

# Untitled

May 2, 2020

```
[228]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="darkgrid")
```

```
[229]: df = pd.read_csv("Data/Time Series COVID-19 Confirmed Global - Regression Task_1_2020_05_02_19_00_00.csv")
df.head()
```

```
[229]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 \
0 NaN Afghanistan 33.0000 65.0000 0 0 0
1 NaN Albania 41.1533 20.1683 0 0 0
2 NaN Algeria 28.0339 1.6596 0 0 0
3 NaN Andorra 42.5063 1.5218 0 0 0
4 NaN Angola -11.2027 17.8739 0 0 0
```

```
1/25/20 1/26/20 1/27/20 ... 3/18/20 3/19/20 3/20/20 3/21/20 \
0 0 0 0 ... 22 22 24 24
1 0 0 0 ... 59 64 70 76
2 0 0 0 ... 74 87 90 139
3 0 0 0 ... 39 53 75 88
4 0 0 0 ... 0 0 1 2
```

```
3/22/20 3/23/20 3/24/20 3/25/20 3/26/20 3/27/20
0 40 40 74 84 94 110
1 89 104 123 146 174 186
2 201 230 264 302 367 409
3 113 133 164 188 224 267
4 2 3 3 3 4 4
```

[5 rows x 70 columns]

```
[230]: df = df[df['Country/Region'] == 'Thailand']
```

```
[231]: df.head()
```

```
[231]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 \
209 NaN Thailand 15.0 101.0 2 3 5

1/25/20 1/26/20 1/27/20 ... 3/18/20 3/19/20 3/20/20 3/21/20 \
209 7 8 8 ... 212 272 322 411

3/22/20 3/23/20 3/24/20 3/25/20 3/26/20 3/27/20
209 599 721 827 934 1045 1136

[1 rows x 70 columns]
```

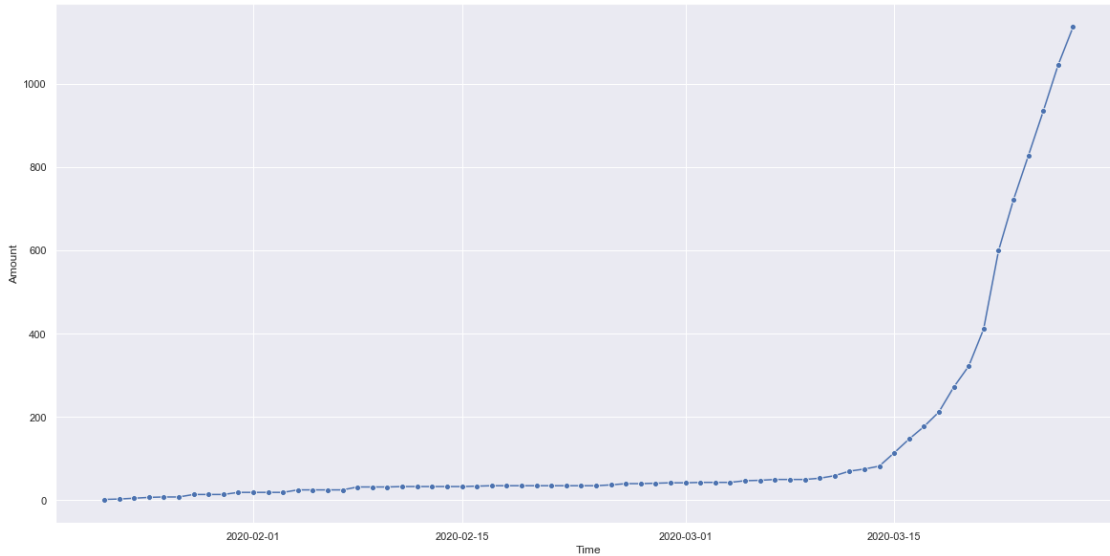
```
[232]: df = pd.DataFrame({
    'Time': pd.to_datetime(df.columns[4:]),
    'n': df.values[0][4:].astype(np.int)
})

df['t'] = df.Time.apply(lambda x: x.toordinal())
```

```
[233]: df.head()
```

```
[233]:      Time  n      t
0 2020-01-22  2  737446
1 2020-01-23  3  737447
2 2020-01-24  5  737448
3 2020-01-25  7  737449
4 2020-01-26  8  737450
```

```
[234]: plt.figure(figsize=(20, 10))
sns.lineplot(marker = 'o', x = df.Time, y = df.n, legend=False)
plt.xlabel('Time')
plt.ylabel('Amount')
plt.show()
```



```
[235]: df.head(10)
```

```
[235]:
```

	Time	n	t
0	2020-01-22	2	737446
1	2020-01-23	3	737447
2	2020-01-24	5	737448
3	2020-01-25	7	737449
4	2020-01-26	8	737450
5	2020-01-27	8	737451
6	2020-01-28	14	737452
7	2020-01-29	14	737453
8	2020-01-30	14	737454
9	2020-01-31	19	737455

```
[236]: df2 = df.copy()
```

```
[237]: df2['Time'] = df2['Time'] - pd.to_timedelta(7, unit='d')
```

```
[250]: df2 = df2.groupby(pd.Grouper(key='Time', freq='W-MON'))['n'].mean().
        ↪reset_index()
```

```
[251]: df2
```

```
[251]:
```

	Time	n
0	2020-01-20	5.500000
1	2020-01-27	16.857143
2	2020-02-03	28.000000
3	2020-02-10	33.428571

```

4 2020-02-17    35.000000
5 2020-02-24    40.714286
6 2020-03-02    47.285714
7 2020-03-09    85.714286
8 2020-03-16   387.714286
9 2020-03-23   985.500000

```

```

[201]: mean_week = []
       for i in df2.n:
           for _ in range(7):
               mean_week.append(i)

```

```

[202]: len(mean_week)

```

```

[202]: 70

```

```

[219]: df['mean_week'] = np.array(mean_week[2:68]).astype(float)

```

```

[222]: df.dtypes

```

```

[222]: Time          datetime64[ns]
       n              int32
       t              int64
       mean_week      float64
       dtype: object

```

## 0.1 Polinomial Regression (2 Degree)

```

[121]: from sklearn.linear_model import LinearRegression
       from sklearn.preprocessing import PolynomialFeatures
       from sklearn.model_selection import GridSearchCV

```

```

[224]: X = df[['t', 'mean_week']].values
       y = df['n'].values.reshape(-1,1)

```

```

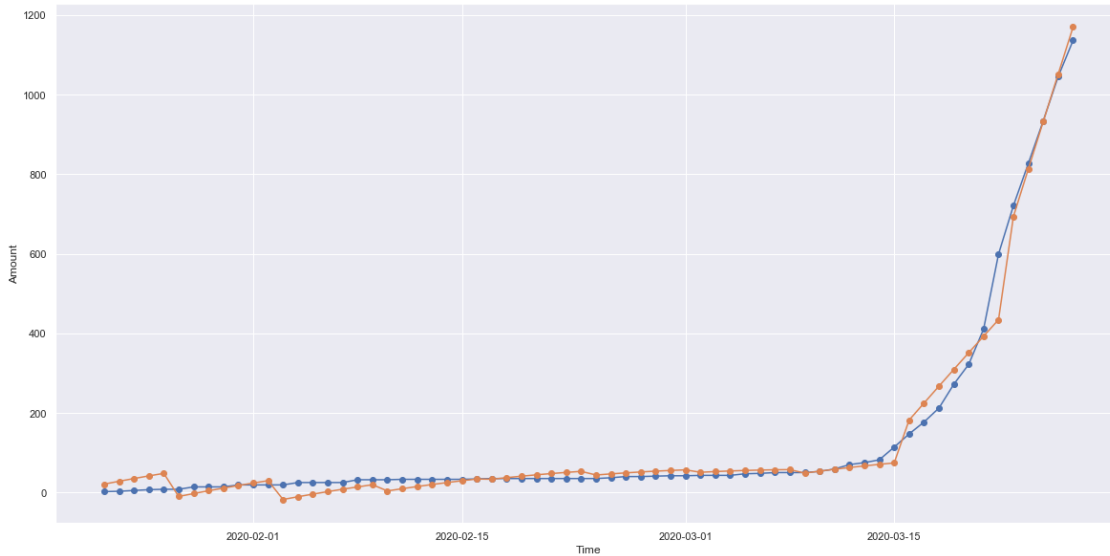
[225]: X_poly = PolynomialFeatures(degree=2).fit_transform(X)

       reg = LinearRegression().fit(X_poly, y)

       pred = reg.predict(X_poly)

       plt.figure(figsize=(20, 10))
       plt.plot(df.Time, df.n, marker = 'o')
       plt.plot(df.Time, pred, marker = 'o')
       plt.xlabel('Time')
       plt.ylabel('Amount')
       plt.show()

```



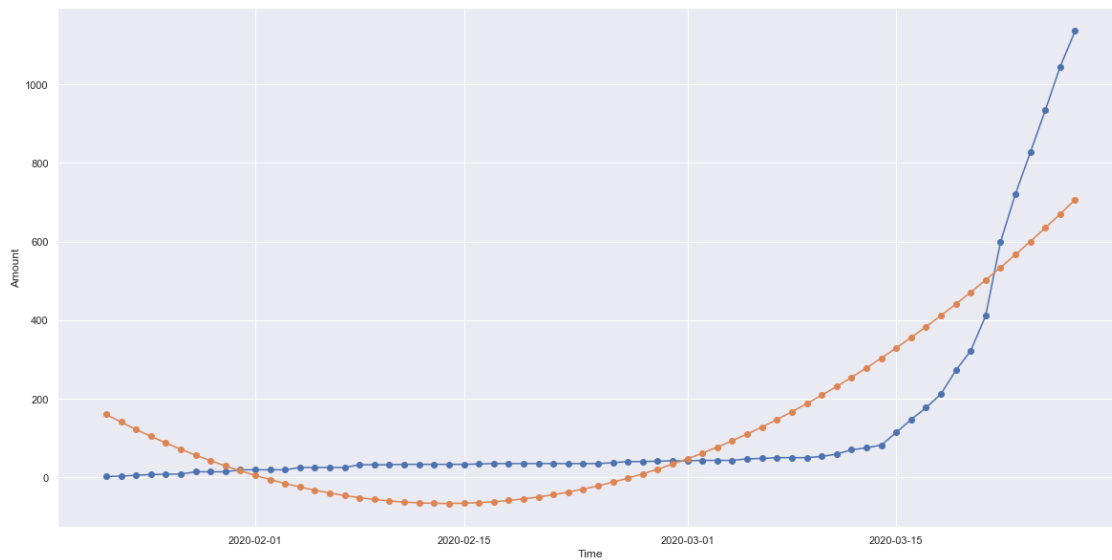
```
[226]: X = df[['t']].values
y = df['n'].values.reshape(-1,1)
```

```
[227]: X_poly = PolynomialFeatures(degree=2).fit_transform(X)

reg = LinearRegression().fit(X_poly, y)

pred = reg.predict(X_poly)

plt.figure(figsize=(20, 10))
plt.plot(df.Time,df.n,marker = 'o')
plt.plot(df.Time,pred,marker = 'o')
plt.xlabel('Time')
plt.ylabel('Amount')
plt.show()
```



[ ]: