Capstone 2 - Predicting Water Pump Condition in Tanzania Data Munging

September 8, 2019

1 Capstone 2 - Predicting Water Pump Condition in Tanzania Data Munging

Kenneth Liao

1.1 Background

The UN publishes and reviews a list of least developed countries (LDC) every 3 years. LDCs are "low-income countries confronting severe structural impediments to sustainable development. They are highly vulnerable to economic and environmental shocks and have low levels of human assets."[1]. Tanzania has been classified as an LDC since the UN published the first list of LDCs in 1971[2]. A common challenge of LDCs is a lack of infrastructure to support the development of the nation, including access to education and healthcare, waste management, and access to potable water.

According to UNICEF, as of 2017, more than 24 million Tanzanians lacked access to basic drinking water[3]. This corresponds to only 56.7% of the country's population having access to basic drinking water. Outside of developed urban areas, much of the potable water is accessed via water pumps.

Taarifa is an open-source platform for crowd-sourced reporting and triaging of infrastructure related issues. Together with the Tanzanian Ministry of Water, data has been collected for thousands of water pumps throughout Tanzania. The goal of this project is to be able to predict the condition of these water pumps to improve maintenance, reduce pump downtime, and ensure basic water access for millions of Tanzanians.

References

1. https://www.un.org/development/desa/dpad/least-developed-country-category.html
2. https://www.un.org/development/desa/dpad/wp-content/uploads/sites/45/publication/ldc_list.pdf
3. https://washwatch.org/en/countries/tanzania/summary/statistics/

1.1.1 Problem Description

Predict the operating condition of water pumps in Tanzania given various metadata on each water pump.

1.1.2 Strategy

The strategy will be to implement an XGBoost model as well as a neural network model for predictions and compare their performance.

1.1.3 Data

The dataset is provided by Taarifa, together with the Tanzanian Ministry of Water and is hosted by DrivenData.org:
https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/

1.2 Data Munging

```
In [1]: import pandas as pd
        import plotly.graph_objs as go
        from plotly.offline import iplot, plot, init_notebook_mode
        import plotly.express as px
        from config import credentials

        init_notebook_mode(connected=True)
```

```
In [2]: # load the data
        train = pd.read_csv('../data/train.csv')
        train_labels = pd.read_csv('../data/train-labels.csv')
```

I'll start by removing the unwanted feature columns we identified in the EDA part of the analysis. This includes duplicate, irrelevant, and single value columns.

```
In [3]: duplicated = ['recorded_by', 'payment_type', 'quantity_group']

        train_clean = train.drop(duplicated, axis=1)
        train_clean.columns
```

```
Out[3]: Index(['id', 'amount_tsh', 'date_recorded', 'funder', 'gps_height',
               'installer', 'longitude', 'latitude', 'wpt_name', 'num_private',
               'basin', 'subvillage', 'region', 'region_code', 'district_code', 'lga',
               'ward', 'population', 'public_meeting', 'scheme_management',
               'scheme_name', 'permit', 'construction_year', 'extraction_type',
               'extraction_type_group', 'extraction_type_class', 'management',
               'management_group', 'payment', 'water_quality', 'quality_group',
               'quantity', 'source', 'source_type', 'source_class', 'waterpoint_type',
               'waterpoint_type_group'],
              dtype='object')
```

```
In [4]: train_clean.set_index(['id', 'date_recorded'], inplace=True)
```

```
In [5]: train_clean.head()
```

Out[5]:
```
                  amount_tsh       funder  gps_height     installer  \
id    date_recorded
69572 2011-03-14      6000.0        Roman        1390         Roman
8776  2013-03-06         0.0      Grumeti        1399       GRUMETI
34310 2013-02-25        25.0  Lottery Club        686  World vision
67743 2013-01-28         0.0       Unicef         263        UNICEF
19728 2011-07-13         0.0   Action In A          0       Artisan

                  longitude   latitude              wpt_name  num_private  \
id    date_recorded
69572 2011-03-14  34.938093  -9.856322                  none            0
8776  2013-03-06  34.698766  -2.147466              Zahanati            0
34310 2013-02-25  37.460664  -3.821329           Kwa Mahundi            0
67743 2013-01-28  38.486161 -11.155298  Zahanati Ya Nanyumbu            0
19728 2011-07-13  31.130847  -1.825359               Shuleni            0

                                     basin  subvillage  \
id    date_recorded
69572 2011-03-14                Lake Nyasa    Mnyusi B
8776  2013-03-06             Lake Victoria     Nyamara
34310 2013-02-25                   Pangani     Majengo
67743 2013-01-28  Ruvuma / Southern Coast  Mahakamani
19728 2011-07-13             Lake Victoria  Kyanyamisa

                          ...          management_group         payment  \
id    date_recorded       ...
69572 2011-03-14          ...                user-group    pay annually
8776  2013-03-06          ...                user-group       never pay
34310 2013-02-25          ...                user-group  pay per bucket
67743 2013-01-28          ...                user-group       never pay
19728 2011-07-13          ...                     other       never pay

                  water_quality quality_group      quantity  \
id    date_recorded
69572 2011-03-14           soft          good        enough
8776  2013-03-06           soft          good  insufficient
34310 2013-02-25           soft          good        enough
67743 2013-01-28           soft          good           dry
19728 2011-07-13           soft          good      seasonal

                                 source           source_type source_class  \
id    date_recorded
69572 2011-03-14                 spring                spring  groundwater
8776  2013-03-06    rainwater harvesting  rainwater harvesting      surface
34310 2013-02-25                    dam                   dam      surface
67743 2013-01-28            machine dbh              borehole  groundwater
19728 2011-07-13    rainwater harvesting  rainwater harvesting      surface
```

```
                                waterpoint_type waterpoint_type_group
id    date_recorded
69572 2011-03-14             communal standpipe    communal standpipe
8776  2013-03-06             communal standpipe    communal standpipe
34310 2013-02-25    communal standpipe multiple    communal standpipe
67743 2013-01-28    communal standpipe multiple    communal standpipe
19728 2011-07-13             communal standpipe    communal standpipe

[5 rows x 35 columns]
```

Next, I need to convert the categorical text features into dummy variables.

```
In [6]: # list of all categorical variables
        cat_cols = []
        for col in train_clean.columns:
            if train_clean[col].dtype == 'object':
                cat_cols.append(col)
        cat_cols
```

```
Out[6]: ['funder',
         'installer',
         'wpt_name',
         'basin',
         'subvillage',
         'region',
         'lga',
         'ward',
         'public_meeting',
         'scheme_management',
         'scheme_name',
         'permit',
         'extraction_type',
         'extraction_type_group',
         'extraction_type_class',
         'management',
         'management_group',
         'payment',
         'water_quality',
         'quality_group',
         'quantity',
         'source',
         'source_type',
         'source_class',
         'waterpoint_type',
         'waterpoint_type_group']
```

```
In [7]: %%time
        cat_dummies = pd.get_dummies(train_clean[cat_cols], dummy_na=True)
```

```
Wall time: 1min 20s
```

I use `pd.get_dummies` with the argument dummy_na=True so that null values are not ignored. They are instead encoded the same as all other values so each feature will have a null dummy variable, indicated whether the sample was null or not for that feature. The resulting categorical feature set now has 65,828 features.

In [8]:
```
cat_dummies.head()
```

Out[8]:
```
                  funder_0  funder_A/co Germany  funder_Aar  \
id    date_recorded
69572 2011-03-14         0                    0           0
8776  2013-03-06         0                    0           0
34310 2013-02-25         0                    0           0
67743 2013-01-28         0                    0           0
19728 2011-07-13         0                    0           0

                  funder_Abas Ka  funder_Abasia  \
id    date_recorded
69572 2011-03-14               0              0
8776  2013-03-06               0              0
34310 2013-02-25               0              0
67743 2013-01-28               0              0
19728 2011-07-13               0              0

                  funder_Abc-ihushi Development Cent  funder_Abd  \
id    date_recorded
69572 2011-03-14                                   0           0
8776  2013-03-06                                   0           0
34310 2013-02-25                                   0           0
67743 2013-01-28                                   0           0
19728 2011-07-13                                   0           0

                  funder_Abdala  funder_Abddwe  funder_Abdul  \
id    date_recorded
69572 2011-03-14              0              0             0
8776  2013-03-06              0              0             0
34310 2013-02-25              0              0             0
67743 2013-01-28              0              0             0
19728 2011-07-13              0              0             0

                            ...              \
id    date_recorded         ...
69572 2011-03-14            ...
8776  2013-03-06            ...
34310 2013-02-25            ...
67743 2013-01-28            ...
```

```
19728 2011-07-13                ...

                     waterpoint_type_improved spring  waterpoint_type_other  \
id    date_recorded
69572 2011-03-14                                   0                      0
8776  2013-03-06                                   0                      0
34310 2013-02-25                                   0                      0
67743 2013-01-28                                   0                      0
19728 2011-07-13                                   0                      0

                     waterpoint_type_nan  waterpoint_type_group_cattle trough  \
id    date_recorded
69572 2011-03-14                       0                                    0
8776  2013-03-06                       0                                    0
34310 2013-02-25                       0                                    0
67743 2013-01-28                       0                                    0
19728 2011-07-13                       0                                    0

                     waterpoint_type_group_communal standpipe  \
id    date_recorded
69572 2011-03-14                                            1
8776  2013-03-06                                            1
34310 2013-02-25                                            1
67743 2013-01-28                                            1
19728 2011-07-13                                            1

                     waterpoint_type_group_dam  \
id    date_recorded
69572 2011-03-14                             0
8776  2013-03-06                             0
34310 2013-02-25                             0
67743 2013-01-28                             0
19728 2011-07-13                             0

                     waterpoint_type_group_hand pump  \
id    date_recorded
69572 2011-03-14                                   0
8776  2013-03-06                                   0
34310 2013-02-25                                   0
67743 2013-01-28                                   0
19728 2011-07-13                                   0

                     waterpoint_type_group_improved spring  \
id    date_recorded
69572 2011-03-14                                         0
8776  2013-03-06                                         0
34310 2013-02-25                                         0
67743 2013-01-28                                         0
```

```
19728 2011-07-13                                           0

                        waterpoint_type_group_other  waterpoint_type_group_nan
id    date_recorded
69572 2011-03-14                                  0                          0
8776  2013-03-06                                  0                          0
34310 2013-02-25                                  0                          0
67743 2013-01-28                                  0                          0
19728 2011-07-13                                  0                          0

[5 rows x 65828 columns]
```

In [9]:
```
# list of all numerical variables
num_cols = []
for col in train_clean.columns:
    if train_clean[col].dtype != 'object':
        num_cols.append(col)
num_cols
```

Out[9]:
```
['amount_tsh',
 'gps_height',
 'longitude',
 'latitude',
 'num_private',
 'region_code',
 'district_code',
 'population',
 'construction_year']
```

In [10]:
```
numerical = train_clean[num_cols]
numerical.head()
```

Out[10]:
```
                        amount_tsh  gps_height  longitude   latitude  \
id    date_recorded
69572 2011-03-14            6000.0        1390  34.938093  -9.856322
8776  2013-03-06               0.0        1399  34.698766  -2.147466
34310 2013-02-25              25.0         686  37.460664  -3.821329
67743 2013-01-28               0.0         263  38.486161 -11.155298
19728 2011-07-13               0.0           0  31.130847  -1.825359

                        num_private  region_code  district_code  population  \
id    date_recorded
69572 2011-03-14                  0           11              5         109
8776  2013-03-06                  0           20              2         280
34310 2013-02-25                  0           21              4         250
67743 2013-01-28                  0           90             63          58
19728 2011-07-13                  0           18              1           0

                        construction_year
```

```
id    date_recorded
69572 2011-03-14                  1999
8776  2013-03-06                  2010
34310 2013-02-25                  2009
67743 2013-01-28                  1986
19728 2011-07-13                     0
```

