基于flask框架的prophet服务：

前提，已经安装了python，最好创建虚拟python环境，建议在anaconda中创建一个python的虚拟环境，以下在虚拟环境下安装和运行。

1. 安装flask： pip install flask， 或者 conda install flask
2. 创建项目目录，比如： flask\_prophet
3. 打开到项目目录: cd \*\*\*\flask\_prophet
4. 创建app.py
5. 在app.py文件中，导入flask

代码参考代码文件app.py和smartpark.py

1. 运行：

通过以下两种方式皆可：

Python app.py

Flask run

Flask run： 默认运行app.py文件

1. 运行后，在浏览器输入：

Localhost:5000/prophet

Localhost:5000/prophet\_hour

1. 之后的工作：
   1. 研究并启用flask的request和response方法
   2. Flask连接数据库（起个名叫分析数据库吧），可以研究下采用哪种数据库合适，选择一种
   3. 设计并实现，每隔几分钟，从大数据库（这个就是老艾说的，三个来源的数据库，provence +traffic+behavior）采集待分析的数据并保存至分析数据库，启用分析和预测功能，并保存结果
   4. 利用flask的request和response根据请求参数，把分析结果返回
   5. 测试流程
   6. 将flask服务部署到阿里服务器

关于Prophet：

中文讲解的很清楚的网址：

https://blog.csdn.net/anshuai\_aw1/article/details/83412058

官方讲解：

https://facebook.github.io/prophet/docs/quick\_start.html#python-api

主要用法有下面这些：

**按照小时，天，月，年进行预测：**

#历史数据（需要带有时间戳的数据），以下代码预测次后300个小时的数据

# predict by hour

df = pd.read\_csv('example\_time.csv')

m = Prophet(changepoint\_prior\_scale=0.01).fit(df)

future = m.make\_future\_dataframe(periods=300, freq='H')

forecast= m.predict(future)

fig = m.plot(forecast)

fig2 = m.plot\_components(forecast)

**限制预测时间段：**

#如果只是想预测6点之前的数据，通过下面的限制

# example: limit result that hour<6, but use all history to do predict

future2 = future.copy()

future2 = future2[future2['ds'].dt.hour < 6]

fcst = m.predict(future2)

fig = m.plot(fcst)

# predict by month

future = m.make\_future\_dataframe(periods=120, freq='M')

# predict by day

future = m.make\_future\_dataframe(periods=365)

future = m.make\_future\_dataframe(periods=365, freq='D' )

#by Year

future = m.make\_future\_dataframe(periods=2, freq='Y')

其他参数：

**趋势变化：**

# 如果趋势变化是过度拟合（灵活性太大）或不足（灵活性不足），则可以在使用输入参数之前调整稀疏强度changepoint\_prior\_scale。默认情况下，此参数设置为0.05

# 调大，更灵活，预测值变化幅度大； 调小，更不灵活，预测变化幅度小

m = Prophet(changepoint\_prior\_scale=0.5)

**增长函数：**

#增长函数： 默认使用分段线性函数，也可以指定成逻辑回归函数.

m = Prophet()

m = Prophet(growth= 'linear')

m = Prophet(growth='logistic')

**指定最大和最小值**

#只有在线性函数的情况下，可以指定最大和最小值

df['cap'] = 100

df['floor'] = 10

future['cap']=100

future['floor']=100

**输出：**

#全部历史数据和预测数据在forecast中,

#只输出预测值，以及预测值的高低范围，采用下面的语法，限定输出字段和倒数50天的结果

forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']][-50:]

#倒数第50~100个结果

forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']][-100:-50]

#打印

print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']][-100:-50])

**格式转换：**

#just get some results from forecast

out\_forecast = forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']][-10:]

# to json

jsondata = out\_forecast.to\_json()

#to html

htmldata = out\_forecast.to\_html()

#forecast所有的列：print(forecast.columns)

Index(['ds', 'trend', 'yhat\_lower', 'yhat\_upper', 'trend\_lower', 'trend\_upper',

'additive\_terms', 'additive\_terms\_lower', 'additive\_terms\_upper',

'weekly', 'weekly\_lower', 'weekly\_upper', 'yearly', 'yearly\_lower',

'yearly\_upper', 'multiplicative\_terms', 'multiplicative\_terms\_lower',

'multiplicative\_terms\_upper', 'yhat'],

dtype='object')

注意：

Flask服务的端口可以在app.py中修改

采用python app.py 可以采用端口和密码来修改，如：192.168.16.70:5001局域网内可以访问

但是，采用flask run，只能用localhost:5000 和127.0.0.1:5000来访问，修改接口和IP不起作用。