Regression

May 4, 2020

[98]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="darkgrid")
```

[99]:
```
df = pd.read_csv("Data/Time Series COVID-19 Confirmed Global - Regression Task
↪- Data.csv")
df.head()
```

[99]:

	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]

[100]:
```
df = df[df['Country/Region'] == 'Thailand']
```

[101]:
```
df.head()
```

[101]:
```
    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]
```

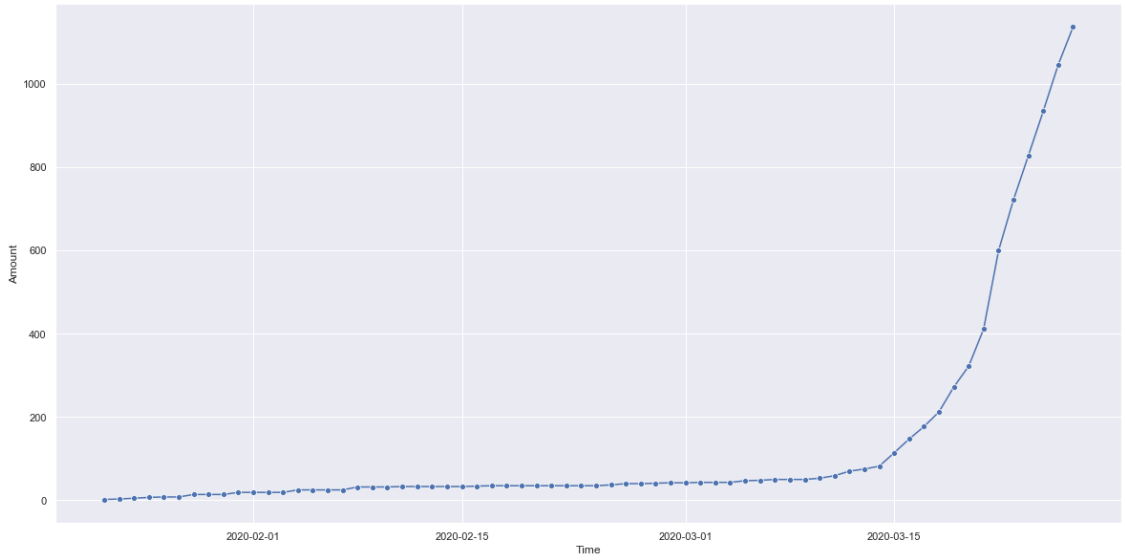
[102]:
```
df = pd.DataFrame({
    'Time': pd.to_datetime(df.columns[4:]),
    'n':df.values[0][4:].astype(np.int)
})

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

[103]:
```
df.head()
```

[103]:
```
        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
```

[104]:
```
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()
```



[105]:
```
df.head(10)
```

[105]:

	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

[106]:
```
df['n'].values[-1]
```

[106]: 1136

[114]:
```
# insert feature n when t-1 and t-2
n = df['n'].values
name_list = ['n1', 'n2', 'n3', 'n4']
df2 = df.drop(np.arange(4)).reset_index(drop = True)
for i in np.arange(4)[::-1]:
    df2[name_list[-i + 3]] = n[i:i-4]
```

[115]:
```
df2
```

[115]:

	Time	n	t	n1	n2	n3	n4
0	2020-01-26	8	737450	7	5	3	2
1	2020-01-27	8	737451	8	7	5	3
2	2020-01-28	14	737452	8	8	7	5
3	2020-01-29	14	737453	14	8	8	7
4	2020-01-30	14	737454	14	14	8	8
..	…	…	…	…	…	…	…
57	2020-03-23	721	737507	599	411	322	272
58	2020-03-24	827	737508	721	599	411	322
59	2020-03-25	934	737509	827	721	599	411
60	2020-03-26	1045	737510	934	827	721	599
61	2020-03-27	1136	737511	1045	934	827	721

[62 rows x 7 columns]

0.1 Polinomial Regression (2 Degree)

[125]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
```

[126]:
```
X = df2[['t','n1', 'n2', 'n3', 'n4']].values
y = df2['n'].values.reshape(-1,1)
```

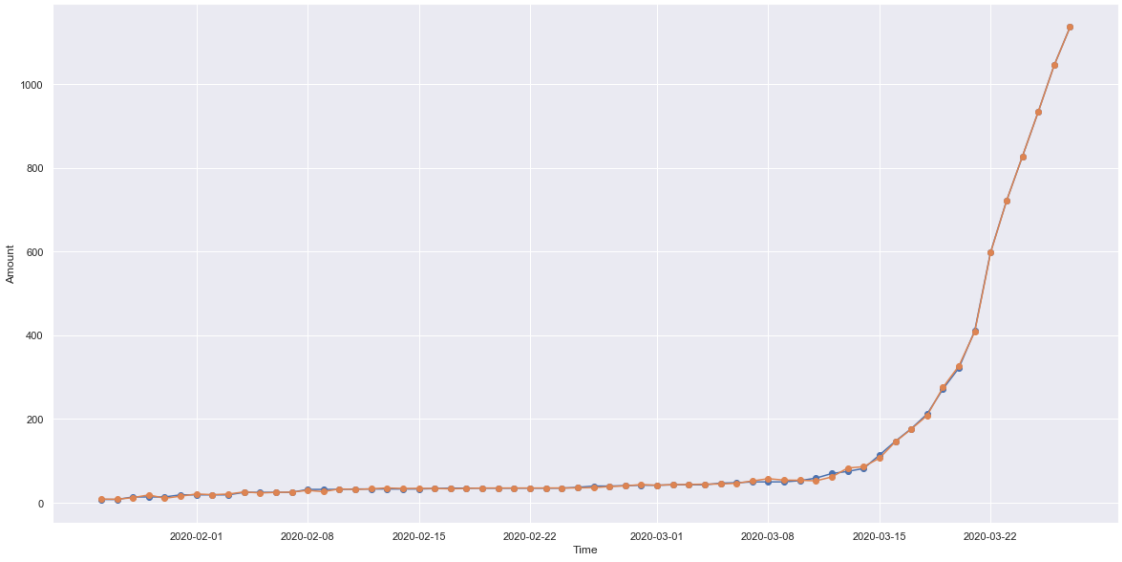
[129]:
```
X_poly = PolynomialFeatures(degree=2).fit_transform(X)

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

pred = reg.predict(X_poly)

plt.figure(figsize=(20, 10))
plt.plot(df2.Time,df2.n,marker = 'o')
plt.plot(df2.Time,pred,marker = 'o')
plt.xlabel('Time')
plt.ylabel('Amount')
plt.show()

print(f'MSE of model : {mean_squared_error(y, pred):.2f}')
print(f'Score of model : {reg.score(X_poly, y):.2f}')
```




```
MSE of model : 8.52
Score of model : 1.00
```

[ ]: