Phase 3 Project

1. Project Setup

- Import necessary libraries (pandas, numpy, sklearn, matplotlib, seaborn)
- Load the dataset
- Set random seed for reproducibility

2. Data Exploration and Preprocessing

- Examine the dataset (head, info, describe)
- Check for missing values and handle them
- Explore data distributions and correlations
- Perform feature engineering if necessary
- Split the data into features (X) and target variable(s) (y)

3. Logistic Regression

- Prepare the data (ensure binary target variable)
- Create and train the model
- Make predictions on the test set
- Evaluate the model (accuracy, precision, recall, F1-score)
- Plot the ROC curve and calculate AUC
- Analyze coefficients and their significance

4. Conclusion and Future Work

- Summarize key findings
- Discuss limitations of the current approach
- Suggest potential improvements or additional models to try

Student details

Nduku Kitenge DSC-PT07

Business Problem

The goal of this project is to predict which pumps are functional, which need some repairs and which dont work at all.

The challenge from DrivenData. (2015). Pump it Up: Data Mining the Water Table. Retrieved [Month Day Year] from https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table with data from Taarifa and the Tanzanian Ministry of Water. The goal of this project is to predict one of these three classes based on a number of variables about what kind of pump is operating, when it was installed, and how it is managed. A smart understanding of which waterpoints will fail can improve maintenance operations and ensure that clean, potable water is available to communities across Tanzania.

1. Project Setup

- Import necessary libraries (pandas, numpy, sklearn, matplotlib, seaborn)
- Load the dataset

In [ ]:
```
#import neccessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [ ]:
```
#Load training dataset
sub_df = pd.read_csv('./SubmissionFormat.csv',index_col=0)
training_df = pd.read_csv('./TrainingSetValues.csv',index_col=0)
test_df = pd.read_csv('./TestSetValues.csv',index_col=0)
t_label_df = pd.read_csv('TrainingSetLabels.csv',index_col=0)
```

2. Data Exploration and Preprocessing

- Examine the dataset (head, info, describe)
- Check for missing values and handle them
- Explore data distributions and correlations
- Perform feature engineering eg. add pump age
- Split the data into features (X) and target variable(s) (y)

In [ ]:
```
# Function to display dataset info
def display_dataset_info(df, name):
    print(f"\n=== {name} ===")
    print(f"Shape: {df.shape}")
    print("\nInfo:")
    print(df.info())
    print("\nDescription:")
    print(df.describe())
    print("\nHead:")
    print(df.head())
    print("\n" + "="*40)

# Display info for each dataset
```

```
display_dataset_info(sub_df, "Submission Format")
display_dataset_info(training_df, "Training Set Values")
display_dataset_info(test_df, "Test Set Values")
display_dataset_info(t_label_df, "Training Set Labels")
```

```
=== Submission Format ===
Shape: (14850, 1)

Info:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14850 entries, 50785 to 68707
Data columns (total 1 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   status_group  14850 non-null  object
dtypes: object(1)
memory usage: 232.0+ KB
None

Description:
           status_group
count             14850
unique                1
top     predicted label
freq              14850

Head:
          status_group
id                    
50785  predicted label
51630  predicted label
17168  predicted label
45559  predicted label
49871  predicted label

========================================

=== Training Set Values ===
Shape: (59400, 39)

Info:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 69572 to 26348
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   amount_tsh             59400 non-null  float64
 1   date_recorded          59400 non-null  object 
 2   funder                 55765 non-null  object 
 3   gps_height             59400 non-null  int64  
 4   installer              55745 non-null  object 
 5   longitude              59400 non-null  float64
 6   latitude               59400 non-null  float64
 7   wpt_name               59400 non-null  object 
 8   num_private            59400 non-null  int64  
 9   basin                  59400 non-null  object 
 10  subvillage             59029 non-null  object 
 11  region                 59400 non-null  object 
 12  region_code            59400 non-null  int64  
 13  district_code          59400 non-null  int64  
 14  lga                    59400 non-null  object 
 15  ward                   59400 non-null  object 
 16  population             59400 non-null  int64  
 17  public_meeting         56066 non-null  object 
 18  recorded_by            59400 non-null  object 
```

```
 19  scheme_management      55523 non-null  object
 20  scheme_name            31234 non-null  object
 21  permit                 56344 non-null  object
 22  construction_year      59400 non-null  int64
 23  extraction_type        59400 non-null  object
 24  extraction_type_group  59400 non-null  object
 25  extraction_type_class  59400 non-null  object
 26  management             59400 non-null  object
 27  management_group       59400 non-null  object
 28  payment                59400 non-null  object
 29  payment_type           59400 non-null  object
 30  water_quality          59400 non-null  object
 31  quality_group          59400 non-null  object
 32  quantity               59400 non-null  object
 33  quantity_group         59400 non-null  object
 34  source                 59400 non-null  object
 35  source_type            59400 non-null  object
 36  source_class           59400 non-null  object
 37  waterpoint_type        59400 non-null  object
 38  waterpoint_type_group  59400 non-null  object
dtypes: float64(3), int64(6), object(30)
memory usage: 18.1+ MB
None

Description:
          amount_tsh    gps_height     longitude      latitude   num_private  \
count   59400.000000  59400.000000  59400.000000  5.940000e+04  59400.000000
mean      317.650385    668.297239     34.077427 -5.706033e+00      0.474141
std      2997.574558    693.116350      6.567432  2.946019e+00     12.236230
min         0.000000    -90.000000      0.000000 -1.164944e+01      0.000000
25%         0.000000      0.000000     33.090347 -8.540621e+00      0.000000
50%         0.000000    369.000000     34.908743 -5.021597e+00      0.000000
75%        20.000000   1319.250000     37.178387 -3.326156e+00      0.000000
max    350000.000000   2770.000000     40.345193 -2.000000e-08   1776.000000

        region_code  district_code    population  construction_year
count  59400.000000   59400.000000  59400.000000       59400.000000
mean      15.297003       5.629747    179.909983        1300.652475
std       17.587406       9.633649    471.482176         951.620547
min        1.000000       0.000000      0.000000           0.000000
25%        5.000000       2.000000      0.000000           0.000000
50%       12.000000       3.000000     25.000000        1986.000000
75%       17.000000       5.000000    215.000000        2004.000000
max       99.000000      80.000000  30500.000000        2013.000000

Head:
       amount_tsh date_recorded        funder  gps_height     installer  \
id
69572      6000.0    2011-03-14         Roman        1390         Roman
8776          0.0    2013-03-06       Grumeti        1399       GRUMETI
34310        25.0    2013-02-25  Lottery Club         686  World vision
67743         0.0    2013-01-28        Unicef         263        UNICEF
19728         0.0    2011-07-13   Action In A           0       Artisan

       longitude   latitude              wpt_name  num_private  \
id
69572  34.938093  -9.856322                  none            0
8776   34.698766  -2.147466               Zahanati            0
34310  37.460664  -3.821329           Kwa Mahundi            0
67743  38.486161 -11.155298  Zahanati Ya Nanyumbu            0
19728  31.130847  -1.825359                Shuleni            0

                        basin  ... payment_type water_quality  quality_group  \
id                             ...
69572               Lake Nyasa  ...     annually          soft           good
```

```
8776             Lake Victoria  ...    never pay          soft           good
34310                 Pangani  ...   per bucket          soft           good
67743  Ruvuma / Southern Coast  ...    never pay          soft           good
19728            Lake Victoria  ...    never pay          soft           good

           quantity quantity_group                source  \
id
69572        enough         enough                spring
8776   insufficient   insufficient  rainwater harvesting
34310        enough         enough                   dam
67743           dry            dry           machine dbh
19728      seasonal       seasonal  rainwater harvesting

                source_type source_class              waterpoint_type  \
id
69572                 spring  groundwater           communal standpipe
8776   rainwater harvesting      surface           communal standpipe
34310                   dam      surface  communal standpipe multiple
67743              borehole  groundwater  communal standpipe multiple
19728  rainwater harvesting      surface           communal standpipe

      waterpoint_type_group
id
69572    communal standpipe
8776     communal standpipe
34310    communal standpipe
67743    communal standpipe
19728    communal standpipe

[5 rows x 39 columns]

========================================

=== Test Set Values ===
Shape: (14850, 39)

Info:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14850 entries, 50785 to 68707
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   amount_tsh             14850 non-null  float64
 1   date_recorded          14850 non-null  object
 2   funder                 13981 non-null  object
 3   gps_height             14850 non-null  int64
 4   installer              13973 non-null  object
 5   longitude              14850 non-null  float64
 6   latitude               14850 non-null  float64
 7   wpt_name               14850 non-null  object
 8   num_private            14850 non-null  int64
 9   basin                  14850 non-null  object
 10  subvillage             14751 non-null  object
 11  region                 14850 non-null  object
 12  region_code            14850 non-null  int64
 13  district_code          14850 non-null  int64
 14  lga                    14850 non-null  object
 15  ward                   14850 non-null  object
 16  population             14850 non-null  int64
 17  public_meeting         14029 non-null  object
 18  recorded_by            14850 non-null  object
 19  scheme_management      13881 non-null  object
 20  scheme_name            7758 non-null   object
 21  permit                 14113 non-null  object
 22  construction_year      14850 non-null  int64
```

```
 23  extraction_type        14850 non-null  object
 24  extraction_type_group  14850 non-null  object
 25  extraction_type_class  14850 non-null  object
 26  management             14850 non-null  object
 27  management_group       14850 non-null  object
 28  payment                14850 non-null  object
 29  payment_type           14850 non-null  object
 30  water_quality          14850 non-null  object
 31  quality_group          14850 non-null  object
 32  quantity               14850 non-null  object
 33  quantity_group         14850 non-null  object
 34  source                 14850 non-null  object
 35  source_type            14850 non-null  object
 36  source_class           14850 non-null  object
 37  waterpoint_type        14850 non-null  object
 38  waterpoint_type_group  14850 non-null  object
dtypes: float64(3), int64(6), object(30)
memory usage: 4.5+ MB
None

Description:
          amount_tsh    gps_height     longitude      latitude   num_private  \
count   14850.000000  14850.000000  14850.000000  1.485000e+04  14850.000000
mean      322.826983    655.147609     34.061605 -5.684724e+00      0.415084
std      2510.968644    691.261185      6.593034  2.940803e+00      8.167910
min         0.000000    -57.000000      0.000000 -1.156459e+01      0.000000
25%         0.000000      0.000000     33.069455 -8.443970e+00      0.000000
50%         0.000000    344.000000     34.901215 -5.049750e+00      0.000000
75%        25.000000   1308.000000     37.196594 -3.320594e+00      0.000000
max    200000.000000   2777.000000     40.325016 -2.000000e-08    669.000000

        region_code  district_code    population  construction_year
count  14850.000000   14850.000000  14850.000000       14850.000000
mean      15.139057       5.626397    184.114209        1289.708350
std       17.191329       9.673842    469.499332         955.241087
min        1.000000       0.000000      0.000000           0.000000
25%        5.000000       2.000000      0.000000           0.000000
50%       12.000000       3.000000     20.000000        1986.000000
75%       17.000000       5.000000    220.000000        2004.000000
max       99.000000      80.000000  11469.000000        2013.000000

Head:
       amount_tsh date_recorded                 funder  gps_height  \
id
50785         0.0    2013-02-04                   Dmdd        1996
51630         0.0    2013-02-04  Government Of Tanzania        1569
17168         0.0    2013-02-01                    NaN        1567
45559         0.0    2013-01-22            Finn Water         267
49871       500.0    2013-03-27                 Bruder        1260

        installer  longitude   latitude                wpt_name  num_private  \
id
50785        DMDD  35.290799  -4.059696  Dinamu Secondary School            0
51630         DWE  36.656709  -3.309214                 Kimnyak            0
17168         NaN  34.767863  -5.004344          Puma Secondary            0
45559  FINN WATER  38.058046  -9.418672          Kwa Mzee Pange            0
49871      BRUDER  35.006123 -10.950412          Kwa Mzee Turuka            0

                        basin  ... payment_type water_quality  quality_group  \
id                             ...
50785                Internal  ...    never pay          soft           good
51630                 Pangani  ...    never pay          soft           good
17168                Internal  ...    never pay          soft           good
45559  Ruvuma / Southern Coast  ...      unknown          soft           good
49871  Ruvuma / Southern Coast  ...      monthly          soft           good
```

```
           quantity quantity_group               source  \
id
50785      seasonal       seasonal  rainwater harvesting
51630  insufficient   insufficient               spring
17168  insufficient   insufficient  rainwater harvesting
45559           dry            dry          shallow well
49871        enough         enough               spring

               source_type source_class     waterpoint_type  \
id
50785  rainwater harvesting      surface               other
51630               spring  groundwater  communal standpipe
17168  rainwater harvesting      surface               other
45559          shallow well  groundwater               other
49871               spring  groundwater  communal standpipe

      waterpoint_type_group
id
50785                 other
51630    communal standpipe
17168                 other
45559                 other
49871    communal standpipe

[5 rows x 39 columns]

========================================

=== Training Set Labels ===
Shape: (59400, 1)

Info:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 69572 to 26348
Data columns (total 1 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   status_group  59400 non-null  object
dtypes: object(1)
memory usage: 928.1+ KB
None

Description:
       status_group
count         59400
unique            3
top      functional
freq          32259

Head:
         status_group
id
69572      functional
8776       functional
34310      functional
67743  non functional
19728      functional

