Tanzanian Well Data Analysis Notebook

The following code will attempt to find correlations between the available data to help determine well funcionality in the nation of Tanzania. Several methods are recommended to do this succesfully, to most accurately acheive results. Since this is quite literally a life or death data problem, careful analysis should be used throughout the process.

In [405]:
```
#Import the relevant libraries
import pandas as pd
from __future__ import print_function, division
import pandas as pd
import numpy as np
import os
from sklearn.metrics import accuracy_score
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score, classification_report
from sklearn.linear_model import LogisticRegression
import warnings
from sklearn.exceptions import DataConversionWarning
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from scipy.cluster import hierarchy as hc
```

OSEMN-Obtain

Here is where we will import the data into our notebook as csv files, generously provided to us by Taarifa, and he Tanzanian Ministry of Water

In [299]:
```
#import the data as csv files
submission_format = pd.read_csv('SubmissionFormat.csv')
test = pd.read_csv('702ddfc5-68cd-4d1d-a0de-f5f566f76d91.csv')
labels = pd.read_csv('0bf8bc6e-30d0-4c50-956a-603fc693d966.csv')
train = pd.read_csv('4910797b-ee55-40a7-8668-10efd5c1b960.csv')
```

In [300]:
```
#Check each data collection to make sure they're appropriately labeled
submission_format.head()
test.head()
train.head()
labels.head()
```

Out[300]:

	id	status_group
0	69572	functional
1	8776	functional
2	34310	functional
3	67743	non functional
4	19728	functional

OSEMN-Explore

Now that we have the data in our notebook, we can start executing functions to explore the size, shape, and types of data we will be working with.

In [301]:
```
print("Train data labels:",len(labels))
print("Train data rows, columns:",train.shape)
print("Test data rows, columns:", test.shape)
```

```
Train data labels: 59400
Train data rows, columns: (59400, 40)
Test data rows, columns: (14850, 40)
```

In [302]:
```
labels.status_group.value_counts()
```

Out[302]:
```
functional                 32259
non functional             22824
functional needs repair     4317
Name: status_group, dtype: int64
```

In [303]:
```
labels.status_group.value_counts(normalize=True)
```

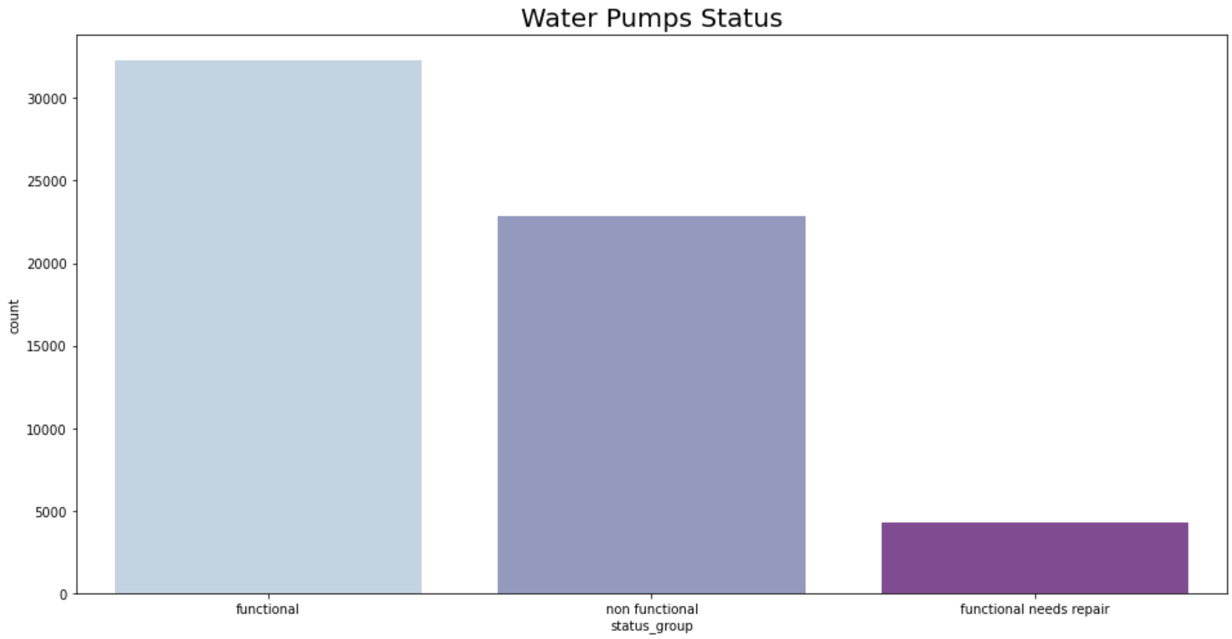
Out[303]:
```
functional                 0.543081
non functional             0.384242
functional needs repair    0.072677
Name: status_group, dtype: float64
```

In [304]:
```
majority = labels['status_group'].mode()[0]
y_pred = np.full(shape=labels['status_group'].shape, fill_value=majority)
```

In [306]:
```
all(y_pred==majority)
```

Out[306]:
```
True
```

In [307]:
```
plt.figure(figsize = (16,8))
plt.title("Water Pumps Status",fontsize=20)
sns.countplot(x = labels['status_group'], data = labels, palette="BuPu");
```



OSEMN-Scrub

A quick look at the data shows us there are quite a few wells in need of repair, removal or relocation. Let's go a little further by removing missing values and changing some of our data types so we can have a clearer picture of wha we'll be dealing with, and make a little less work for our model later on!

In [308]:
```
df = train.merge(labels, on = train.index)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59400 entries, 0 to 59399
Data columns (total 43 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   key_0                  59400 non-null  int64  
 1   id_x                   59400 non-null  int64  
 2   amount_tsh             59400 non-null  float64
 3   date_recorded          59400 non-null  object 
 4   funder                 55765 non-null  object 
 5   gps_height             59400 non-null  int64  
 6   installer              55745 non-null  object 
 7   longitude              59400 non-null  float64
 8   latitude               59400 non-null  float64
 9   wpt_name               59400 non-null  object 
 10  num_private            59400 non-null  int64  
 11  basin                  59400 non-null  object 
 12  subvillage             59029 non-null  object 
 13  region                 59400 non-null  object 
 14  region_code            59400 non-null  int64  
 15  district_code          59400 non-null  int64  
 16  lga                    59400 non-null  object 
 17  ward                   59400 non-null  object 
 18  population             59400 non-null  int64  
 19  public_meeting         56066 non-null  object 
 20  recorded_by            59400 non-null  object 
 21  scheme_management      55523 non-null  object 
 22  scheme_name            31234 non-null  object 
 23  permit                 56344 non-null  object 
 24  construction_year      59400 non-null  int64  
 25  extraction_type        59400 non-null  object 
 26  extraction_type_group  59400 non-null  object 
 27  extraction_type_class  59400 non-null  object 
 28  management             59400 non-null  object 
 29  management_group       59400 non-null  object 
 30  payment                59400 non-null  object 
 31  payment_type           59400 non-null  object 
 32  water_quality          59400 non-null  object 
 33  quality_group          59400 non-null  object 
 34  quantity               59400 non-null  object 
 35  quantity_group         59400 non-null  object 
 36  source                 59400 non-null  object 
 37  source_type            59400 non-null  object 
 38  source_class           59400 non-null  object 
 39  waterpoint_type        59400 non-null  object 
 40  waterpoint_type_group  59400 non-null  object 
 41  id_y                   59400 non-null  int64  
 42  status_group           59400 non-null  object 
dtypes: float64(3), int64(9), object(31)
memory usage: 19.9+ MB
```

In [309]:
```
!pip install category_encoders
!pip install geopandas
import geopandas
```

```
Requirement already satisfied: category_encoders in ./opt/anaconda3/envs/
learn-env/lib/python3.8/site-packages (2.2.2)
Requirement already satisfied: pandas>=0.21.1 in ./opt/anaconda3/envs/lea
rn-env/lib/python3.8/site-packages (from category_encoders) (1.1.3)
Requirement already satisfied: statsmodels>=0.9.0 in ./opt/anaconda3/env
s/learn-env/lib/python3.8/site-packages (from category_encoders) (0.12.0)
Requirement already satisfied: scipy>=1.0.0 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from category_encoders) (1.5.2)
Requirement already satisfied: patsy>=0.5.1 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from category_encoders) (0.5.1)
Requirement already satisfied: numpy>=1.14.0 in ./opt/anaconda3/envs/lear
n-env/lib/python3.8/site-packages (from category_encoders) (1.18.5)
Requirement already satisfied: scikit-learn>=0.20.0 in ./opt/anaconda3/en
vs/learn-env/lib/python3.8/site-packages (from category_encoders) (0.23.
2)
Requirement already satisfied: python-dateutil>=2.7.3 in ./opt/anaconda3/
envs/learn-env/lib/python3.8/site-packages (from pandas>=0.21.1->category
_encoders) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from pandas>=0.21.1->category_encoders)
(2020.1)
Requirement already satisfied: six in ./opt/anaconda3/envs/learn-env/lib/
python3.8/site-packages (from patsy>=0.5.1->category_encoders) (1.15.0)
Requirement already satisfied: joblib>=0.11 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from scikit-learn>=0.20.0->category_enc
oders) (0.17.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in ./opt/anaconda3/en
vs/learn-env/lib/python3.8/site-packages (from scikit-learn>=0.20.0->cate
gory_encoders) (2.1.0)
Requirement already satisfied: geopandas in ./opt/anaconda3/envs/learn-en
v/lib/python3.8/site-packages (0.8.2)
Requirement already satisfied: fiona in ./opt/anaconda3/envs/learn-env/li
b/python3.8/site-packages (from geopandas) (1.8.18)
Requirement already satisfied: shapely in ./opt/anaconda3/envs/learn-env/
lib/python3.8/site-packages (from geopandas) (1.7.1)
Requirement already satisfied: pandas>=0.23.0 in ./opt/anaconda3/envs/lea
rn-env/lib/python3.8/site-packages (from geopandas) (1.1.3)
Requirement already satisfied: pyproj>=2.2.0 in ./opt/anaconda3/envs/lear
n-env/lib/python3.8/site-packages (from geopandas) (3.0.0.post1)
Requirement already satisfied: cligj>=0.5 in ./opt/anaconda3/envs/learn-e
nv/lib/python3.8/site-packages (from fiona->geopandas) (0.7.1)
Requirement already satisfied: munch in ./opt/anaconda3/envs/learn-env/li
b/python3.8/site-packages (from fiona->geopandas) (2.5.0)
Requirement already satisfied: attrs>=17 in ./opt/anaconda3/envs/learn-en
v/lib/python3.8/site-packages (from fiona->geopandas) (20.2.0)
Requirement already satisfied: click-plugins>=1.0 in ./opt/anaconda3/env
s/learn-env/lib/python3.8/site-packages (from fiona->geopandas) (1.1.1)
Requirement already satisfied: click<8,>=4.0 in ./opt/anaconda3/envs/lear
n-env/lib/python3.8/site-packages (from fiona->geopandas) (7.1.2)
Requirement already satisfied: six>=1.7 in ./opt/anaconda3/envs/learn-en
v/lib/python3.8/site-packages (from fiona->geopandas) (1.15.0)
Requirement already satisfied: certifi in ./opt/anaconda3/envs/learn-env/
```

```
lib/python3.8/site-packages (from fiona->geopandas) (2020.6.20)
Requirement already satisfied: python-dateutil>=2.7.3 in ./opt/anaconda3/
envs/learn-env/lib/python3.8/site-packages (from pandas>=0.23.0->geopanda
s) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from pandas>=0.23.0->geopandas) (2020.
1)
Requirement already satisfied: numpy>=1.15.4 in ./opt/anaconda3/envs/lear
n-env/lib/python3.8/site-packages (from pandas>=0.23.0->geopandas) (1.18.
5)
```

In [310]:

```
train['construction_year'].median()
```

Out[310]:

```
1986.0
```

In [311]:

```
train['construction_year'] = train['construction_year'].replace({0:1986})
train['age'] = train['date_recorded'].astype(str).str[:4].astype(int) - tra
train['pop/year'] = train['population'].replace({0:1}) / train['age'].repla
```

In [312]:

```
X_test = test
```

In [313]:

```
mean_year = full_df[full_df['construction_year']>0]['construction_year'].me
full_df.loc[full_df['construction_year']==0, 'construction_year'] = int(mea
X_test['age'] = X_test['date_recorded'].astype(str).str[:4].astype(int) - X
X_test['pop/year'] = X_test['population'].replace({0:1}) / X_test['age'].re
```

In [314]:
```
train.head(10)
```

Out[314]:

	id	amount_tsh	date_recorded	funder	gps_height	installer	longitude	latitude	wpt_
0	69572	6000.0	2011-03-14	Roman	1390	Roman	34.938093	-9.856322	
1	8776	0.0	2013-03-06	Grumeti	1399	GRUMETI	34.698766	-2.147466	Zal
2	34310	25.0	2013-02-25	Lottery Club	686	World vision	37.460664	-3.821329	Ma
3	67743	0.0	2013-01-28	Unicef	263	UNICEF	38.486161	-11.155298	Zahan Nany
4	19728	0.0	2011-07-13	Action In A	0	Artisan	31.130847	-1.825359	Sl
5	9944	20.0	2011-03-13	Mkinga Distric Coun	0	DWE	39.172796	-4.765587	
6	19816	0.0	2012-10-01	Dwsp	0	DWSP	33.362410	-3.766365	Ng
7	54551	0.0	2012-10-09	Rwssp	0	DWE	32.620617	-4.226198	Tushir
8	53934	0.0	2012-11-03	Wateraid	0	Water Aid	32.711100	-5.146712	Rama
9	46144	0.0	2011-08-03	Isingiro Ho	0	Artisan	30.626991	-1.257051	Kw

10 rows × 42 columns

In [315]:
```
train['water_/_person'] = train['amount_tsh'].replace({0:1}) / train['popul
```

In [316]:
```
X_test['water_/_person'] = X_test['amount_tsh'].replace({0:1}) / X_test['po
```

In [318]:
```
imputer = SimpleImputer(missing_values = np.nan, strategy = 'mean')
imputer.fit(features, labels['status_group'])
features = imputer.transform(features)
```

In [319]:
```
scaler = RobustScaler()
scaler.fit(features, labels['status_group'])
features = scaler.transform(features)
```

In [320]:
```
from pylab import rcParams
rcParams['figure.figsize'] = 30, 20
```

OSEMN-Model

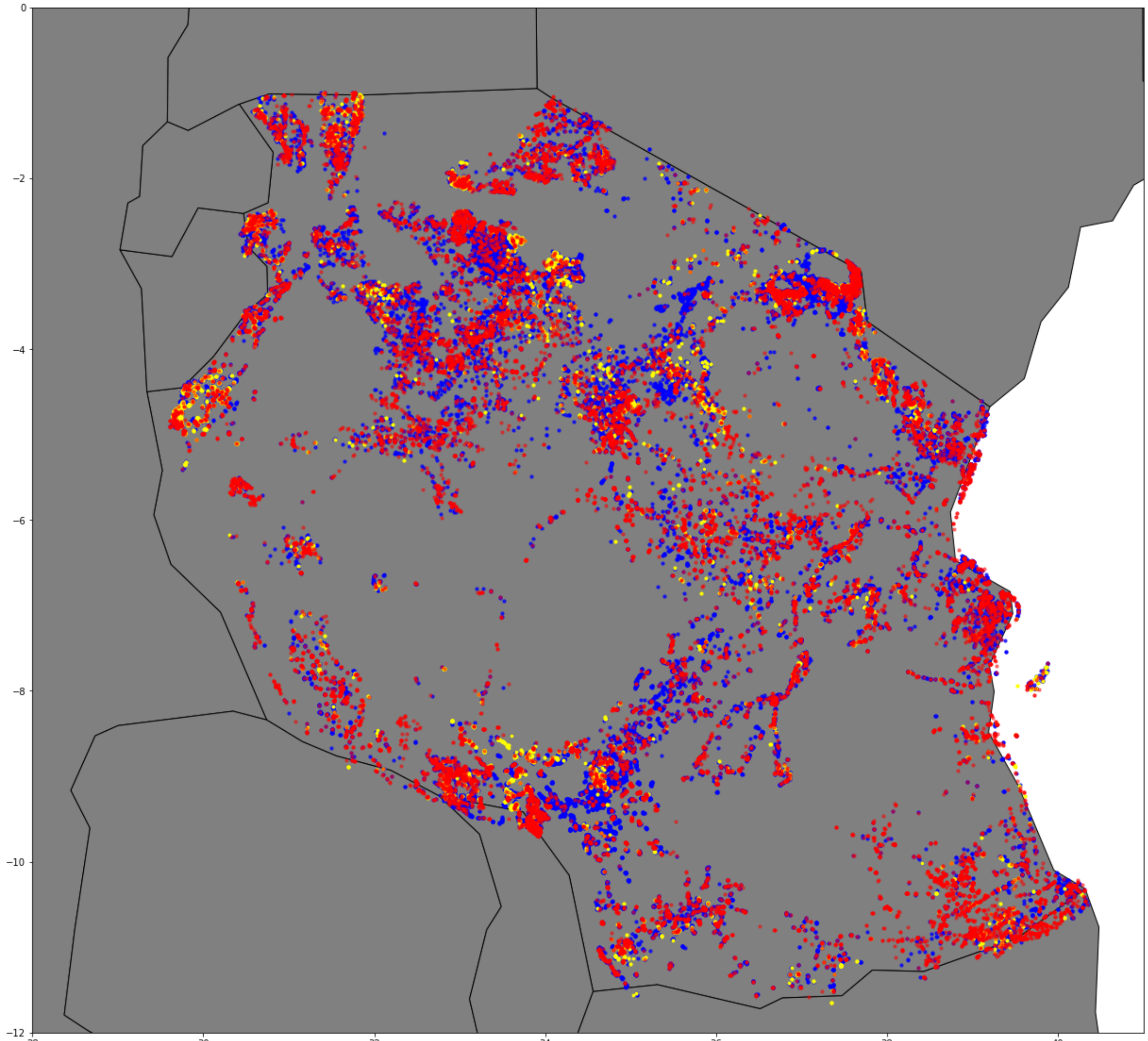
The model below is a rough scatterplot to showcase the different locations and functionoalities of the wells geographically, it also pints a much more interesting picture of how certain locations are not having their needs met, which would certainly be something worth exploring in future work.

We will also be trying our hand at some logistic regression, and tree classification later on, but first le's fit our data to better serve our purpose.

In [321]:
```
!pip install descartes
```

```
Requirement already satisfied: descartes in ./opt/anaconda3/envs/learn-en
v/lib/python3.8/site-packages (1.1.0)
Requirement already satisfied: matplotlib in ./opt/anaconda3/envs/learn-e
nv/lib/python3.8/site-packages (from descartes) (3.3.1)
Requirement already satisfied: kiwisolver>=1.0.1 in ./opt/anaconda3/envs/
learn-env/lib/python3.8/site-packages (from matplotlib->descartes) (1.2.
0)
Requirement already satisfied: python-dateutil>=2.1 in ./opt/anaconda3/en
vs/learn-env/lib/python3.8/site-packages (from matplotlib->descartes) (2.
8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 i
n ./opt/anaconda3/envs/learn-env/lib/python3.8/site-packages (from matplo
tlib->descartes) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in ./opt/anaconda3/envs/lear
n-env/lib/python3.8/site-packages (from matplotlib->descartes) (7.2.0)
Requirement already satisfied: numpy>=1.15 in ./opt/anaconda3/envs/learn-
env/lib/python3.8/site-packages (from matplotlib->descartes) (1.18.5)
Requirement already satisfied: cycler>=0.10 in ./opt/anaconda3/envs/learn
-env/lib/python3.8/site-packages (from matplotlib->descartes) (0.10.0)
Requirement already satisfied: certifi>=2020.06.20 in ./opt/anaconda3/env
s/learn-env/lib/python3.8/site-packages (from matplotlib->descartes) (202
0.6.20)
Requirement already satisfied: six>=1.5 in ./opt/anaconda3/envs/learn-en
v/lib/python3.8/site-packages (from python-dateutil>=2.1->matplotlib->des
cartes) (1.15.0)
```

In [322]:
```
gdf = geopandas.GeoDataFrame(
    df, geometry=geopandas.points_from_xy(df.longitude, df.latitude))

functional = gdf.where(labels['status_group'] == 'functional')
repair = gdf.where(labels['status_group'] == 'functional needs repair')
broken = gdf.where(labels['status_group'] == 'non functional')

world = geopandas.read_file(geopandas.datasets.get_path('naturalearth_lowre

ax = world[world.continent == 'Africa'].plot(
    color='grey', edgecolor='black')
ax.scatter(functional['longitude'], functional['latitude'],
           c='blue',alpha=.75, s=10)
ax.scatter(repair['longitude'], repair['latitude'],
           c='yellow', alpha=1, s=10)
ax.scatter(broken['longitude'], broken['latitude'],
           c='red', alpha=.5, s=10)

plt.ylim(-12, 0)
plt.xlim(28,41)

plt.show()

#blue is functional, yellow needs repair, red is broken
```



In [324]:
```
full_df = pd.concat([train, test])
full_df.shape
#59400 + 14358
```

Out[324]: (74250, 43)

In [325]:
```
full_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74250 entries, 0 to 14849
Data columns (total 43 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   id                     74250 non-null  int64  
 1   amount_tsh             74250 non-null  float64
 2   date_recorded          74250 non-null  object 
 3   funder                 69746 non-null  object 
 4   gps_height             74250 non-null  int64  
 5   installer              69718 non-null  object 
 6   longitude              74250 non-null  float64
 7   latitude               74250 non-null  float64
 8   wpt_name               74250 non-null  object 
 9   num_private            74250 non-null  int64  
 10  basin                  74250 non-null  object 
 11  subvillage             73780 non-null  object 
 12  region                 74250 non-null  object 
 13  region_code            74250 non-null  int64  
 14  district_code          74250 non-null  int64  
 15  lga                    74250 non-null  object 
 16  ward                   74250 non-null  object 
 17  population             74250 non-null  int64  
 18  public_meeting         70095 non-null  object 
 19  recorded_by            74250 non-null  object 
 20  scheme_management      69404 non-null  object 
 21  scheme_name            38992 non-null  object 
 22  permit                 70457 non-null  object 
 23  construction_year      74250 non-null  int64  
 24  extraction_type        74250 non-null  object 
 25  extraction_type_group  74250 non-null  object 
 26  extraction_type_class  74250 non-null  object 
 27  management             74250 non-null  object 
 28  management_group       74250 non-null  object 
 29  payment                74250 non-null  object 
 30  payment_type           74250 non-null  object 
 31  water_quality          74250 non-null  object 
 32  quality_group          74250 non-null  object 
 33  quantity               74250 non-null  object 
 34  quantity_group         74250 non-null  object 
 35  source                 74250 non-null  object 
 36  source_type            74250 non-null  object 
 37  source_class           74250 non-null  object 
 38  waterpoint_type        74250 non-null  object 
 39  waterpoint_type_group  74250 non-null  object 
 40  age                    74250 non-null  int64  
 41  pop/year               74250 non-null  float64
 42  water_/_person         74250 non-null  float64
dtypes: float64(5), int64(8), object(30)
memory usage: 24.9+ MB
```

In [326]:
```
for column in full_df.columns:
    full_df[column].fillna(full_df[column].mode()[0], inplace=True)
```

In [327]:
```
full_df.isna().sum()
```

Out[327]:
```
id                       0
amount_tsh               0
date_recorded            0
funder                   0
gps_height               0
installer                0
longitude                0
latitude                 0
wpt_name                 0
num_private              0
basin                    0
subvillage               0
region                   0
region_code              0
district_code            0
lga                      0
ward                     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
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
age                      0
pop/year                 0
water_/_person           0
dtype: int64
```

In [254]:
```
test = (test.drop(['date_recorded'], axis = 1),
test.drop(['construction_year'], axis = 1))
```

In [256]:
```
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
```

In [259]:
```
X_matrix, y_vector = train, labels['status_group']
```

In [260]:
```
majority_class = y_vector.mode()
y_vector.value_counts(normalize=True)
```

Out[260]:
```
functional                 0.543081
non functional             0.384242
functional needs repair    0.072677
Name: status_group, dtype: float64
```

In [261]:
```
majority_prediction = [majority_class] * len(y_vector)
accuracy_score(y_vector, majority_prediction)
```

Out[261]:
```
0.543080808080808
```

In [265]:
```
X_cleaned = full_df[:-14358]
X_test_cleaned = full_df[-14358:]
y = labels['status_group']
```

In [266]:
```
X_cleaned.shape, X_test_cleaned.shape, y.shape
```

Out[266]:
```
((59892, 43), (14358, 43), (59400,))
```

In [270]:
```
X_matrix.head()
y_vector.head()
```

Out[270]:
```
0        functional
1        functional
2        functional
3    non functional
4        functional
Name: status_group, dtype: object
```

In [271]:
```
X_matrix.dtypes
```

Out[271]:
```
id                         int64
amount_tsh               float64
date_recorded             object
funder                    object
gps_height                 int64
installer                 object
longitude                float64
latitude                 float64
wpt_name                  object
num_private                int64
basin                     object
subvillage                object
region                    object
region_code                int64
district_code              int64
lga                       object
ward                      object
population                 int64
public_meeting            object
recorded_by               object
scheme_management         object
scheme_name               object
permit                    object
construction_year          int64
extraction_type           object
extraction_type_group     object
extraction_type_class     object
management                object
management_group          object
payment                   object
payment_type              object
water_quality             object
quality_group             object
quantity                  object
quantity_group            object
source                    object
source_type               object
source_class              object
waterpoint_type           object
waterpoint_type_group     object
age                        int64
pop/year                 float64
water_/_person           float64
dtype: object
```

In [377]:
```
len(y_vector)
```

Out[377]:
```
14850
```

In [275]:
```
X_train_numeric = X_matrix.select_dtypes(np.number)
X_test_numeric = X_test.select_dtypes(np.number)
```

In [281]:
```
train.head()
```

Out[281]:

	id	amount_tsh	date_recorded	funder	gps_height	installer	longitude	latitude	wpt_na
0	69572	6000.0	2011-03-14	Roman	1390	Roman	34.938093	-9.856322	r
1	8776	0.0	2013-03-06	Grumeti	1399	GRUMETI	34.698766	-2.147466	Zaha
2	34310	25.0	2013-02-25	Lottery Club	686	World vision	37.460664	-3.821329	Mah
3	67743	0.0	2013-01-28	Unicef	263	UNICEF	38.486161	-11.155298	Zaha Nanyu
4	19728	0.0	2011-07-13	Action In A	0	Artisan	31.130847	-1.825359	Shu

5 rows × 43 columns

In [328]:
```
majority_class = labels['status_group'].mode()[0]

y_pred = np.full(shape = labels['status_group'].shape, fill_value = majorit
```

In [329]:
```
accuracy_score(labels['status_group'], y_pred)
```

Out[329]: 0.543080808080808

In [332]:
```
print(classification_report(labels['status_group'], y_pred))
#oof that could look a lot better...
```

```
                         precision    recall  f1-score   support

             functional       0.54      1.00      0.70     32259
functional needs repair       0.00      0.00      0.00      4317
         non functional       0.00      0.00      0.00     22824

               accuracy                           0.54     59400
              macro avg       0.18      0.33      0.23     59400
           weighted avg       0.29      0.54      0.38     59400
```

In [369]:
```
len(X_train_numeric)
```

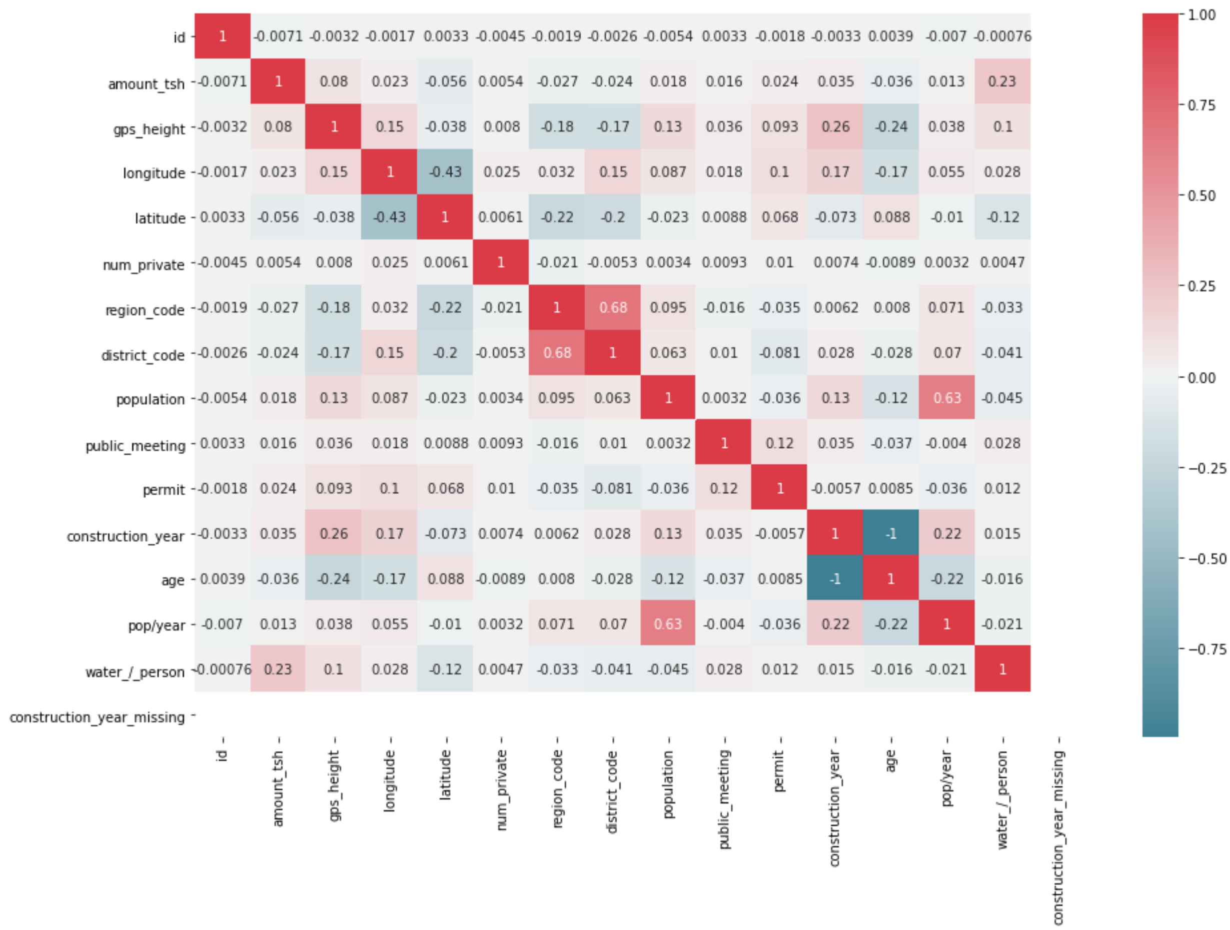
Out[369]: 59400

In [370]:
```
X_matrix, X_train_numeric, y, y_vector = train_test_split(X_train_numeric,
```

In [398]:
```
clf = tree.DecisionTreeClassifier()
clf.fit(X = X_matrix, y = y)
clf.feature_importances_
clf.score(X = X_test_numeric, y = y_vector)
```

Out[398]: 0.44195286195286193

In [423]:
```
def correlation_heatmap(labels):
    ax=plt.subplots(figsize=(15,10))
    colormap=sns.diverging_palette(220,10,as_cmap=True)
    sns.heatmap(full_df.corr(),annot=True,cmap=colormap)

correlation_heatmap(labels)
```



In [426]:
```
#new_df = pd.concat([train, labels])
new_df = pd.merge(train, labels, on = 'id')
```

In [421]:
```
new_df.head()
```

Out[421]:

ne	num_private	...	quantity_group	source	source_type	source_class	waterpoint_type	waterpoint
ne	0	...	enough	spring	spring	groundwater	communal standpipe	commu
ati	0	...	insufficient	rainwater harvesting	rainwater harvesting	surface	communal standpipe	commu
/a di	0	...	enough	dam	dam	surface	communal standpipe multiple	commu
ati /a ou	0	...	dry	machine dbh	borehole	groundwater	communal standpipe multiple	commu
ni	0	...	seasonal	rainwater harvesting	rainwater harvesting	surface	communal standpipe	commu

In [349]:
```
y.shape, X_matrix.shape
```

Out[349]: ((59400,), (59400, 42))

In [396]:
```
len(y_vector)
```

Out[396]: 14850

OSEMN-Interpret

Our model didn't exactly live up to our expectations, 44% would be worse than flipping a coin, especially considering that over 50% of the wells in Tanzania are currently operational. We did however get to take a look at geographic distributions, and took an opportunity to experiment with logistic regression and classification.

Unfortunately, the interpretation of this data would be more speculative than most any analysts would be comfortable with. The interpretation of this data could be most accurately described as needing further work, if we want to achieve actionable results. However, this will provide opportunities to practice ternary classification and predictive modeling, which can help solve future problems.

In [ ]: