家乐福

销售预测与需求计划

项目文档

目录

[1. 项目背景与解决方案 5](#_Toc19709075)

[1.1. 项目背景 5](#_Toc19709076)

[1.1.1. 销售预测模型 5](#_Toc19709077)

[1.1.2. 需求计划模型 5](#_Toc19709078)

[1.2. 解决方案 5](#_Toc19709079)

[1.2.1. 销售预测模型 5](#_Toc19709080)

[1.2.2. 需求计划模型 6](#_Toc19709081)

[2. 系统设计与架构 8](#_Toc19709082)

[2.1. 系统概要 8](#_Toc19709083)

[2.2. 硬件架构 9](#_Toc19709084)

[2.2.1. Hadoop集群 9](#_Toc19709085)

[2.2.2. 高性能的独立服务器 9](#_Toc19709086)

[2.3. 工作流程(图示) 10](#_Toc19709087)

[2.4. 数据流程(图示) 10](#_Toc19709088)

[3. 数据处理与特征工程 11](#_Toc19709089)

[3.1. 数据源 11](#_Toc19709090)

[3.1.1. 交易记录数据 11](#_Toc19709091)

[3.1.2. 商品信息数据 11](#_Toc19709092)

[3.1.3. 库存信息数据 11](#_Toc19709093)

[3.1.4. 订货及运输信息数据 11](#_Toc19709094)

[3.1.5. 门店信息数据 11](#_Toc19709095)

[3.1.6. DM信息数据 11](#_Toc19709096)

[3.1.7. Coupon 信息数据 11](#_Toc19709097)

[3.1.8. 门店商品清单信息数据 11](#_Toc19709098)

[3.1.9. 日期信息数据 11](#_Toc19709099)

[3.1.10. 竞争对手信息数据 11](#_Toc19709100)

[3.1.11. 外部数据 12](#_Toc19709101)

[3.1.12. 门店单品信息 （forecast\_store\_item\_details） 12](#_Toc19709102)

[3.1.13. 存库单品送货信息 (forecast\_onstock\_order\_delivery\_mapping) 13](#_Toc19709103)

[3.1.14. 越库单品送货信息 （forecast\_xdock\_order\_mapping） 13](#_Toc19709104)

[3.1.15. 大仓单品信息 （forecast\_store\_item\_details） 13](#_Toc19709105)

[3.1.16. 系统日 （forecast\_calendar） 14](#_Toc19709106)

[3.1.17. 大仓订货日 （forecast\_dc\_order\_delivery\_mapping） 14](#_Toc19709107)

[3.1.18. 门店送货到店时间 （forecast\_stores\_delv\_time） 15](#_Toc19709108)

[3.1.19. 门店可订货单品信息（v\_forecast\_inscope\_store\_item\_details） 15](#_Toc19709109)

[3.1.20. 大仓可订货单品信息（v\_forecast\_inscope\_dc\_item\_details） 15](#_Toc19709110)

[3.1.21. 当日应在门店下订单的存库单品信息（v\_forecast\_daily\_onstock\_order\_items） 15](#_Toc19709111)

[3.1.22. 当日应在门店下订单的越库单品信息（v\_forecast\_daily\_xdock\_order\_items） 15](#_Toc19709112)

[3.2. 数据处理目标以及方法 15](#_Toc19709113)

[3.2.1. 区分DM销售以及非DM销售 15](#_Toc19709114)

[3.2.2. 剔除有计划的BP(基于销售小票) 15](#_Toc19709115)

[3.2.3. 训练数据出现缺货时的日销量 16](#_Toc19709116)

[3.2.4. 周销量预测到天销量预测的转化 16](#_Toc19709117)

[3.2.5. DM销售预测到天销售预测的变化 17](#_Toc19709118)

[3.3. 数据处理流程 17](#_Toc19709119)

[3.3.1. 普通模型数据处理（由上至下顺序执行） 17](#_Toc19709120)

[3.3.2. DM模型数据处理（由上至下顺序执行） 18](#_Toc19709121)

[3.4. 输出表说明 18](#_Toc19709122)

[3.4.1. 模型特征信息表 18](#_Toc19709123)

[3.4.2. 需求计划信息表 24](#_Toc19709124)

[4. 销售预测的建模 27](#_Toc19709125)

[4.1. 非DM期间的销售预测模型 27](#_Toc19709126)

[4.1.1. 模型介绍 27](#_Toc19709127)

[4.1.2. 模型训练使用的特征 27](#_Toc19709128)

[4.1.3. 模型训练的逻辑 29](#_Toc19709129)

[4.2. DM期间的销售预测模型 32](#_Toc19709130)

[4.2.1. 模型介绍 32](#_Toc19709131)

[4.2.2. 模型训练使用的特征 32](#_Toc19709132)

[4.2.3. 模型训练的逻辑 33](#_Toc19709133)

[5. 需求计划的设计 36](#_Toc19709134)

[5.1. 需求计划介绍 36](#_Toc19709135)

[5.2. 需求计划的架构 36](#_Toc19709136)

[5.3. 需求计划系统模块 37](#_Toc19709137)

[5.4. 模块结构 37](#_Toc19709138)

[5.4.1. 订单生成 37](#_Toc19709139)

[5.4.2. 系统监控 38](#_Toc19709140)

[5.5. 详细设计 39](#_Toc19709141)

[5.5.1. 数据库设计 39](#_Toc19709142)

[5.6. DM订单模块设计 41](#_Toc19709143)

[5.7. 常规订单模块设计 42](#_Toc19709144)

[6. 使用说明 43](#_Toc19709145)

[6.1. 销售预测与需求计划项目文件夹内容介绍 43](#_Toc19709146)

[6.2. 执行与设定方法 44](#_Toc19709147)

[6.2.1. 开启自动化执行 44](#_Toc19709148)

[6.2.2. 代码中的参数设定 45](#_Toc19709149)

[6.3. 运维方法 47](#_Toc19709150)

[6.3.1. 检查运行状态 47](#_Toc19709151)

[6.3.2. 检查运行结果 47](#_Toc19709152)

[6.3.3. 订单逻辑调试 48](#_Toc19709153)

[6.3.4. 示例 49](#_Toc19709154)

[6.3.5. 重新生成 49](#_Toc19709155)

[7. 测试方法 51](#_Toc19709156)

[7.1. 测试环境 51](#_Toc19709157)

[7.2. 测试方案 51](#_Toc19709158)

[7.2.1. 系统环境检查 51](#_Toc19709159)

[7.2.2. 销售预测程序测试 52](#_Toc19709160)

[7.2.3. 需求计划的测试 54](#_Toc19709161)

[8. 安装和部署方法 57](#_Toc19709162)

[8.1. 系统环境要求 57](#_Toc19709163)

[8.2. 安装和配置 57](#_Toc19709164)

[8.2.1. Airflow 57](#_Toc19709165)

[8.2.2. Python包 58](#_Toc19709166)

[8.3. 初始化订单参数 58](#_Toc19709167)

[8.4. 初始化数据库表 59](#_Toc19709168)

# 项目背景与解决方案

## 项目背景

数据咨询公司Artefact为家乐福开发一套销售预测与需求计划的模型，旨在提高供应商到货率，降低大仓和门店库存，已提高大仓和门店的商品上架率。此模型分为销售预测模型和需求计划模型两部分。

### 销售预测模型

Artefact为家乐福开发一个基于机器学习的销售预测模型。此模型将主要食用家乐福数据池历史数据，按周预测华东区49家大型超市3个供应商（宝洁，联合利华和雀巢）的未来9周的销售预测。产品范围涵盖正常商品和海报单品。此项目模型为概念验证型模型。

### 需求计划模型

Artefact将结合供应商和大仓订单参数，把销售预测转化成大仓和门店的订单。此项目模型为概念验证型模型。

## 解决方案

项目解决方案主要分为两个板块：销售预测模型和需求计划模型。

### 销售预测模型

销售预测模型主要分为5个步骤：商业数据源获取，数据清理和转化，模型特征转化，模型训练，结果输出。

步骤一 商业数据源获取

在项目前期，Artefact的咨询顾问对家乐福业务团队与IT团队进行访谈，明确影响销售的商业因素，并得知商业数据是如何储存在数据池中。采访记录与采访报告参见交付物3《访谈报告》。

步骤二 数据清理和转化

Artefact的数据工程师对于数据池中的数据库进行处理。具体处理过程参照技术文档。

步骤三 模型特征转化

Artefact的数据科学家将表中数据提炼成DM与非DM模型的特征，具体转化过程参照技术文档。

步骤四 模型训练

模型的特征将会加入到DM和非DM模型中进行模型训练：非DM模型为单周单品模型，即每周每单品一个模型；DM模型为一个模型。具体训练过程参照技术文档。

步骤五 结果输出

模型将会产生销售预测的结果，非DM模型输出结果为单品单店每周的销量预测，DM模型输出结果为单品单店DM档期的销量预测。具体输出过程参照技术文档。

### 需求计划模型

需求计划模型分为非DM模型和DM模型

#### 非DM模型

非DM需求计划模型主要分为5个步骤：门店每周非DM销售量分成每天销售量，门店订货参数识别，门店未来库存估计，门店下单量计算，大仓下单量计算，具体处理过程参照技术文档。

步骤一 门店每周非DM销售量分成每天销售量

根据单品单店的历史销售规律将门店每周非DM销售量分解成每天销售量，具体处理过程参照技术文档。

步骤二 门店订货参数识别

把供应链团队提供的订货参数倒入模型中，具体处理过程参照技术文档。

步骤三 门店未来库存估计

根据未来销售预测和现有库存情况，推算每个订货周期以及到货周期的库存情况，具体处理过程参照技术文档。

步骤四 门店下单量计算

根据未来预测的库存情况计算门店下单量，并按订货单位（PCB）转换，具体处理过程参照技术文档。

步骤五 大仓下单量计算

汇总门店下单量到华东区所有门店总下单量。

针对于越库单品，东区总下单量即为所有门店加总下单量。

对于存库单品，大仓库存水平在12-24天区间波动。

具体处理过程参照技术文档。

#### DM模型

DM需求计划模型主要分为5个步骤：DM期间销售预测总量按天分解，越库单品订货，存库单品A类促销进价（PPP）大仓囤货，大仓发送给门店存库单品75%量，存库单品剩余25%量遵循正常订货逻辑

步骤一DM期间销售总量按天分解

将DM期间销售预测总总量在门店层面按天分解。

步骤二 越库单品订货

针对于越库单品，东区总下单量即为所有门店加总销售总量。

步骤三 存库单品A类促销进价（PPP）大仓囤货

针对于存库单品B类，汇总东区门店所有销售总量即为大仓订货量

针对于存库单品A类，如果促销进价小于等与非促销进价的90%，计算DM档期结束后4周的正常门店销量，加总到此档DM销售量得到订货总量

步骤四 大仓发送给门店存库单品75%量

计算存库单品在DM开档前，大仓第一次推送给门店75%的量。

步骤五 存库单品剩余25%量遵循正常订货逻辑

存库单品剩余25%量遵循正常订货逻辑，DM开档后，有需求的门店订货。如果无需求，则转化成大仓库存。

# 系统设计与架构

## 系统概要

系统每次运行需要产生百万条以上的数据，逻辑较为复杂。系统需在每日8点30分前生成运算结果。数据池中数据最早在每日5点30分可以准备完毕。系统可用运行时间较少。鉴于以上两点和现有环境，系统基于Spark 批处理技术进行搭建。Spark由加州大学伯克利分校AMP实验室开发，是一个围绕速度、易用性和复杂分析构建的大数据处理框架。十分适合用来构建大型的、低延迟的数据应用程序。

系统包含以下几个组成

1. 数据载入部分

Spark 运行环境对Kudu表的支持并不理想。家福乐数据池中的Kudu表多以全量备份形式更新，难以获取过去某一时间点的数据状态。数据载入部分使用Impala运行环境获取数据池表当前快照，并写入需求计划自用表中，以便以后使用。

逻辑部分

此部分为系统核心。在数据载入后此部分运行订单逻辑，生成订单信息，生成需求计划。生成的信息会写入数据池表中。此部分有个分为两个子类，DM订单逻辑和日常订单逻辑。

DM订单逻辑较为简单，逻辑分支较少，使用Pyspark技术基于Spark运行环境开发。日常订单逻辑复杂，分支众多，使用Scala技术基于Spark运行环境开发。

1. 输出部分

此部分读取订单逻辑部分输出，将其按照规定的格式生成订单文件和需求计划文件，并放置于指定位置。此部分基于Python技术开发。使用openpyxl包以生成规定格式文件

1. 监控部分

此部分用以计算所需的可测量KPI。此部分使用Jupyter Notebook技术为基础进行开发。使用运行环境获取所需数据。

4. 运行部分

此部分用以实现系统的自动化运行。使用Airflow技术进行开发。

## 硬件架构

### Hadoop集群

配置：基于家乐福的集群配置，此项目一共使用了41TB的集群硬盘空间资源。

需安装工具：Hue, Hive, Impala

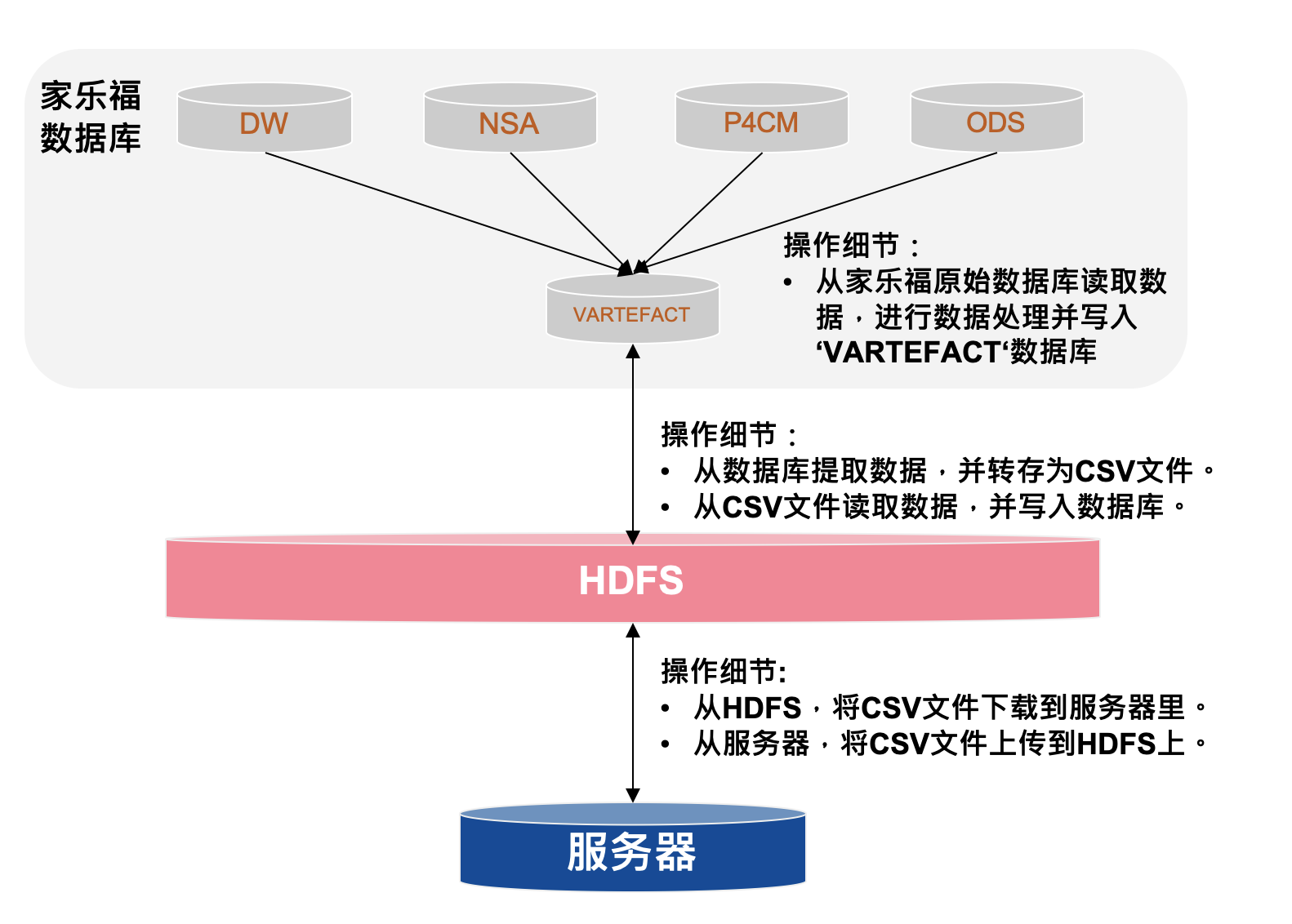
### 高性能的独立服务器

配置：32个 2.4GHz的CPU，256GB内存，1TB硬盘空间。

需安装工具：Jupyter Lab, Python3 (以及相关Python包), Spark, Scala, Airflow。

## 工作流程(图示)

## 数据流程(图示)



# 数据处理与特征工程

## 数据源

### 交易记录数据

dw.trxns\_sales\_daily\_kudu, ods.data8\_media

### 商品信息数据

p4md.mstms,

ods.p4md\_itmgld,

bidata.dim\_itemsubgl,

p4md.itmbar,

fds.p4cm\_item\_map

### 库存信息数据

fds.p4cm\_daily\_stock

### 订货及运输信息数据

lfms.daily\_shippment

### 门店信息数据

ods.p4md\_stogld,

ods.p4md\_stoatt

### DM信息数据

nsa.dm\_extract\_log,

ods.nsa\_dm\_theme

### Coupon 信息数据

nsa.coupon\_activity，

nsa.coupon\_activity\_dtl,

nsa.coupon\_dtl\_setdate,

ods.nsa\_coupon\_item\_detail,

nsa.coupon\_activity\_his,

nsa.coupon\_activity\_dtl\_his,

nsa.coupon\_dtl\_setdate\_his,

ods.nsa\_coupon\_item\_detail\_his；

### 门店商品清单信息数据

nsa.daily\_corresponding,

ods.p4cm\_store\_item

### 日期信息数据

ods.dim\_calendar

### 竞争对手信息数据

pricing.competitors\_dm\_list

### 外部数据

法定节假日数据：vartefact.public\_holidays

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | date\_key |  | string |  |  |  |  |
| 2 | holiday |  | string |  |  |  | 如果是法定假日标1，否则标0 |

中国农历节假日数据：vartefact.chinese\_festival

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | date\_key |  | string |  |  |  |  |
| 2 | festival\_type |  | string |  |  |  | 每一个节日有一个对应的数字，1为新年，2是中国新年，3是清明，劳动节4，端午节5，中秋节6，国庆节7 |

历史气温数据：vartefact.temperature\_info\_fc1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | city |  | string |  |  |  |  |
| 2 | city\_code |  | string |  |  |  |  |
| 3 | territory\_code |  | string |  |  |  |  |
| 4 | month |  | string |  |  |  |  |
| 5 | high\_temp |  | string |  |  |  | 过去三年该城市该月最高气温平均值 |
| 6 | low\_temp |  | string |  |  |  | 过去三年该城市该月最低气温平均值 |

### 门店单品信息 （forecast\_store\_item\_details）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | store\_code |  | string |  |  |  | 门店代码 |
| 2 | dept\_code |  | string |  |  |  | 部门代码 |
| 3 | item\_code |  | string |  |  |  | 商品一级代码 |
| 4 | sub\_code |  | string |  |  |  | 部门代码 |
| 5 | cn\_name |  | string |  |  |  |  |
| 6 | store\_status |  | string |  |  |  |  |
| 7 | dc\_supplier\_code |  | string |  |  |  |  |
| 8 | ds\_supplier\_code |  | string |  |  |  |  |
| 9 | repl\_type |  | string |  |  |  |  |
| 10 | order\_day |  | string |  |  |  |  |
| 11 | lead\_time |  | int |  |  |  |  |
| 12 | order\_by |  | string |  |  |  |  |
| 13 | qty\_per\_pack |  | int |  |  |  |  |
| 14 | pack\_per\_box |  | int |  |  |  |  |
| 15 | dc\_status |  | string |  |  |  |  |
| 16 | rotation |  | string |  |  |  |  |
| 17 | item\_type |  | string |  |  |  |  |
| 18 | remark |  | string |  |  |  |  |
| 19 | risk\_item\_unilever |  | string |  |  |  |  |
| 20 | qty\_per\_unit |  | int |  |  |  |  |
| 21 | con\_holding |  | string |  |  |  |  |

### 存库单品送货信息 (forecast\_onstock\_order\_delivery\_mapping)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | store\_code |  | string |  |  |  |  |
| 2 | dept\_code |  | string |  |  |  |  |
| 3 | rotation |  | string |  |  |  |  |
| 4 | lead\_time |  | string |  |  |  |  |
| 5 | order\_days |  | string |  |  |  |  |
| 6 | order\_iso\_weekday | | string |  |  |  |  |
| 7 | order\_weekday\_short | | string |  |  |  |  |
| 8 | delivery\_iso\_weekday | | string |  |  |  |  |
| 9 | week\_shift |  | int |  |  |  |  |

### 越库单品送货信息 （forecast\_xdock\_order\_mapping）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | store\_code |  | string |  |  |  |  |
| 2 | dept\_code |  | string |  |  |  |  |
| 3 | item\_code |  | string |  |  |  |  |
| 4 | sub\_code |  | string |  |  |  |  |
| 5 | lead\_time |  | string |  |  |  |  |
| 6 | dc\_to\_store\_time | | int |  |  |  |  |
| 7 | order\_days |  | string |  |  |  |  |
| 8 | order\_iso\_weekday | | string |  |  |  |  |
| 9 | order\_weekday\_short | | string |  |  |  |  |

### 大仓单品信息 （forecast\_store\_item\_details）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | dc |  | string |  |  |  |  |
| 2 | dc\_site |  | string |  |  |  |  |
| 3 | full\_item\_code |  | string |  |  |  |  |
| 4 | dc\_status |  | string |  |  |  |  |
| 5 | item\_name\_english |  | string |  |  |  |  |
| 6 | item\_name\_local |  | string |  |  |  |  |
| 7 | current\_warehouse |  | string |  |  |  |  |
| 8 | primary\_ds\_supplier |  | string |  |  |  |  |
| 9 | primary\_ds\_supplier\_name |  | string |  |  |  |  |
| 10 | qty\_per\_box |  | int |  |  |  |  |
| 11 | primary\_barcode |  | string |  |  |  |  |
| 12 | rotation |  | string |  |  |  |  |
| 13 | box\_per\_layer\_ti |  | int |  |  |  |  |
| 14 | layer\_per\_pallet\_hi |  | int |  |  |  |  |
| 15 | stop\_start\_date |  | string |  |  |  |  |
| 16 | stop\_reason |  | string |  |  |  |  |
| 17 | qty\_per\_pack |  | int |  |  |  |  |
| 18 | pack\_per\_box |  | int |  |  |  |  |
| 19 | holding\_supplier\_code |  | string |  |  |  |  |
| 20 | holding\_code |  | string |  |  |  |  |
| 21 | risk\_item\_unilever |  | string |  |  |  |  |
| 22 | order\_uint |  | string |  |  |  |  |
| 23 | seasonal |  | string |  |  |  |  |
| 24 | item\_type |  | string |  |  |  |  |
| 25 | dept\_code |  | string |  |  |  |  |
| 26 | item\_code |  | string |  |  |  |  |
| 27 | sub\_code |  | string |  |  |  |  |
| 28 | qty\_per\_unit |  | int |  |  |  |  |

### 系统日 （forecast\_calendar）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | week\_index |  | int |  |  |  |  |
| 2 | date\_key |  | string |  |  |  |  |
| 3 | iso\_day |  | string |  |  |  |  |
| 4 | weekday |  | string |  |  |  |  |
| 5 | iso\_weekday | | string |  |  |  |  |
| 6 | weekday\_short | | string |  |  |  |  |
| 7 | weekday\_long | | string |  |  |  |  |

### 大仓订货日 （forecast\_dc\_order\_delivery\_mapping）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | con\_holding |  | string |  |  |  |  |
| 2 | holding\_name | | string |  |  |  |  |
| 3 | order\_date |  | string |  |  |  |  |
| 4 | order\_weekday | | string |  |  |  |  |
| 5 | delivery\_date | | string |  |  |  |  |
| 6 | delivery\_weekday | | string |  |  |  |  |
| 7 | week\_shift |  | bigint |  |  |  |  |
| 8 | risk\_item\_unilever | | string |  |  |  |  |

### 门店送货到店时间 （forecast\_stores\_delv\_time）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | store\_code |  | string |  |  |  |  |
| 2 | delivery\_time | | string |  |  |  |  |

### 门店可订货单品信息（v\_forecast\_inscope\_store\_item\_details）

### 大仓可订货单品信息（v\_forecast\_inscope\_dc\_item\_details）

### 当日应在门店下订单的存库单品信息（v\_forecast\_daily\_onstock\_order\_items）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | dept\_code |  | string |  |  |  |  |
| 2 | item\_code |  | string |  |  |  |  |
| 3 | sub\_code |  | string |  |  |  |  |
| 4 | cn\_name |  | string |  |  |  |  |
| 5 | con\_holding |  | string |  |  |  |  |
| 6 | store\_code |  | string |  |  |  |  |
| 7 | rotation |  | string |  |  |  |  |
| 8 | ds\_supplier\_code | | string |  |  |  |  |
| 9 | dc\_supplier\_code | | string |  |  |  |  |
| 10 | order\_day |  | string |  |  |  |  |
| 11 | item\_stop\_start\_date | | string |  |  |  |  |
| 12 | item\_stop\_end\_date | | string |  |  |  |  |

### 当日应在门店下订单的越库单品信息（v\_forecast\_daily\_xdock\_order\_items）

## 数据处理目标以及方法

### 区分DM销售以及非DM销售

根据nsa.dm\_extract\_log中extract\_order = max(40，50)的DM计划: 如果一个单品在DM的范围内（DM城市范围内，DM开始和结束时间内），就会被认为是DM销售。

### 剔除有计划的BP(基于销售小票)

1. 将不同的购物小票进行分组：ticket\_id相差小于等于5的小票会被标记成同组。
2. 计算单品销售量的90分位数(无折扣和有折扣期间分别计算)：

* 90分位数：作为次数分布中的一个点，一组数列按从小到大排列，分为100个单位，90百分位数表示在此次数分布中小于该特定个案的个案数量占总数量的90%
* 无折扣期间：单品不在DM上，会员折扣为0， 促销折扣为0，优惠券折扣为0）：根据每个单品自从2017-01-01至今的无折扣销售记录以小时为单位分组计算其90分位数 — P90
* 有折扣期间, 即非无折扣期间: 根据每个单品自从2017-01-01至今的有折扣销售记录以小时为单位分组，将属于同一个小分类的小组合并计算其90分位数 — P90

为防止由于销售记录过少导致的P90过大，如果P90大于50将会别50替代。

1. 根据第一步中对于每张小票的分组，将第二步中的P90对应到每组小票：

为了防止直接使用P90设置阈值（不大于50）过于激进：

对于折扣期间：选取200 \* log( P90 + 1) 作为阈值

对于非折扣期间：选取 175 \* log( P90 + 1) 作为阈值

1. 选取有计划的BP的候选销售记录：

* 销售数量大于第三部中定义的阈值或者大于250
* 或者被标记使用支票支付

1. 将第四步中标记的候选销售记录与BP订货记录相匹配

从lfms.daily\_shipment中选取有计划的BP订单到货记录，与到货日前后35天且被标记为候选的销售记录数量相对应。能够对应出的或者单张小票销售同一单品超过250个的会被标记为有计划的BP销售。

### 训练数据出现缺货时的日销量

* + - 1. 无折扣期间：过去90天内相同星期数且无折扣销售期间的日销售量的中位数，如果不存在用0补充。
      2. 有折扣期间：过去90天内相同相同星期数且有折扣销售期间的日销售量的中位数，如果不存在用0补充。如果一共出现小于5次，则选取无折扣期间的中位数代替。
      3. 根据第一步和第二步，每家门店每个单品在每一个缺货的日期都会对应一个历史日销量中位数，选取这个中位数和当天真实日销量中的较大值作为缺货日的真实日销量。

### 周销量预测到天销量预测的转化

1. 对于每个单品在每家店选取没有法定节日或者DM的完整周，计算每个星期数的销售量占总销售量的比值。
2. 对于每一个周期性的对于销售有明显影响的活动，计算一个影响因子。这些影响因子会被叠加到第一步中计算的每个星期数的比值上，从而得到加成过的比值。得到加成过的比值之后对于每个星期，进行变基以保证一个星期内的比值的和为100%。
3. 将变基过的比值应用于每个周销量预测上得到最后的天销量预测。如果某个单品过去没有过销售记录，则选取平均分配的方式对周销量进行划分，用于之后的需求计划模型。

### DM销售预测到天销售预测的变化

1. 根据过去一年出现的所有DM的特征（DM持续时长，DM开始的星期数，DM结束的星期数）, 分析得出最常出现的DM形式， 并为不同的DM形式编号。对于每一个DM形式，在小分类和门店力度计算每天的销量占总销量的比值。
2. 将第一步中计算所得比值应用到DM销量预测上得到天销量预测。如果某一个小分类在某一家门店没有历史销量，则选取平均分配的方式对DM预测销量进行划分，用于之后的需求计划模型。

## 数据处理流程

先执行普通模型数据处理，后执行DM模型数据处理。

### 普通模型数据处理（由上至下顺序执行）

|  |  |  |
| --- | --- | --- |
| 表分类 | 表名 | 解释 |
| 模型适用范围 | vartefact.forecast\_store\_code\_scope\_sprint4 | 模型适用的门店范围 |
| vartefact.forecast\_itemid\_list\_threebrands\_sprint4 | 模型适用的单品的范围 |
| 单品信息数据 | vartefact.forecast\_item\_id\_family\_codes\_sprint4 | 单品描述信息 |
| vartefact.lastest\_active\_status | 单品active情况 |
| DM信息数据 | vartefact.forecast\_dm\_plans\_sprint4 | DM单品列表计划（全部） |
| vartefact.forecast\_add\_future\_dms\_sprint4 | 未来DM单品计划 |
| vartefact.forecast\_next\_dm\_sprint4 | 下一次DM单品计划 |
| 交易信息数据 | vartefact.forecast\_trxn\_v7\_sprint4 | 单品交易数据及当前和下一档DM数据 |
| vartefact.forecast\_trxn\_v7\_full\_item\_id\_sprint4 | 加入属于模型范围但不存在销售记录的单品 |
| 识别BP | vartefact.forecast\_trxn\_v7\_full\_item\_id\_sprint4\_group\_id\_new | 小票重新分组信息 |
| vartefact.art\_filter\_non\_promo | 非促销期间销售量的九十分位数信息 |
| vartefact.art\_filter\_promo | 促销期间销售量的九十分位数信息 |
| vartefact.grouped\_to\_be\_shipment | 标记有可能为BP的候选销售交易记录 |
| vartefact.p4cm\_item\_map\_complete | 单品id和code的对应 |
| vartefact.shipment\_scope\_map\_corrected | BP订单的到货信息 |
| vartefact.grouped\_to\_be\_shipment\_groupped | BP订单与实际销售交易记录的对应 |
| vartefact.forecast\_item\_store\_perc\_flagged | 被标记BP数量与BP到货数量的比例 |
| vartefact.forecast\_trxn\_flag\_v1\_sprint4 | 清除计划中的BP之后的交易记录 |
| 日销售记录 | vartefact.forecast\_sprint4\_full\_date\_daily\_sales | 日销售量信息 |
| vartefact.forecast\_sprint4\_out\_of\_stock\_median | 通过历史销售记录修正出现断货时的销量信息 |
| vartefact.forecast\_sprint4\_trxn\_to\_day | 经过调整的日销售信息 |
| vartefact.forecast\_sprint4\_daily\_future\_dms | 未来所有DM的信息 |
| vartefact.forecast\_sprint4\_daily\_next\_dm | 选择下一个DM的信息 |
| vartefact.forecast\_sprint4\_add\_dm\_to\_daily | 加入DM信息的日销售信息 |
| 周销售记录 | vartefact.forecast\_sprint4\_day\_to\_week | 周销售记录信息 |
| vartefact.forecast\_sprint4\_day\_to\_week\_test | 周销售记录信息（加入在范围中但没有销售记录的单品） |
| vartefact.forecast\_sprint2\_festival\_ticket\_count\_flag | 每个单品在每个中国节日的交易笔数 |
| vartefact.forecast\_sprint2\_trxn\_week\_features\_flag\_sprint4 | 加入历史销售信息特征的周销售记录 |
| vartefact.forecast\_sprint2\_final\_flag\_sprint4 | 加入单品信息的周销量记录 |
| 优惠券信息 | vartefact.forecast\_assortment\_full | 单品分类信息 |
| vartefact.coupon\_mapping | 优惠券信息 |
| vartefact.forecast\_sprint3\_v3\_flag\_sprint4 | 周力度下的单品分类信息 |
| vartefact.forecast\_sprint3\_coupon\_item\_link\_flag\_sprint4 | 周力度下的优惠券信息 |
| vartefact.coupon\_city\_store\_union\_flag\_sprint4 | 优惠券信息重构 |
| vartefact.forecast\_sprint3\_v5\_flag\_sprint4 | 加入优惠券信息的单品周销售记录 |
| 周销售记录加入历史交易信息 | vartefact.forecast\_sprint3\_v6\_flag\_sprint4 | 加入小分类单品历史销售记录信息 |
| vartefact.forecast\_sprint3\_v9\_flag\_sprint4 | 加入上一年单品历史销售记录信息 |
| vartefact.forecast\_out\_of\_stock\_temp | 周力度下的断货标记 |
| vartefact.forecast\_sprint3\_v10\_flag\_sprint4 | 加入周力度下断货标记的周销售记录 |

### DM模型数据处理（由上至下顺序执行）

|  |  |  |
| --- | --- | --- |
| 表分类 | 表名 | 解释 |
| DM聚合表 | vartefact.forecast\_sprint4\_dm\_agg\_v2 | 将销售记录按DM聚合 |
| vartefact.forecast\_sprint4\_promo\_past\_features | 加入来自历史DM的特征 |
| 节日特征 | vartefact.chinese\_festival vartefact.forecast\_sprint4\_festival\_lunar\_feat | 来自外部文件 加入与农历节假日相关的特征 |
| DM信息提取 | vartefact.dm\_mapping\_1719\_dates\_last\_version vartefact.last\_year\_dm\_sales\_vrai\_exact | 来自家乐福提供外部文件 计算去年同期DM的销售特征 |
| vartefact.promo\_dataset\_feat\_combine\_exact\_vrai | 加入去年同期DM的销售特征 |
| vartefact.uplift\_promo | 加入计算销量提升的日期限制 |
| vartefact.forecast\_sprint4\_promo\_uplift | 加入DM带来的销量提升 |
| vartefact.forecast\_sprint4\_promo\_with\_baseline | 加入来自历史销售记录的特征 |
| vartefact.forecast\_sprint4\_promo\_with\_coupon | 加入来自优惠券的折扣信息 |
| vartefact.forecast\_sprint4\_promo\_mecha\_v4 | 加入来自DM中slot\_type的折扣信息 |