========================================
```

In [ ]:
```
#check for missing values in training dataset
training_df.isna().sum()
```

Out[ ]:
```
amount_tsh                 0
date_recorded              0
funder                  3635
gps_height                 0
installer               3655
longitude                  0
latitude                   0
wpt_name                   0
num_private                0
basin                      0
subvillage               371
region                     0
region_code                0
district_code              0
lga                        0
ward                       0
population                 0
public_meeting          3334
recorded_by                0
scheme_management       3877
scheme_name            28166
permit                  3056
construction_year          0
extraction_type            0
extraction_type_group      0
extraction_type_class      0
management                 0
management_group           0
payment                    0
payment_type               0
water_quality              0
quality_group              0
quantity                   0
quantity_group             0
source                     0
source_type                0
source_class               0
waterpoint_type            0
waterpoint_type_group      0
dtype: int64
```

Dropping columns and reasons why:

- Scheme name - Too many null values
- wpt_name - Too many unique values
- management - management group covers this
- quality group - quality covers this
- quantity group
- extracton type group

Numerical columns:

- num private - dont know what this means.

Observations

- Funder and installer are similar, but funder has less null values than installer.
- Waterpoint type and water point are similar - Will drop waterpoint type.

In [ ]:
```
#Check categorical columns
training_df.select_dtypes(include=['object']).columns
```

Out[ ]:
```
Index(['date_recorded', 'funder', 'installer', 'wpt_name', 'basin',
       'subvillage', 'region', 'lga', 'ward', 'public_meeting', 'recorded_by',
       'scheme_management', 'scheme_name', 'permit', 'extraction_type',
       'extraction_type_group', 'extraction_type_class', 'management',
       'management_group', 'payment', 'payment_type', 'water_quality',
       'quality_group', 'quantity', 'quantity_group', 'source', 'source_type',
       'source_class', 'waterpoint_type', 'waterpoint_type_group'],
      dtype='object')
```

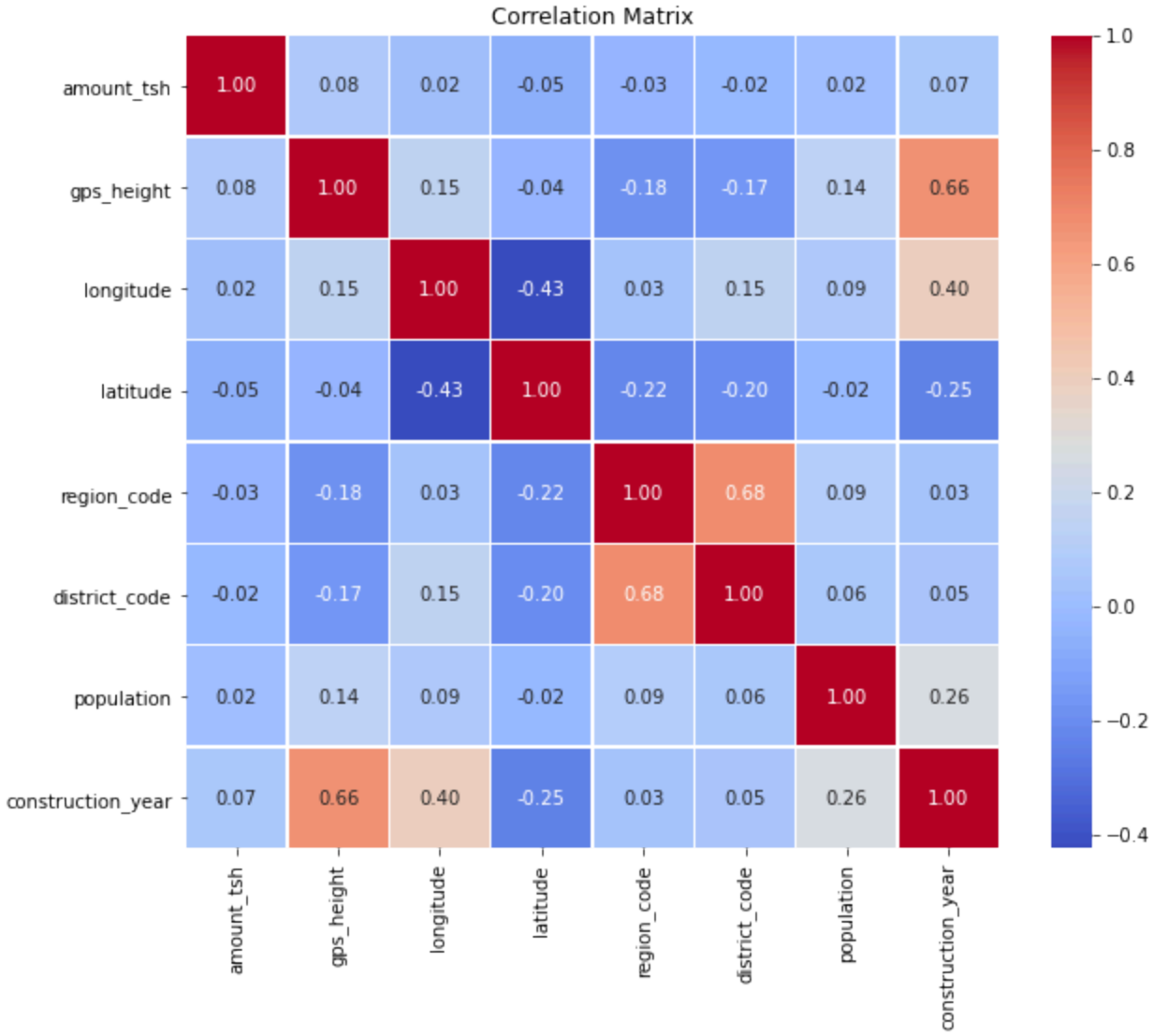
In [ ]:
```
#check numerical columns
training_df.select_dtypes(include=['int64', 'float64']).columns
```

Out[ ]:
```
Index(['amount_tsh', 'gps_height', 'longitude', 'latitude', 'num_private',
       'region_code', 'district_code', 'population', 'construction_year'],
      dtype='object')
```

In [ ]:
```
#Define the list of columns to drop
columns_drop = ['wpt_name','lga','ward','quality_group','extraction_type_class','manage

# Drop unnecessary columns
training_df = training_df.drop(columns_drop,axis=1)

#Print remaining columns
print(training_df.columns)
```

```
Index(['amount_tsh', 'date_recorded', 'funder', 'gps_height', 'installer',
       'longitude', 'latitude', 'basin', 'subvillage', 'region', 'region_code',
       'district_code', 'population', 'public_meeting', 'recorded_by',
       'scheme_management', 'scheme_name', 'permit', 'construction_year',
       'extraction_type', 'extraction_type_group', 'management_group',
       'payment', 'payment_type', 'water_quality', 'quantity',
       'quantity_group', 'source_type', 'source_class', 'waterpoint_type'],
      dtype='object')
```

In [ ]:
```
#explore data distrubitions and correlations of the numerical- training data
corr_matrix = training_df.corr()

# Plot the correlation matrix using Seaborn for numerical values.
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm', square=True, linewidth
plt.title('Correlation Matrix')
plt.show()
```



In [ ]:
```
# merge with training labels data using ID as connector
merged_df = pd.merge(training_df, t_label_df, on='id', how='outer', indicator=True)
print(merged_df.head(20))
```

```
       amount_tsh date_recorded                  funder  gps_height  \
id
69572      6000.0    2011-03-14                   Roman        1390
8776          0.0    2013-03-06                 Grumeti        1399
34310        25.0    2013-02-25             Lottery Club         686
67743         0.0    2013-01-28                  Unicef         263
19728         0.0    2011-07-13             Action In A           0
9944         20.0    2011-03-13      Mkinga Distric Coun           0
19816         0.0    2012-10-01                    Dwsp           0
54551         0.0    2012-10-09                   Rwssp           0
53934         0.0    2012-11-03                Wateraid           0
46144         0.0    2011-08-03             Isingiro Ho           0
49056         0.0    2011-02-20                 Private          62
50409       200.0    2013-02-18                  Danida        1062
36957         0.0    2012-10-14           World Vision           0
50495         0.0    2013-03-15  Lawatefuka Water Supply        1368
53752         0.0    2012-10-20                   Biore           0
61848         0.0    2011-08-04                   Rudep        1645
48451       500.0    2011-07-04                  Unicef        1703
58155         0.0    2011-09-04                  Unicef        1656
34169         0.0    2011-07-22                  Hesawa        1162
18274       500.0    2011-02-22                  Danida        1763
```

id	installer	longitude	latitude	basin	subvillage	region	...	payment	payment_type	water_quality	quantity	quantity_group	source_type
69572	Roman	34.938093	-9.856322	Lake Nyasa	Mnyusi B	Iringa	...	pay annually	annually	soft	enough	enough	spring
8776	GRUMETI	34.698766	-2.147466	Lake Victoria	Nyamara	Mara	...	never pay	never pay	soft	insufficient	insufficient	rainwater harvesting
34310	World vision	37.460664	-3.821329	Pangani	Majengo	Manyara	...	pay per bucket	per bucket	soft	enough	enough	dam
67743	UNICEF	38.486161	-11.155298	Ruvuma / Southern Coast	Mahakamani	Mtwara	...	never pay	never pay	soft	dry	dry	borehole
19728	Artisan	31.130847	-1.825359	Lake Victoria	Kyanyamisa	Kagera	...	never pay	never pay	soft	seasonal	seasonal	rainwater harvesting
9944	DWE	39.172796	-4.765587	Pangani	Moa/Mwereme	Tanga	...	pay per bucket	per bucket	salty	enough	enough	other
19816	DWSP	33.362410	-3.766365	Internal	Ishinabulandi	Shinyanga	...	never pay	never pay	soft	enough	enough	borehole
54551	DWE	32.620617	-4.226198	Lake Tanganyika	Nyawishi Center	Shinyanga	...	unknown	unknown	milky	enough	enough	shallow well
53934	Water Aid	32.711100	-5.146712	Lake Tanganyika	Imalauduki	Tabora	...	never pay	never pay	salty	seasonal	seasonal	borehole
46144	Artisan	30.626991	-1.257051	Lake Victoria	Mkonomre	Kagera	...	never pay	never pay	soft	enough	enough	shallow well
49056	Private	39.209518	-7.034139	Wami / Ruvu	Mizugo	Pwani	...	never pay	never pay	salty	enough	enough	borehole
50409	DANIDA	35.770258	-10.574175	Lake Nyasa	Ngondombwito	Ruvuma	...	pay when scheme fails	on failure	soft	insufficient	insufficient	shallow well
36957	World vision	33.798106	-3.290194	Internal	Nkilifa	Shinyanga	...	other	other	soft	enough	enough	shallow well
50495	Lawatefuka water sup	37.092574	-3.181783	Pangani	Omarini	Kilimanjaro	...	pay monthly	monthly	soft	enough	enough	spring
53752	WEDECO	34.364073	-3.629333	Internal	Mwabasabi	Shinyanga	...	never pay	never pay	soft	enough	enough	shallow well
61848	DWE	31.444121	-8.274962	Lake Tanganyika	Tunzi	Rukwa	...	never pay	never pay	soft	enough	enough	borehole
48451	DWE	34.642439	-9.106185	Rufiji	Kidudumo	Iringa	...	pay monthly	monthly	soft	enough	enough	borehole
58155	DWE	34.569266	-9.085515	Rufiji	Yeriko	Iringa	...	pay when scheme fails	on failure	soft	enough	enough	borehole
34169	DWE	32.920154	-1.947868	Lake Victoria	Center	Mwanza	...	never pay	never pay	soft	enough	enough	borehole
18274	Danid	34.508967	-9.894412	Lake Nyasa	Manyanya	Iringa	...	pay annually	annually	soft	enough	enough	borehole

```
48451           soft           dry            dry            river/lake
58155           soft           dry            dry            river/lake
34169          milky  insufficient   insufficient                spring
18274           soft        enough         enough                spring

      source_class              waterpoint_type             status_group  \
id
69572  groundwater           communal standpipe               functional
8776       surface           communal standpipe               functional
34310      surface  communal standpipe multiple               functional
67743  groundwater  communal standpipe multiple           non functional
19728      surface           communal standpipe               functional
9944       unknown  communal standpipe multiple               functional
19816  groundwater                   hand pump           non functional
54551  groundwater                   hand pump           non functional
53934  groundwater                   hand pump           non functional
46144  groundwater                   hand pump               functional
49056  groundwater                       other               functional
50409  groundwater                   hand pump               functional
36957  groundwater                   hand pump               functional
50495  groundwater           communal standpipe               functional
53752  groundwater                   hand pump               functional
61848  groundwater                   hand pump               functional
48451      surface           communal standpipe           non functional
58155      surface           communal standpipe           non functional
34169  groundwater                       other  functional needs repair
18274  groundwater           communal standpipe               functional

      _merge
id
69572   both
8776    both
34310   both
67743   both
19728   both
9944    both
19816   both
54551   both
53934   both
46144   both
49056   both
50409   both
36957   both
50495   both
53752   both
61848   both
48451   both
58155   both
34169   both
18274   both

[20 rows x 32 columns]
```

In [ ]:
```
#check for missing values in merged dataset
merged_df.isna().sum()
```

Out[ ]:
```
amount_tsh                0
date_recorded             0
funder                 3635
gps_height                0
installer              3655
longitude                 0
latitude                  0
basin                     0
subvillage              371
```

```
region                     0
region_code                0
district_code              0
population                 0
public_meeting          3334
recorded_by                0
scheme_management       3877
scheme_name            28166
permit                  3056
construction_year          0
extraction_type            0
extraction_type_group      0
management_group           0
payment                    0
payment_type               0
water_quality              0
quantity                   0
quantity_group             0
source_type                0
source_class               0
waterpoint_type            0
status_group               0
_merge                     0
dtype: int64
```

In [ ]:
```
#Treat null values
missing_value_columns = ['funder', 'installer', 'subvillage', 'public_meeting','scheme_

# Check the value counts
for col in missing_value_columns:
    print(merged_df[col].value_counts())
```

```
Government Of Tanzania           9084
Danida                           3114
Hesawa                           2202
Rwssp                            1374
World Bank                       1349
                                 ...
Tanzania Egypt Technical Co Op      1
Resolute Mininggolden Pride         1
Mbozi Hospital                      1
Isf / Tasaff                        1
Dv                                  1
Name: funder, Length: 1897, dtype: int64
DWE              17402
Government        1825
RWE               1206
Commu             1060
DANIDA            1050
                 ...
Safari Roya          1
REGWA                1
Mwakabalula          1
Mamaz                1
Prof. Saluati        1
Name: installer, Length: 2145, dtype: int64
Madukani       508
Shuleni        506
Majengo        502
Kati           373
Mtakuja        262
              ...
Shigangama       1
Mpimba           1
Nyakichacha      1
```

```
Ilyema           1
Itembe           1
Name: subvillage, Length: 19287, dtype: int64
True     51011
False     5055
Name: public_meeting, dtype: int64
K                               682
None                            644
Borehole                        546
Chalinze wate                   405
M                               400
                               ...
Sola                              1
Singino                           1
Igongolo gravity water sch        1
Shimassawe branch line            1
Imbaseny gravity water supply     1
Name: scheme_name, Length: 2696, dtype: int64
VWC                  36793
WUG                   5206
Water authority       3153
WUA                   2883
Water Board           2748
Parastatal            1680
Private operator      1063
Company               1061
Other                  766
SWC                     97
Trust                   72
None                     1
Name: scheme_management, dtype: int64
True     38852
False    17492
Name: permit, dtype: int64
```

In [ ]:
```
# Remove rows with missing values in 'funder', 'installer' and 'scheme_management' colu
merged_df.dropna(subset=['funder','installer', 'scheme_management'], axis=0, inplace=Tr
```

Replacing the missing values for public meeting and permit with False. Assuming that the information doesnt exist.

In [ ]:
```
# Fill missing values in public meeting and permit'
for col in ['public_meeting', 'permit']:
    merged_df[col] = merged_df[col].fillna(False)

#Fill in missing values in 'scheme_name', subvillage
for col2 in ['scheme_name', 'subvillage']:
    merged_df[col2] = merged_df[col2].fillna('None')
```

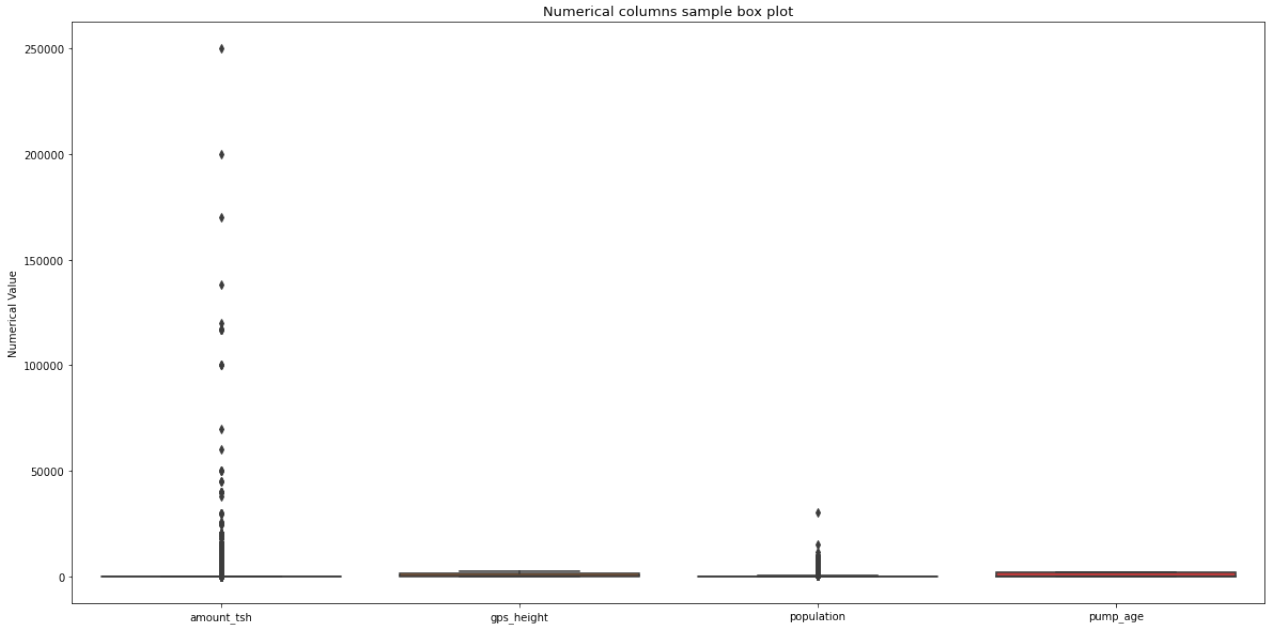
In [ ]:
```
# Confirm there are no more missing values
merged_df.isna().sum()
```

Out[ ]:
```
amount_tsh              0
date_recorded           0
funder                  0
gps_height              0
installer               0
longitude               0
latitude                0
basin                   0
subvillage              0
region                  0
```

```
region_code              0
district_code            0
population               0
public_meeting           0
recorded_by              0
scheme_management        0
scheme_name              0
permit                   0
construction_year        0
extraction_type          0
extraction_type_group    0
management_group         0
payment                  0
payment_type             0
water_quality            0
quantity                 0
quantity_group           0
source_type              0
source_class             0
waterpoint_type          0
status_group             0
_merge                   0
dtype: int64
```

In [ ]:
```
# Ensure the 'installation_year' is in numeric format (not datetime, just the year)
merged_df['construction_year'] = pd.to_numeric(training_df['construction_year'], errors

# Calculate pump age
current_year = pd.Timestamp.now().year
merged_df['pump_age'] = current_year - merged_df['construction_year']

# Handle cases where installation_year might be missing or incorrect (e.g., 0 or negati
merged_df['pump_age'] = merged_df['pump_age'].apply(lambda x: x if x > 0 else None)
```

In [ ]:
```
# Plotting box plots of some numerical columns
columns = ['amount_tsh', 'gps_height', 'population','pump_age']
plt.figure(figsize=(20, 10))
sns.boxplot(data=[merged_df[col] for col in columns])
plt.title("Numerical columns sample box plot", fontsize=13)
plt.ylabel("Numerical Value")
plt.xticks(range(0,4), columns)
plt.show()
```



In [ ]:
```
# Check whether there are duplicates
merged_df.duplicated(keep = 'first').sum()
```

Out[ ]:
```
129
```

In [ ]:
```
# Change the data type of public_meeting and permit columns to binary for classificatio
merged_df[['public_meeting', 'permit']] = merged_df[['public_meeting', 'permit']].astyp
# Check the new data types
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51880 entries, 69572 to 26348
Data columns (total 33 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   amount_tsh             51880 non-null  float64
 1   date_recorded          51880 non-null  object
 2   funder                 51880 non-null  object
 3   gps_height             51880 non-null  int64
 4   installer              51880 non-null  object
 5   longitude              51880 non-null  float64
 6   latitude               51880 non-null  float64
 7   basin                  51880 non-null  object
 8   subvillage             51880 non-null  object
 9   region                 51880 non-null  object
 10  region_code            51880 non-null  int64
 11  district_code          51880 non-null  int64
 12  population             51880 non-null  int64
 13  public_meeting         51880 non-null  int32
 14  recorded_by            51880 non-null  object
 15  scheme_management      51880 non-null  object
 16  scheme_name            51880 non-null  object
 17  permit                 51880 non-null  int32
 18  construction_year      51880 non-null  int64
 19  extraction_type        51880 non-null  object
 20  extraction_type_group  51880 non-null  object
 21  management_group       51880 non-null  object
 22  payment                51880 non-null  object
 23  payment_type           51880 non-null  object
 24  water_quality          51880 non-null  object
 25  quantity               51880 non-null  object
 26  quantity_group         51880 non-null  object
```

```
 27  source_type            51880 non-null  object
 28  source_class           51880 non-null  object
 29  waterpoint_type        51880 non-null  object
 30  status_group           51880 non-null  object
 31  _merge                 51880 non-null  category
 32  pump_age               51880 non-null  int64
dtypes: category(1), float64(3), int32(2), int64(6), object(21)
memory usage: 12.7+ MB
```

3. Logistic Regression

- Prepare the data (ensure binary target variable)
- Create and train the model
- Make predictions on the test set
- Evaluate the model (accuracy, precision, recall, F1-score)
- Plot the ROC curve and calculate AUC
- Analyze coefficients and their significance

In [ ]:
```
# Assign status_group column to y series
y = merged_df['status_group']

# Drop status_group and _merge to create X dataframe
X = merged_df.drop(['status_group','_merge'], axis=1)

# Print first 5 rows of X
X.head()
```

Out[ ]:

id	amount_tsh	date_recorded	funder	gps_height	installer	longitude	latitude	basin	
69572	6000.0	2011-03-14	Roman	1390	Roman	34.938093	-9.856322	Lake Nyasa	
8776	0.0	2013-03-06	Grumeti	1399	GRUMETI	34.698766	-2.147466	Lake Victoria	
34310	25.0	2013-02-25	Lottery Club	686	World vision	37.460664	-3.821329	Pangani	
67743	0.0	2013-01-28	Unicef	263	UNICEF	38.486161	-11.155298	Ruvuma / Southern Coast	M
9944	20.0	2011-03-13	Mkinga Distric Coun	0	DWE	39.172796	-4.765587	Pangani	Moa

5 rows × 31 columns

In [ ]:
```
#Check categorical columns in merged set
merged_df.select_dtypes(include=['object']).columns
```

Out[ ]:
```
Index(['date_recorded', 'funder', 'installer', 'basin', 'subvillage', 'region',
       'recorded_by', 'scheme_management', 'scheme_name', 'extraction_type',
```

```
       'extraction_type_group', 'management_group', 'payment', 'payment_type',
       'water_quality', 'quantity', 'quantity_group', 'source_type',
       'source_class', 'waterpoint_type', 'status_group'],
      dtype='object')
```

In [ ]:
```
#check numerical columns in merged set
merged_df.select_dtypes(include=['int64', 'float64']).columns
```

Out[ ]:
```
Index(['amount_tsh', 'gps_height', 'longitude', 'latitude', 'region_code',
       'district_code', 'population', 'construction_year', 'pump_age'],
      dtype='object')
```

In [ ]:
```
# Create lists of categorical, numerical, and binary columns
category_column = ['funder', 'installer', 'basin', 'region', 'scheme_management', 'sche
       'extraction_type_group', 'management_group', 'payment_type',
       'water_quality', 'quantity_group', 'source_type',
       'source_class', 'waterpoint_type']

numerical_column = ['amount_tsh', 'gps_height', 'longitude', 'latitude', 'region_code',
       'district_code', 'population', 'construction_year', 'pump_age']

binary_column = ['public_meeting', 'permit']
```

In [ ]:
```
#create dummies for categorical colums
X= pd.get_dummies(X, columns=category_column)
X
```

Out[ ]:

id	amount_tsh	gps_height	longitude	latitude	subvillage	region_code	district_code	popula
69572	6000.0	1390	34.938093	-9.856322	Mnyusi B	11	5	
8776	0.0	1399	34.698766	-2.147466	Nyamara	20	2	
34310	25.0	686	37.460664	-3.821329	Majengo	21	4	
67743	0.0	263	38.486161	-11.155298	Mahakamani	90	63	
9944	20.0	0	39.172796	-4.765587	Moa/Mwereme	4	8	
...	...	...	...	...	...	...	...	
11164	500.0	351	37.634053	-6.124830	Komstari	5	6	
60739	10.0	1210	37.169807	-3.253847	Kiduruni	3	5	

id	amount_tsh	gps_height	longitude	latitude	subvillage	region_code	district_code	popula
27263	4700.0	1212	35.249991	-9.070629	Igumbilo	11	4	
31282	0.0	0	35.861315	-6.378573	Mwinyi	1	4	
26348	0.0	191	38.104048	-6.747464	Kikatanyemba	5	2	

51880 rows × 6540 columns

In [ ]:
```
# Split the data into training and test sets
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4


# Initialize the Decision Tree model
#dt = DecisionTreeClassifier(random_state=42)

# Train the model
#dt.fit(X_train, y_train)

# Predict on the test set
#y_pred = dt.predict(X_test)

# Evaluate the model
#print("Accuracy:", accuracy_score(y_test, y_pred))
#print("Classification Report:\n", classification_report(y_test, y_pred))
#print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

In [ ]:
```

```