In [11]:
```python
numerical.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 59400 entries, (69572, 2011-03-14) to (26348, 2011-03-23)
Data columns (total 9 columns):
amount_tsh           59400 non-null float64
gps_height           59400 non-null int64
longitude            59400 non-null float64
latitude             59400 non-null float64
num_private          59400 non-null int64
region_code          59400 non-null int64
district_code        59400 non-null int64
population           59400 non-null int64
construction_year    59400 non-null int64
dtypes: float64(3), int64(6)
memory usage: 4.9+ MB
```

Luckily, none of the numerical columns have null values. We also don't need to normalize the numerical columns if using a tree-based model. However, for a neural network model, normalization will be necessary. I'll leave the data as-is for now and we can apply normalization when working with the neural network model specifically.

In [12]:
```python
# merge data back together.
```

In [13]:
```python
train_full = pd.concat([cat_dummies, numerical], axis=1)
train_full.head()
```

Out[13]:
```
                     funder_0  funder_A/co Germany  funder_Aar  \
id    date_recorded
69572 2011-03-14            0                    0           0
8776  2013-03-06            0                    0           0
34310 2013-02-25            0                    0           0
67743 2013-01-28            0                    0           0
19728 2011-07-13            0                    0           0

                     funder_Abas Ka  funder_Abasia  \
id    date_recorded
69572 2011-03-14                  0              0
8776  2013-03-06                  0              0
34310 2013-02-25                  0              0
67743 2013-01-28                  0              0
```

```
19728 2011-07-13                  0             0

                    funder_Abc-ihushi Development Cent  funder_Abd  \
id    date_recorded
69572 2011-03-14                                     0           0
8776  2013-03-06                                     0           0
34310 2013-02-25                                     0           0
67743 2013-01-28                                     0           0
19728 2011-07-13                                     0           0

                    funder_Abdala  funder_Abddwe  funder_Abdul  \
id    date_recorded
69572 2011-03-14                0              0             0
8776  2013-03-06                0              0             0
34310 2013-02-25                0              0             0
67743 2013-01-28                0              0             0
19728 2011-07-13                0              0             0

                          ...          waterpoint_type_group_nan  amount_tsh  \
id    date_recorded       ...
69572 2011-03-14          ...                                  0      6000.0
8776  2013-03-06          ...                                  0         0.0
34310 2013-02-25          ...                                  0        25.0
67743 2013-01-28          ...                                  0         0.0
19728 2011-07-13          ...                                  0         0.0

                    gps_height  longitude   latitude  num_private  \
id    date_recorded
69572 2011-03-14          1390  34.938093  -9.856322            0
8776  2013-03-06          1399  34.698766  -2.147466            0
34310 2013-02-25           686  37.460664  -3.821329            0
67743 2013-01-28           263  38.486161 -11.155298            0
19728 2011-07-13             0  31.130847  -1.825359            0

                    region_code  district_code  population  construction_year
id    date_recorded
69572 2011-03-14             11              5         109               1999
8776  2013-03-06             20              2         280               2010
34310 2013-02-25             21              4         250               2009
67743 2013-01-28             90             63          58               1986
19728 2011-07-13             18              1           0                  0

[5 rows x 65837 columns]
```

In [15]:
```
train_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 59400 entries, (69572, 2011-03-14) to (26348, 2011-03-23)
```

```
Columns: 65837 entries, funder_0 to construction_year
dtypes: float64(3), int64(6), uint8(65828)
memory usage: 3.6+ GB
```

```
In [19]: train_full.to_pickle('../data/train_full.pkl')
```

The full dataset is now ready to train on. There may be issues with the dimension of this dataset after converting to dummy variables. The shape of the dataset is now 59400 X 69572. If the model shows poor performance, it may benefit by using another model to reduce the number of features to those which are most important. This can be done with a number of techniques including PCA, step-wise feature selection, and genetic algorithms for feature selection.