## 输出表说明

### 模型特征信息表

#### 普通模型vartefact.forecast\_sprint3\_v10\_flag\_sprint4

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段类型 | 字段中文名称 |
| 1 | active\_flag | tinyint | 单品是否active |
| 2 | barcode | string | 条形码 |
| 3 | city\_code | string | 城市代码 |
| 4 | coupon\_disc\_max | decimal(15,2) | 周最大折扣 |
| 5 | coupon\_disc\_mean | decimal(19,6) | 周平均折扣 |
| 6 | coupon\_disc\_min | decimal(15,2) | 周最小折扣 |
| 7 | coupon\_disc\_sum | decimal(38,2) | 周总折扣 |
| 8 | curr\_psp\_days | int | 当前促销价格持续时长 |
| 9 | curr\_psp\_end\_dow | int | 当前促销价格结束日（周几） |
| 10 | curr\_psp\_end\_week\_count | bigint | 距离当前DM结束还有多少周 |
| 11 | curr\_psp\_start\_dow | int | 当前促销价格开始日（周几） |
| 12 | curr\_psp\_start\_week\_count | bigint | 距离当前DM开始有多少周 |
| 13 | current\_dm\_end\_date | string | 当前DM结束日期 |
| 14 | current\_dm\_msp | decimal(12,1) | 当前DM会员售价 |
| 15 | current\_dm\_nsp | decimal(15,4) | 当前DM的名义销售价格 |
| 16 | current\_dm\_page\_no | int | 单品在当前DM的页数 |
| 17 | current\_dm\_page\_strategy | string | 单品在当前DM的page \_strategy |
| 18 | current\_dm\_psp | decimal(15,4) | 单品在当前DM的促销价格 |
| 19 | current\_dm\_psp\_end\_date | string | 单品在当前DM的促销价格结束时间 |
| 20 | current\_dm\_psp\_start\_date | string | 单品在当前DM的促销价格开始时间 |
| 21 | current\_dm\_slot\_type\_code | string | 单品在当前DM的slot\_type\_code |
| 22 | current\_dm\_theme\_id | int | 单品当前所在的DM的ID |
| 23 | current\_prom\_type | int | 单品当前所在的DM的prom\_type |
| 24 | current\_theme\_end\_date | string | 单品当前所在的DM的结束时间 |
| 25 | current\_theme\_start\_date | string | 单品当前所在的DM的开始时间 |
| 26 | disc\_ratio | double | 单品在促销期间出售的周数与所有有销售的周数之比 |
| 27 | ext\_amt\_sum | decimal(38,2) | 名义销售额的总和 |
| 28 | fam\_sales\_qty\_max\_roll | decimal(38,3) | 同分类历史最高周销量 |
| 29 | fam\_sales\_qty\_mean\_12w | decimal(38,6) | 历史12周的移动平均周销量 |
| 30 | fam\_sales\_qty\_mean\_24w | decimal(38,6) | 历史24周的移动平均周销量 |
| 31 | fam\_sales\_qty\_mean\_4w | decimal(38,6) | 历史4周的移动平均周销量 |
| 32 | fam\_sales\_qty\_mean\_52w | decimal(38,6) | 历史52周的移动平均周销量 |
| 33 | fam\_sales\_qty\_min\_roll | decimal(38,3) | 同分类历史最低周销量 |
| 34 | family\_code | string | 分类代码 |
| 35 | family\_edesc | string | 分类英文描述 |
| 36 | family\_ldesc | string | 分类中文描述 |
| 37 | festival\_type | int | 节日类型 |
| 38 | group\_family\_code | string | 大分类代码 |
| 39 | group\_family\_edesc | string | 大分类英文描述 |
| 40 | high\_temp\_month | float | 历史三年当月平均最高气温 |
| 41 | holiday\_count\_new | bigint | 当周包含的公共假日天数 |
| 42 | ind\_competitor | tinyint | 同城是否有其他竞争对手销售该单品 |
| 43 | ind\_current\_on\_dm\_flag | tinyint | 该单品当前是否在DM出现 |
| 44 | item\_id | int | item\_id |
| 45 | item\_seasonal\_code | string | 单品季节性代码 |
| 46 | item\_store\_mean\_12points | decimal(38,6) | 单品同门店历史12周的移动平均周销量 |
| 47 | item\_store\_mean\_24points | decimal(38,6) | 单品同门店历史24周的移动平均周销量 |
| 48 | item\_store\_mean\_4points | decimal(38,6) | 单品同门店历史4周的移动平均周销量 |
| 49 | item\_store\_mean\_52points | decimal(38,6) | 单品同门店历史52周的移动平均周销量 |
| 50 | itmcapa | string | 单品单位含量，例如100g |
| 51 | itmcaput | string | 单品单位，例如g |
| 52 | itmedesc | string | 单品英文描述 |
| 53 | itmldesc | string | 单品中文描述 |
| 54 | itmpack | decimal(6,2) | 单品中包装包含个数 |
| 55 | itmstkun | string | 中文单位 |
| 56 | last\_year\_festival\_ticket\_count | bigint | 去年同节日期间销售小票数 |
| 57 | last\_year\_sales\_qty | decimal(38,3) | 去年同期销售量 |
| 58 | low\_temp\_month | float | 历史三年当月平均最低气温 |
| 59 | mpsp\_disc\_sum | decimal(38,2) | 会员折扣总和 |
| 60 | next\_dm\_days | int | 下一次DM持续的时长（天数） |
| 61 | next\_dm\_end\_date | string | 下一次DM结束的时间 |
| 62 | next\_dm\_msp | decimal(12,1) | 下一次DM的会员销售价格 |
| 63 | next\_dm\_msp\_end\_date | string | 下一次DM会员销售价格结束时间 |
| 64 | next\_dm\_msp\_start\_date | string | 下一次DM会员销售价格开始时间 |
| 65 | next\_dm\_nl | string | 下一次DM是全国范围还是地区范围 |
| 66 | next\_dm\_nsp | decimal(15,4) | 下一次DM的名义销售价格 |
| 67 | next\_dm\_page\_no | int | 该单品在下一次DM的页数 |
| 68 | next\_dm\_page\_strategy | string | 该单品在下一次DM的page\_strategy |
| 69 | next\_dm\_page\_strategy\_name | string | 该单品在下一次DM的page\_strategy\_name |
| 70 | next\_dm\_prom\_type | int | 该单品在下一次DM的prom\_type |
| 71 | next\_dm\_psp | decimal(15,4) | 该单品在下一次DM的促销价格 |
| 72 | next\_dm\_slot\_type\_code | string | 该单品在下一次DM的slot\_type\_code |
| 73 | next\_dm\_slot\_type\_name | string | 该单品在下一次DM的slot\_type\_code\_name |
| 74 | next\_dm\_start\_date | string | 下一次DM的开始时间 |
| 75 | next\_dm\_start\_week\_count | bigint | 距离下一次DM的周数 |
| 76 | next\_dm\_theme\_id | int | 下一次DM的ID |
| 77 | next\_dm\_v5\_extract\_datetime | string | 下一次DM第五版计划上传时间 |
| 78 | next\_dm\_w1\_sales\_qty | decimal(38,3) | 下一次DM期间的销售量,不会被用于模型训练 |
| 79 | next\_dm\_w2\_sales\_qty | decimal(38,3) | 下一次DM开始一周之后的周销售量，不会被用于模型训练 |
| 80 | out\_stock\_flag | int | 是否出现脱销 |
| 81 | promo\_disc\_sum | decimal(38,2) | 促销折扣总和 |
| 82 | psp\_nsp\_ratio | decimal(35,20) | 当前DM促销价格与名义销售价格的比值 |
| 83 | sales\_amt\_sum | decimal(38,2) | 周总销售额 |
| 84 | sales\_qty\_fam\_store\_wgt | decimal(38,6) | 该单品在此门店当年销量占该分类当年在此门店销量的比例 |
| 85 | sales\_qty\_fam\_wgt | decimal(38,6) | 该单品当年销量占该分类当年销量的比例 |
| 86 | sales\_qty\_sum | decimal(38,3) | 销量之和 |
| 87 | store\_code | string | 门店代码 |
| 88 | sub\_code | string | sub\_code |
| 89 | sub\_family\_code | string | 小分类代码 |
| 90 | sub\_family\_edesc | string | 小分类英文描述 |
| 91 | sub\_family\_ldesc | string | 小分类中文描述 |
| 92 | sub\_id | int | sub\_id |
| 93 | subfam\_sales\_qty\_mean\_12w | decimal(38,6) | 所属小分类过去12周的移动平均销量 |
| 94 | subfam\_sales\_qty\_mean\_24w | decimal(38,6) | 所属小分类过去24周的移动平均销量 |
| 95 | subfam\_sales\_qty\_mean\_4w | decimal(38,6) | 所属小分类过去4周的移动平均销量 |
| 96 | subfam\_sales\_qty\_mean\_52w | decimal(38,6) | 所属小分类过去52周的移动平均销量 |
| 97 | three\_brand\_item\_list\_holding\_code | string | Holding code |
| 98 | three\_brand\_item\_list\_id | int | item\_id |
| 99 | trxn\_month | string | 交易月份 |
| 100 | trxn\_year | string | 交易年份 |
| 101 | week\_begin\_date | string | 周开始时间 |
| 102 | week\_end\_date | string | 周结束时间 |
| 103 | week\_key | string | 交易week\_key |
| 104 | assortment\_avg\_nsp | decimal(26,13) | 来自单品分类表主表的周平均名义销售价格 |
| 105 | assortment\_active\_flag | tinyint | 来自单品分类表主表的active状态 |
| 106 | coupon\_activity\_type\_id\_full | string | 单品在该周适用的所有的优惠券的type\_id |
| 107 | coupon\_typecode\_full | string | 单品在该周适用的所有的优惠券的typecode |
| 108 | coupon\_count1\_full | string | 单品在该周适用的所有的优惠券的count1 |
| 109 | ndv\_coupon\_activity\_type\_id | bigint | 单品在该周适用的所有的优惠券的type\_id的种类数 |
| 110 | ndv\_coupon\_typecode | bigint | 单品在该周适用的所有的优惠券的typecode的种类数 |
| 111 | ndv\_coupon\_count1 | bigint | 单品在该周适用的所有的优惠券的count1的种类数 |
| 112 | ind\_coupon\_typeid\_dd | tinyint | 对该单品是否存在type\_id为DD种类的优惠券 |
| 113 | ind\_coupon\_typeid\_np | tinyint | 对该单品是否存在type\_id为NP种类的优惠券 |
| 114 | ind\_coupon\_typeid\_ac | tinyint | 对该单品是否存在type\_id为AC种类的优惠券 |
| 115 | ind\_coupon\_typeid\_cc | tinyint | 对该单品是否存在type\_id为CC种类的优惠券 |
| 116 | ind\_coupon\_typecode\_cp | tinyint | 对该单品是否存在type\_code为CP种类的优惠券 |
| 117 | ind\_coupon\_typecode\_mpm | tinyint | 对该单品是否存在type\_code为MPM种类的优惠券 |
| 118 | ind\_coupon\_typecode\_mp | tinyint | 对该单品是否存在type\_code为MP种类的优惠券 |
| 119 | ind\_coupon\_typecode\_other | tinyint | 对该单品是否存在type\_code为CP，MPM或者MP外的种类的优惠券 |
| 120 | ind\_coupon\_count1\_27 | tinyint | 对该单品是否存在count\_1为27种类的优惠券 |
| 121 | ind\_coupon\_count1\_02\_25 | tinyint | 对该单品是否存在count\_1为02\_25种类的优惠券 |
| 122 | ind\_coupon\_count1\_60 | tinyint | 对该单品是否存在count\_1为60种类的优惠券 |
| 123 | ind\_coupon\_count1\_4 | tinyint | 对该单品是否存在count\_1为4种类的优惠券 |
| 124 | ind\_coupon\_count1\_1 | tinyint | 对该单品是否存在count\_1为1种类的优惠券 |
| 125 | ind\_coupon\_count1\_58 | tinyint | 对该单品是否存在count\_1为58种类的优惠券 |
| 126 | ind\_coupon\_count1\_11 | tinyint | 对该单品是否存在count\_1为11种类的优惠券 |
| 127 | ind\_coupon\_count1\_16\_34 | tinyint | 对该单品是否存在count\_1为16\_34种类的优惠券 |
| 128 | ind\_coupon\_count1\_56 | tinyint | 对该单品是否存在count\_1为56种类的优惠券 |
| 129 | ind\_coupon\_count1\_37 | tinyint | 对该单品是否存在count\_1为37种类的优惠券 |
| 130 | ind\_coupon\_count1\_44 | tinyint | 对该单品是否存在count\_1为44种类的优惠券 |
| 131 | ind\_coupon\_count1\_9 | tinyint | 对该单品是否存在count\_1为9种类的优惠券 |
| 132 | ind\_coupon\_count1\_10 | tinyint | 对该单品是否存在count\_1为10种类的优惠券 |
| 133 | ind\_coupon\_count1\_other | tinyint | 对该单品是否存在count\_1为其他种类的优惠券 |
| 134 | fam\_sales\_item\_ratio | decimal(38,6) | 该分类在该门店当周销量与交易笔数的比值 |
| 135 | fam\_sales\_item\_ratio\_4w | decimal(38,6) | 历史4周该分类移动平均周销量与交易笔数的比值 |
| 136 | fam\_sales\_item\_ratio\_12w | decimal(38,6) | 历史12周该分类移动平均周销量与交易笔数的比值 |
| 137 | fam\_sales\_item\_ratio\_24w | decimal(38,6) | 历史24周该分类移动平均周销量与交易笔数的比值 |
| 138 | fam\_sales\_item\_ratio\_52w | decimal(38,6) | 历史52周该分类移动平均周销量与交易笔数的比值 |
| 139 | subfam\_sales\_item\_ratio | decimal(38,6) | 该小分类在该门店当周销量与交易笔数的比值 |
| 140 | subfam\_sales\_item\_ratio\_4w | decimal(38,6) | 历史4周该小分类移动平均周销量与交易笔数的比值 |
| 141 | subfam\_sales\_item\_ratio\_12w | decimal(38,6) | 历史12周该小分类移动平均周销量与交易笔数的比值 |
| 142 | subfam\_sales\_item\_ratio\_24w | decimal(38,6) | 历史24周该小分类移动平均周销量与交易笔数的比值 |
| 143 | subfam\_sales\_item\_ratio\_52w | decimal(38,6) | 历史52周该小分类移动平均周销量与交易笔数的比值 |
| 144 | last\_year\_week\_key | string | 上一年对应的week\_key |
| 145 | last\_year\_week\_key\_m1bef | string | 上一年对应时间向前一个月对应的week\_key |
| 146 | last\_year\_week\_key\_m1aft | string | 上一年对应时间向后一个月对应的week\_key |
| 147 | week\_key\_before | string | 上一年对应的week\_key用于选取早于对应周的销售量 |
| 148 | last\_year\_sales\_qty\_bef | decimal(38,3) | 上一年早于对应周且距离对应周最近的周销售量 |
| 149 | rownum | bigint | 选取上一年早于对应周且距离对应周最近的周销售量的条件 |
| 150 | week\_key\_after | string | 上一年对应的week\_key用于选取晚于对应周的销售量 |
| 151 | last\_year\_sales\_qty\_aft | decimal(38,3) | 上一年晚于对应周且距离对应周最近的周销售量 |
| 152 | rownum\_bis | bigint | 选取上一年晚于对应周且距离对应周最近的周销售量的条件 |
| 153 | last\_year\_sales\_qty\_new | decimal(38,3) | 上一年对应周的近似周销售量 |
| 154 | ind\_out\_of\_stock | tinyint | 该周中是否出现了脱销的情况 |

#### DM模型 vartefact.forecast\_sprint4\_promo\_mecha\_v4

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段类型 | 字段中文名称 |
| 1 | item\_id | int | 单品ID |
| 2 | sub\_id | int | 单品sub\_id |
| 3 | store\_code | string | 门店代码 |
| 4 | current\_dm\_theme\_id | int | 当前所在DM的代码 |
| 5 | sub\_code | string | 单品sub\_code |
| 6 | city\_code | string | 城市代码 |
| 7 | dm\_sales\_qty | decimal(38,3) | DM期间的销量 |
| 8 | out\_of\_stock\_flag | tinyint | 是否出现了脱销 |
| 9 | nl | string | DM全国范围还是地区范围 |
| 10 | current\_dm\_theme\_en\_desc | string | 当前DM的主题（英文） |
| 11 | current\_theme\_start\_date | string | 当前DM开始的时间 |
| 12 | current\_theme\_end\_date | string | 当前DM结束的时间 |
| 13 | current\_dm\_psp | decimal(14,2) | 当前DM下的促销价格 |
| 14 | current\_dm\_nsp | decimal(14,2) | 当前DM下的名义销售价格 |
| 15 | current\_dm\_psp\_nsp\_ratio | decimal(19,3) | 当前DM下促销价格和名义销售价格的比值 |
| 16 | current\_dm\_psp\_start\_date | string | 当前DM下促销价格生效起始时间 |
| 17 | current\_dm\_psp\_end\_date | string | 当前DM下促销价格生效结束时间 |
| 18 | current\_dm\_page\_strategy\_code | string | 当前DM下单品所在页的策略代码 |
| 19 | current\_dm\_slot\_type\_code | string | 当前DM下单品所在位置代码 |
| 20 | current\_dm\_slot\_type\_name | string | 当前DM下单品所在位置名称 |
| 21 | current\_dm\_page\_no | int | 当前DM下单品所在页数 |
| 22 | item\_seasonal\_code | string | 单品季节性代码 |
| 23 | sub\_family\_code | string | 单品小分类代码 |
| 24 | family\_code | string | 单品分类代码 |
| 25 | psp\_start\_week | string | 促销价格开始生效的周代码 |
| 26 | psp\_start\_month | string | 促销价格开始生效的月代码 |
| 27 | psp\_end\_week | string | 促销价格结束的周代码 |
| 28 | last\_year\_dm\_sales | decimal(38,3) | 上一年同月份DM的销售量 |
| 29 | last\_year\_dm\_psp\_nsp\_ratio | decimal(19,3) | 上一年同月份DM的促销价与名义售价的比值 |
| 30 | last\_year\_fam\_dm\_sales\_avg | decimal(38,6) | 上一年同月份DM整个分类下的单品的平均销量 |
| 31 | last\_5dm\_sales\_avg | decimal(38,6) | 过去五个DM期间的平均销量 |
| 32 | fam\_last\_5dm\_sales\_avg | decimal(38,6) | 过去五个DM期间整个分类的平均销量 |
| 33 | festival\_type | string | 标记节日类别 |
| 34 | lunar\_last\_year\_week\_key | string | 标记农历上一年对应周代码 |
| 35 | lunar\_2w\_bef | string | 标记农历上一年对应周两周前的周代码 |
| 36 | lunar\_2w\_aft | string | 标记农历上一年对应周两周后的周代码 |
| 37 | lunar\_6w\_bef | string | 标记农历上一年对应周六周前的周代码 |
| 38 | lunar\_6w\_aft | string | 标记农历上一年对应周六周后的周代码 |
| 39 | last\_year\_lunar\_sales\_qty\_1m\_avg | decimal(38,6) | 农历上一年对应前后两周的平均销量 |
| 40 | last\_year\_lunar\_sales\_qty\_1m\_sum | decimal(38,3) | 农历上一年对应前后两周的销量之和 |
| 41 | last\_year\_lunar\_ratio\_1m | decimal(22,6) | 农历上一年对应前后两周的促销价与名义售价比值的平均值 |
| 42 | last\_year\_lunar\_sales\_qty\_3m\_sum | decimal(38,3) | 农历上一年对应前后六周的销量之和 |
| 43 | last\_year\_lunar\_sales\_qty\_3m\_avg | decimal(38,6) | 农历上一年对应前后六周的平均销量 |
| 44 | last\_year\_lunar\_ratio\_3m | decimal(22,6) | 农历上一年对应前后六周的促销价与名义售价比值的平均值 |
| 45 | last\_year\_dm\_sales\_vrai\_exact | decimal(38,3) | 上一年对应DM期间的销量（根据家乐福DM对应表） |
| 46 | vrai\_exact\_or\_lunar\_1m | decimal(38,6) | 上一年对应DM期间的销量，若为空则选取农历上一年对应前后两周的平均销量 |
| 47 | vrai\_exact\_or\_lunar\_3m | decimal(38,6) | 上一年对应DM期间的销量，若为空则选取农历上一年对应前后六周的平均销量 |
| 48 | three\_brand\_item\_list\_holding\_code | string | 供应商holding代码 |
| 49 | uplift | decimal(38,6) | 上一年DM期间日销量与DM开始之前两周日销量的比值 |
| 50 | 4w\_sales\_4w\_bef | decimal(38,3) | 上一年对应DM开始之前八周到前四周的总销量 |
| 51 | coupon\_disc\_ratio\_avg\_max | decimal(38,6) | 优惠券count1平均折扣的最大值 |
| 52 | coup\_disc\_ratio\_mech\_max | decimal(33,18) | DM折扣slot类型平均折扣最大的折扣率 |

### 需求计划信息表

#### 存库单品门店订单 （forecast\_onstock\_orders）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | item\_id |  | int |  |  |  |  |
| 2 | sub\_id |  | int |  |  |  |  |
| 3 | dept\_code |  | string |  |  |  |  |
| 4 | item\_code |  | string |  |  |  |  |
| 5 | sub\_code |  | string |  |  |  |  |
| 6 | con\_holding |  | string |  |  |  |  |
| 7 | store\_code |  | string |  |  |  |  |
| 8 | supplier\_code | | string |  |  |  |  |
| 9 | delivery\_day | | string |  |  |  |  |
| 10 | minimum\_stock\_required | | double |  |  |  |  |
| 11 | order\_qty |  | int |  |  |  |  |
| 12 | order\_without\_pcb | | double |  |  |  |  |
| 13 | order\_day |  | string |  |  |  |  |

#### 存库单品大仓订单（forecast\_dc\_orders）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | item\_id |  | int |  |  |  |  |
| 2 | sub\_id |  | int |  |  |  |  |
| 3 | dept\_code |  | string |  |  |  |  |
| 4 | item\_code |  | string |  |  |  |  |
| 5 | sub\_code |  | string |  |  |  |  |
| 6 | con\_holding |  | string |  |  |  |  |
| 7 | supplier\_code | | string |  |  |  |  |
| 8 | delivery\_day | | string |  |  |  |  |
| 9 | average\_sales | | double |  |  |  |  |
| 10 | order\_qty |  | int |  |  |  |  |
| 11 | order\_without\_pcb | | double |  |  |  |  |
| 12 | order\_day |  | string |  |  |  |  |

