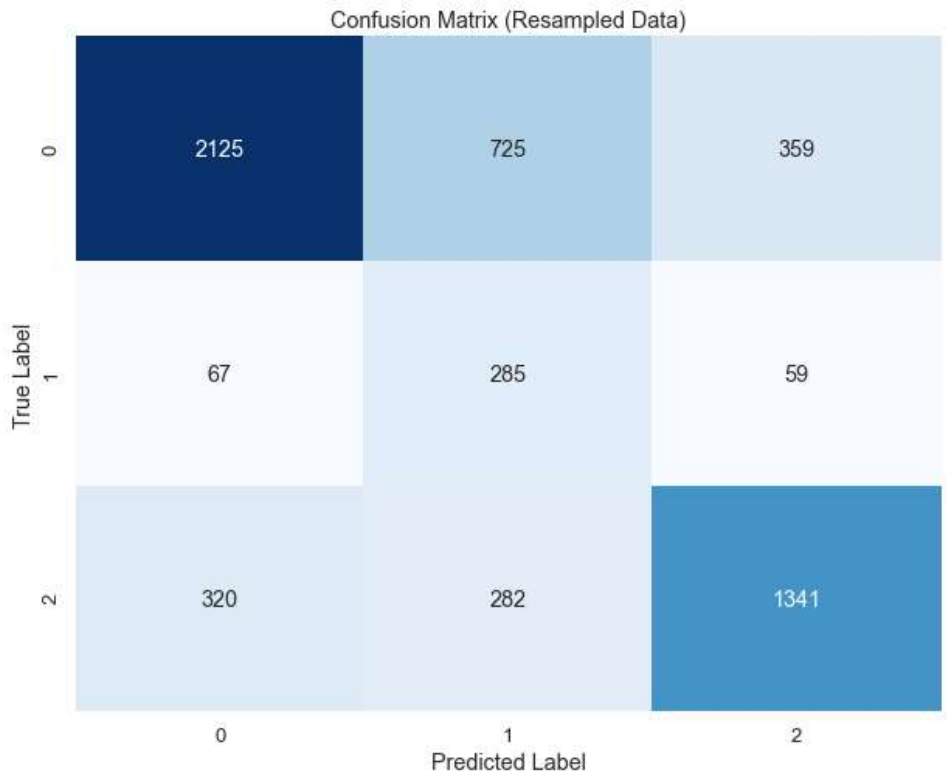
Why this is Important:

Addressing Class Imbalance:

- Class imbalance is a common problem in machine learning, especially in scenarios where one class is rare compared to others.
- By using SMOTE to oversample the minority class, we ensure that the model is exposed to sufficient examples of the minority class during training, improving its ability to recognize and classify such instances accurately. **Enhancing Model Performance**:
- Resampling techniques like SMOTE can lead to better model performance, especially in situations where class imbalance negatively impacts the model's ability to learn from the data.
- Training the logistic regression model on the resampled data allows it to capture the underlying patterns more effectively, leading to more reliable predictions on unseen data.

Model performance

- **Confusion Matrix Visualization**:
    - The confusion matrix provides insight into the performance of the model by showing the count of true positive, true negative, false positive, and false negative predictions for each class.
- **Assessing Model Performance**:
    - The confusion matrix visualization allows us to evaluate how well the logistic regression model performs on the validation data after being trained on resampled data.
    - It provides valuable information about the model's ability to correctly classify instances belonging to different classes and identify any potential misclassifications.



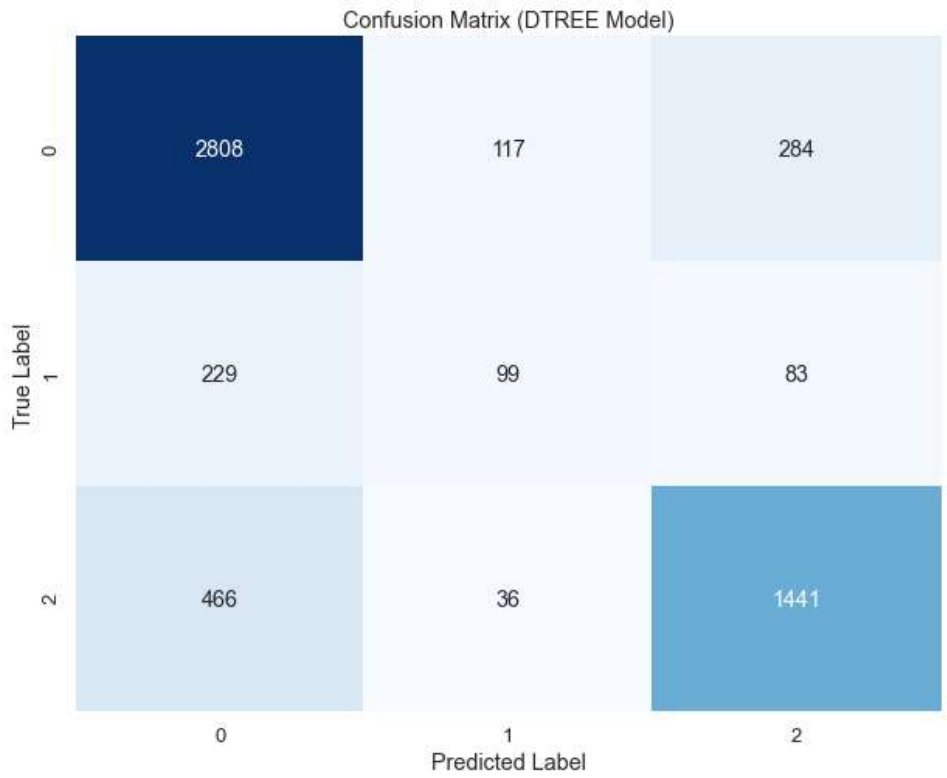
The regression model with SMOTE achieved a lower score, primarily because the size of the training dataset was reduced.

Third Model: Decision Tree

Initializing, training, and evaluating a Decision Tree classification model:

- The best parameters for the model are determined through a grid search, with the following results:
  - Max Depth: None
  - Min Samples Split: 20
  - Min Samples Leaf: 1
- The best accuracy achieved by the model is 78.48%.

This process helps in understanding the optimal configuration of the Decision Tree model and its performance on the dataset.



Fourth Model: K-Nearest Neighbors (KNN)

Initializing, training, and evaluating K-nearest neighbors (KNN) classification model

Evaluate the Model:

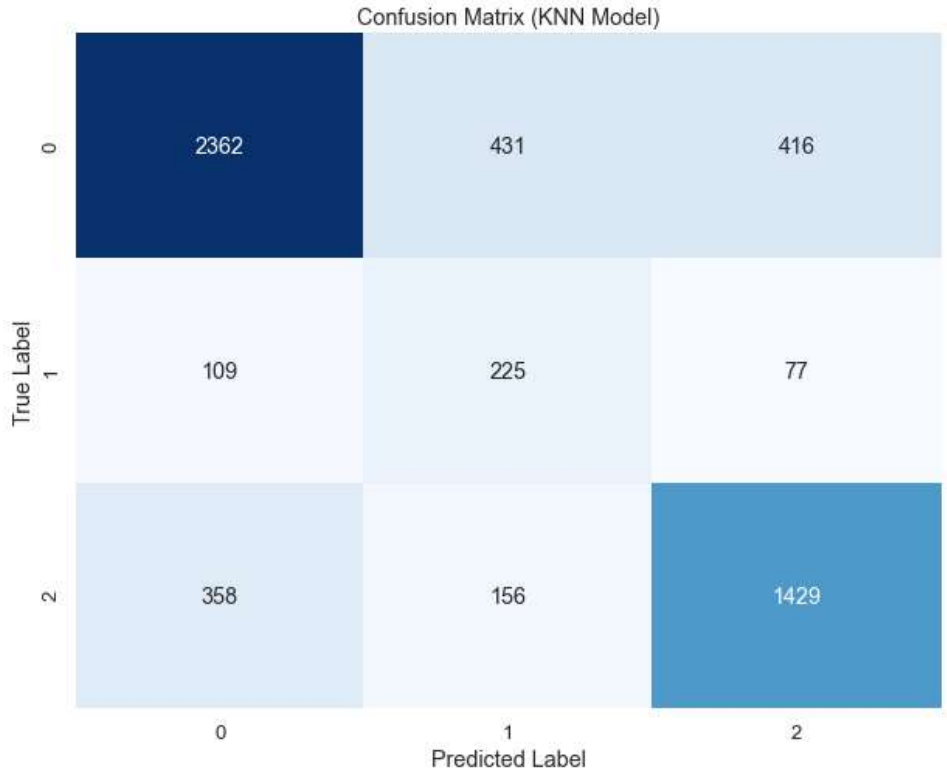
- A classification report is generated using `classification_report(y_val, y_pred_knn)`.
- The classification report provides metrics such as precision, recall, F1-score, and support for each class, offering insights into the model's performance.

This process helps assess the effectiveness of the KNN model in classifying the data.

Confusion Matrix: The confusion matrix heatmap visualizes the performance of the K-nearest neighbors (KNN) model in classifying the data. Each cell in the heatmap represents the count of data points that fall into a specific combination of true and predicted labels.

True Label (Rows): The rows of the heatmap represent the true labels of the data points. **Predicted Label (Columns)**: The columns represent the predicted labels by the KNN model. **Diagonal Cells**: These cells represent the instances where the true label matches the predicted label. Higher values along the diagonal indicate accurate predictions. **Off-diagonal Cells**: These cells represent instances of misclassification. The values in these cells indicate how many data points were incorrectly classified. **Color Gradient**: The color intensity of the cells indicates the count of data points. Darker shades typically represent higher counts.

By examining the confusion matrix heatmap, we can assess the KNN model's performance across different classes. We can identify which classes are more accurately predicted and which ones have higher misclassification rates. This information helps in understanding the strengths and weaknesses of the model and can guide further improvements or adjustments to the model parameters.



The output of the evaluation metrics provides insights into the performance of the K-nearest neighbors (KNN) model:

Accuracy: The accuracy score indicates the proportion of correctly classified instances among all instances. In this case, the KNN model achieved an accuracy of approximately 72.19%, which suggests that about 72.19% of the predictions made by the model were correct.

Precision: Precision measures the ability of the classifier not to label a negative sample as positive. It is the ratio of true positive predictions to the total number of instances predicted as positive. The weighted average precision score here is approximately 76.18%, indicating that, on average, 76.18% of the instances predicted as positive were indeed positive.

Recall: Recall, also known as sensitivity, measures the ability of the classifier to find all positive instances. It is the ratio of true positive predictions to the total number of actual positive instances. The weighted average recall score is approximately 72.19%, suggesting that the model correctly identified about 72.19% of the actual positive instances.

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is a single metric that combines both precision and recall. The weighted average F1-score is approximately 73.68%, indicating a balanced performance between precision and recall.

The output of the GridSearchCV process reveals the best parameters and the corresponding accuracy score achieved by the K-nearest neighbors (KNN) classifier:

Best Parameters: The best combination of hyperparameters found by the grid search consists of using a Manhattan distance metric (`metric='manhattan'`), considering 3 nearest neighbors (`n_neighbors=3`), and applying distance-based weights for predictions (`weights='distance'`).

Best Score (Accuracy): The accuracy score associated with the best parameters is approximately 83.67%. This indicates that the KNN classifier achieved an accuracy of around 83.67% on the training data when using the optimal combination of hyperparameters identified by the grid search.

Fifth Model: KNN with GridSearchCV

The output of the GridSearchCV process for the Random Forest classifier reveals the following results:

Best Parameters: The best combination of hyperparameters found by the grid search includes:

- `max_depth` : None
- `max_features` : 'auto'
- `min_samples_leaf` : 1
- `min_samples_split` : 2
- `n_estimators` : 200 These parameters indicate that the best-performing Random Forest model was built with 200 estimators (trees), no limit on the maximum depth of each tree (`max_depth=None`), and other settings for controlling node splitting and leaf conditions.

Best Score (Accuracy): The accuracy score associated with the best parameters is approximately 87.95%. This indicates that the Random Forest classifier achieved an accuracy of around 87.95% on the training data when using the optimal combination of hyperparameters identified by the grid search.

Accuracy: The accuracy achieved by the Random Forest model on the test set is approximately 88.62%. This suggests that the model generalized well to unseen data.

Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's performance across different classes. It shows the number of true positives, true negatives, false positives, and false negatives for each class. In this case, the confusion matrix reveals that the model correctly classified a majority of instances across all classes, with some misclassifications.

Sixth Model: Random Forest with GridserachCV

The output of the evaluation metrics for the Random Forest classifier on the test set

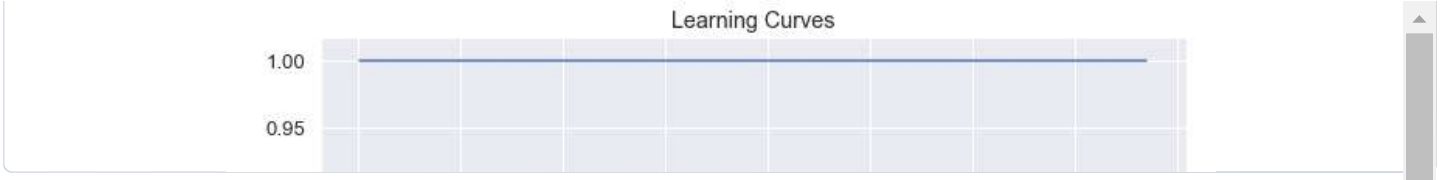
Accuracy: The accuracy of the model is approximately 88.29%. This indicates the proportion of correctly classified instances out of all instances in the test set.

Precision: The precision of the model is approximately 80.43%. Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It provides insight into the model's ability to avoid false positives.

Recall: The recall of the model is approximately 80.38%. Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual positive instances in the test set. It indicates the model's ability to capture all positive instances.

F1-score: The F1-score of the model is approximately 80.40%. The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when the class distribution is imbalanced.

Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's predictions across different classes. It shows the number of true positives, true negatives, false positives, and false negatives for each class. In this case, the confusion matrix reveals that the model correctly classified a majority of instances across all classes, with some misclassifications.



Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%