```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import pandas as pd
import numpy as np
import scipy.stats as st
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import cross_validation, metrics
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import AdaBoostClassifier, GradientBoostingClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.decomposition import PCA, KernelPCA
from sklearn.decomposition import FactorAnalysis
from sklearn.feature_selection import RFECV
```

```
%load_ext rpy2.ipynb
```

```
%matplotlib inline
%load_ext rpy2.ipynb
```

```
!ls -lrh data/
```

```
total 2.8M
-rwxrwxrwx 1 root root 2.8M Feb 12 05:19 train_ubiquant.parquet
```

```
df = pd.read_parquet('data/train_ubiquant.parquet')
df.shape
```

```
(4412499, 4)
```

	row_id	time_id	investment_id	target
0	0_0_0	0	0	0.390125
1	0_0_1	0	1	-0.059772
2	0_0_2	0	2	-0.099326
3	0_0_3	0	3	-0.062455
4	0_0_4	0	4	0.091643

```
df.tail(5)
```

	row_id	time_id	investment_id	target
4412494	1043_9_2996	1043	2996	NaN
4412495	1043_9_3000	1043	3000	NaN
4412496	1043_9_3001	1043	3001	NaN
4412497	1043_9_3002	1043	3002	NaN
4412498	1043_9_3003	1043	3003	NaN

```
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 4412499 entries, 0 to 4412498
Data columns (total 4 columns):
 0   row_id         object
 1   time_id        int64
 2   investment_id  int64
 3   target         float64
dtype: object, int64, int64, float64
memory usage: 134.7 MB
```

```
df.describe()
```

	time_id	investment_id	target
count	4412499.000000	4412499.000000	4412499.000000
mean	1000.000000	1000.000000	0.000000
std	0.000000	0.000000	0.000000
min	1000.000000	1000.000000	0.000000
25%	1000.000000	1000.000000	0.000000
50%	1000.000000	1000.000000	0.000000
75%	1000.000000	1000.000000	0.000000
max	1000.000000	1000.000000	0.000000

```
df.describe(include=['O'])
```

	row_id
count	4412499
unique	4412499
top	0_0_0
freq	1

```
df.describe(include=['object', 'float64'])
```

	row_id	target
count	4412499	4412499.000000
unique	4412499	NaN
top	0_0_0	NaN
freq	1	NaN
mean	NaN	0.000000
std	NaN	0.000000
min	NaN	0.000000
25%	NaN	0.000000
50%	NaN	0.000000
75%	NaN	0.000000
max	NaN	0.000000

```
df.describe(include=['int64', 'float64'])
```

	time_id	investment_id	target
count	4412499.000000	4412499.000000	4412499.000000
mean	1000.000000	1000.000000	0.000000
std	0.000000	0.000000	0.000000
min	1000.000000	1000.000000	0.000000
25%	1000.000000	1000.000000	0.000000
50%	1000.000000	1000.000000	0.000000
75%	1000.000000	1000.000000	0.000000
max	1000.000000	1000.000000	0.000000

```
df.describe(include=['object'])
```

	row_id
count	4412499
unique	4412499
top	0_0_0
freq	1

```
df.describe(exclude=['object'])
```

	time_id	investment_id	target
count	4412499.000000	4412499.000000	4412499.000000
mean	1000.000000	1000.000000	0.000000
std	0.000000	0.000000	0.000000
min	1000.000000	1000.000000	0.000000
25%	1000.000000	1000.000000	0.000000
50%	1000.000000	1000.000000	0.000000
75%	1000.000000	1000.000000	0.000000
max	1000.000000	1000.000000	0.000000

```
df.columns.values
```

```
array(['row_id', 'time_id', 'investment_id', 'target'],
      dtype=object)
```

```
df.dtypes
```

```
row_id         object
time_id        int64
investment_id  int64
target         float64
dtype: object, int64, int64, float64
```

```
df.isnull().sum()
```

```
row_id         0
time_id        0
investment_id  0
target         0
dtype: int64
```

```
df.isnull().sum/df.count()
```

```
row_id         0.000000
time_id        0.000000
investment_id  0.000000
target         0.000000
dtype: int64, int64, int64, float64
```

```
df.isnull().sum/df.shape[0]
```

```
row_id         0.000000
time_id        0.000000
investment_id  0.000000
target         0.000000
dtype: int64, int64, int64, float64
```

```
df.isnull().sum/df.shape[1]
```

```
row_id         1.103125e+06
time_id        2.206250e+06
investment_id  2.206250e+06
target         1.103125e+06
dtype: int64, int64, int64, float64
```

```
df.isnull().sum()/df.shape[1]
```

```
row_id         1.103125e+06
time_id        2.206250e+06
investment_id  2.206250e+06
target         1.103125e+06
dtype: int64, int64, int64, float64
```

```
df.isnull().sum()/df.shape[0]
```

```
row_id         0.000000
time_id        0.000000
investment_id  0.000000
target         0.000000
dtype: int64, int64, int64, float64
```

```
df.isnull().sum()/df.count()
```

```
row_id         0.000000
time_id        0.000000
investment_id  0.000000
target         0.000000
```

