中芯国际数据分析

Demo1：

原始数据POC\_20210901\_Raw.CSV

计算出芯片的四个时间段差：

pod\_arrive - pod\_remove、trackin\_time - trackout\_time、load\_time - unload\_time、lotstart\_time - lotend\_time

Demo2：

处理好的数据：poc\_2021\_9.10.csv

计算各个时间差的均值、最小值、最大值、标准差

Demo3：

原始数据POC\_20210901\_Raw.CSV

计算出芯片的八个个时间段差：

track\_in-pod\_arrive、load-track\_in、lot\_start-load、lot\_end-lot\_start、unload-lot\_end、track\_out-unload、pod\_remove-track\_out、pod\_interval

Demo4：

处理好的数据：T09.csv

将每个机台每天arrive-remove时间段的折线图

Demo5：

处理好的数据：T02.csv

找到pod\_arrive-pod\_remove这一时间段内没有加工交集的数据，将表存成csv格式

Demo6：

处理好的数据：T01.csv

对start\_times和load\_times分别进行排序，分析未按照排队顺序加工的工件

LinearRegression.py：

处理好的数据：poc\_2021\_9.10.csv

处理好的数据用线性回归模型测试效果

MLP.py:

处理好的数据：poc\_2021\_9.10.csv

处理好的数据用MLP模型测试效果

中软.py:

处理好的数据：中软.xlsx

数据预处理成字典格式

并行度.py:

处理好的数据：T02.csv

查找每个机台产品加工时间段内有并集的数据，并记录并集加工的个数

折线图.py:

将csv文件数据画成折线图

Test.py:

1、对于n批要加工的对象，这里定义一个规则，然后找到在这个规则下最优的方案。

2、将n个对象排成一个序列作为函数参数，最终函数输出一个n个对象的加工总时间。序列中靠前的对象将有更高的优先级，这体现在一个tank在空闲时将优先分配给优先级更高的对象占用，这样也使得一台机器尽可能地处在饱和状态工作。

3、对于n个对象共有n!个优先级序列，最终找到一个函数输出最小即耗时最短的序列。