Project2 总结

- **Python 变量命名规范**

1. 模块名：小写字母，单词之间用_分割,eg: ad_stats.py
2. 包名：同模块名
3. 类名：单词首字母大写，中间需要分割的也需要大写，如：ConfigUtil
4. 全局变量名：大写字母，中间用_分割 eg: COLOR_WRITE
5. 普通变量：小写字母，单词之间用_分割 eg: this_is_a_var
6. 实例变量：，以_开头，其他和普通变量一样
7. 私有实例变量：以__开头（两个下划线），其他和普通变量一样
8. 专有变量，__开头,__结尾，一般为 python 的自有变量，
9. 普通函数：和普通变量一样
10. 私有函数：以__开头，其他和普通函数一样

- **框架结构规范**

1. FastRearchData：从多个不同文件（文件类型可以为 csv、database、pickle）中读取数据，进行数据的拼接和简单处理，生成结果对象，数据类型可以为 DF、NP 或者 HDF5 等;

2. IndicatorGallexy：存放指标计算方法，可以划分为多个类，传入数据，计算不同指标并返回计算得到的指标;

3. ModelEngine：模型的训练和评估（注意：在进行模型训练之前，需要预处理数据集，但是每次可以有不同的预处理操作，因此可以将 DataPreprocessing 写为一个新类，并将其装配到 ModelEngine 中;

- **argparse 的使用**

1. argparse 是一个命令行解析包，argparse 会从 sys.argv 中解析出定义好的参数，自动上呢工程才呢过帮助和实用信息。基本语法包括创建一个 ArgumentParse()参对象，调用 add_argument()方法添加参数，使用 parse_args()添加的参数，如下所示：

```
parser = argparse.ArgumentParser()
parser.add_argument('integer', type=int, help='display an integer')
args = parser.parse_args()
```

2. argument 对象：

class argparse.ArgumentParser(prog=None, usage=None, description=None, epilog=None, parents=[],formatter_class=argparse.HelpFormatter, prefix_chars='-', fromfile_prefix_cha

rs=None, argument_default=None,conflict_handler='error', add_help=True)

创建一个新的 ArgumentParser 对象。所有的参数应该以关键字参数传递。参数详细描述如下：

- prog - 程序的名字（默认：sys.argv[0]）
- usage - 描述程序用法的字符串（默认：从解析器的参数生成）
- description - 参数帮助信息之前的文本（默认：空）
- epilog - 参数帮助信息之后的文本（默认：空）
- parents - ArgumentParser 对象的一个列表，这些对象的参数应该包括进去
- formatter_class - 定制化帮助信息的类
- prefix_chars - 可选参数的前缀字符集（默认：‘-‘）
- fromfile_prefix_chars - 额外的参数应该读取的文件的前缀字符集（默认：None）
- argument_default - 参数的全局默认值（默认：None）
- conflict_handler - 解决冲突的可选参数的策略（通常没有必要）

add_help - 给解析器添加-h/–help 选项（默认：True）

3. add_argument()方法：

基本语法：add_argument(name or flags...[, action][, nargs][, const]

[, default][, type][, choices][, required][, help][, metavar][, dest])

必须区分可选参数和位置参数：可选的参数将以- 前缀标识，剩余的参数将被假定为位置参数：

4. Parse_args()方法：

基本语法：arse_args(args=None, namespace=None)

作用：将参数字符串转换成对象并设置成命名空间的属性，返回构成的命名空间

- **泛化处理**

对属性的访问（显示给出字符串）应放在类外执行，在类定义内，不能给出字符串访问属性名。函数的输入部分也要考虑实际，不能只传一个文件，要考虑到更多情况处理的可能。

- **个人总结**

通过本次练习，掌握了上一个练习中没怎么熟悉的 DataFrame 的使用，而且进一步的对 sklearn 和 pandas 包的使用有了了解，第一次尝试写类，写一个简单的类没有什么问题，但是复杂的还要后面更多的学习。这次练习中存在的最大问题就是命名规范和代码规范的问题，尤其是后者，在后面的学习中还要多多注意。