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

	Anomaly
Date	
2020-01-01	1.0
2020-01-02	1.0
2020-01-03	1.0
2020-01-04	1.0
2020-01-05	1.0
2020-01-06	1.0
2020-01-07	1.0
2020-01-08	1.0
2020-01-09	1.0
2020-01-10	1.0
2020-01-11	1.0
2020-01-12	1.0
2020-01-13	1.0
2020-01-14	1.0
2020-01-15	1.0
2020-01-16	1.0
2020-01-17	1.0
2020-01-18	1.0
2020-01-19	1.0
2020-01-20	1.0
2020-01-21	1.0
2020-01-22	1.0
2020-01-23	1.0
2020-01-24	1.0
2020-01-25	1.0
2020-01-26	1.0
2020-01-27	1.0
2020-01-28	1.0
2020-01-29	1.0
2020-01-30	1.0
2020-01-31	1.0
2020-02-01	1.0
2020-02-02	1.0
2020-02-03	1.0
2020-02-04	1.0
2020-02-05	1.0
2020-02-06	1.0
2020-02-07	1.0
2020-02-08	1.0
2020-02-09	1.0
2020-02-10	1.0
2020-02-11	1.0
2020-02-12	1.0
2020-02-13	1.0
2020-02-14	1.0
2020-02-15	1.0
2020-02-16	1.0
2020-02-17	1.0
2020-02-18	1.0
2020-02-19	1.0
2020-02-20	1.0
2020-02-21	1.0
2020-02-22	1.0
2020-02-23	1.0
2020-02-24	1.0
2020-02-25	1.0
2020-02-26	1.0
2020-02-27	1.0
2020-02-28	1.0
2020-02-29	1.0
2020-03-01	1.0
2020-03-02	1.0
2020-03-03	1.0
2020-03-04	1.0
2020-03-05	1.0
2020-03-06	1.0
2020-03-07	1.0
2020-03-08	1.0
2020-03-09	1.0
2020-03-10	1.0
2020-03-11	1.0
2020-03-12	1.0
2020-03-13	1.0
2020-03-14	1.0
2020-03-15	1.0
2020-03-16	1.0
2020-03-17	1.0
2020-03-18	1.0
2020-03-19	1.0
2020-03-20	1.0
2020-03-21	1.0
2020-03-22	1.0
2020-03-23	1.0
2020-03-24	1.0
2020-03-25	1.0
2020-03-26	1.0
2020-03-27	1.0
2020-03-28	1.0
2020-03-29	1.0
2020-03-30	1.0
2020-03-31	1.0
2020-04-01	1.0
2020-04-02	1.0
2020-04-03	1.0
2020-04-04	1.0
2020-04-05	1.0
2020-04-06	1.0
2020-04-07	1.0
2020-04-08	1.0
2020-04-09	1.0
2020-04-10	1.0
2020-04-11	1.0
2020-04-12	1.0
2020-04-13	1.0
2020-04-14	1.0
2020-04-15	1.0
2020-04-16	1.0
2020-04-17	1.0
2020-04-18	1.0
2020-04-19	1.0
2020-04-20	1.0
2020-04-21	1.0
2020-04-22	1.0
2020-04-23	1.0
2020-04-24	1.0
2020-04-25	1.0
2020-04-26	1.0
2020-04-27	1.0
2020-04-28	1.0
2020-04-29	1.0
2020-04-30	1.0
2020-05-01	1.0
2020-05-02	1.0
2020-05-03	1.0
2020-05-04	1.0
2020-05-05	1.0
2020-05-06	1.0
2020-05-07	1.0
2020-05-08	1.0
2020-05-09	1.0
2020-05-10	1.0
2020-05-11	1.0
2020-05-12	1.0
2020-05-13	1.0
2020-05-14	1.0
2020-05-15	1.0
2020-05-16	1.0
2020-05-17	1.0
2020-05-18	1.0
2020-05-19	1.0
2020-05-20	1.0
2020-05-21	1.0
2020-05-22	1.0
2020-05-23	1.0
2020-05-24	1.0
2020-05-25	1.0
2020-05-26	1.0
2020-05-27	1.0
2020-05-28	1.0
2020-05-29	1.0
2020-05-30	1.0
2020-05-31	1.0
2020-06-01	1.0
2020-06-02	1.0
2020-06-03	1.0
2020-06-04	1.0
2020-06-05	1.0
2020-06-06	1.0
2020-06-07	1.0
2020-06-08	1.0
2020-06-09	1.0
2020-06-10	1.0
2020-06-11	1.0
2020-06-12	1.0
2020-06-13	1.0
2020-06-14	1.0
2020-06-15	1.0
2020-06-16	1.0
2020-06-17	1.0
2020-06-18	1.0
2020-06-19	1.0
2020-06-20	1.0
2020-06-21	1.0
2020-06-22	1.0
2020-06-23	1.0
2020-06-24	1.0
2020-06-25	1.0
2020-06-26	1.0
2020-06-27	1.0
2020-06-28	1.0
2020-06-29	1.0
2020-06-30	1.0
2020-07-01	1.0
2020-07-02	1.0
2020-07-03	1.0
2020-07-04	1.0
2020-07-05	1.0
2020-07-06	1.0
2020-07-07	1.0
2020-07-08	1.0
2020-07-09	1.0
2020-07-10	1.0
2020-07-11	1.0
2020-07-12	1.0
2020-07-13	1.0
2020-07-14	1.0
2020-07-15	1.0
2020-07-16	1.0
2020-07-17	1.0
2020-07-18	1.0
2020-07-19	1.0
2020-07-20	1.0
2020-07-21	1.0
2020-07-22	1.0
2020-07-23	1.0
2020-07-24	1.0
2020-07-25	1.0
2020-07-26	1.0
2020-07-27	1.0
2020-07-28	1.0
2020-07-29	1.0
2020-07-30	1.0
2020-07-31	1.0
2020-08-01	1.0
2020-08-02	1.0
2020-08-03	1.0
2020-08-04	1.0
2020-08-05	1.0
2020-08-06	1.0
2020-08-07	1.0
2020-08-08	1.0
2020-08-09	1.0
2020-08-10	1.0
2020-08-11	1.0
2020-08-12	1.0
2020-08-13	1.0
2020-08-14	1.0
2020-08-15	1.0
2020-08-16	1.0
2020-08-17	1.0
2020-08-18	1.0
2020-08-19	1.0
2020-08-20	1.0
2020-08-21	1.0
2020-08-22	1.0
2020-08-23	1.0
2020-08-24	1.0
2020-08-25	1.0
2020-08-26	1.0
2020-08-27	1.0
2020-08-28	1.0
2020-08-29	1.0
2020-08-30	1.0
2020-08-31	1.0
2020-09-01	1.0
2020-09-02	1.0
2020-09-03	1.0
2020-09-04	1.0
2020-09-05	1.0
2020-09-06	1.0
2020-09-07	1.0
2020-09-08	1.0
2020-09-09	1.0
2020-09-10	1.0
2020-09-11	1.0
2020-09-12	1.0
2020-09-13	1.0
2020-09-14	1.0
2020-09-15	1.0
2020-09-16	1.0
2020-09-17	1.0
2020-09-18	1.0
2020-09-19	1.0
2020-09-20	1.0
2020-09-21	1.0
2020-09-22	1.0
2020-09-23	1.0
2020-09-24	1.0
2020-09-25	1.0
2020-09-26	1.0
2020-09-27	1.0
2020-09-28	1.0
2020-09-29	1.0
2020-09-30	1.0
2020-10-01	1.0
2020-10-02	1.0
2020-10-03	1.0
2020-10-04	1.0
2020-10-05	1.0
2020-10-06	1.0
2020-10-07	1.0
2020-10-08	1.0
2020-10-09	1.0
2020-10-10	1.0
2020-10-11	1.0
2020-10-12	1.0
2020-10-13	1.0
2020-10-14	1.0
2020-10-15	1.0
2020-10-16	1.0
2020-10-17	1.0
2020-10-18	1.0
2020-10-19	1.0
2020-10-20	1.0
2020-10-21	1.0
2020-10-22	1.0
2020-10-23	1.0
2020-10-24	1.0
2020-10-25	1.0
2020-10-26	1.0
2020-10-27	1.0
2020-10-28	1.0
2020-10-29	1.0
2020-10-30	1.0
2020-10-31	1.0
2020-11-01	1.0
2020-11-02	1.0
2020-11-03	1.0
2020-11-04	1.0
2020-11-05	1.0
2020-11-06	1.0
2020-11-07	1.0
2020-11-08	1.0
2020-11-09	1.0
2020-11-10	1.0
2020-11-11	1.0
2020-11-12	1.0
2020-11-13	1.0
2020-11-14	1.0
2020-11-15	1.0
2020-11-16	1.0
2020-11-17	1.0
2020-11-18	1.0
2020-11-19	1.0
2020-11-20	1.0
2020-11-21	1.0
2020-11-22	1.0
2020-11-23	1.0
2020-11-24	1.0
2020-11-25	1.0
2020-11-26	1.0
2020-11-27	1.0
2020-11-28	1.0
2020-11-29	1.0
2020-11-30	1.0
2020-12-01	1.0
2020-12-02	1.0
2020-12-03	1.0
2020-12-04	1.0
2020-12-05	1.0
2020-12-06	1.0
2020-12-07	1.0
2020-12-08	1.0
2020-12-09	1.0
2020-12-10	1.0
2020-12-11	1.0
2020-12-12	1.0
2020-12-13	1.0
2020-12-14	1.0
2020-12-15	1.0
2020-12-16	1.0
2020-12-17	1.0
2020-12-18	1.0
2020-12-19	1.0
2020-12-20	1.0
2020-12-21	1.0
2020-12-22	1.0
2020-12-23	1.0
2020-12-24	1.0
2020-12-25	1.0
2020-12-26	1.0
2020-12-27	1.0
2020-12-28	1.0
2020-12-29	1.0
2020-12-30	1.0
2020-12-31	1.0
2021-01-01	1.0
2021-01-02	1.0
2021-01-03	1.0
2021-01-04	1.0
2021-01-05	1.0
2021-01-06	1.0
2021-01-07	1.0
2021-01-08	1.0
2021-01-09	1.0
2021-01-10	1.0
2021-01-11	1.0
2021-01-12	1.0
2021-01-13	1.0
2021-01-14	1.0
2021-01-15	1.0
2021-01-16	1.0
2021-01-17	1.0
2021-01-18	1.0
2021-01-19	1.0
2021-01-20	1.0
2021-01-21	1.0
2021-01-22	1.0
2021-01-23	1.0
2021-01-24	1.0
2021-01-25	1.0
2021-01-26	1.0
2021-01-27	1.0
2021-01-28	1.0
2021-01-29	1.0
2021-01-30	1.0
2021-01-31	1.0
2021-02-01	1.0
2021-02-02	1.0
2021-02-03	1.0
2021-02-04	1.0
2021-02-05	1.0
2021-02-06	1.0
2021-02-07	1.0
2021-02-08	1.0
2021-02-09	1.0
2021-02-10	1.0
2021-02-11	1.0
2021-02-12	1.0
2021-02-13	1.0
2021-02-14	1.0
2021-02-15	1.0
2021-02-16	1.0
2021-02-17	1.0
2021-02-18	1.0
2021-02-19	1.0
2021-02-20	1.0
2021-02-21	1.0
2021-02-22	1.0
2021-02-23	1.0
2021-02-24	1.0
2021-02-25	1.0
2021-02-26	1.0
2021-02-27	1.0
2021-02-28	1.0
2021-03-01	1.0
2021-03-02	1.0
2021-03-03	1.0
2021-03-04	1.0
2021-03-05	1.0
2021-03-06	1.0
2021-03-07	1.0
2021-03-08	1.0
2021-03-09	1.0
2021-03-10	1.0
2021-03-11	1.0
2021-03-12	1.0
2021-03-13	1.0
2021-03-14	1.0
2021-03-15	1.0
2021-03-16	1.0
2021-03-17	1.0
2021-03-18	1.0
2021-03-19	1.0
2021-03-20	1.0
2021-03-21	1.0
2021-03-22	1.0
2021-03-23	1.0
2021-03-24	1.0
2021-03-25	1.0
2021-03-26	1.0
2021-03-27	1.0
2021-03-28	1.0
2021-03-29	1.0
2021-03-30	1.0
2021-03-31	1.0
2021-04-01	1.0
2021-04-02	1.0
2021-04-03	1.0
2021-04-04	1.0
2021-04-05	1.0
2021-04-06	1.0
2021-04-07	1.0
2021-04-08	1.0
2021-04-09	1.0
2021-04-10	1.0
2021-04-11	1.0
2021-04-12	1.0
2021-04-13	1.0
2021-04-14	1.0
2021-04-15	1.0
2021-04-16	1.0
2021-04-17	1.0
2021-04-18	1.0
2021-04-19	1.0
2021-04-20	1.0
2021-04-21	1.0
2021-04-22	1.0
2021-04-23	1.0
2021-04-24	1.0
2021-04-25	1.0
2021-04-26	1.0
2021-04-27	1.0
2021-04-28	1.0
2021-04-29	1.0
2021-04-30	1.0
2021-05-01	1.0
2021-05-02	1.0
2021-05-03	1.0
2021-05-04	1.0
2021-05-05	1.0
2021-05-06	1.0
2021-05-07	1.0
2021-05-08	1.0
2021-05-09	1.0
2021-05-10	1.0
2021-05-11	1.0
2021-05-12	1.0
2021-05-13	1.0
2021-05-14	1.0
2021-05-15	1.0
2021-05-16	1.0
2021-05-17	1.0
2021-05-18	1.0
2021-05-19	1.0
2021-05-20	1.0
2021-05-21	1.0
2021-05-22	1.0
2021-05-23	1.0
2021-05-24	1.0
2021-05-25	1.0
2021-05-26	1.0
2021-05-27	1.0
2021-05-28	1.0
2021-05-29	1.0
2021-05-30	1.0
2021-05-31	1.0
2021-06-01	1.0
2021-06-02	1.0
2021-06-03	1.0
2021-06-04	1.0
2021-06-05	1.0
2021-06-06	1.0
2021-06-07	1.0
2021-06-08	1.0
2021-06-09	1.0
2021-06-10	1.0
2021-06-11	1.0
2021-06-12	1.0
2021-06-13	1.0
2021-06-14	1.0
2021-06-15	1.0
2021-06-16	1.0
2021-06-17	1.0
2021-06-18	1.0
2021-06-19	1.0
2021-06-20	1.0
2021-06-21	1.0
2021-06-22	1.0
2021-06-23	1.0
2021-06-24	1.0
2021-06-25	1.0
2021-06-26	1.0
2021-06-27	1.0
2021-06-28	1.0
2021-06-29	1.0
2021-06-30	1.0
2021-07-01	1.0
2021-07-02	1.0
2021-07-03	1.0
2021-07-04	1.0
2021-07-05	1.0
2021-07-06	1.0
2021-07-07	1.0
2021-07-08	1.0
2021-07-09	1.0
2021-07-10	1.0
2021-07-11	1.0
2021-07-12	1.0
2021-07-13	1.0
2021-07-14	1.0
2021-07-15	1.0
2021-07-16	1.0
2021-07-17	1.0
2021-07-18	1.0
2021-07-19	1.0
2021-07-20	1.0
2021-07-21	1.0
2021-07-22	1.0
2021-07-23	1.0
2021-07-24	1.0
2021-07-25	1.0
2021-07-26	1.0
2021-07-27	1.0
2021-07-28	1.0
2021-07-29	1.0
2021-07-30	1.0
2021-07-31	1.0
2021-08-01	1.0
2021-08-02	1.0
2021-08-03	1.0
2021-08-04	1.0
2021-08-05	1.0
2021-08-06	1.0
2021-08-07	1.0
2021-08-08	1.0
2021-08-09	1.0
2021-08-10	1.0
2021-08-11	1.0
2021-08-12	1.0
2021-08-13	1.0
2021-08-14	1.0
2021-08-15	1.0
2021-08-16	1.0
2021-08-17	1.0
2021-08-18	1.0
2021-08-19	1.0
2021-08-20	1.0
2021-08-21	1.0
2021-08-22	1.0
2021-08-23	1.0
2021-08-24	1.0
2021-08-25	1.0
2021-08-26	1.0
2021-08-27	1.0
2021-08-28	1.0
2021-08-29	1.0
2021-08-30	1.0
2021-08-31	1.0
2021-09-01	1.0
2021-09-02	1.0
2021-09-03	1.0
2021-09-04	1.0
2021-09-05	1.0
2021-09-06	1.0
2021-09-07	1.0
2021-09-08	1.0
2021-09-09	1.0
2021-09-10	1.0
2021-09-11	1.0
2021-09-12	1.0
2021-09-13	1.0
2021-09-14	1.0
2021-09-15	1.0
2021-09-16	1.0
2021-09-17	1.0
2021-09-18	1.0
2021-09-19	1.0
2021-09-20	1.0
2021-09-21	1.0
2021-09-22	1.0
2021-09-23	1.0
2021-09-24	1.0
2021-09-25	1.0
2021-09-26	1.0
2021-09-27	1.0
2021-09-28	1.0
2021-09-29	1.0
2021-09-30	1.0
2021-10-01	1.0
2021-10-02	1.0
2021-10-03	1.0
2021-10-04	1.0
2021-10-05	1.0
2021-10-06	1.0
2021-10-07	1.0
2021-10-08	1.0
2021-10-09	1.0
2021-10-10	1.0
2021-10-11	1.0
2021-10-12	1.0
2021-10-13	1.0
2021-10-14	1.0
2021-10-15	1.0
2021-10-16	1.0
2021-10-17	1.0
2021-10-18	1.0
2021-10-19	1.0
2021-10-20	1.0
2021-10-21	1.0
2021-10-22	1.0
2021-10-23	1.0
2021-10-24	1.0
2021-10-25	1.0
2021-10-26	1.0
2021-10-27	1.0
2021-10-28	1.0
2021-10-29	1.0
2021-10-30	1.0
2021-10-31	1.0
2021-11-01	1.0
2021-11-02	1.0
2021-11-03	1.0
2021-11-04	1.0
2021-11-05	1.0
2021-11-06	1.0
2021-11-07	1.0
2021-11-08	1.0
2021-11-09	1.0
2021-11-10	1.0
2021-11-11	1.0
2021-11-12	1.0
2021-11-13	1.0
2021-11-14	1.0
2021-11-15	1.0
2021-11-16	1.0
2021-11-17	1.0
2021-11-18	1.0
2021-11-19	1.0
2021-11-20	1.0
2021-11-21	1.0
2021-11-22	1.0
2021-11-23	1.0
2021-11-24	1.0
2021-11-25	1.0
2021-11-26	1.0
2021-11-27	1.0
2021-11-28	1.0
2021-11-29	1.0
2021-11-30	1.0
2021-12-01	1.0
2021-12-02	1.0
2021-12-03	1.0
2021-12-04	1.0
2021-12-05	1.0
2021-12-06	1.0
2021-12-07	1.0
2021-12-08	1.0
2021-12-09	1.0
2021-12-10	1.0
2021-12-11	1.0
2021-12-12	1.0
2021-12-13	1.0
2021-12-14	1.0
2021-12-15	1.0
2021-12-16	1.0
2021-12-17	1.0
2021-12-18	1.0
2021-12-19	1.0
2021-12-20	1.0
2021-12-21	1.0
2021-12-22	1.0
2021-12-23	1.0
2021-12-24	1.0
2021-12-25	1.0
2021-12-26	1.0
2021-12-27	1.0
2021-12-28	1.0
2021-12-29	1.0
2021-12-30	1.0
2021-12-31	1.0

[11] [12]

```
df['Anomaly'] = df['Anomaly'].fillna(0)
df['Anomaly'] = df['Anomaly'].astype(int)
df['Anomaly'].value_counts()
```

Out[10]:

	date	Name
0	2020-01-01	0.0
1	2020-01-02	0.0
2	2020-01-03	0.0
3	2020-01-04	0.0
4	2020-01-05	0.0
5	2020-01-06	0.0
6	2020-01-07	0.0
7	2020-01-08	0.0
8	2020-01-09	0.0
9	2020-01-10	0.0
10	2020-01-11	0.0
11	2020-01-12	0.0
12	2020-01-13	0.0
13	2020-01-14	0.0
14	2020-01-15	0.0
15	2020-01-16	0.0
16	2020-01-17	0.0
17	2020-01-18	0.0
18	2020-01-19	0.0
19	2020-01-20	0.0
20	2020-01-21	0.0
21	2020-01-22	0.0
22	2020-01-23	0.0
23	2020-01-24	0.0
24	2020-01-25	0.0
25	2020-01-26	0.0
26	2020-01-27	0.0
27	2020-01-28	0.0
28	2020-01-29	0.0
29	2020-01-30	0.0
30	2020-01-31	0.0
31	2020-02-01	0.0
32	2020-02-02	0.0
33	2020-02-03	0.0
34	2020-02-04	0.0
35	2020-02-05	0.0
36	2020-02-06	0.0
37	2020-02-07	0.0
38	2020-02-08	0.0
39	2020-02-09	0.0
40	2020-02-10	0.0
41	2020-02-11	0.0
42	2020-02-12	0.0
43	2020-02-13	0.0
44	2020-02-14	0.0
45	2020-02-15	0.0
46	2020-02-16	0.0
47	2020-02-17	0.0
48	2020-02-18	0.0
49	2020-02-19	0.0
50	2020-02-20	0.0
51	2020-02-21	0.0
52	2020-02-22	0.0
53	2020-02-23	0.0
54	2020-02-24	0.0
55	2020-02-25	0.0
56	2020-02-26	0.0
57	2020-02-27	0.0
58	2020-02-28	0.0
59	2020-02-29	0.0
60	2020-03-01	0.0
61	2020-03-02	0.0
62	2020-03-03	0.0
63	2020-03-04	0.0
64	2020-03-05	0.0
65	2020-03-06	0.0
66	2020-03-07	0.0
67	2020-03-08	0.0
68	2020-03-09	0.0
69	2020-03-10	0.0
70	2020-03-11	0.0
71	2020-03-12	0.0
72	2020-03-13	0.0
73	2020-03-14	0.0
74	2020-03-15	0.0
75	2020-03-16	0.0
76	2020-03-17	0.0
77	2020-03-18	0.0
78	2020-03-19	0.0
79	2020-03-20	0.0
80	2020-03-21	0.0
81	2020-03-22	0.0
82	2020-03-23	0.0
83	2020-03-24	0.0
84	2020-03-25	0.0
85	2020-03-26	0.0
86	2020-03-27	0.0
87	2020-03-28	0.0
88	2020-03-29	0.0
89	2020-03-30	0.0
90	2020-03-31	0.0
91	2020-04-01	0.0
92	2020-04-02	0.0
93	2020-04-03	0.0
94	2020-04-04	0.0
95	2020-04-05	0.0
96	2020-04-06	0.0
97	2020-04-07	0.0
98	2020-04-08	0.0
99	2020-04-09	0.0
100	2020-04-10	0.0
101	2020-04-11	0.0
102	2020-04-12	0.0
103	2020-04-13	0.0
104	2020-04-14	0.0
105	2020-04-15	0.0
106	2020-04-16	0.0
107	2020-04-17	0.0
108	2020-04-18	0.0
109	2020-04-19	0.0
110	2020-04-20	0.0
111	2020-04-21	0.0
112	2020-04-22	0.0
113	2020-04-23	0.0
114	2020-04-24	0.0
115	2020-04-25	0.0
116	2020-04-26	0.0
117	2020-04-27	0.0
118	2020-04-28	0.0
119	2020-04-29	0.0
120	2020-04-30	0.0
121	2020-05-01	0.0
122	2020-05-02	0.0
123	2020-05-03	0.0
124	2020-05-04	0.0
125	2020-05-05	0.0
126	2020-05-06	0.0
127	2020-05-07	0.0
128	2020-05-08	0.0
129	2020-05-09	0.0
130	2020-05-10	0.0
131	2020-05-11	0.0
132	2020-05-12	0.0
133	2020-05-13	0.0
134	2020-05-14	0.0
135	2020-05-15	0.0
136	2020-05-16	0.0
137	2020-05-17	0.0
138	2020-05-18	0.0
139	2020-05-19	0.0
140	2020-05-20	0.0
141	2020-05-21	0.0
142	2020-05-22	0.0
143	2020-05-23	0.0
144	2020-05-24	0.0
145	2020-05-25	0.0
146	2020-05-26	0.0
147	2020-05-27	0.0
148	2020-05-28	0.0
149	2020-05-29	0.0
150	2020-05-30	0.0
151	2020-05-31	0.0
152	2020-06-01	0.0
153	2020-06-02	0.0
154	2020-06-03	0.0
155	2020-06-04	0.0
156	2020-06-05	0.0
157	2020-06-06	0.0
158	2020-06-07	0.0
159	2020-06-08	0.0
160	2020-06-09	0.0
161	2020-06-10	0.0
162	2020-06-11	0.0
163	2020-06-12	0.0
164	2020-06-13	0.0
165	2020-06-14	0.0
166	2020-06-15	0.0
167	2020-06-16	0.0
168	2020-06-17	0.0
169	2020-06-18	0.0
170	2020-06-19	0.0
171	2020-06-20	0.0
172	2020-06-21	0.0
173	2020-06-22	0.0
174	2020-06-23	0.0
175	2020-06-24	0.0
176	2020-06-25	0.0
177	2020-06-26	0.0
178	2020-06-27	0.0
179	2020-06-28	0.0
180	2020-06-29	0.0
181	2020-06-30	0.0
182	2020-07-01	0.0
183	2020-07-02	0.0
184	2020-07-03	0.0
185	2020-07-04	0.0
186	2020-07-05	0.0
187	2020-07-06	0.0
188	2020-07-07	0.0
189	2020-07-08	0.0
190	2020-07-09	0.0
191	2020-07-10	0.0
192	2020-07-11	0.0
193	2020-07-12	0.0
194	2020-07-13	0.0
195	2020-07-14	0.0
196	2020-07-15	0.0
197	2020-07-16	0.0
198	2020-07-17	0.0
199	2020-07-18	0.0
200	2020-07-19	0.0
201	2020-07-20	0.0
202	2020-07-21	0.0
203	2020-07-22	0.0
204	2020-07-23	0.0
205	2020-07-24	0.0
206	2020-07-25	0.0
207	2020-07-26	0.0
208	2020-07-27	0.0
209	2020-07-28	0.0
210	2020-07-29	0.0
211	2020-07-30	0.0
212	2020-07-31	0.0
213	2020-08-01	0.0
214	2020-08-02	0.0
215	2020-08-03	0.0
216	2020-08-04	0.0
217	2020-08-05	0.0
218	2020-08-06	0.0
219	2020-08-07	0.0
220	2020-08-08	0.0
221	2020-08-09	0.0
222	2020-08-10	0.0
223	2020-08-11	0.0
224	2020-08-12	0.0
225	2020-08-13	0.0
226	2020-08-14	0.0
227	2020-08-15	0.0
228	2020-08-16	0.0
229	2020-08-17	0.0
230	2020-08-18	0.0
231	2020-08-19	0.0
232	2020-08-20	0.0
233	2020-08-21	0.0
234	2020-08-22	0.0
235	2020-08-23	0.0
236	2020-08-24	0.0
237	2020-08-25	0.0
238	2020-08-26	0.0
239	2020-08-27	0.0
240	2020-08-28	0.0
241	2020-08-29	0.0
242	2020-08-30	0.0
243	2020-08-31	0.0
244	2020-09-01	0.0
245	2020-09-02	0.0
246	2020-09-03	0.0
247	2020-09-04	0.0
248	2020-09-05	0.0
249	2020-09-06	0.0
250	2020-09-07	0.0
251	2020-09-08	0.0
252	2020-09-09	0.0
253	2020-09-10	0.0
254	2020-09-11	0.0
255	2020-09-12	0.0
256	2020-09-13	0.0
257	2020-09-14	0.0
258	2020-09-15	0.0
259	2020-09-16	0.0
260	2020-09-17	0.0
261	2020-09-18	0.0
262	2020-09-19	0.0
263	2020-09-20	0.0
264	2020-09-21	0.0
265	2020-09-22	0.0
266	2020-09-23	0.0
267	2020-09-24	0.0
268	2020-09-25	0.0
269	2020-09-26	0.0
270	2020-09-27	0.0
271	2020-09-28	0.0
272	2020-09-29	0.0
273	2020-09-30	0.0
274	2020-10-01	0.0
275	2020-10-02	0.0
276	2020-10-03	0.0
277	2020-10-04	0.0
278	2020-10-05	0.0
279	2020-10-06	0.0
280	2020-10-07	0.0
281	2020-10-08	0.0
282	2020-10-09	0.0
283	2020-10-10	0.0
284	2020-10-11	0.0
285	2020-10-12	0.0
286	2020-10-13	0.0
287	2020-10-14	0.0
288	2020-10-15	0.0
289	2020-10-16	0.0
290	2020-10-17	0.0
291	2020-10-18	0.0
292	2020-10-19	0.0
293	2020-10-20	0.0
294	2020-10-21	0.0
295	2020-10-22	0.0
296	2020-10-23	0.0
297	2020-10-24	0.0
298	2020-10-25	0.0
299	2020-10-26	0.0
300	2020-10-27	0.0
301	2020-10-28	0.0
302	2020-10-29	0.0
303	2020-10-30	0.0
304	2020-10-31	0.0
305	2020-11-01	0.0
306	2020-11-02	0.0
307	2020-11-03	0.0
308	2020-11-04	0.0
309	2020-11-05	0.0
310	2020-11-06	0.0
311	2020-11-07	0.0
312	2020-11-08	0.0
313	2020-11-09	0.0
314	2020-11-10	0.0
315	2020-11-11	0.0
316	2020-11-12	0.0
317	2020-11-13	0.0
318	2020-11-14	0.0
319	2020-11-15	0.0
320	2020-11-16	0.0
321	2020-11-17	0.0
322	2020-11-18	0.0
323	2020-11-19	0.0
324	2020-11-20	0.0
325	2020-11-21	0.0
326	2020-11-22	0.0
327	2020-11-23	0.0
328	2020-11-24	0.0
329	2020-11-25	0.0
330	2020-11-26	0.0
331	2020-11-27	0.0
332	2020-11-28	0.0
333	2020-11-29	0.0
334	2020-11-30	0.0
335	2020-12-01	0.0
336	2020-12-02	0.0
337	2020-12-03	0.0
338	2020-12-04	0.0
339	2020-12-05	0.0
340	2020-12-06	0.0
341	2020-12-07	0.0
342	2020-12-08	0.0
343	2020-12-09	0.0
344	2020-12-10	0.0
345	2020-12-11	0.0
346	2020-12-12	0.0
347	2020-12-13	0.0
348	2020-12-14	0.0
349	2020-12-15	0.0
350	2020-12-16	0.0
351	2020-12-17	0.0
352	2020-12-18	0.0
353	2020-12-19	0.0
354	2020-12-20	0.0
355	2020-12-21	0.0
356	2020-12-22	0.0
357	2020-12-23	0.0
358	2020-12-24	0.0
359	2020-12-25	0.0
360	2020-12-26	0.0
361	2020-12-27	0.0
362	2020-12-28	0.0
363	2020-12-29	0.0
364	2020-12-30	0.0
365	2020-12-31	0.0

In[11]:

```
# Create a stacked area plot with less space.
df.plot(kind='area', figsize=(20,10), fontsize=20, linewidth=0, stacked=True)
plt.legend(loc='upper left', bbox_to_anchor=(1.0, 1.0), fontsize=20)
```

Out[11]:

In[12]:

```
df.dtypes
```

Out[12]:

```
date                                    object
Name                                   float64
dtype: object
```

In[13]:

```
df[(df['Name'] > 0.5) & (df['date'] == '2020-01-01')]
```

Out[13]:

	date	Name
0	2020-01-01	0.0

In[14]:

```
df['Name'].sum(axis=0).sum(axis=0)/df['Name'].count()
```

Out[14]:

```
0.0
```

In[15]:

```
df['Name'].sum(axis=0).sum(axis=0)/df['Name'].count()*100
```

Out[15]:

```
0.0
```

In[16]:

```
df['Name'].sum(axis=0).sum(axis=0)/df['Name'].count()*100*365
```

Out[16]:

```
0.0
```

```
Out[12]:
```

	positive	recovered	death	Daily Diff
date				
2020-03-14	3.0	0.0	0.0	3.0daily
2020-03-15	3.0	0.0	0.0	0.0daily
2020-03-16	3.0	0.0	0.0	0.0daily
2020-03-17	3.0	0.0	0.0	0.0daily
2020-03-18	3.0	0.0	0.0	0.0daily
2020-03-19	3.0	0.0	0.0	0.0daily
2020-03-20	3.0	0.0	0.0	0.0daily
2020-03-21	3.0	0.0	0.0	0.0daily
2020-03-22	3.0	0.0	0.0	0.0daily
2020-03-23	3.0	0.0	0.0	0.0daily
2020-03-24	3.0	0.0	0.0	0.0daily
2020-03-25	3.0	0.0	0.0	0.0daily
2020-03-26	3.0	0.0	0.0	0.0daily
2020-03-27	3.0	0.0	0.0	0.0daily
2020-03-28	3.0	0.0	0.0	0.0daily
2020-03-29	3.0	0.0	0.0	0.0daily
2020-03-30	3.0	0.0	0.0	0.0daily
2020-03-31	3.0	0.0	0.0	0.0daily
2020-04-01	3.0	0.0	0.0	0.0daily
2020-04-02	3.0	0.0	0.0	0.0daily
2020-04-03	3.0	0.0	0.0	0.0daily

Growth of different types of cases

```
In [108]: #Calculating the Mortality Rate and Recovery Rate
          print("Mortality Rate : {0:.3f}".format(Mortality_Rate*100),"%")
          print("Recovery Rate : {0:.3f}".format(Recovery_Rate*100),"%")
          #Calculating the Death Rate and Recovery Rate for cases in the last week
          Death_Rate_Last_Week = Death_Last_Week/Confirmed_Last_Week
          Recovery_Rate_Last_Week = Recovery_Last_Week/Confirmed_Last_Week
          print("Mortality Rate last week : {0:.3f}".format(Death_Rate_Last_Week*100),"%")
          print("Recovery Rate last week : {0:.3f}".format(Recovery_Rate_Last_Week*100),"%")
          #Calculating the Death Rate and Recovery Rate for cases in the last 2 weeks
          Death_Rate_Last_2_Weeks = Death_Last_2_Weeks/Confirmed_Last_2_Weeks
          print("Mortality Rate last 2 weeks : {0:.3f}".format(Death_Rate_Last_2_Weeks*100),"%")
          print("Recovery Rate last 2 weeks : {0:.3f}".format(Recovery_Rate_Last_2_Weeks*100),"%")

Out[108]: Mortality Rate : 3.117 %
          Recovery Rate : 96.883 %
          Mortality Rate last week : 0.000 %
          Recovery Rate last week : 100.000 %
          Mortality Rate last 2 weeks : 0.000 %
          Recovery Rate last 2 weeks : 100.000 %
```

Recovery Rate

Mortality Rate

Mortality rate = (Number of Death Cases / Number of Confirmed Cases) x 100

Recovery Rate = (Number of Recovered Cases / Number of Confirmed Cases) x 100

Mortality Rate last week gives us an idea of how the country is fighting the virus, and vice versa.

Recovery Rate last week and last 2 weeks gives us a good idea, whether superficial reason to why number of Closed Cases increasing.

```
In [109]: #Calculating the Mortality Rate and Recovery Rate for cases recorded in the last week
          print("Mortality Rate last week : {0:.3f}".format(Death_Rate_Last_Week*100),"%")
          print("Recovery Rate last week : {0:.3f}".format(Recovery_Rate_Last_Week*100),"%")
          #Calculating the Mortality Rate and Recovery Rate for cases recorded in the last 2 weeks
          print("Mortality Rate last 2 weeks : {0:.3f}".format(Death_Rate_Last_2_Weeks*100),"%")
          print("Recovery Rate last 2 weeks : {0:.3f}".format(Recovery_Rate_Last_2_Weeks*100),"%")

Out[109]: Mortality Rate last week : 0.000 %
          Recovery Rate last week : 100.000 %
          Mortality Rate last 2 weeks : 0.000 %
          Recovery Rate last 2 weeks : 100.000 %
```

Daily increase in different types of Cases

7 Days Rolling Mean of Daily Increase of Confirmed, Recovered and Death Cases

Growth Factor

Growth factor is the factor by which a quantity multiplies itself over time. The formula used is:

Formula: Every day's new (Confirmed,Recovered,Death) cases /(Confirmed,Recovered,Death) cases the previous day.

A growth factor above 1 indicates an increase in new cases.

A growth factor **above 1 but trending downward is a positive sign, whereas **a growth factor constantly above 1 is the sign of a pandemic.**

A growth factor **between 0 and 1 indicates a decrease in new cases but **a trend that is trending upward is a bad sign.**

A growth factor **constant at 1 indicates there is no change in any kind of cases.

A growth factor **below 1 indicates an increase in new cases and Closed Cases on the previous day.

A growth factor **above 1 but trending downward is a positive sign.

A growth factor **constant at 1 indicates there is no change in any kind of cases.

A growth factor **below 1 indicates an increase in new cases, **but trending from positive one getting recovered or dying as compared to the Confirmed Cases.

```
In [110]: #Calculating Growth Factor for different types of cases
          cnf = lambda i,confirmed,confirmed_minus_one,frac=False:(confirmed[i]/confirmed_minus_one[i])
          def gf(confirmed,confirmed_minus_one,frac=False):
              return [cnf(i,confirmed,confirmed_minus_one,frac) for i in range(len(confirmed))]
          #Calculating Growth Factor for cases recorded in the last week
          Confirmed_Growth = gf(Confirmed,Confirmed[-1:-1-len(Confirmed):-1])
          Recovered_Growth = gf(Recovered,Recovered[-1:-1-len(Recovered):-1])
          Death_Growth = gf(Death,Death[-1:-1-len(Death):-1])
          Closed_Growth = gf(Closed,Closed[-1:-1-len(Closed):-1])
          #Calculating Growth Factor for cases recorded in the last 2 weeks
          Confirmed_Growth_2 = gf(Confirmed,Confirmed[-2:-2-len(Confirmed):-1])
          Recovered_Growth_2 = gf(Recovered,Recovered[-2:-2-len(Recovered):-1])
          Death_Growth_2 = gf(Death,Death[-2:-2-len(Death):-1])
          Closed_Growth_2 = gf(Closed,Closed[-2:-2-len(Closed):-1])
```

Gatesone Growth Factor of different types of cases

Growth Factor for Active and Closed Cases

Growth factor is the factor by which a quantity multiplies itself over time. The formula used is:

Formula:Every day's new (Active and Closed) cases /(Active and Closed) cases the previous day.

A growth factor above 1 indicates an increase corresponding cases.

A growth factor **between 0 and 1 indicates a decrease but **trending in a positive sign.**

A growth factor **constant at 1 indicates there is no change in any kind of cases.

A growth factor **below 1 indicates an increase in cases, but **trending from positive one getting recovered or dying as compared to the Confirmed Cases.

```
In [111]: #Calculating Growth Factor for Active and Closed Cases
          Active_Growth = gf(Active,Active[-1:-1-len(Active):-1])
          Closed_Growth = gf(Closed,Closed[-1:-1-len(Closed):-1])
          #Calculating Growth Factor for cases recorded in the last 2 weeks
          Active_Growth_2 = gf(Active,Active[-2:-2-len(Active):-1])
          Closed_Growth_2 = gf(Closed,Closed[-2:-2-len(Closed):-1])
```

Gatesone Growth Factor of Active and Closed Cases

Growth Factor constantly above 1 is an clear indication of Exponential increase in all form of cases.

Rate of Doubling for Confirmed Cases

```
In [112]: #pip install
          #pip install sympy
          #pip install numpy
          #pip install matplotlib
          #pip install pandas
          #pip install sklearn
          #pip install xgboost
          #pip install lightgbm
          #pip install fbprophet
          #pip install pyramidal
          #pip install statsmodels
          #pip install keras
          #pip install tensorflow
          #pip install pmdarima
          #pip install plotly
          #pip install cufflinks
          #pip install chart_studio
          #pip install plotly_express
          #pip install dtale
          #pip install jupyter_dash
          #pip install dash_core_components
          #pip install dash_html_components
          #pip install dash_bootstrap_components
          #pip install dash_cytoscape
          #pip install dash_dangerous_components
          #pip install dash_daq
          #pip install dash_gif_components
          #pip install dash_gif_components
          #pip install dash_interactive_graphviz
          #pip install dash_reusable_components
          #pip install dash_table_experiments
          #pip install dash_table_experiments
          #pip install dash_vtkjs
          #pip install dash_vtk
          #pip install dash_auth
          #pip install dash_dangerous_components
          #pip install dash_daq
          #pip install dash_gif_components
          #pip install dash_interactive_graphviz
          #pip install dash_reusable_components
```

In [101]:
```
print(y_pred_test_prediction)
y_pred_test_prediction.tail()
```

```
[2.07000000 2.07000000 2.07000000 2.07000000 2.07000000 2.07000000
 
```