# 01_import_inspect_convert

March 10, 2024

## 1 DengAI: Predicting Disease Spread

Predicting local epidemics of dengue fever to help fight life-threatening pandemics.

This project predicts the number of dengue fever cases reported each week in the following locations:

- San Juan (Puerto Rico)
- Iquitos (Peru)

The predictor variables include environmental variables describing changes in temperature, precipitation, vegitation, and more.

This is a time series project using **Random Forest** and **Negative Binomial** regression models to predict the total cases of Dengue fever over time.

## 2 Understand Problem

The DrivenData website has a useful Problem description.

Goal is to predict `total_cases` label for each `city`, `year`, `weekofyear` in test set.

This is a prediction problem with `3` target variables:

- `city`
- `year`
- `weekofyear`

### 2.1 `city`

Categorical variable with two levels:

- San Juan (Puerto Rico) recorded as `sj`
- Iquitos (Peru) recorded as `iq`

Missing values are recorded as `NaN`

## 3 Performance Metric

Performance evaluated using `mean absolute error` (MAE): Mean Absolute Error

## 4 Submission Format

A `.csv` file with the following columns:

- `city` (`sj` or `iq`)
- `year` (year)
- `weekofyear` (integer 1-52)
- `total_cases` (integer values)

## 5 Benchmark Walkthrough

There's a Benchmark Walkthrough guide, which I'll only check if I run into trouble.

## 6 Download Data

- Create an account on Driven Data
- Join the practice competition **DengAI: Predicting Disease Spread**
- Download the data from the data page.
- Move the files to `../data/`

## 7 Load Modules

[1]:
```
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

## 8 Inspect Data

## 9 Inspect Submission Format

The submission format is a `.csv` file:

- 417 rows
- 1st row is columns: `city`, `year`, `weekofyear`, `total_cases`

[2]:
```
!head -10 '../data/submission_format.csv'
```

```
city,year,weekofyear,total_cases
sj,2008,18,0
sj,2008,19,0
sj,2008,20,0
sj,2008,21,0
sj,2008,22,0
sj,2008,23,0
sj,2008,24,0
```

```
sj,2008,25,0
sj,2008,26,0
```

[3]:
```
!wc '../data/submission_format.csv'
```

```
 417  417 5369 ../data/submission_format.csv
```

## 10 Inspect Target Values (labels)

The `dengue_labels_train.csv` file:

- 1457 rows (cases)
- 1st row is columns: `city`, `year`, `weekofyear`, `total_cases` (same as `submission_format.csv`)

[4]:
```
!head -10 '../data/dengue_labels_train.csv'
```

```
city,year,weekofyear,total_cases
sj,1990,18,4
sj,1990,19,5
sj,1990,20,4
sj,1990,21,3
sj,1990,22,6
sj,1990,23,2
sj,1990,24,4
sj,1990,25,5
sj,1990,26,10
```

[5]:
```
!wc '../data/dengue_labels_train.csv'
```

```
1457  1457 19582 ../data/dengue_labels_train.csv
```

## 11 Inspect Training Set Features

- 1457 rows (same as `dengue_labels_train.csv`)
- Many columns! The first three are the same as the other files: `city`, `year`, `weekofyear`, excluding `total_cases`

[6]:
```
!head -10 '../data/dengue_features_train.csv'
```

```
city,year,weekofyear,week_start_date,ndvi_ne,ndvi_nw,ndvi_se,ndvi_sw,precipitati
on_amt_mm,reanalysis_air_temp_k,reanalysis_avg_temp_k,reanalysis_dew_point_temp_
k,reanalysis_max_air_temp_k,reanalysis_min_air_temp_k,reanalysis_precip_amt_kg_p
er_m2,reanalysis_relative_humidity_percent,reanalysis_sat_precip_amt_mm,reanalys
is_specific_humidity_g_per_kg,reanalysis_tdtr_k,station_avg_temp_c,station_diur_
temp_rng_c,station_max_temp_c,station_min_temp_c,station_precip_mm
sj,1990,18,1990-04-
30,0.1226,0.103725,0.1984833,0.1776167,12.42,297.572857143,297.742857143,292.414
285714,299.8,295.9,32.0,73.3657142857,12.42,14.0128571429,2.62857142857,25.44285
71429,6.9,29.4,20.0,16.0
```

```
sj,1990,19,1990-05-
07,0.1699,0.142175,0.1623571,0.1554857,22.82,298.211428571,298.442857143,293.951
428571,300.9,296.4,17.94,77.3685714286,22.82,15.3728571429,2.37142857143,26.7142
857143,6.37142857143,31.7,22.2,8.6
sj,1990,20,1990-05-
14,0.03225,0.1729667,0.1572,0.1708429,34.54,298.781428571,298.878571429,295.4342
85714,300.5,297.3,26.1,82.0528571429,34.54,16.8485714286,2.3,26.7142857143,6.485
71428571,32.2,22.8,41.4
sj,1990,21,1990-05-
21,0.1286333,0.2450667,0.2275571,0.2358857,15.36,298.987142857,299.228571429,295
.31,301.4,297.0,13.9,80.3371428571,15.36,16.6728571429,2.42857142857,27.47142857
14,6.77142857143,33.3,23.3,4.0
sj,1990,22,1990-05-
28,0.1962,0.2622,0.2512,0.24734,7.52,299.518571429,299.664285714,295.821428571,3
01.9,297.5,12.2,80.46,7.52,17.21,3.01428571429,28.9428571429,9.37142857143,35.0,
23.9,5.8
sj,1990,23,1990-06-
04,,0.17485,0.2543143,0.1817429,9.58,299.63,299.764285714,295.851428571,302.4,29
8.1,26.49,79.8914285714,9.58,17.2128571429,2.1,28.1142857143,6.94285714286,34.4,
23.9,39.1
sj,1990,24,1990-06-
11,0.1129,0.0928,0.2050714,0.2102714,3.48,299.207142857,299.221428571,295.865714
286,301.3,297.7,38.6,82.0,3.48,17.2342857143,2.04285714286,27.4142857143,6.77142
857143,32.2,23.3,29.7
sj,1990,25,1990-06-
18,0.0725,0.0725,0.1514714,0.1330286,151.12,299.591428571,299.528571429,296.5314
28571,300.6,298.4,30.0,83.3757142857,151.12,17.9771428571,1.57142857143,28.37142
85714,7.68571428571,33.9,22.8,21.1
sj,1990,26,1990-06-
25,0.10245,0.146175,0.1255714,0.1236,19.32,299.578571429,299.557142857,296.37857
1429,302.1,297.7,37.51,82.7685714286,19.32,17.79,1.88571428571,28.3285714286,7.3
8571428571,33.9,22.8,21.1
```

[7]:
```
!wc '../data/dengue_features_train.csv'
```

```
  1457   1457 287139 ../data/dengue_features_train.csv
```

## 12 Inspect Test Set

- 417 rows (same as `submission_format.csv`)
- Looks like the same columns as `dengue_features_train.csv`

[8]:
```
!head -10 '../data/dengue_features_test.csv'
```

```
city,year,weekofyear,week_start_date,ndvi_ne,ndvi_nw,ndvi_se,ndvi_sw,precipitati
on_amt_mm,reanalysis_air_temp_k,reanalysis_avg_temp_k,reanalysis_dew_point_temp_
k,reanalysis_max_air_temp_k,reanalysis_min_air_temp_k,reanalysis_precip_amt_kg_p
er_m2,reanalysis_relative_humidity_percent,reanalysis_sat_precip_amt_mm,reanalys
```

```
is_specific_humidity_g_per_kg,reanalysis_tdtr_k,station_avg_temp_c,station_diur_
temp_rng_c,station_max_temp_c,station_min_temp_c,station_precip_mm
sj,2008,18,2008-04-29,-0.0189,-
0.0189,0.1027286,0.0912,78.6,298.492857143,298.55,294.527142857,301.1,296.4,25.3
7,78.7814285714,78.6,15.9185714286,3.12857142857,26.5285714286,7.05714285714,33.
3,21.7,75.2
sj,2008,19,2008-05-06,-0.018,-
0.0124,0.08204286,0.07231429,12.56,298.475714286,298.557142857,294.395714286,300
.8,296.7,21.83,78.23,12.56,15.7914285714,2.57142857143,26.0714285714,5.557142857
14,30.0,22.2,34.3
sj,2008,20,2008-05-13,-
0.0015,,0.1510833,0.09152857,3.66,299.455714286,299.357142857,295.308571429,302.
2,296.4,4.12,78.27,3.66,16.6742857143,4.42857142857,27.9285714286,7.78571428571,
32.8,22.8,3.0
sj,2008,21,2008-05-20,,-
0.01986667,0.1243286,0.1256857,0.0,299.69,299.728571429,294.402857143,303.0,296.
9,2.2,73.0157142857,0.0,15.7757142857,4.34285714286,28.0571428571,6.27142857143,
33.3,24.4,0.3
sj,2008,22,2008-05-
27,0.0568,0.03983333,0.06226667,0.07591429,0.76,299.78,299.671428571,294.76,302.
3,297.3,4.36,74.0842857143,0.76,16.1371428571,3.54285714286,27.6142857143,7.0857
1428571,33.3,23.3,84.1
sj,2008,23,2008-06-03,-0.044,-
0.03046667,0.132,0.08352857,71.17,299.768571429,299.728571429,295.314285714,301.
9,297.6,22.55,76.5571428571,71.17,16.6671428571,2.85714285714,28.0,5.17142857143
,32.8,25.0,27.7
sj,2008,24,2008-06-10,-0.0443,-
0.024925,0.1322714,0.1591571,48.99,300.062857143,300.007142857,295.65,302.4,297.
5,13.1,76.8442857143,48.99,17.01,3.15714285714,27.4,6.04285714286,31.1,23.3,91.7
sj,2008,25,2008-06-
17,,0.08215,0.1443714,0.1167286,30.81,300.484285714,300.578571429,295.997142857,
303.5,297.5,7.2,76.87,30.81,17.42,3.9,28.7571428571,6.98571428571,34.4,24.4,0.3
sj,2008,26,2008-06-
24,0.0108,0.0499,0.1005714,0.1173286,8.02,300.601428571,300.621428571,296.268571
429,302.5,298.5,17.1,77.3957142857,8.02,17.6785714286,2.78571428571,28.657142857
1,6.24285714286,32.8,23.9,28.7
```

[9]:
```
!wc '../data/dengue_features_test.csv'
```

```
  417   417 82465 ../data/dengue_features_test.csv
```

## 13 Import Data

[10]:
```
submission_format = pd.read_csv('../data/submission_format.csv')
train_features = pd.read_csv('../data/dengue_features_train.csv')
train_labels = pd.read_csv('../data/dengue_labels_train.csv')
test_features = pd.read_csv('../data/dengue_features_test.csv')
```

## 14 Inspect Data with Pandas

- `year` and `weekofyear` have same summaries for `train_features` and `train_labels`
- there are many columns (features):

```
['city', 'year', 'weekofyear', 'week_start_date', 'ndvi_ne', 'ndvi_nw',
       'ndvi_se', 'ndvi_sw', 'precipitation_amt_mm', 'reanalysis_air_temp_k',
       'reanalysis_avg_temp_k', 'reanalysis_dew_point_temp_k',
       'reanalysis_max_air_temp_k', 'reanalysis_min_air_temp_k',
       'reanalysis_precip_amt_kg_per_m2',
       'reanalysis_relative_humidity_percent', 'reanalysis_sat_precip_amt_mm',
       'reanalysis_specific_humidity_g_per_kg', 'reanalysis_tdtr_k',
       'station_avg_temp_c', 'station_diur_temp_rng_c', 'station_max_temp_c',
       'station_min_temp_c', 'station_precip_mm'],
```

[11]:
```
train_features
```

[11]:
```
      city  year  weekofyear week_start_date   ndvi_ne   ndvi_nw   ndvi_se  \
0       sj  1990          18      1990-04-30  0.122600  0.103725  0.198483
1       sj  1990          19      1990-05-07  0.169900  0.142175  0.162357
2       sj  1990          20      1990-05-14  0.032250  0.172967  0.157200
3       sj  1990          21      1990-05-21  0.128633  0.245067  0.227557
4       sj  1990          22      1990-05-28  0.196200  0.262200  0.251200
…    …   …  …          …              …       …         …         …
1451    iq  2010          21      2010-05-28  0.342750  0.318900  0.256343
1452    iq  2010          22      2010-06-04  0.160157  0.160371  0.136043
1453    iq  2010          23      2010-06-11  0.247057  0.146057  0.250357
1454    iq  2010          24      2010-06-18  0.333914  0.245771  0.278886
1455    iq  2010          25      2010-06-25  0.298186  0.232971  0.274214

        ndvi_sw  precipitation_amt_mm  reanalysis_air_temp_k  …  \
0      0.177617                 12.42             297.572857  …
1      0.155486                 22.82             298.211429  …
2      0.170843                 34.54             298.781429  …
3      0.235886                 15.36             298.987143  …
4      0.247340                  7.52             299.518571  …
…         …                     …                      …  …
1451   0.292514                 55.30             299.334286  …
1452   0.225657                 86.47             298.330000  …
1453   0.233714                 58.94             296.598571  …
1454   0.325486                 59.67             296.345714  …
1455   0.315757                 63.22             298.097143  …

       reanalysis_precip_amt_kg_per_m2  reanalysis_relative_humidity_percent  \
0                                32.00                             73.365714
1                                17.94                             77.368571
2                                26.10                             82.052857
3                                13.90                             80.337143
```

```
4                                 12.20                                 80.460000
…                                    …                                      …
1451                              45.00                                 88.765714
1452                             207.10                                 91.600000
1453                              50.60                                 94.280000
1454                              62.33                                 94.660000
1455                              36.90                                 89.082857

      reanalysis_sat_precip_amt_mm  reanalysis_specific_humidity_g_per_kg  \
0                            12.42                                 14.012857
1                            22.82                                 15.372857
2                            34.54                                 16.848571
3                            15.36                                 16.672857
4                             7.52                                 17.210000
…                                …                                      …
1451                         55.30                                 18.485714
1452                         86.47                                 18.070000
1453                         58.94                                 17.008571
1454                         59.67                                 16.815714
1455                         63.22                                 17.355714

      reanalysis_tdtr_k  station_avg_temp_c  station_diur_temp_rng_c  \
0              2.628571           25.442857                 6.900000
1              2.371429           26.714286                 6.371429
2              2.300000           26.714286                 6.485714
3              2.428571           27.471429                 6.771429
4              3.014286           28.942857                 9.371429
…                 …                  …                        …
1451           9.800000           28.633333                11.933333
1452           7.471429           27.433333                10.500000
1453           7.500000           24.400000                 6.900000
1454           7.871429           25.433333                 8.733333
1455          11.014286           27.475000                 9.900000

      station_max_temp_c  station_min_temp_c  station_precip_mm
0                   29.4                20.0               16.0
1                   31.7                22.2                8.6
2                   32.2                22.8               41.4
3                   33.3                23.3                4.0
4                   35.0                23.9                5.8
…                      …                   …                  …
1451                35.4                22.4               27.0
1452                34.7                21.7               36.6
1453                32.2                19.2                7.4
1454                31.2                21.0               16.0
1455                33.7                22.2               20.4
```

```
[1456 rows x 24 columns]
```

[12]:
```
submission_format.describe()
```

[12]:
```
              year  weekofyear  total_cases
count   416.000000  416.000000        416.0
mean   2010.766827   26.439904          0.0
std       1.434835   14.978257          0.0
min    2008.000000    1.000000          0.0
25%    2010.000000   13.750000          0.0
50%    2011.000000   26.000000          0.0
75%    2012.000000   39.000000          0.0
max    2013.000000   53.000000          0.0
```

[13]:
```
submission_format.columns
```

[13]:
```
Index(['city', 'year', 'weekofyear', 'total_cases'], dtype='object')
```

[14]:
```
train_features.describe()
```

[14]:
```
              year   weekofyear      ndvi_ne      ndvi_nw      ndvi_se  \
count  1456.000000  1456.000000  1262.000000  1404.000000  1434.000000
mean   2001.031593    26.503434     0.142294     0.130553     0.203783
std       5.408314    15.019437     0.140531     0.119999     0.073860
min    1990.000000     1.000000    -0.406250    -0.456100    -0.015533
25%    1997.000000    13.750000     0.044950     0.049217     0.155087
50%    2002.000000    26.500000     0.128817     0.121429     0.196050
75%    2005.000000    39.250000     0.248483     0.216600     0.248846
max    2010.000000    53.000000     0.508357     0.454429     0.538314

           ndvi_sw  precipitation_amt_mm  reanalysis_air_temp_k  \
count  1434.000000           1443.000000            1446.000000
mean      0.202305             45.760388             298.701852
std       0.083903             43.715537               1.362420
min      -0.063457              0.000000             294.635714
25%       0.144209              9.800000             297.658929
50%       0.189450             38.340000             298.646429
75%       0.246982             70.235000             299.833571
max       0.546017            390.600000             302.200000

       reanalysis_avg_temp_k  reanalysis_dew_point_temp_k  …  \
count            1446.000000                  1446.000000  …
mean              299.225578                   295.246356  …
std                 1.261715                     1.527810  …
min               294.892857                   289.642857  …
25%               298.257143                   294.118929  …
50%               299.289286                   295.640714  …
```

```
75%              300.207143                  296.460000  …
max              302.928571                  298.450000  …

       reanalysis_precip_amt_kg_per_m2  reanalysis_relative_humidity_percent  \
count                      1446.000000                           1446.000000
mean                         40.151819                             82.161959
std                          43.434399                              7.153897
min                           0.000000                             57.787143
25%                          13.055000                             77.177143
50%                          27.245000                             80.301429
75%                          52.200000                             86.357857
max                         570.500000                             98.610000

       reanalysis_sat_precip_amt_mm  reanalysis_specific_humidity_g_per_kg  \
count                   1443.000000                            1446.000000
mean                      45.760388                              16.746427
std                       43.715537                               1.542494
min                        0.000000                              11.715714
25%                        9.800000                              15.557143
50%                       38.340000                              17.087143
75%                       70.235000                              17.978214
max                      390.600000                              20.461429

       reanalysis_tdtr_k  station_avg_temp_c  station_diur_temp_rng_c  \
count        1446.000000         1413.000000              1413.000000
mean            4.903754           27.185783                 8.059328
std             3.546445            1.292347                 2.128568
min             1.357143           21.400000                 4.528571
25%             2.328571           26.300000                 6.514286
50%             2.857143           27.414286                 7.300000
75%             7.625000           28.157143                 9.566667
max            16.028571           30.800000                15.800000

       station_max_temp_c  station_min_temp_c  station_precip_mm
count         1436.000000         1442.000000       1434.000000
mean            32.452437           22.102150         39.326360
std              1.959318            1.574066         47.455314
min             26.700000           14.700000          0.000000
25%             31.100000           21.100000          8.700000
50%             32.800000           22.200000         23.850000
75%             33.900000           23.300000         53.900000
max             42.200000           25.600000        543.300000

[8 rows x 22 columns]
```

[15]:
```
train_features.columns
```

[15]:
```
Index(['city', 'year', 'weekofyear', 'week_start_date', 'ndvi_ne', 'ndvi_nw',
       'ndvi_se', 'ndvi_sw', 'precipitation_amt_mm', 'reanalysis_air_temp_k',
       'reanalysis_avg_temp_k', 'reanalysis_dew_point_temp_k',
       'reanalysis_max_air_temp_k', 'reanalysis_min_air_temp_k',
       'reanalysis_precip_amt_kg_per_m2',
       'reanalysis_relative_humidity_percent', 'reanalysis_sat_precip_amt_mm',
       'reanalysis_specific_humidity_g_per_kg', 'reanalysis_tdtr_k',
       'station_avg_temp_c', 'station_diur_temp_rng_c', 'station_max_temp_c',
       'station_min_temp_c', 'station_precip_mm'],
      dtype='object')
```

[16]:
```
train_features.dtypes
```

[16]:
```
city                                    object
year                                     int64
weekofyear                               int64
week_start_date                         object
ndvi_ne                                float64
ndvi_nw                                float64
ndvi_se                                float64
ndvi_sw                                float64
precipitation_amt_mm                   float64
reanalysis_air_temp_k                  float64
reanalysis_avg_temp_k                  float64
reanalysis_dew_point_temp_k            float64
reanalysis_max_air_temp_k              float64
reanalysis_min_air_temp_k              float64
reanalysis_precip_amt_kg_per_m2        float64
reanalysis_relative_humidity_percent   float64
reanalysis_sat_precip_amt_mm           float64
reanalysis_specific_humidity_g_per_kg  float64
reanalysis_tdtr_k                      float64
station_avg_temp_c                     float64
station_diur_temp_rng_c                float64
station_max_temp_c                     float64
station_min_temp_c                     float64
station_precip_mm                      float64
dtype: object
```

[17]:
```
train_labels.describe()
```

[17]:

| | year | weekofyear | total_cases |
|---|---|---|---|
| count | 1456.000000 | 1456.000000 | 1456.000000 |
| mean | 2001.031593 | 26.503434 | 24.675137 |
| std | 5.408314 | 15.019437 | 43.596000 |
| min | 1990.000000 | 1.000000 | 0.000000 |
| 25% | 1997.000000 | 13.750000 | 5.000000 |

```
50%    2002.000000    26.500000    12.000000
75%    2005.000000    39.250000    28.000000
max    2010.000000    53.000000   461.000000
```

[18]:
```
train_labels.columns
```

[18]:
```
Index(['city', 'year', 'weekofyear', 'total_cases'], dtype='object')
```

[19]:
```
test_features.describe()
```

[19]:
```
              year  weekofyear     ndvi_ne     ndvi_nw     ndvi_se  \
count   416.000000  416.000000  373.000000  405.000000  415.000000
mean   2010.766827   26.439904    0.126050    0.126803    0.207702
std       1.434835   14.978257    0.164353    0.141420    0.079102
min    2008.000000    1.000000   -0.463400   -0.211800    0.006200
25%    2010.000000   13.750000   -0.001500    0.015975    0.148670
50%    2011.000000   26.000000    0.110100    0.088700    0.204171
75%    2012.000000   39.000000    0.263329    0.242400    0.254871
max    2013.000000   53.000000    0.500400    0.649000    0.453043

          ndvi_sw  precipitation_amt_mm  reanalysis_air_temp_k  \
count  415.000000            414.000000             414.000000
mean     0.201721             38.354324             298.818295
std      0.092028             35.171126               1.469501
min     -0.014671              0.000000             294.554286
25%      0.134079              8.175000             297.751429
50%      0.186471             31.455000             298.547143
75%      0.253243             57.772500             300.240357
max      0.529043            169.340000             301.935714

       reanalysis_avg_temp_k  reanalysis_dew_point_temp_k  …  \
count             414.000000                   414.000000  …
mean              299.353071                   295.419179  …
std                 1.306233                     1.523099  …
min               295.235714                   290.818571  …
25%               298.323214                   294.335714  …
50%               299.328571                   295.825000  …
75%               300.521429                   296.643571  …
max               303.328571                   297.794286  …

       reanalysis_precip_amt_kg_per_m2  reanalysis_relative_humidity_percent  \
count                       414.000000                            414.000000
mean                         42.171135                             82.499810
std                          48.909514                              7.378243
min                           0.000000                             64.920000
25%                           9.430000                             77.397143
50%                          25.850000                             80.330000
```

```
75%                          56.475000                                88.328929
max                         301.400000                                97.982857

       reanalysis_sat_precip_amt_mm  reanalysis_specific_humidity_g_per_kg  \
count                    414.000000                             414.000000
mean                      38.354324                              16.927088
std                       35.171126                               1.557868
min                        0.000000                              12.537143
25%                        8.175000                              15.792857
50%                       31.455000                              17.337143
75%                       57.772500                              18.174643
max                      169.340000                              19.598571

       reanalysis_tdtr_k  station_avg_temp_c  station_diur_temp_rng_c  \
count         414.000000          404.000000               404.000000
mean            5.124569           27.369587                 7.810991
std             3.542870            1.232608                 2.449718
min             1.485714           24.157143                 4.042857
25%             2.446429           26.514286                 5.928571
50%             2.914286           27.483333                 6.642857
75%             8.171429           28.319048                 9.812500
max            14.485714           30.271429                14.725000

       station_max_temp_c  station_min_temp_c  station_precip_mm
count          413.000000          407.000000         411.000000
mean            32.534625           22.368550          34.278589
std              1.920429            1.731437          34.655966
min             27.200000           14.200000           0.000000
25%             31.100000           21.200000           9.100000
50%             32.800000           22.200000          23.600000
75%             33.900000           23.300000          47.750000
max             38.400000           26.700000         212.000000

[8 rows x 22 columns]
```

[20]:
```
test_features.columns
```

[20]:
```
Index(['city', 'year', 'weekofyear', 'week_start_date', 'ndvi_ne', 'ndvi_nw',
       'ndvi_se', 'ndvi_sw', 'precipitation_amt_mm', 'reanalysis_air_temp_k',
       'reanalysis_avg_temp_k', 'reanalysis_dew_point_temp_k',
       'reanalysis_max_air_temp_k', 'reanalysis_min_air_temp_k',
       'reanalysis_precip_amt_kg_per_m2',
       'reanalysis_relative_humidity_percent', 'reanalysis_sat_precip_amt_mm',
       'reanalysis_specific_humidity_g_per_kg', 'reanalysis_tdtr_k',
       'station_avg_temp_c', 'station_diur_temp_rng_c', 'station_max_temp_c',
       'station_min_temp_c', 'station_precip_mm'],
      dtype='object')
```

## 15 Inspect NA Values

We examined `train_features` and `train_labels` for NA (Not Available) values

We discovered:

- `NA` values for all features except `city`, `year`, `weekofyear`, `week_start_date`
- Total `NA` sum in `train_features`: `548`
- Total `NA` sum in `test_features`: `119`
- Largest `NA` count for `ndvi_ne` (Pixedl northweast of city centroid)
- Range between `10` and `194` `NA` values

Without domain knowledge, it's difficult to know if these `NA` values matter!

Let's plot `NA` values

[21]:
```
train_features.isna().sum().sum()
```

[21]:
```
548
```

[22]:
```
train_features.isna().sum().sort_values()
```

[22]:
```
city                                    0
year                                    0
weekofyear                              0
week_start_date                         0
reanalysis_tdtr_k                      10
reanalysis_specific_humidity_g_per_kg  10
reanalysis_relative_humidity_percent   10
reanalysis_precip_amt_kg_per_m2        10
reanalysis_min_air_temp_k              10
reanalysis_max_air_temp_k              10
reanalysis_dew_point_temp_k            10
reanalysis_air_temp_k                  10
reanalysis_avg_temp_k                  10
precipitation_amt_mm                   13
reanalysis_sat_precip_amt_mm           13
station_min_temp_c                     14
station_max_temp_c                     20
ndvi_sw                                22
ndvi_se                                22
station_precip_mm                      22
station_avg_temp_c                     43
station_diur_temp_rng_c                43
ndvi_nw                                52
ndvi_ne                               194
dtype: int64
```

[23]:
```
train_labels.isna().sum()
```

[23]:
```
city           0
year           0
weekofyear     0
total_cases    0
dtype: int64
```

[24]:
```
test_features.isna().sum().sum()
```

[24]:
```
119
```

[25]:
```
test_features.isna().sum().sort_values()
```

[25]:
```
city                                   0
year                                   0
weekofyear                             0
week_start_date                        0
ndvi_se                                1
ndvi_sw                                1
reanalysis_tdtr_k                      2
reanalysis_specific_humidity_g_per_kg  2
reanalysis_sat_precip_amt_mm           2
reanalysis_relative_humidity_percent   2
reanalysis_precip_amt_kg_per_m2        2
reanalysis_min_air_temp_k              2
reanalysis_dew_point_temp_k            2
reanalysis_avg_temp_k                  2
reanalysis_air_temp_k                  2
precipitation_amt_mm                   2
reanalysis_max_air_temp_k              2
station_max_temp_c                     3
station_precip_mm                      5
station_min_temp_c                     9
ndvi_nw                               11
station_avg_temp_c                    12
station_diur_temp_rng_c               12
ndvi_ne                               43
dtype: int64
```

## 16 Plot Heatmap of Missing Values

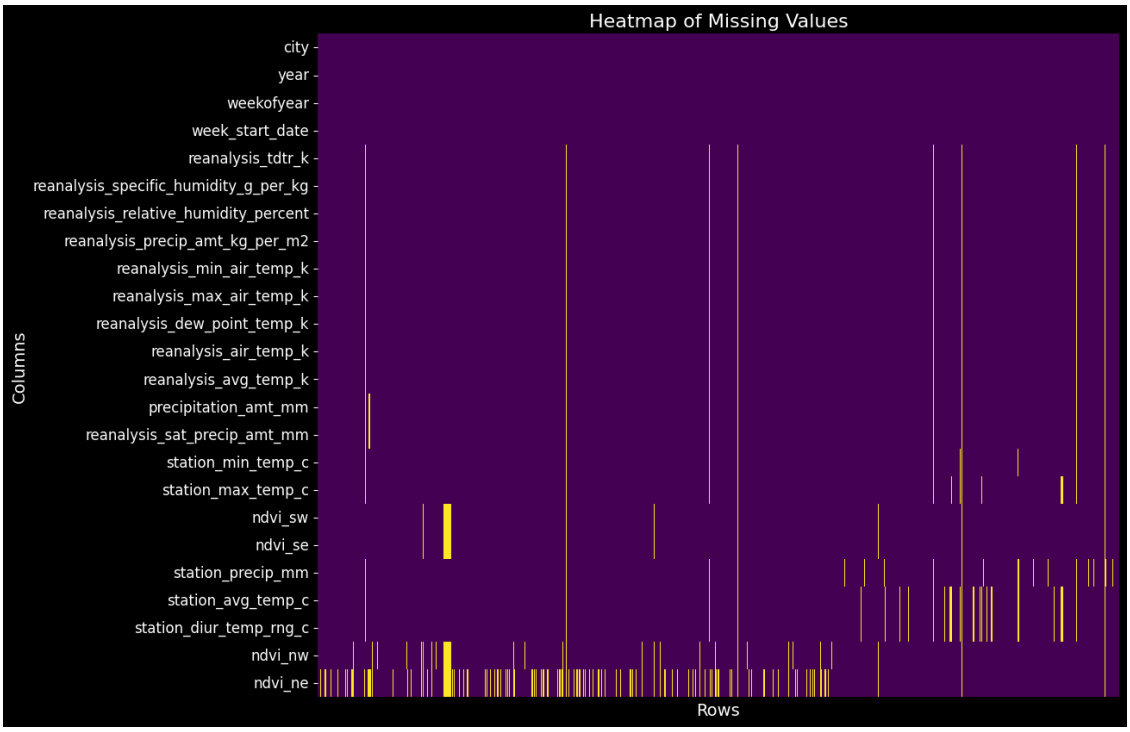
The following plots show some potential clustering of `NA` values for these variables:

- `nvdi_ne`: Pixel northeast of city centroid
- `nvdi_nw`: Pixel northwest of city centroid
- `nvdi_se`: Pixel southeast of city centroid
- `nvdi_sw`: Pixel southwest of city centroid

We would need to speak with domain experts and/or stakeholders to make sense of these findings.

Perhaps this is due to changes in the satellite imaging for these specific samples?

[26]:
```
# Create a boolean DataFrame of missing values for all columns except␣
 ↪'respondent_id'
missing_values = train_features.isna()

# Calculate the sum of missing values for each variable and sort them in␣
 ↪ascending order
sorted_columns = missing_values.sum().sort_values().index

# Reorder the missing_values DataFrame based on the sorted variables
sorted_missing_values = missing_values[sorted_columns]

# Create the figure and axis for the heatmap
plt.figure(figsize=(12, 10))  # Adjust figure size as needed
ax = sns.heatmap(sorted_missing_values.T, cbar=False, xticklabels=False,␣
 ↪cmap='viridis')

# Set the background color of the figure (outside the heatmap)
plt.gcf().set_facecolor('black')

# Set the title and labels with white text
plt.title('Heatmap of Missing Values', color='white', fontsize=16)
plt.xlabel('Rows', color='white', fontsize=14)
plt.ylabel('Columns', color='white', fontsize=14)

# Set tick colors to white and increase font size for better readability
ax.tick_params(axis='y', colors='white', labelsize=12)

# Set the color of the axis (spine) to white
for spine in ax.spines.values():
    spine.set_edgecolor('white')

# Optionally, rotate the y-axis labels for better readability if needed
plt.yticks(rotation=0)

# Save the figure with a black background
plt.savefig('../images/nan_heatmap.png', facecolor='black',␣
 ↪bbox_inches='tight', pad_inches=1.0)
plt.show()
```




[27]:
```
# Create a boolean DataFrame of missing values for all columns except␣
 ↪'respondent_id'
missing_values = train_features.isna()

# Calculate the sum of missing values for each variable and sort them in␣
 ↪ascending order
sorted_columns = missing_values.sum().sort_values().index

# Reorder the missing_values DataFrame based on the sorted variables
sorted_missing_values = missing_values[sorted_columns]

# Convert the sorted boolean DataFrame of missing values to a numeric format
numeric_missing_values = sorted_missing_values.astype(int)  # 1 for True␣
 ↪(missing), 0 for False (not missing)

# Increase the figure's height to give more space for y-axis labels
# The height might need further adjustment based on the actual number of labels
fig_height = max(600, 30 * len(sorted_columns))

fig = px.imshow(numeric_missing_values.T, color_continuous_scale='Viridis',
                labels=dict(x="Rows", y="Columns", color="Missing Values"),
                title="Heatmap of Missing Values",
                height=fig_height)  # Set custom height
```

```
# Update layout to improve aesthetics and readability of y-axis labels
fig.update_layout(
    plot_bgcolor='black',
    paper_bgcolor='black',
    title_font=dict(size=16, color='white'),
    xaxis=dict(showticklabels=False),
    yaxis=dict(
        tickmode='array',
        tickvals=list(range(len(sorted_columns))),
        ticktext=sorted_columns,
        tickfont=dict(size=10, color='white')  # Adjust font size as needed
    ),
    yaxis_title="Columns",
    xaxis_title="Rows"
)

# Show the figure
fig.show()
```

## 17 Drop Missing Values

We decided to drop `NA` values from the whole dataset

But we're mindful that this would need to happen in communication with domain experts!

[28]:
```
#train_features = train_features.dropna()
#test_features = test_features.dropna()
```

[29]:
```
train_features.isna().sum()
```

[29]:
```
city                                     0
year                                     0
weekofyear                               0
week_start_date                          0
ndvi_ne                                194
ndvi_nw                                 52
ndvi_se                                 22
ndvi_sw                                 22
precipitation_amt_mm                    13
reanalysis_air_temp_k                   10
reanalysis_avg_temp_k                   10
reanalysis_dew_point_temp_k             10
reanalysis_max_air_temp_k               10
reanalysis_min_air_temp_k               10
reanalysis_precip_amt_kg_per_m2         10
reanalysis_relative_humidity_percent    10
```

```
reanalysis_sat_precip_amt_mm              13
reanalysis_specific_humidity_g_per_kg     10
reanalysis_tdtr_k                         10
station_avg_temp_c                        43
station_diur_temp_rng_c                   43
station_max_temp_c                        20
station_min_temp_c                        14
station_precip_mm                         22
dtype: int64
```

[30]:
```
test_features.isna().sum()
```

[30]:
```
city                                   0
year                                   0
weekofyear                             0
week_start_date                        0
ndvi_ne                               43
ndvi_nw                               11
ndvi_se                                1
ndvi_sw                                1
precipitation_amt_mm                   2
reanalysis_air_temp_k                  2
reanalysis_avg_temp_k                  2
reanalysis_dew_point_temp_k            2
reanalysis_max_air_temp_k              2
reanalysis_min_air_temp_k              2
reanalysis_precip_amt_kg_per_m2        2
reanalysis_relative_humidity_percent   2
reanalysis_sat_precip_amt_mm           2
reanalysis_specific_humidity_g_per_kg  2
reanalysis_tdtr_k                      2
station_avg_temp_c                    12
station_diur_temp_rng_c               12
station_max_temp_c                     3
station_min_temp_c                     9
station_precip_mm                      5
dtype: int64
```

## 18 Understanding Data Types

## 19 City and Date

`city` – City abbreviations: `sj` for San Juan and `iq` for Iquitos

`week_start_date` – Date given in `yyyy-mm-dd` format

## 20 Daily Climate Data

`station_max_temp_c` – Maximum temperature

`station_min_temp_c` – Minimum temperature

`station_avg_temp_c` – Average temperature

`station_precip_mm` – Total precipitation

`station_diur_temp_rng_c` – Diurnal temperature range

## 21 Satellite Precipitation Measurements

`precipitation_amt_mm` – Total precipitation

## 22 Climate Forecast System Reanalysis

`reanalysis_sat_precip_amt_mm` – Total precipitation

`reanalysis_dew_point_temp_k` – Mean dew point temperature

`reanalysis_air_temp_k` – Mean air temperature

`reanalysis_relative_humidity_percent` – Mean relative humidity

`reanalysis_specific_humidity_g_per_kg` – Mean specific humidity

`reanalysis_precip_amt_kg_per_m2` – Total precipitation

`reanalysis_max_air_temp_k` – Maximum air temperature

`reanalysis_min_air_temp_k` – Minimum air temperature

`reanalysis_avg_temp_k` – Average air temperature

`reanalysis_tdtr_k` – Diurnal temperature range

## 23 Satellite Vegetation

`ndvi_se` – Pixel southeast of city centroid

`ndvi_sw` – Pixel southwest of city centroid

`ndvi_ne` – Pixel northeast of city centroid

`ndvi_nw` – Pixel northwest of city centroid

## 24 Data Type Conversions

### 24.1 Target Variables

- `city`, `year` and `total_cases` are all `int64` data type

[31]:
```
train_labels.dtypes
```

[31]:
```
city           object
year            int64
weekofyear      int64
total_cases     int64
dtype: object
```

[32]:
```
train_labels.describe()
```

[32]:
```
                year   weekofyear  total_cases
count  1456.000000  1456.000000  1456.000000
mean   2001.031593    26.503434    24.675137
std       5.408314    15.019437    43.596000
min    1990.000000     1.000000     0.000000
25%    1997.000000    13.750000     5.000000
50%    2002.000000    26.500000    12.000000
75%    2005.000000    39.250000    28.000000
max    2010.000000    53.000000   461.000000
```

[33]:
```
train_labels['year'].describe()
```

[33]:
```
count    1456.000000
mean     2001.031593
std         5.408314
min      1990.000000
25%      1997.000000
50%      2002.000000
75%      2005.000000
max      2010.000000
Name: year, dtype: float64
```

[34]:
```
train_labels.groupby(['year']).value_counts()
```

[34]:
```
year  city  weekofyear  total_cases
1990  sj    18          4              1
            19          5              1
            20          4              1
            21          3              1
            22          6              1
                                      ..
2010  iq    22          8              1
            23          1              1
            24          1              1
            25          4              1
            53          0              1
Name: count, Length: 1456, dtype: int64
```

[35]:
```
train_labels['year'].value_counts()
```

[35]:
```
year
2001    104
2007    104
2006    104
2005    104
2004    104
2003    104
2002    104
2000     78
2008     69
2009     52
1991     52
1998     52
1997     52
1996     52
1995     52
1994     52
1993     52
1992     52
1999     52
1990     35
2010     26
Name: count, dtype: int64
```

[36]:
```
train_labels['weekofyear']
```

[36]:
```
0       18
1       19
2       20
3       21
4       22
       ..
1451    21
1452    22
1453    23
1454    24
1455    25
Name: weekofyear, Length: 1456, dtype: int64
```

[37]:
```
train_labels['weekofyear'].describe()
```

[37]:
```
count    1456.000000
mean       26.503434
std        15.019437
min         1.000000
```

```
25%        13.750000
50%        26.500000
75%        39.250000
max        53.000000
Name: weekofyear, dtype: float64
```

[38]:
```
train_labels['weekofyear'].value_counts()
```

[38]:
```
weekofyear
18    28
19    28
46    28
47    28
48    28
49    28
50    28
51    28
1     28
2     28
3     28
4     28
5     28
6     28
7     28
8     28
9     28
10    28
11    28
12    28
13    28
14    28
15    28
16    28
17    28
45    28
44    28
43    28
42    28
20    28
21    28
22    28
23    28
24    28
25    28
26    28
27    28
28    28
```

```
29    28
30    28
31    28
32    28
33    28
34    28
35    28
36    28
37    28
38    28
39    28
40    28
41    28
52    23
53     5
Name: count, dtype: int64
```

## 25 city, year, weekofyear, weekstartdate

## 26 city

[39]:
```
train_features['city']
```

[39]:
```
0       sj
1       sj
2       sj
3       sj
4       sj
       ..
1451    iq
1452    iq
1453    iq
1454    iq
1455    iq
Name: city, Length: 1456, dtype: object
```

## 27 Convert city to category Data Type

[40]:
```
train_features.loc[:,'city'] = train_features.loc[:,'city'].astype("category")
train_features['city'].describe()
```

[40]:
```
count     1456
unique       2
top         sj
freq       936
```

```
Name: city, dtype: object
```

[41]:
```
train_features['city'].value_counts()
```

[41]:
```
city
sj    936
iq    520
Name: count, dtype: int64
```

[42]:
```
type(train_features['city'][0])
```

[42]:
```
str
```

## 28 year

- There are two cities, San Juan and Iquitos, with test data for each city spanning 5 and 3 years respectively.
- year is int64 data type

[43]:
```
train_features.dtypes
```

[43]:
```
city                                    object
year                                     int64
weekofyear                               int64
week_start_date                         object
ndvi_ne                                float64
ndvi_nw                                float64
ndvi_se                                float64
ndvi_sw                                float64
precipitation_amt_mm                   float64
reanalysis_air_temp_k                  float64
reanalysis_avg_temp_k                  float64
reanalysis_dew_point_temp_k            float64
reanalysis_max_air_temp_k              float64
reanalysis_min_air_temp_k              float64
reanalysis_precip_amt_kg_per_m2        float64
reanalysis_relative_humidity_percent   float64
reanalysis_sat_precip_amt_mm           float64
reanalysis_specific_humidity_g_per_kg  float64
reanalysis_tdtr_k                      float64
station_avg_temp_c                     float64
station_diur_temp_rng_c                float64
station_max_temp_c                     float64
station_min_temp_c                     float64
station_precip_mm                      float64
dtype: object
```

[44]:
```
train_features['year'].describe()
```

[44]:
```
count    1456.000000
mean     2001.031593
std         5.408314
min      1990.000000
25%      1997.000000
50%      2002.000000
75%      2005.000000
max      2010.000000
Name: year, dtype: float64
```

[45]:
```
print(train_features['year'].dtype)
```

```
int64
```

[46]:
```
train_features['year'].value_counts()
```

[46]:
```
year
2001    104
2007    104
2006    104
2005    104
2004    104
2003    104
2002    104
2000     78
2008     69
2009     52
1991     52
1998     52
1997     52
1996     52
1995     52
1994     52
1993     52
1992     52
1999     52
1990     35
2010     26
Name: count, dtype: int64
```

## 29 week_start_date

[47]:
```
train_features['week_start_date']
```

[47]:
```
0       1990-04-30
1       1990-05-07
2       1990-05-14
3       1990-05-21
4       1990-05-28
           …
1451    2010-05-28
1452    2010-06-04
1453    2010-06-11
1454    2010-06-18
1455    2010-06-25
Name: week_start_date, Length: 1456, dtype: object
```

[48]:
```
type(train_features['week_start_date'][0])
```

[48]:
```
str
```

## 30 Data Conversions

Convert `week_start_date` to Pandas Timestamp (`datetime64[ns]`) Data Type

[49]:
```
train_features.loc[:,'week_start_date'] = train_features.loc[:
 ↪,'week_start_date'].astype("datetime64[ns]")
```

[50]:
```
train_features['week_start_date']
```

[50]:
```
0       1990-04-30 00:00:00
1       1990-05-07 00:00:00
2       1990-05-14 00:00:00
3       1990-05-21 00:00:00
4       1990-05-28 00:00:00
               …
1451    2010-05-28 00:00:00
1452    2010-06-04 00:00:00
1453    2010-06-11 00:00:00
1454    2010-06-18 00:00:00
1455    2010-06-25 00:00:00
Name: week_start_date, Length: 1456, dtype: object
```

[51]:
```
type(train_features['week_start_date'][0])
```

[51]:
```
pandas._libs.tslibs.timestamps.Timestamp
```

## 31 Save Cleaned Datasets

[52]:
```
train_features.to_csv('../data/clean/train_features.csv', index=False)
train_labels.to_csv('../data/clean/train_labels.csv', index=False)
train_labels.to_pickle('../data/clean/train_labels.pkl')
train_features.to_pickle('../data/clean/train_features.pkl')
!cp '../data/dengue_features_test.csv' '../data/clean/test_features.csv'
```

## 32 TODO

- Exploratory Data Analysis
- Check data types
  - `weekofyear` to pandas `datetime`
- Check NaNs
- Clean and transform datasets
- Save cleaned datasets
- Check for class imbalance in `train_labels`
- Split `train_features` and `train_labels` into `train` and `validation`
- Choose baseline model based on prediction task and data types
- Feature engineering and preprocessing of data for modeling
- Decide hyperparameters
- Build model
- Train/fit model
- Make predictions on `validation` set
- Validate model using visualisation; tune hyperparameters
- Make predictions on `test` set
- Make a submission

## 33 Decision

- `NA` values were dropped from all datasets
- `year`, `weekofyear`, and `total_cases` were kept as `int64`
- `week_start_date` was converted to Pandas Timestamp (`datetime64[ns]`) Data Type in both `train_features`; this variable is not present in `train_labels`