Python史上最难包的安装—fbprophet

序言:

网上截图称史上最难真不为过，目前安装过的最难最不容易成功的包

python 安装fbprophet模块的艰辛历程

fbprophet这个模块是我目前见过最难装的一个模块，我安装这个包安装了3天，气死我了，需求的依赖包太多，而且对依赖包的版本有极高的要求，所以建议在装这个模块的时候在一个空的虚拟环境下安装，这样依赖包就直接可以下载自己想要的版本依赖包。刚开始的时候用pip安装，出现各种错误，由于忘截图了，就不一一展示所报的错误，后来同事说用anaconda安装比较简单，就自己下载了anaconda，anaconda自己有不熟，硬着头皮上。安装好anaconda后。在python3.6.2版本上安装，总是包一个错误
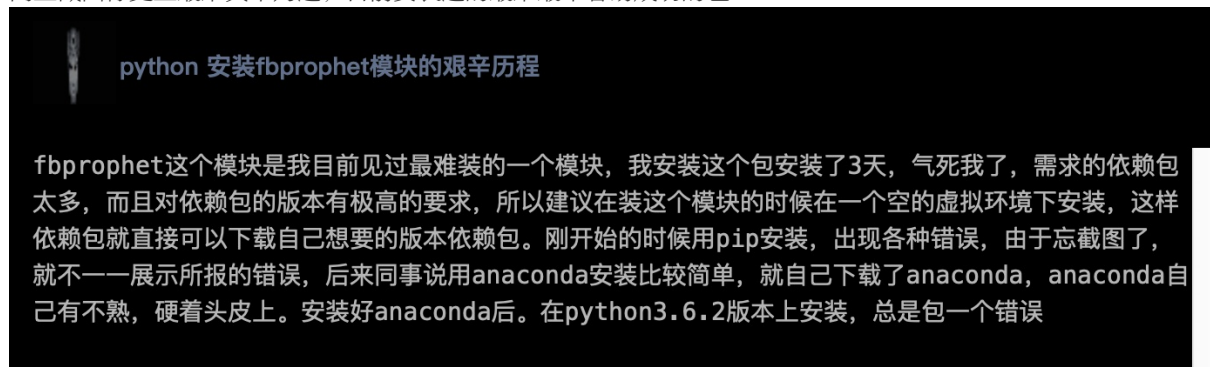
Fbprophet安装步骤（CentOS）,fbprophet,centOS

史上最难装的库。。。
最后成解决办法是，在anaconda建立一个新的环境，这样比较好安装

严格按照以下步骤安装即可（不要考虑什么pip，屡试屡败）

1.先安装Anaconda

http://agroup.baidu.com/codingarts/md/article/3545456

2.依据官方文档

https://facebook.github.io/prophet/docs/installation.html#python

Anaconda

Use `conda install gcc` to set up gcc. The easiest way to install Prophet is through conda-forge: `conda install -c conda-forge fbprophet`.

`conda install gcc` 这一步很好安装，基本都能成功
坑爹的是`conda install -c conda-forge fbprophet`，想一下成功那是不可能的；

按照如下步骤：

- 用conda先建一个新的环境

```
[root@malexa-input01 ~]# python --version
Python 3.7.0
[root@malexa-input01 ~]# conda create -n my_env python=3.7
Solving environment: \
Warning: 4 possible package resolutions (only showing differing packages):
  - https://mirrors.ustc.edu.cn/anaconda/pkgs/free::libffi-3.2.1-1, https://mirrors.ustc.edu.cn/anaconda/pkgs/free::zlib-1.2.11-0
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free::libffi-3.2.1-1, https://mirrors.ustc.edu.cn/anaconda/pkgs/free::zlib-1.2.11-0
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free::zlib-1.2.11-0, https://mirrors.ustc.edu.cn/anaconda/pkgs/free::libffi-3.2.1-1
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free::libffi-3.2.1-1, https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free::zlib-1.2.11done


==> WARNING: A newer version of conda exists. <==
  current version: 4.5.11
  latest version: 4.9.2
```

成功后激活新环境再执行安装即可：

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate my_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate

[root@malexa-input01 ~]# source activate my_env
(my_env) [root@malexa-input01 ~]# conda install -c conda-forge fbprophet
```

3.测试

```
1.  注：首次导入包的时候直接用 from fbprophet import Prophet 可能存在问题；
2.  可直接 import fbprophet
```

```
(my_env) [root@malexa-input01 ~]# python
Python 3.7.8 | packaged by conda-forge | (default, Nov 17 2020, 23:45:15)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from fbprophet import Prophet
Importing plotly failed. Interactive plots will not work.
>>> import fbprophet
>>> from fbprophet import Prophet
>>> import pandas as pd
>>> df = pd.DataFrame({
...     'ds': pd.date_range(start='1/1/2018', periods=30, freq='M'),
...     'y':(1,2,1,5,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2)
... })
>>> print(df)
           ds  y
0  2018-01-31  1
1  2018-02-28  2
2  2018-03-31  1
3  2018-04-30  5
4  2018-05-31  1
5  2018-06-30  2
6  2018-07-31  1
7  2018-08-31  2
8  2018-09-30  1
9  2018-10-31  2
10 2018-11-30  1
11 2018-12-31  2
12 2019-01-31  1
13 2019-02-28  2
14 2019-03-31  1
15 2019-04-30  2
16 2019-05-31  1
17 2019-06-30  2
18 2019-07-31  1
19 2019-08-31  2
20 2019-09-30  1
21 2019-10-31  2
22 2019-11-30  1
23 2019-12-31  2
24 2020-01-31  1
25 2020-02-29  2
26 2020-03-31  1
27 2020-04-30  2
28 2020-05-31  1
29 2020-06-30  2
>>> m=Prophet()
>>> m.fit(df)
```

```
>>> future = m.make_future_dataframe(periods=2,freq='M')
>>> forecast = m.predict(future)
>>> forecast_df=forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
>>> print(forecast_df)
           ds      yhat  yhat_lower  yhat_upper
0   2018-01-31  1.034688    0.500833    1.556040
1   2018-02-28  2.238416    1.729246    2.779713
2   2018-03-31  1.240932    0.734910    1.760734
3   2018-04-30  3.313429    2.794666    3.844734
4   2018-05-31  1.239989    0.715980    1.752164
5   2018-06-30  2.234264    1.700027    2.779004
6   2018-07-31  1.030062    0.528217    1.537835
7   2018-08-31  2.103231    1.581311    2.634401
8   2018-09-30  1.160127    0.624908    1.727085
9   2018-10-31  2.187858    1.682038    2.660982
10  2018-11-30  1.177730    0.648688    1.723117
11  2018-12-31  2.129200    1.641710    2.667505
12  2019-01-31  1.000271    0.465081    1.540812
13  2019-02-28  1.846800    1.345283    2.391661
14  2019-03-31  0.874330    0.344608    1.408088
15  2019-04-30  3.156200    2.619403    3.689912
16  2019-05-31  0.860295    0.345758    1.419572
17  2019-06-30  1.854416    1.339787    2.376479
18  2019-07-31  0.970741    0.403580    1.486050
19  2019-08-31  1.896893    1.394130    2.413538
20  2019-09-30  0.840345    0.352435    1.398993
21  2019-10-31  1.811964    1.295988    2.370440
22  2019-11-30  0.822436    0.321615    1.334290
23  2019-12-31  1.870507    1.326478    2.382360
24  2020-01-31  0.965621    0.414856    1.493814
25  2020-02-29  1.914260    1.383070    2.431282
26  2020-03-31  0.885650    0.355508    1.406699
27  2020-04-30  2.530719    1.977749    3.079410
28  2020-05-31  0.901713    0.373804    1.391459
29  2020-06-30  1.912630    1.410687    2.439980
30  2020-07-31  0.063806   -0.447209    0.611724
31  2020-08-31  1.410300    0.884302    1.895421
```