explore

May 24, 2023

1 Tanzania Water Wells Classificaton Data Exploration and Cleaning

1.1 Problem Overview

The Tanzania Ministry of Water along with Taarifa, a crowd-source platform, have commisioned the development of a predictive model that is supposed to be able to predict with **water wells** are likely to fail. While much of Tanzanias population has access to basic water services, a large 39% of households still lack this basic need. An estimated 10% of preventable deaths in the country can be attributed to inadequate *wash services*. A predictive model can enable quick **predictive maintenance** on water wells and help ensure water security in many of the rural communities that are disporportionately affected by this problem.

-

1.1.1 Project Objectives

1. To conduct exploratory analysis and determine which features to include in our model
2. Determine the cleaning steps to be included in building the model pipeline

-

1.1.2 Success Metric

- `Accuracy`: 75%
- `Recall`: 80%

1.2 EDA and Cleaning

[1]:
```python
# import relevant libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from functions import drop_artefacts_and_nulls, ternary_to_binary, calculate_age

warnings.filterwarnings('ignore')
%matplotlib inline
```

```python
sns.set_style('darkgrid')
```

[2]:
```python
# import and view data
train_set = pd.read_csv('Data/train_set.csv')
train_set_labels = pd.read_csv('Data/train_set_labels.csv')
train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59400 entries, 0 to 59399
Data columns (total 40 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     59400 non-null  int64
 1   amount_tsh             59400 non-null  float64
 2   date_recorded          59400 non-null  object
 3   funder                 55765 non-null  object
 4   gps_height             59400 non-null  int64
 5   installer              55745 non-null  object
 6   longitude              59400 non-null  float64
 7   latitude               59400 non-null  float64
 8   wpt_name               59400 non-null  object
 9   num_private            59400 non-null  int64
 10  basin                  59400 non-null  object
 11  subvillage             59029 non-null  object
 12  region                 59400 non-null  object
 13  region_code            59400 non-null  int64
 14  district_code          59400 non-null  int64
 15  lga                    59400 non-null  object
 16  ward                   59400 non-null  object
 17  population             59400 non-null  int64
 18  public_meeting         56066 non-null  object
 19  recorded_by            59400 non-null  object
 20  scheme_management      55523 non-null  object
 21  scheme_name            31234 non-null  object
 22  permit                 56344 non-null  object
 23  construction_year      59400 non-null  int64
 24  extraction_type        59400 non-null  object
 25  extraction_type_group  59400 non-null  object
 26  extraction_type_class  59400 non-null  object
 27  management             59400 non-null  object
 28  management_group       59400 non-null  object
 29  payment                59400 non-null  object
 30  payment_type           59400 non-null  object
 31  water_quality          59400 non-null  object
 32  quality_group          59400 non-null  object
 33  quantity               59400 non-null  object
 34  quantity_group         59400 non-null  object
 35  source                 59400 non-null  object
 36  source_type            59400 non-null  object
 37  source_class           59400 non-null  object
 38  waterpoint_type        59400 non-null  object
 39  waterpoint_type_group  59400 non-null  object
dtypes: float64(3), int64(7), object(30)
memory usage: 18.1+ MB
```

[3]:

```
#labels
train_set_labels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59400 entries, 0 to 59399
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   id            59400 non-null  int64
 1   status_group  59400 non-null  object
dtypes: int64(1), object(1)
memory usage: 928.2+ KB
```

[4]:

```
#examin test set
test_set = pd.read_csv('Data/test_set.csv')
test_set.head()
```

[4]:

```
      id  amount_tsh date_recorded                  funder  gps_height  \
0  50785         0.0    2013-02-04                    Dmdd        1996
1  51630         0.0    2013-02-04  Government Of Tanzania        1569
2  17168         0.0    2013-02-01                     NaN        1567
3  45559         0.0    2013-01-22              Finn Water         267
4  49871       500.0    2013-03-27                  Bruder        1260

    installer  longitude  latitude                wpt_name  num_private  \
0        DMDD  35.290799  -4.059696  Dinamu Secondary School            0
1         DWE  36.656709  -3.309214                  Kimnyak            0
2         NaN  34.767863  -5.004344           Puma Secondary            0
3  FINN WATER  38.058046  -9.418672           Kwa Mzee Pange            0
4      BRUDER  35.006123 -10.950412          Kwa Mzee Turuka            0

   … payment_type water_quality quality_group     quantity  quantity_group  \
0  …    never pay          soft          good     seasonal        seasonal
1  …    never pay          soft          good  insufficient    insufficient
2  …    never pay          soft          good  insufficient    insufficient
3  …      unknown          soft          good          dry             dry
4  …      monthly          soft          good       enough          enough
```

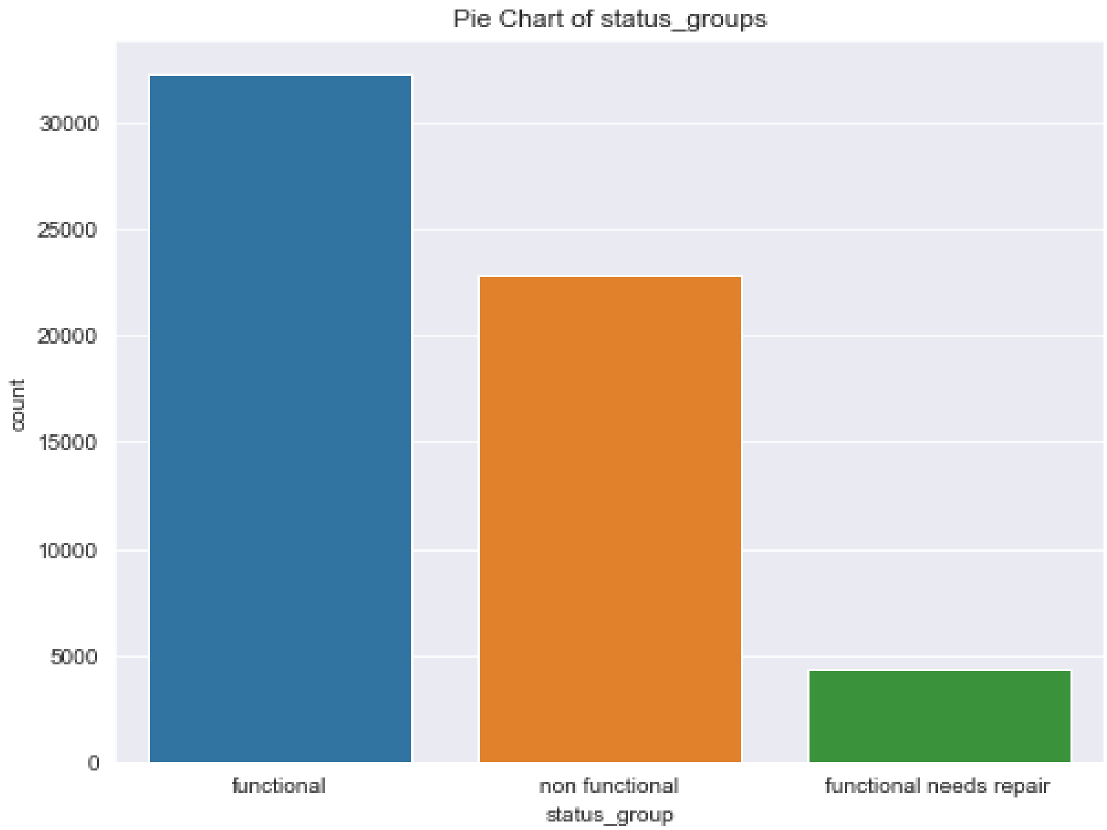
```
                source            source_type  source_class  \
0  rainwater harvesting  rainwater harvesting       surface
1                spring                spring   groundwater
2  rainwater harvesting  rainwater harvesting       surface
3          shallow well          shallow well   groundwater
4                spring                spring   groundwater

      waterpoint_type waterpoint_type_group
0               other                 other
1  communal standpipe    communal standpipe
2               other                 other
3               other                 other
4  communal standpipe    communal standpipe

[5 rows x 40 columns]
```

[5]:
```
print('Labels:', train_set_labels['status_group'].unique())
```

```
Labels: ['functional' 'non functional' 'functional needs repair']
```

- The problem is *ternary*. Need to modify it to approach it as a *binary* classification problem.

[6]:
```
#plot pie chart of status_groups
plt.figure(figsize=(8,6))
sns.countplot(x='status_group', data=train_set_labels)
plt.title('Pie Chart of status_groups')
plt.show()
```



- It would be best to convert `functional needs repair` and `non functional` into a single column to make the problem binary

[7]:
```
#transform labels
train_set_labels = ternary_to_binary(train_set_labels)
```

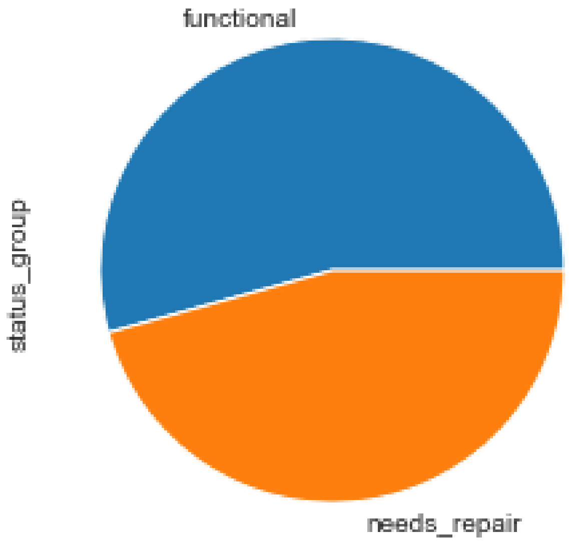
[8]:
```
#merge predictors and labels for eda
labelled_train_set = pd.merge(train_set, train_set_labels, on='id')

labelled_train_set.head()
```

[8]:
```
      id  amount_tsh date_recorded        funder  gps_height     installer  \
0  69572      6000.0    2011-03-14         Roman        1390         Roman
1   8776         0.0    2013-03-06       Grumeti        1399       GRUMETI
2  34310        25.0    2013-02-25  Lottery Club         686  World vision
3  67743         0.0    2013-01-28        Unicef         263        UNICEF
4  19728         0.0    2011-07-13   Action In A           0       Artisan

   longitude   latitude              wpt_name  num_private  … water_quality  \
0  34.938093  -9.856322                  none            0  …          soft
1  34.698766  -2.147466              Zahanati            0  …          soft
```

```
2  37.460664  -3.821329           Kwa Mahundi            0   …          soft
3  38.486161 -11.155298  Zahanati Ya Nanyumbu            0   …          soft
4  31.130847  -1.825359               Shuleni            0   …          soft

  quality_group      quantity  quantity_group                source  \
0          good        enough          enough                spring
1          good  insufficient    insufficient  rainwater harvesting
2          good        enough          enough                   dam
3          good           dry             dry           machine dbh
4          good      seasonal        seasonal  rainwater harvesting

            source_type source_class             waterpoint_type  \
0                spring  groundwater          communal standpipe
1  rainwater harvesting      surface          communal standpipe
2                   dam      surface  communal standpipe multiple
3              borehole  groundwater  communal standpipe multiple
4  rainwater harvesting      surface          communal standpipe

  waterpoint_type_group  status_group
0    communal standpipe    functional
1    communal standpipe    functional
2    communal standpipe    functional
3    communal standpipe  needs_repair
4    communal standpipe    functional

[5 rows x 41 columns]
```

[9]:

```
#plot class distribution of new target
labelled_train_set['status_group'].value_counts().plot(kind='pie');
```



- There is an acceptable level of class imbalance

[10]:
```
#save labelled train set
labelled_train_set.to_csv('Data/labelled_train_set.csv', index=False)
```

[11]:
```
labelled_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 0 to 59399
Data columns (total 41 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     59400 non-null  int64
 1   amount_tsh             59400 non-null  float64
 2   date_recorded          59400 non-null  object
 3   funder                 55765 non-null  object
 4   gps_height             59400 non-null  int64
 5   installer              55745 non-null  object
 6   longitude              59400 non-null  float64
 7   latitude               59400 non-null  float64
 8   wpt_name               59400 non-null  object
 9   num_private            59400 non-null  int64
 10  basin                  59400 non-null  object
 11  subvillage             59029 non-null  object
 12  region                 59400 non-null  object
 13  region_code            59400 non-null  int64
 14  district_code          59400 non-null  int64
 15  lga                    59400 non-null  object
 16  ward                   59400 non-null  object
 17  population             59400 non-null  int64
 18  public_meeting         56066 non-null  object
 19  recorded_by            59400 non-null  object
 20  scheme_management      55523 non-null  object
 21  scheme_name            31234 non-null  object
 22  permit                 56344 non-null  object
 23  construction_year      59400 non-null  int64
 24  extraction_type        59400 non-null  object
 25  extraction_type_group  59400 non-null  object
 26  extraction_type_class  59400 non-null  object
 27  management             59400 non-null  object
 28  management_group       59400 non-null  object
 29  payment                59400 non-null  object
 30  payment_type           59400 non-null  object
 31  water_quality          59400 non-null  object
 32  quality_group          59400 non-null  object
 33  quantity               59400 non-null  object
 34  quantity_group         59400 non-null  object
 35  source                 59400 non-null  object
 36  source_type            59400 non-null  object
 37  source_class           59400 non-null  object
 38  waterpoint_type        59400 non-null  object
 39  waterpoint_type_group  59400 non-null  object
 40  status_group           59400 non-null  object
dtypes: float64(3), int64(7), object(31)
memory usage: 19.0+ MB
```

[12]:

```
#shape and column types
categorical = labelled_train_set.select_dtypes(include='object').columns
numerical = labelled_train_set.select_dtypes(include='number').columns
print('shape:', labelled_train_set.shape)
print('categorical columns:\n', categorical.values)
print('numerical columns:\n', numerical.values)
```

```
shape: (59400, 41)
categorical columns:
 ['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' 'status_group']
numerical columns:
 ['id' 'amount_tsh' 'gps_height' 'longitude' 'latitude' 'num_private'
 'region_code' 'district_code' 'population' 'construction_year']
```

1.2.1 Examine some columns to determine which ones are irrelevant

From the data documentation features such as `wpt_name` and `id` are artefacts and irrelevant to predictive modelling thus shall be added to a list of features to be dropped by the function *drop_irrelevant_cols*

[13]:

```
#examine 'recorded_by'
labelled_train_set['recorded_by'].value_counts()
```

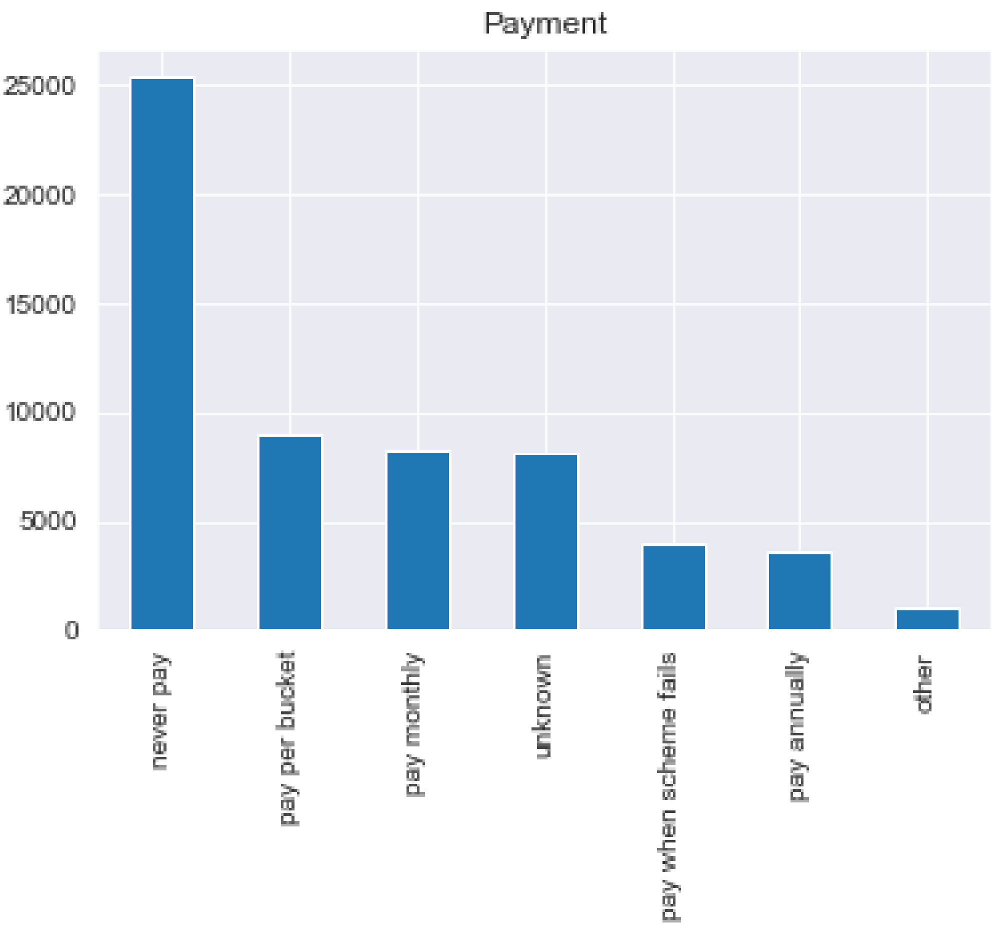
[13]:

```
GeoData Consultants Ltd    59400
Name: recorded_by, dtype: int64
```

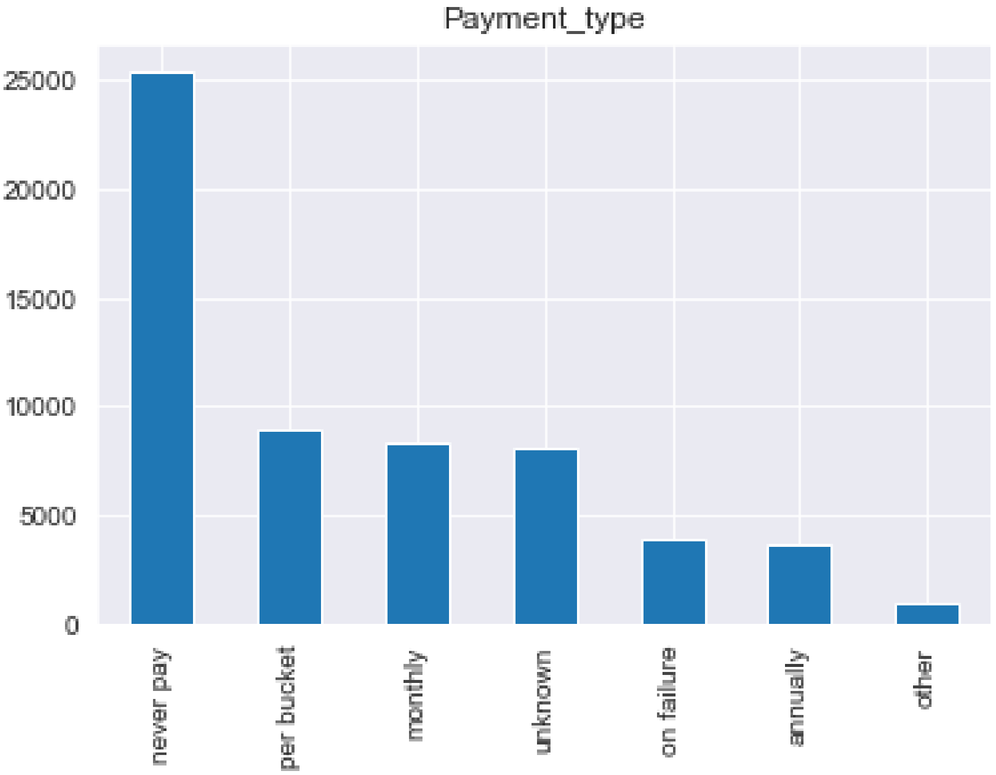
- `recorded_by` should be dropped as it is irrelevant

[14]:

```
#Examine 'payment'
labelled_train_set['payment'].value_counts().plot(kind='bar')
plt.title('Payment');
```



```
[15]: #visualize `payment_type`
      train_set['payment_type'].value_counts().plot(kind='bar')
      plt.title('Payment_type');
```



- `payment` and `payment_type` have the same information and are both irrelevant for modelling.

1.2.2 Redundant Columns

[16]:

```
#print unique values in categorical columns
for col in categorical:
    print(f'{col} values:\n{labelled_train_set[col].unique()}\n')
```

```
date_recorded values:
['2011-03-14' '2013-03-06' '2013-02-25' '2013-01-28' '2011-07-13'
 '2011-03-13' '2012-10-01' '2012-10-09' '2012-11-03' '2011-08-03'
 '2011-02-20' '2013-02-18' '2012-10-14' '2013-03-15' '2012-10-20'
 '2011-08-04' '2011-07-04' '2011-09-04' '2011-07-22' '2011-02-22'
 '2011-02-27' '2013-02-10' '2011-10-04' '2013-11-03' '2013-01-21'
 '2013-01-16' '2011-07-11' '2013-03-05' '2013-03-16' '2011-03-23'
 '2011-03-16' '2013-03-19' '2011-03-11' '2011-02-23' '2013-03-28'
 '2011-07-16' '2011-03-27' '2013-02-11' '2013-10-03' '2011-03-12'
 '2011-07-07' '2013-01-15' '2013-03-18' '2012-10-22' '2013-02-05'
 '2011-07-27' '2011-04-04' '2013-02-21' '2011-08-18' '2011-07-31'
 '2011-08-01' '2011-07-14' '2013-02-22' '2013-07-03' '2013-08-03'
 '2013-01-22' '2011-03-22' '2013-05-03' '2013-01-19' '2013-02-09'
 '2011-01-04' '2013-02-04' '2011-03-05' '2011-03-31' '2013-02-27'
```

```
'2011-08-07' '2011-03-30' '2012-10-23' '2011-03-02' '2011-03-04'
'2013-03-14' '2012-10-18' '2011-08-08' '2011-09-05' '2011-04-11'
'2013-01-17' '2013-01-20' '2013-01-27' '2013-02-19' '2012-10-06'
'2013-02-13' '2013-02-26' '2013-02-16' '2013-02-17' '2011-07-21'
'2011-03-07' '2011-07-12' '2011-07-17' '2011-04-07' '2011-04-10'
'2012-10-29' '2011-02-25' '2012-10-11' '2013-02-14' '2012-10-05'
'2011-07-19' '2011-04-15' '2011-03-17' '2013-01-25' '2011-10-07'
'2011-02-24' '2013-03-21' '2011-08-02' '2011-02-03' '2013-01-24'
'2012-10-21' '2011-04-16' '2013-02-03' '2013-02-24' '2011-04-02'
'2012-10-19' '2013-01-30' '2011-03-03' '2011-08-17' '2011-03-28'
'2011-07-23' '2013-02-06' '2011-08-11' '2011-03-09' '2011-03-18'
'2013-03-07' '2011-08-14' '2013-09-03' '2011-02-16' '2011-04-03'
'2011-07-09' '2011-04-12' '2011-04-14' '2011-03-10' '2013-03-25'
'2013-02-28' '2013-01-18' '2012-10-10' '2011-07-03' '2011-08-05'
'2011-07-20' '2013-03-13' '2011-03-15' '2011-07-18' '2013-03-03'
'2011-11-07' '2013-04-04' '2012-10-16' '2013-03-23' '2013-04-03'
'2013-02-08' '2011-03-21' '2011-04-05' '2012-10-15' '2011-03-19'
'2013-06-03' '2013-03-29' '2012-10-28' '2011-07-15' '2012-10-12'
'2011-07-29' '2011-08-06' '2012-10-13' '2013-03-02' '2013-02-12'
'2013-01-29' '2013-01-04' '2012-10-25' '2012-11-13' '2013-02-01'
'2011-08-10' '2013-03-17' '2011-07-30' '2011-02-21' '2011-02-17'
'2011-08-19' '2013-02-15' '2013-02-02' '2013-01-26' '2011-04-06'
'2011-08-21' '2013-03-24' '2013-10-02' '2011-04-01' '2013-02-23'
'2013-02-20' '2011-04-08' '2011-03-29' '2011-03-25' '2013-03-01'
'2013-05-04' '2012-11-05' '2011-03-24' '2011-03-20' '2013-03-04'
'2012-10-26' '2013-06-04' '2011-08-20' '2013-02-07' '2011-07-24'
'2011-07-25' '2013-03-22' '2013-08-02' '2011-07-28' '2013-03-12'
'2013-03-30' '2013-12-03' '2011-03-26' '2011-03-08' '2013-01-23'
'2012-11-04' '2012-10-02' '2012-10-07' '2011-04-18' '2012-11-15'
'2011-08-12' '2011-08-23' '2012-10-08' '2011-02-26' '2013-03-26'
'2011-03-01' '2012-12-14' '2011-02-14' '2013-01-14' '2012-10-04'
'2011-07-26' '2012-10-27' '2012-10-17' '2013-09-02' '2012-10-03'
'2013-03-20' '2012-11-08' '2011-02-15' '2012-10-24' '2013-03-10'
'2011-07-06' '2011-07-08' '2012-12-13' '2011-08-25' '2004-08-01'
'2011-04-09' '2012-10-31' '2011-03-06' '2013-07-02' '2012-11-12'
'2011-02-28' '2011-02-02' '2013-01-31' '2011-07-10' '2011-04-17'
'2011-07-05' '2011-06-04' '2011-08-22' '2011-01-03' '2013-01-13'
'2012-11-09' '2013-01-11' '2011-01-08' '2013-07-04' '2011-02-18'
'2011-08-13' '2012-11-06' '2011-06-03' '2013-01-12' '2013-03-08'
'2004-12-01' '2012-12-16' '2011-04-13' '2012-11-01' '2013-03-27'
'2011-12-03' '2013-01-08' '2011-04-19' '2012-12-15' '2012-10-30'
'2011-08-16' '2013-01-07' '2013-01-03' '2013-05-02' '2011-08-15'
'2011-02-19' '2011-11-03' '2011-04-21' '2013-01-10' '2012-11-10'
'2011-12-07' '2012-11-11' '2011-10-03' '2011-08-26' '2011-04-22'
'2011-08-09' '2011-06-07' '2002-10-14' '2013-03-09' '2011-02-04'
'2013-01-09' '2012-12-12' '2012-11-14' '2012-12-11' '2011-04-20'
'2012-12-18' '2011-08-27' '2013-12-02' '2013-11-02' '2011-09-27'
'2011-08-24' '2011-09-03' '2012-11-29' '2011-09-19' '2012-12-21'
```

```
 '2012-11-02' '2013-03-11' '2012-11-19' '2011-05-03' '2012-12-10'
 '2011-11-04' '2004-05-01' '2004-04-05' '2012-12-17' '2012-11-07'
 '2012-11-30' '2004-06-01' '2011-05-07' '2012-12-23' '2011-09-09'
 '2012-12-24' '2011-05-04' '2011-04-23' '2013-04-02' '2011-02-01'
 '2011-09-18' '2011-09-06' '2011-09-20' '2004-03-01' '2011-09-17'
 '2013-01-01' '2004-01-07' '2004-07-01' '2011-09-11' '2011-08-31'
 '2011-09-21' '2011-08-30' '2011-08-28' '2011-09-01' '2011-09-28'
 '2011-09-16' '2011-09-13' '2011-09-08' '2011-09-23' '2013-01-06'
 '2011-09-14' '2004-03-06' '2012-01-21' '2012-01-25' '2011-09-15'
 '2011-09-25' '2004-09-01' '2004-04-01' '2011-09-26' '2011-09-12'
 '2013-12-01']

funder values:
['Roman' 'Grumeti' 'Lottery Club' … 'Dina' 'Brown' 'Samlo']

installer values:
['Roman' 'GRUMETI' 'World vision' … 'Dina' 'brown' 'SELEPTA']

wpt_name values:
['none' 'Zahanati' 'Kwa Mahundi' … 'Kwa Yahona Kuvala' 'Mshoro'
 'Kwa Mzee Lugawa']

basin values:
['Lake Nyasa' 'Lake Victoria' 'Pangani' 'Ruvuma / Southern Coast'
 'Internal' 'Lake Tanganyika' 'Wami / Ruvu' 'Rufiji' 'Lake Rukwa']

subvillage values:
['Mnyusi B' 'Nyamara' 'Majengo' … 'Itete B' 'Maore Kati' 'Kikatanyemba']

region values:
['Iringa' 'Mara' 'Manyara' 'Mtwara' 'Kagera' 'Tanga' 'Shinyanga' 'Tabora'
 'Pwani' 'Ruvuma' 'Kilimanjaro' 'Rukwa' 'Mwanza' 'Kigoma' 'Lindi' 'Dodoma'
 'Arusha' 'Mbeya' 'Singida' 'Morogoro' 'Dar es Salaam']

lga values:
['Ludewa' 'Serengeti' 'Simanjiro' 'Nanyumbu' 'Karagwe' 'Mkinga'
 'Shinyanga Rural' 'Kahama' 'Tabora Urban' 'Mkuranga' 'Namtumbo' 'Maswa'
 'Siha' 'Meatu' 'Sumbawanga Rural' 'Njombe' 'Ukerewe' 'Bariadi' 'Same'
 'Kigoma Rural' 'Moshi Rural' 'Lindi Rural' 'Rombo' 'Chamwino' 'Bagamoyo'
 'Mafia' 'Arusha Rural' 'Kyela' 'Kondoa' 'Kilolo' 'Kibondo' 'Makete'
 'Singida Rural' 'Masasi' 'Rungwe' 'Moshi Urban' 'Geita' 'Mbulu'
 'Bukoba Rural' 'Muheza' 'Lushoto' 'Meru' 'Iramba' 'Kilombero' 'Mbarali'
 'Kasulu' 'Bukoba Urban' 'Korogwe' 'Bukombe' 'Morogoro Rural' 'Kishapu'
 'Musoma Rural' 'Sengerema' 'Iringa Rural' 'Muleba' 'Dodoma Urban'
 'Ruangwa' 'Hanang' 'Misenyi' 'Missungwi' 'Songea Rural' 'Tanga' 'Tunduru'
 'Hai' 'Mwanga' 'Chato' 'Biharamulo' 'Ileje' 'Mpwapwa' 'Mvomero' 'Bunda'
 'Kiteto' 'Longido' 'Urambo' 'Mbozi' 'Sikonge' 'Ilala' 'Tarime' 'Temeke'
 'Mbeya Rural' 'Magu' 'Manyoni' 'Igunga' 'Kilosa' 'Babati' 'Chunya'
 'Mufindi' 'Mtwara Rural' 'Ngara' 'Karatu' 'Mpanda' 'Kibaha'
 'Singida Urban' 'Newala' 'Nzega' 'Nkasi' 'Bahi' 'Mbinga' 'Ulanga'
 'Sumbawanga Urban' 'Morogoro Urban' 'Tandahimba' 'Kisarawe'
 'Mtwara Urban' 'Kilwa' 'Liwale' 'Kongwa' 'Uyui' 'Rufiji' 'Kwimba'
 'Monduli' 'Shinyanga Urban' 'Ngorongoro' 'Handeni' 'Rorya' 'Pangani'
 'Lindi Urban' 'Nachingwea' 'Kinondoni' 'Kigoma Urban' 'Ilemela' 'Kilindi'
 'Arusha Urban' 'Songea Urban' 'Nyamagana']

ward values:
['Mundindi' 'Natta' 'Ngorika' … 'Chinugulu' 'Nyamtinga' 'Kinungu']

public_meeting values:
[True nan False]

recorded_by values:
['GeoData Consultants Ltd']

scheme_management values:
['VWC' 'Other' nan 'Private operator' 'WUG' 'Water Board' 'WUA'
 'Water authority' 'Company' 'Parastatal' 'Trust' 'SWC' 'None']

scheme_name values:
['Roman' nan 'Nyumba ya mungu pipe scheme' … 'BL Nsherehehe'
 'Magati  gravity spri' 'Mtawanya']

permit values:
[False True nan]

extraction_type values:
['gravity' 'submersible' 'swn 80' 'nira/tanira' 'india mark ii' 'other'
 'ksb' 'mono' 'windmill' 'afridev' 'other - rope pump' 'india mark iii'
 'other - swn 81' 'other - play pump' 'cemo' 'climax' 'walimi'
 'other - mkulima/shinyanga']

extraction_type_group values:
['gravity' 'submersible' 'swn 80' 'nira/tanira' 'india mark ii' 'other'
 'mono' 'wind-powered' 'afridev' 'rope pump' 'india mark iii'
 'other handpump' 'other motorpump']

extraction_type_class values:
['gravity' 'submersible' 'handpump' 'other' 'motorpump' 'wind-powered'
 'rope pump']

management values:
['vwc' 'wug' 'other' 'private operator' 'water board' 'wua' 'company'
 'water authority' 'parastatal' 'unknown' 'other - school' 'trust']

management_group values:
```

```
['user-group' 'other' 'commercial' 'parastatal' 'unknown']

payment values:
['pay annually' 'never pay' 'pay per bucket' 'unknown'
 'pay when scheme fails' 'other' 'pay monthly']

payment_type values:
['annually' 'never pay' 'per bucket' 'unknown' 'on failure' 'other'
 'monthly']

water_quality values:
['soft' 'salty' 'milky' 'unknown' 'fluoride' 'coloured' 'salty abandoned'
 'fluoride abandoned']

quality_group values:
['good' 'salty' 'milky' 'unknown' 'fluoride' 'colored']

quantity values:
['enough' 'insufficient' 'dry' 'seasonal' 'unknown']

quantity_group values:
['enough' 'insufficient' 'dry' 'seasonal' 'unknown']

source values:
['spring' 'rainwater harvesting' 'dam' 'machine dbh' 'other'
 'shallow well' 'river' 'hand dtw' 'lake' 'unknown']

source_type values:
['spring' 'rainwater harvesting' 'dam' 'borehole' 'other' 'shallow well'
 'river/lake']

source_class values:
['groundwater' 'surface' 'unknown']

waterpoint_type values:
['communal standpipe' 'communal standpipe multiple' 'hand pump' 'other'
 'improved spring' 'cattle trough' 'dam']

waterpoint_type_group values:
['communal standpipe' 'hand pump' 'other' 'improved spring'
 'cattle trough' 'dam']

status_group values:
['functional' 'needs_repair']
```

- `source_type` and `source_class` store redudunt information already in `source`. They should be added to the irrelevant columns in **functions.py**

- water_quality and quality_group have redundant information. We can drop the former and keep quality_group as it is cleaner
- waterpoint_type_group and waterpoint_type have similar values. We drop the former and keep waterpoint_type
- quantity_group and quantity have the same information. We drop the former and keep quantity

[17]:

```
labelled_train_set.info()
```

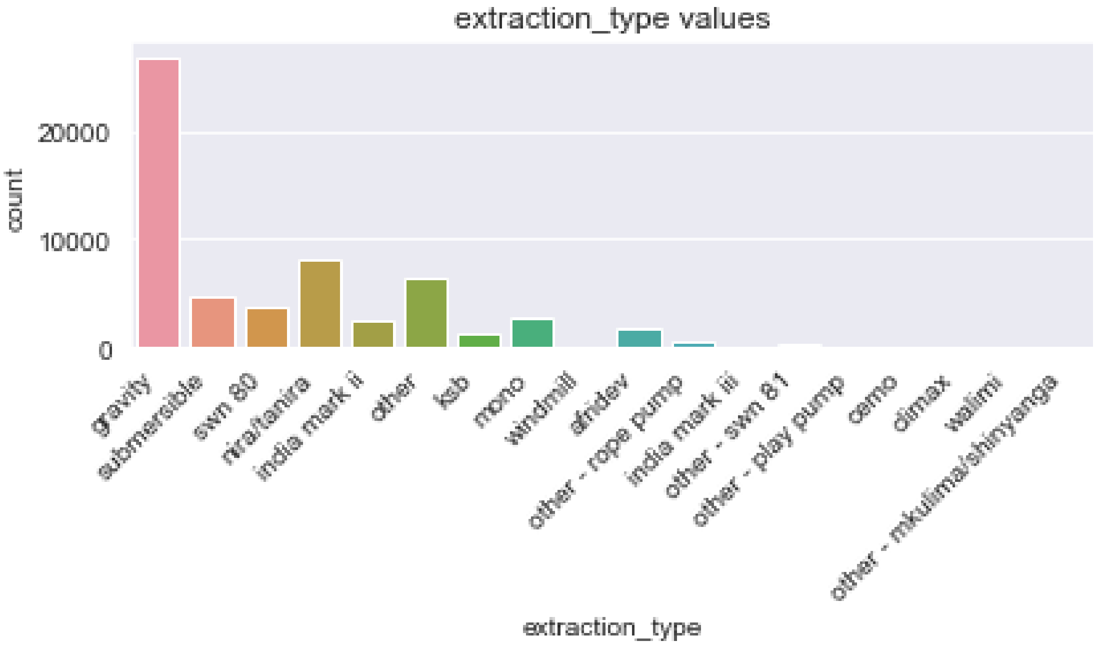
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 0 to 59399
Data columns (total 41 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   id                     59400 non-null  int64  
 1   amount_tsh             59400 non-null  float64
 2   date_recorded          59400 non-null  object 
 3   funder                 55765 non-null  object 
 4   gps_height             59400 non-null  int64  
 5   installer              55745 non-null  object 
 6   longitude              59400 non-null  float64
 7   latitude               59400 non-null  float64
 8   wpt_name               59400 non-null  object 
 9   num_private            59400 non-null  int64  
 10  basin                  59400 non-null  object 
 11  subvillage             59029 non-null  object 
 12  region                 59400 non-null  object 
 13  region_code            59400 non-null  int64  
 14  district_code          59400 non-null  int64  
 15  lga                    59400 non-null  object 
 16  ward                   59400 non-null  object 
 17  population             59400 non-null  int64  
 18  public_meeting         56066 non-null  object 
 19  recorded_by            59400 non-null  object 
 20  scheme_management      55523 non-null  object 
 21  scheme_name            31234 non-null  object 
 22  permit                 56344 non-null  object 
 23  construction_year      59400 non-null  int64  
 24  extraction_type        59400 non-null  object 
 25  extraction_type_group  59400 non-null  object 
 26  extraction_type_class  59400 non-null  object 
 27  management             59400 non-null  object 
 28  management_group       59400 non-null  object 
 29  payment                59400 non-null  object 
 30  payment_type           59400 non-null  object 
 31  water_quality          59400 non-null  object 
 32  quality_group          59400 non-null  object 
 33  quantity               59400 non-null  object 
 34  quantity_group         59400 non-null  object
 35  source                 59400 non-null  object
 36  source_type            59400 non-null  object
 37  source_class           59400 non-null  object
 38  waterpoint_type        59400 non-null  object
 39  waterpoint_type_group  59400 non-null  object
 40  status_group           59400 non-null  object
dtypes: float64(3), int64(7), object(31)
memory usage: 19.0+ MB
```

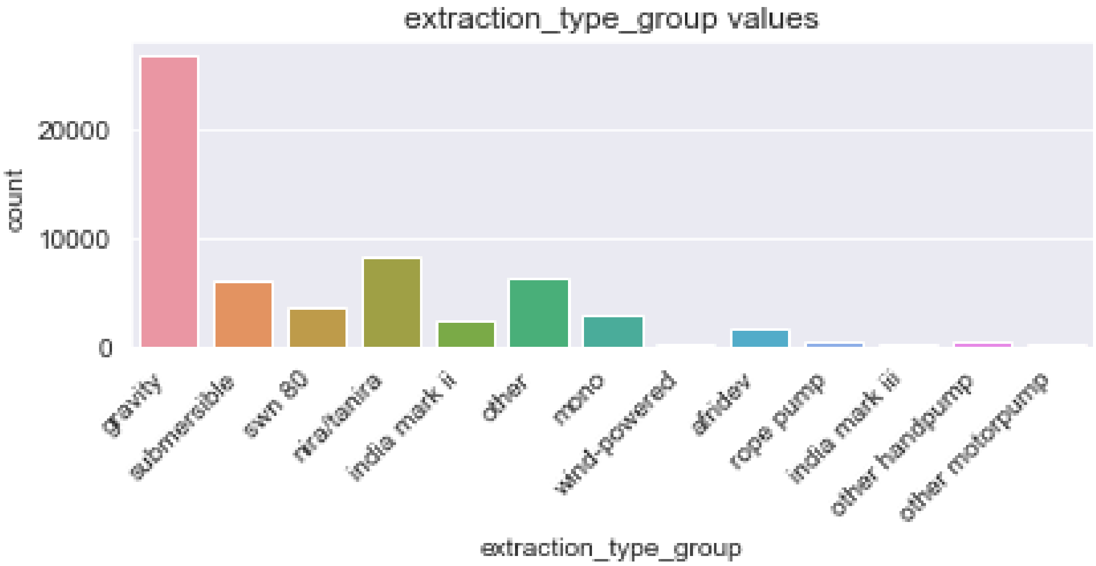
[18]:

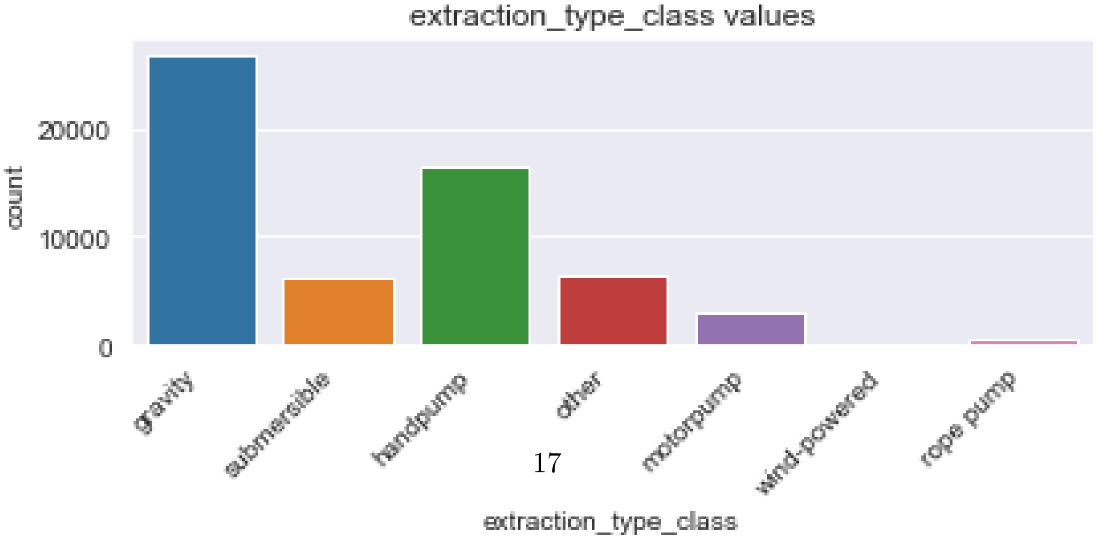
```
#fucntion to plot comparison barcharts
def count_plot_compare(data):
    """
    Plots countplots for subset of columns for comparisons
    Params: data, pandas.DataFrame
    Returns: None
    """
    # #fig and axes
    fig, axes = plt.subplots(nrows=len(data.columns), ncols=1, figsize=(6, 10))

    # #loop over subset
    for ind, col in enumerate(data.columns):
        ax = sns.countplot(x=col, data=data, ax=axes[ind])
        ax.set_title(f'{col} values')
        ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')

    # #adjust spacing
    plt.tight_layout()

    # #show
    plt.show()
    return None
```

[19]:

```
#countplot for 'extrction_type' columns
col_subset_1 = labelled_train_set.iloc[:, 24:27]
count_plot_compare(col_subset_1)
```


extraction_type values
count
20000
10000
0
gravity
submersible
swn 80
nira/tanira
india mark ii
other
ksb
mono
windmill
afridev
other - rope pump
india mark iii
other - swn 81
other - play pump
cemo
climax
walimi
other - mkulima/shinyanga
extraction_type


extraction_type_group values
count
20000
10000
0
gravity
submersible
swn 80
nira/tanira
india mark ii
other
mono
wind-powered
afridev
rope pump
india mark iii
other handpump
other motorpump
extraction_type_group


extraction_type_class values
count
20000
10000
0
gravity
submersible
handpump
other
motorpump
wind-powered
rope pump
extraction_type_class

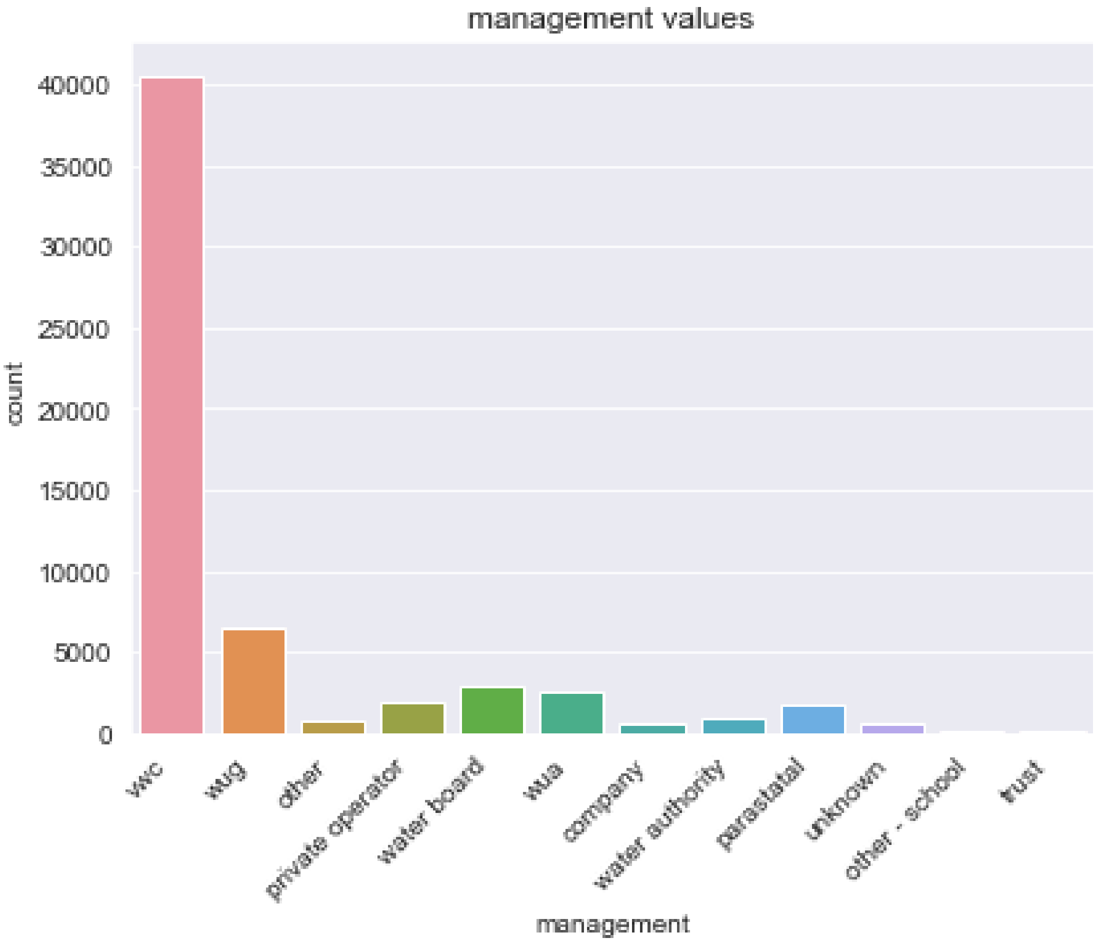
- Seems that `extraction_type_class` better generalizes this feature. The rest shall be dropped.

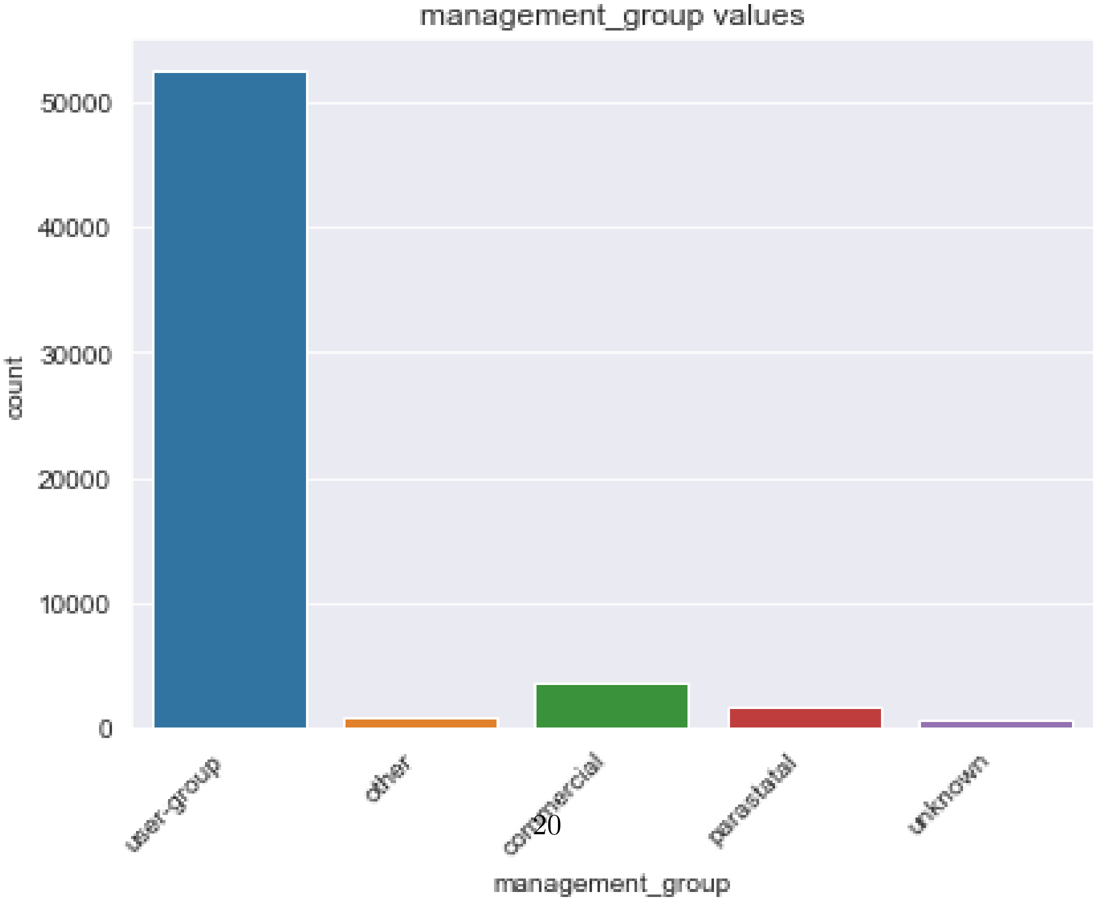
[20]:

```
labelled_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 0 to 59399
Data columns (total 41 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     59400 non-null  int64
 1   amount_tsh             59400 non-null  float64
 2   date_recorded          59400 non-null  object
 3   funder                 55765 non-null  object
 4   gps_height             59400 non-null  int64
 5   installer              55745 non-null  object
 6   longitude              59400 non-null  float64
 7   latitude               59400 non-null  float64
 8   wpt_name               59400 non-null  object
 9   num_private            59400 non-null  int64
 10  basin                  59400 non-null  object
 11  subvillage             59029 non-null  object
 12  region                 59400 non-null  object
 13  region_code            59400 non-null  int64
 14  district_code          59400 non-null  int64
 15  lga                    59400 non-null  object
 16  ward                   59400 non-null  object
 17  population             59400 non-null  int64
 18  public_meeting         56066 non-null  object
 19  recorded_by            59400 non-null  object
 20  scheme_management      55523 non-null  object
 21  scheme_name            31234 non-null  object
 22  permit                 56344 non-null  object
 23  construction_year      59400 non-null  int64
 24  extraction_type        59400 non-null  object
 25  extraction_type_group  59400 non-null  object
 26  extraction_type_class  59400 non-null  object
 27  management             59400 non-null  object
 28  management_group       59400 non-null  object
 29  payment                59400 non-null  object
 30  payment_type           59400 non-null  object
 31  water_quality          59400 non-null  object
 32  quality_group          59400 non-null  object
 33  quantity               59400 non-null  object
 34  quantity_group         59400 non-null  object
 35   source                 59400 non-null  object 
 36   source_type            59400 non-null  object 
 37   source_class           59400 non-null  object 
 38   waterpoint_type        59400 non-null  object 
 39   waterpoint_type_group  59400 non-null  object 
 40   status_group           59400 non-null  object 
dtypes: float64(3), int64(7), object(31)
memory usage: 21.5+ MB
```

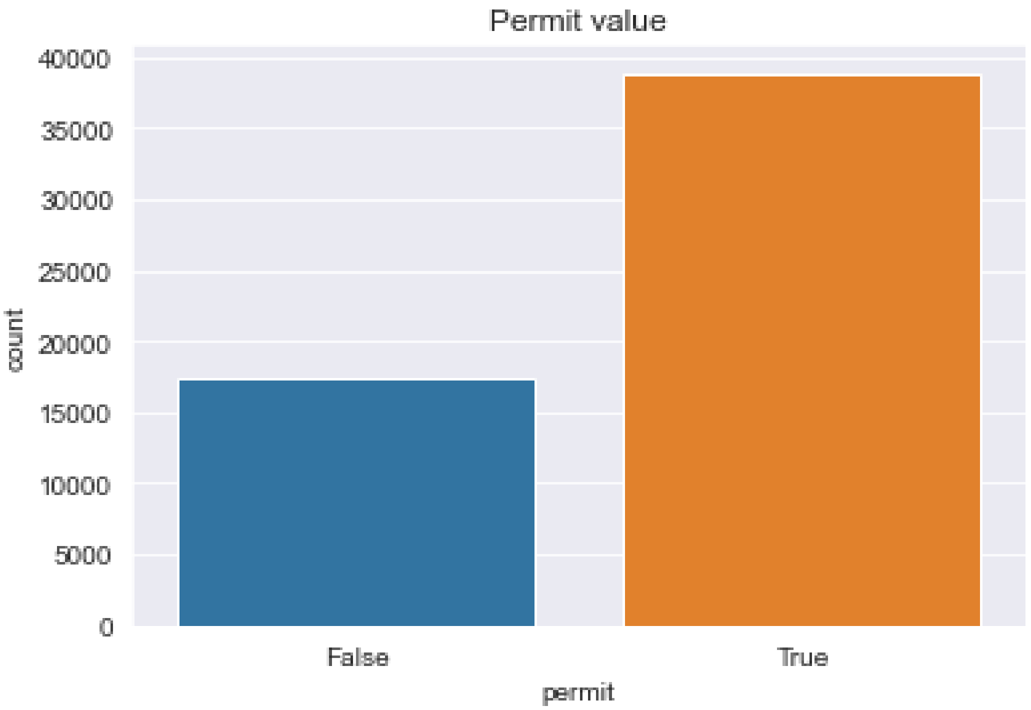
[21]:
```
#countplot comparison for 'management' columns
col_subset_2 = labelled_train_set.iloc[:, 27:29]
count_plot_compare(col_subset_2)
```


management values
count
40000
35000
30000
25000
20000
15000
10000
5000
0
vwc
wug
other
private operator
water board
wua
company
water authority
parastatal
unknown
other - school
trust
management


management_group values
count
50000
40000
30000
20000
10000
0
user-group
other
commercial
parastatal
unknown
management_group

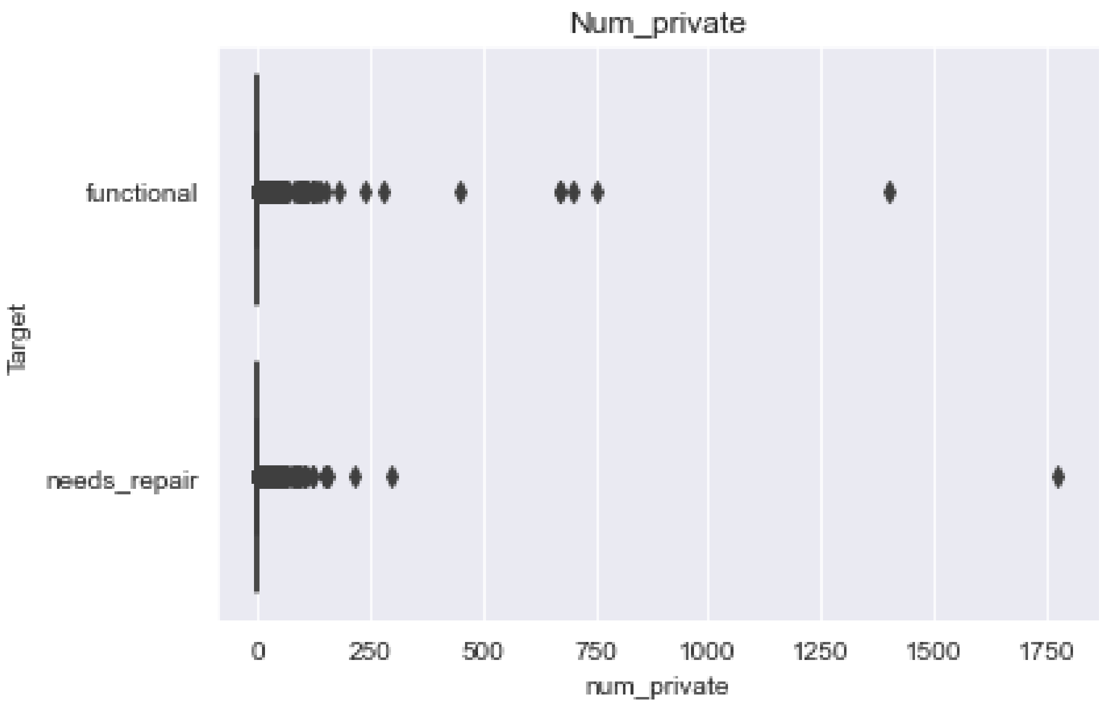
- `management_group` better generalizes this attribute.

```
[22]: #plot 'permit' values
sns.countplot(x='permit', data=labelled_train_set)
plt.title('Permit value');
```



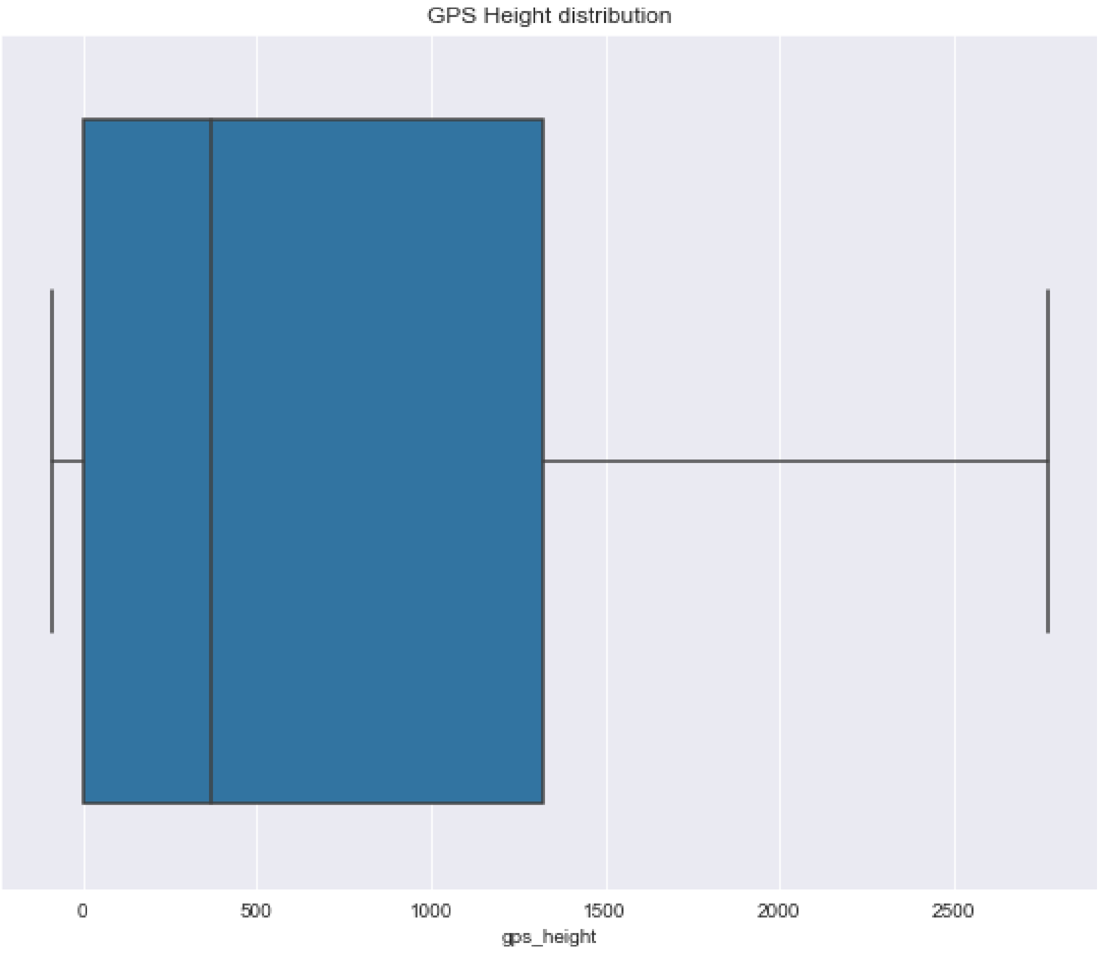
- `permit_value` is a binary value. It might be worth maintaining.

```
[23]: #visualize 'num_private'
vis = sns.boxplot(data=labelled_train_set, x='num_private', y='status_group')
vis.set_title('Num_private')
plt.ylabel('Target');
```



- num_private seems to be a continuous categorical variable by looking at the distribution. From the documentation it seems to be a miscellaneous column. It shall be dropped.

[24]:
```
#visualize 'gps_height' in boxplot
plt.figure(figsize=(10, 8))
sns.boxplot(x='gps_height', data=labelled_train_set)
plt.title('GPS Height distribution')
plt.show();
```



- Check for Validity, Completeness, Consistency and Uniformity

1.2.3 Duplicates

[25]:
```
print(f'Duplicates: ', labelled_train_set['id'].duplicated().sum())
```

```
Duplicates:  0
```

- There are no duplicate entries

1.2.4 Missing Values and Irrelevant columns

[26]:
```
#Local function to print percentage missing errors
def print_missing_perc(data):
    """
    Print percentage missing values
    Parameters: data
    Returns: None
    """
    cols_with_null = []
    for col in data.columns:
        missing_perc = float(data[col].isna().sum()/len(data[col]))
        if(missing_perc > 0):
            cols_with_null.append((col, missing_perc))
        if(col == data.columns[-1]):
            for null_col in cols_with_null:
                print(f'{null_col[0]} missing: {null_col[1]*100}%')
    if not len(cols_with_null):
        print('No null values')
    return None
```

[27]:

```
print_missing_perc(labelled_train_set)
```

```
funder missing: 6.11952861952862%
installer missing: 6.153198653198653%
subvillage missing: 0.6245791245791246%
public_meeting missing: 5.612794612794613%
scheme_management missing: 6.526936026936027%
scheme_name missing: 47.41750841750842%
permit missing: 5.144781144781145%
```

[28]:

```
tww_df = drop_artefacts_and_nulls(labelled_train_set, thresh=.2)
# print_missing_percentage(tww_df)
print_missing_perc(tww_df)
```

```
No null values
```

[29]:

```
tww_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50956 entries, 0 to 59399
Data columns (total 23 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   amount_tsh          50956 non-null  float64
 1   date_recorded       50956 non-null  object
 2   gps_height          50956 non-null  int64
 3   installer           50956 non-null  object
 4   longitude           50956 non-null  float64
 5   latitude            50956 non-null  float64
 6   basin               50956 non-null  object
 7   subvillage          50956 non-null  object
 8   region_code         50956 non-null  int64
 9   district_code       50956 non-null  int64
 10  lga                 50956 non-null  object
 11  ward                50956 non-null  object
 12  population             50956 non-null  int64
 13  scheme_management      50956 non-null  object
 14  permit                 50956 non-null  object
 15  construction_year      50956 non-null  int64
 16  extraction_type_class  50956 non-null  object
 17  management_group       50956 non-null  object
 18  quality_group          50956 non-null  object
 19  quantity               50956 non-null  object
 20  source                 50956 non-null  object
 21  waterpoint_type        50956 non-null  object
 22  status_group           50956 non-null  object
dtypes: float64(3), int64(5), object(15)
memory usage: 9.3+ MB
```

[30]:

```
#new categorical columns
new_categorical = tww_df.select_dtypes(include='object').columns
new_numerical = tww_df.select_dtypes(include='number').columns
print(f'categorical columns:\n{new_categorical.values}')
print(f'numerical columns:\n{new_numerical.values}')
```

```
categorical columns:
['date_recorded' 'installer' 'basin' 'subvillage' 'lga' 'ward'
 'scheme_management' 'permit' 'extraction_type_class' 'management_group'
 'quality_group' 'quantity' 'source' 'waterpoint_type' 'status_group']
numerical columns:
['amount_tsh' 'gps_height' 'longitude' 'latitude' 'region_code'
 'district_code' 'population' 'construction_year']
```

[31]:

```
#print unique values in categorical columns
for col in new_categorical:
    print(f'{col} values:\n{tww_df[col].unique()}\n')
```

```
date_recorded values:
['2011-03-14' '2013-03-06' '2013-02-25' '2013-01-28' '2011-03-13'
 '2012-10-01' '2012-11-03' '2011-02-20' '2013-02-18' '2012-10-14'
 '2013-03-15' '2012-10-20' '2011-08-04' '2011-07-04' '2011-09-04'
 '2011-02-22' '2011-02-27' '2013-02-10' '2011-10-04' '2013-11-03'
 '2013-01-21' '2013-01-16' '2013-03-05' '2013-03-16' '2011-03-23'
 '2011-03-16' '2013-03-19' '2011-03-11' '2011-07-16' '2011-03-27'
 '2013-02-11' '2013-10-03' '2011-03-12' '2011-07-07' '2013-03-18'
 '2012-10-22' '2013-02-05' '2011-04-04' '2013-02-21' '2011-08-18'
 '2011-07-31' '2011-08-01' '2013-02-22' '2013-07-03' '2013-08-03'
 '2013-01-22' '2011-03-22' '2013-05-03' '2013-01-19' '2011-01-04'
 '2013-02-04' '2013-02-27' '2011-03-30' '2012-10-23' '2011-03-02'
 '2011-03-04' '2013-03-14' '2012-10-18' '2011-08-08' '2011-04-11'
 '2013-01-17' '2013-01-20' '2011-07-11' '2013-01-27' '2013-02-19'
 '2012-10-06' '2013-02-13' '2013-02-26' '2013-02-16' '2013-02-17'
 '2011-07-21' '2011-03-07' '2011-07-17' '2011-04-07' '2011-04-10'
```

```
'2011-02-25' '2011-07-14' '2011-07-22' '2012-10-11' '2011-07-27'
'2011-07-19' '2011-03-05' '2011-03-17' '2013-01-25' '2011-10-07'
'2011-08-03' '2011-02-24' '2013-03-21' '2011-08-02' '2011-02-03'
'2013-02-03' '2013-02-24' '2011-04-02' '2012-10-19' '2011-03-03'
'2011-08-17' '2011-03-28' '2011-07-23' '2013-02-06' '2013-01-30'
'2011-08-11' '2011-03-09' '2013-03-07' '2011-08-14' '2013-09-03'
'2011-02-16' '2011-04-03' '2011-07-09' '2011-04-12' '2011-03-10'
'2011-04-14' '2013-03-25' '2013-02-28' '2013-01-18' '2012-10-10'
'2011-07-03' '2011-07-20' '2013-03-13' '2011-03-15' '2011-07-18'
'2013-03-03' '2011-11-07' '2013-04-04' '2012-10-16' '2013-03-23'
'2013-04-03' '2013-02-08' '2011-03-21' '2011-04-05' '2012-10-15'
'2011-08-07' '2013-02-14' '2011-03-19' '2013-06-03' '2013-03-29'
'2011-07-15' '2012-10-12' '2011-03-18' '2011-08-06' '2012-10-13'
'2013-03-02' '2013-02-12' '2013-01-29' '2013-01-04' '2012-11-13'
'2013-02-01' '2011-08-10' '2013-03-17' '2011-07-30' '2011-02-17'
'2011-08-19' '2011-07-29' '2013-01-26' '2011-04-06' '2012-10-05'
'2013-02-09' '2011-08-21' '2013-03-24' '2012-10-29' '2013-10-02'
'2011-04-01' '2013-02-23' '2013-02-20' '2011-03-29' '2013-03-01'
'2011-03-31' '2013-05-04' '2012-11-05' '2011-03-24' '2013-03-04'
'2012-10-21' '2011-04-08' '2012-10-26' '2013-06-04' '2011-08-20'
'2011-04-16' '2013-02-07' '2011-07-25' '2011-07-12' '2013-03-22'
'2013-08-02' '2011-02-23' '2011-07-28' '2013-03-12' '2013-03-30'
'2013-12-03' '2011-07-13' '2011-03-26' '2013-01-23' '2012-11-04'
'2012-10-02' '2012-10-07' '2011-04-18' '2012-11-15' '2011-08-12'
'2012-10-08' '2011-03-20' '2011-02-26' '2013-03-26' '2011-03-01'
'2011-07-24' '2013-01-24' '2012-12-14' '2011-02-14' '2013-01-14'
'2012-10-04' '2011-07-26' '2011-03-25' '2011-08-05' '2011-04-15'
'2011-03-08' '2012-10-27' '2011-02-21' '2012-10-17' '2013-09-02'
'2012-10-03' '2013-03-20' '2012-11-08' '2011-02-15' '2012-10-24'
'2013-03-10' '2011-07-06' '2011-07-08' '2012-12-13' '2011-08-25'
'2011-04-09' '2012-10-31' '2011-03-06' '2013-07-02' '2012-11-12'
'2011-02-28' '2013-02-15' '2011-02-02' '2013-03-28' '2013-01-31'
'2011-07-10' '2011-04-17' '2011-07-05' '2011-06-04' '2011-08-22'
'2011-01-03' '2012-11-09' '2011-01-08' '2013-07-04' '2011-02-18'
'2011-08-13' '2012-11-06' '2011-06-03' '2012-10-25' '2013-03-08'
'2013-01-13' '2013-02-02' '2012-10-28' '2004-12-01' '2012-12-16'
'2013-03-27' '2012-11-01' '2011-12-03' '2011-04-19' '2013-01-15'
'2012-12-15' '2012-10-30' '2011-08-16' '2013-01-07' '2013-01-03'
'2013-05-02' '2011-08-15' '2011-04-13' '2011-02-19' '2011-11-03'
'2011-04-21' '2011-12-07' '2012-11-11' '2012-10-09' '2011-10-03'
'2011-08-26' '2011-08-09' '2011-06-07' '2002-10-14' '2013-03-09'
'2011-08-23' '2013-01-11' '2011-02-04' '2013-01-09' '2012-12-12'
'2012-11-14' '2012-12-11' '2004-08-01' '2011-04-20' '2012-12-18'
'2011-08-27' '2013-12-02' '2013-11-02' '2011-09-03' '2012-12-21'
'2012-11-02' '2013-03-11' '2012-11-19' '2011-05-03' '2012-12-10'
'2012-11-10' '2011-11-04' '2004-05-01' '2004-04-05' '2012-12-17'
'2012-11-07' '2011-04-22' '2013-01-12' '2013-01-08' '2004-06-01'
'2011-05-07' '2011-08-24' '2012-12-24' '2011-05-04' '2011-04-23'
```

```
 '2013-04-02' '2011-02-01' '2012-12-23' '2004-03-01' '2004-01-07'
 '2012-11-30' '2013-01-06' '2004-03-06' '2012-01-21' '2004-09-01'
 '2004-04-01' '2013-12-01']

installer values:
['Roman' 'GRUMETI' 'World vision' … 'Dina' 'brown' 'SELEPTA']

basin values:
['Lake Nyasa' 'Lake Victoria' 'Pangani' 'Ruvuma / Southern Coast'
 'Internal' 'Lake Tanganyika' 'Wami / Ruvu' 'Rufiji' 'Lake Rukwa']

subvillage values:
['Mnyusi B' 'Nyamara' 'Majengo' … 'Itete B' 'Maore Kati' 'Kikatanyemba']

lga values:
['Ludewa' 'Serengeti' 'Simanjiro' 'Nanyumbu' 'Mkinga' 'Shinyanga Rural'
 'Tabora Urban' 'Mkuranga' 'Namtumbo' 'Maswa' 'Siha' 'Meatu'
 'Sumbawanga Rural' 'Njombe' 'Bariadi' 'Same' 'Kigoma Rural' 'Moshi Rural'
 'Lindi Rural' 'Rombo' 'Chamwino' 'Bagamoyo' 'Kyela' 'Kondoa' 'Kilolo'
 'Kibondo' 'Makete' 'Arusha Rural' 'Masasi' 'Moshi Urban' 'Geita' 'Mbulu'
 'Bukoba Rural' 'Muheza' 'Lushoto' 'Meru' 'Iramba' 'Karagwe' 'Kasulu'
 'Korogwe' 'Bukombe' 'Morogoro Rural' 'Kishapu' 'Musoma Rural' 'Sengerema'
 'Iringa Rural' 'Dodoma Urban' 'Ruangwa' 'Hanang' 'Misenyi' 'Missungwi'
 'Songea Rural' 'Tanga' 'Tunduru' 'Hai' 'Mwanga' 'Chato' 'Biharamulo'
 'Ileje' 'Mpwapwa' 'Mvomero' 'Bunda' 'Kiteto' 'Urambo' 'Mbozi' 'Sikonge'
 'Ilala' 'Muleba' 'Temeke' 'Mbeya Rural' 'Magu' 'Manyoni' 'Igunga'
 'Kilosa' 'Babati' 'Chunya' 'Mufindi' 'Mtwara Rural' 'Ngara' 'Karatu'
 'Mpanda' 'Kibaha' 'Ukerewe' 'Newala' 'Nzega' 'Nkasi' 'Bahi' 'Mbinga'
 'Ulanga' 'Sumbawanga Urban' 'Morogoro Urban' 'Tandahimba' 'Kisarawe'
 'Liwale' 'Longido' 'Kilombero' 'Uyui' 'Rufiji' 'Kwimba' 'Shinyanga Urban'
 'Kilwa' 'Ngorongoro' 'Handeni' 'Mtwara Urban' 'Rorya' 'Pangani'
 'Nachingwea' 'Kinondoni' 'Kahama' 'Kigoma Urban' 'Tarime' 'Ilemela'
 'Singida Urban' 'Kilindi' 'Songea Urban' 'Singida Rural' 'Nyamagana']

ward values:
['Mundindi' 'Natta' 'Ngorika' … 'Miteja' 'Jana' 'Ngaya']

scheme_management values:
['VWC' 'Other' 'Private operator' 'WUG' 'Water Board' 'WUA'
 'Water authority' 'Company' 'Parastatal' 'Trust' 'SWC' 'None']

permit values:
[False True]

extraction_type_class values:
['gravity' 'submersible' 'handpump' 'wind-powered' 'other' 'rope pump'
 'motorpump']
```

```
management_group values:
['user-group' 'commercial' 'other' 'parastatal' 'unknown']

quality_group values:
['good' 'salty' 'unknown' 'milky' 'fluoride' 'colored']

quantity values:
['enough' 'insufficient' 'dry' 'seasonal' 'unknown']

source values:
['spring' 'rainwater harvesting' 'dam' 'machine dbh' 'other'
 'shallow well' 'river' 'hand dtw' 'lake' 'unknown']

waterpoint_type values:
['communal standpipe' 'communal standpipe multiple' 'hand pump' 'other'
 'improved spring' 'cattle trough' 'dam']

status_group values:
['functional' 'needs_repair']
```

Feature Engineering

[32]:

```
#engineer new feature 'age'
tww_df = calculate_age(tww_df)
tww_df.head()
```

[32]:

```
   amount_tsh  gps_height     installer  longitude   latitude  \
0      6000.0        1390         Roman  34.938093  -9.856322
1         0.0        1399       GRUMETI  34.698766  -2.147466
2        25.0         686  World vision  37.460664  -3.821329
3         0.0         263        UNICEF  38.486161 -11.155298
5        20.0           0           DWE  39.172796  -4.765587

                     basin   subvillage  region_code  district_code  \
0               Lake Nyasa     Mnyusi B           11              5
1            Lake Victoria      Nyamara           20              2
2                  Pangani      Majengo           21              4
3  Ruvuma / Southern Coast   Mahakamani           90             63
5                  Pangani  Moa/Mwereme            4              8

         lga  … permit  construction_year extraction_type_class  \
0     Ludewa  …  False               1999               gravity
1  Serengeti  …   True               2010               gravity
2  Simanjiro  …   True               2009               gravity
3   Nanyumbu  …   True               1986           submersible
5     Mkinga  …   True               2009           submersible
```

```
  management_group  quality_group      quantity                source  \
0       user-group           good        enough                spring   
1       user-group           good  insufficient  rainwater harvesting   
2       user-group           good        enough                   dam   
3       user-group           good           dry           machine dbh   
5       user-group          salty        enough                 other   

               waterpoint_type  status_group age  
0           communal standpipe    functional  12  
1           communal standpipe    functional   3  
2  communal standpipe multiple    functional   4  
3  communal standpipe multiple  needs_repair  27  
5  communal standpipe multiple    functional   2  

[5 rows x 23 columns]
```

1.3 Conclusion

- Some features are irrelevant to the model and should be dropped. Such as `id`, `wpt_name` etc
- Some features are duplicate columns storing the same information. Some of these generalize the data set better than the rest.