#### DM单品门店订单（forecast\_dm\_orders）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | item\_id |  | int |  |  |  |  |
| 2 | sub\_id |  | int |  |  |  |  |
| 3 | store\_code |  | string |  |  |  |  |
| 4 | con\_holding |  | string |  |  |  |  |
| 5 | theme\_start\_date | | string |  |  |  |  |
| 6 | theme\_end\_date | | string |  |  |  |  |
| 7 | npp |  | decimal(15,4) | |  |  |  |
| 8 | ppp |  | decimal(15,4) | |  |  |  |
| 9 | ppp\_start\_date | | string |  |  |  |  |
| 10 | ppp\_end\_date | | string |  |  |  |  |
| 11 | city\_code |  | string |  |  |  |  |
| 12 | dept\_code |  | string |  |  |  |  |
| 13 | item\_code |  | string |  |  |  |  |
| 14 | sub\_code |  | string |  |  |  |  |
| 15 | pcb |  | string |  |  |  |  |
| 16 | dc\_supplier\_code | | string |  |  |  |  |
| 17 | ds\_supplier\_code | | string |  |  |  |  |
| 18 | rotation |  | string |  |  |  |  |
| 19 | run\_date |  | string |  |  |  |  |
| 20 | first\_order\_date | | string |  |  |  |  |
| 21 | first\_delivery\_date | | string |  |  |  |  |
| 22 | regular\_sales\_before\_dm | | double |  |  |  |  |
| 23 | four\_weeks\_after\_dm | | double |  |  |  |  |
| 24 | dm\_sales |  | double |  |  |  |  |
| 25 | order\_qty |  | int |  |  |  |  |
| 26 | first\_dm\_order\_qty | | int |  |  |  |  |
| 27 | order\_without\_pcb | | double |  |  |  |  |
| 28 | dm\_theme\_id | | int |  |  |  |  |

#### DM单品大仓订单（forecast\_dm\_dc\_orders）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | item\_id |  | int |  |  |  |  |
| 2 | sub\_id |  | int |  |  |  |  |
| 3 | con\_holding |  | string |  |  |  |  |
| 4 | theme\_start\_date | | string |  |  |  |  |
| 5 | theme\_end\_date | | string |  |  |  |  |
| 6 | npp |  | decimal(15,4) | |  |  |  |
| 7 | ppp |  | decimal(15,4) | |  |  |  |
| 8 | ppp\_start\_date | | string |  |  |  |  |
| 9 | ppp\_end\_date | | string |  |  |  |  |
| 10 | dept\_code |  | string |  |  |  |  |
| 11 | item\_code |  | string |  |  |  |  |
| 12 | sub\_code |  | string |  |  |  |  |
| 13 | pcb |  | string |  |  |  |  |
| 14 | ds\_supplier\_code | | string |  |  |  |  |
| 15 | rotation |  | string |  |  |  |  |
| 16 | run\_date |  | string |  |  |  |  |
| 17 | first\_order\_date | | string |  |  |  |  |
| 18 | first\_delivery\_date | | string |  |  |  |  |
| 19 | regular\_sales\_before\_dm | | double |  |  |  |  |
| 20 | four\_weeks\_after\_dm | | double |  |  |  |  |
| 21 | dm\_sales |  | double |  |  |  |  |
| 22 | order\_qty |  | int |  |  |  |  |
| 23 | order\_without\_pcb | | double |  |  |  |  |
| 24 | dm\_theme\_id | | int |  |  |  |  |

# 销售预测的建模

## 非DM期间的销售预测模型

### 模型介绍

家乐福的基准使用滑动平均模型作为预测。当异常高销售发生时，滑动平均模型会显示异常高的结果，因此不能准确反映商店的未来销售。 需要人工干预来仔细检查那些预测，这是有效的但是却可能导致错误。

在DM之外，我们使用一个模型在商品/周级别预测销售额。

该范围内约有2,500个商品，预测时间为10周，因此大约有25,000个模型。

培训范围：从2017-01-01开始，直到最后一个数据可用。

根据可用的最新数据预测未来10周。

### 模型训练使用的特征

|  |  |  |
| --- | --- | --- |
| **独热编码** | | |
| **序号** | **列名** | **说明** |
| 1 | store\_code | 商店代码 |
| 2 | city\_code | 城市代码 |
| 3 | sub\_family\_code | 三级类目代码 |
| 4 | three\_brand\_item\_list\_holding\_code | 供应商Holding代码 |
| 5 | current\_dm\_slot\_type\_code | 现在DM上的Slot |
| 6 | next\_dm\_slot\_type\_code | 下次DM上的Slot |
| 7 | next\_dm\_prom\_type | 下次DM 上的促销代码 |
| 8 | item\_seasonal\_code | 商品的季节性代码 |
| 9 | festival\_type | 节假日类型 |
| 10 | current\_prom\_type | 现在的促销类型 |
| 11 | current\_dm\_page\_strategy | 现在的页面策略 |
| 12 | next\_dm\_page\_strategy | 下一次DM上的页面策略 |

|  |  |  |
| --- | --- | --- |
| **时间特征** | | |
| **序号** | **列名** | **描述** |
| 1 | disc\_ratio | 促销中销售商品的周数与销售的总周数相比 |
| 2 | sales\_qty\_fam\_wgt | 与二级类目销售相比的项目销售数量的年度权重 |
| 3 | sales\_qty\_fam\_store\_wgt | 二级类目/商品/商店的销售数量的年度权重 |
| 4 | fam\_sales\_qty\_mean\_4w | 二级类目销售数量的4周移动平均值 |
| 5 | fam\_sales\_qty\_mean\_12w | 二级类目销售数量的12周移动平均值 |
| 6 | fam\_sales\_qty\_mean\_24w | 二级类目销售数量的24周移动平均值 |
| 7 | fam\_sales\_qty\_mean\_52w | 二级类目销售数量的52周移动平均值 |
| 8 | fam\_sales\_qty\_min\_roll | 从一开始，滚动最低每周家庭销售数量 |
| 9 | fam\_sales\_qty\_max\_roll | 从一开始，滚动窗口中最大的每周销售数量的二级类目 |
| 10 | subfam\_sales\_qty\_mean\_4w | 子家庭销售数量的4周移动平均值 |
| 11 | subfam\_sales\_qty\_mean\_12w | 子家庭销售数量的12周移动平均值 |
| 12 | subfam\_sales\_qty\_mean\_24w | 子家庭销售数量的24周移动平均值 |
| 13 | subfam\_sales\_qty\_mean\_52w | 子家庭销售数量的52周移动平均值 |
| 14 | item\_store\_mean\_4points | 过去4个周点的商品/商店的平均销售数量 |
| 15 | item\_store\_mean\_12points | 过去12个周点的商品/商店的平均销售数量 |
| 16 | item\_store\_mean\_24points | 过去24个周点的商品/商店的平均销售数量 |
| 17 | item\_store\_mean\_52points | 最近52个周点的商品/商店的平均销售数量 |
| 18 | ind\_competitor | 如果竞争对手在同一城市代码中出售同一物品，则旗帜等于1 |
| 19 | fam\_sales\_item\_ratio\_4w | 二级类目销售数量的4周移动平均值除以交易数量 |
| 20 | fam\_sales\_item\_ratio\_12w | 二级类目销售数量的12周移动平均值除以交易数量 |
| 21 | fam\_sales\_item\_ratio\_24w | 二级类目销售数量的24周移动平均值除以交易数量 |
| 22 | fam\_sales\_item\_ratio\_52w | 二级类目销售数量的52周移动平均值除以交易数量 |
| 23 | subfam\_sales\_item\_ratio\_4w | 三级类目销售数量的4周移动平均值除以交易数量 |
| 24 | subfam\_sales\_item\_ratio\_12w | 三级类目销售数量的12周移动平均值除以交易数量 |
| 25 | subfam\_sales\_item\_ratio\_24w | 三级类目销售数量的24周移动平均值除以交易数量 |
| 26 | subfam\_sales\_item\_ratio\_52w | 三级类目销售数量的移动平均值为52周除以交易数量 |
| 27 | assortment\_avg\_nsp | 品类表中的平均名义销售价格 |

|  |  |  |
| --- | --- | --- |
| **普通特征** | | |
| **序号** | **列名** | **描述** |
| 1 | trxn\_year | 交易年份 |
| 2 | trxn\_month | 交易月份 |
| 3 | trxn\_week | 交易周 |
| 4 | current\_dm\_page\_no | 现在的DM的页面策略 |
| 5 | current\_dm\_nsp | 现在的DM上的正常售价 |
| 6 | current\_dm\_psp | 现在的DM上的促销售价 |
| 7 | current\_dm\_msp | 现在DM上的会员售价 |
| 8 | ind\_current\_on\_dm\_flag | 是否在本期DM上 |
| 9 | assortment\_active\_flag | 这个商品是否依旧处于存活状态 |
| 10 | holiday\_count\_new | 节假日总数 |
| 11 | next\_dm\_psp | 下期DM上的促销售价 |
| 12 | next\_dm\_nsp | 下期DM上的正常售价 |
| 13 | next\_dm\_page\_no | 下期DM上的页面策略 |
| 14 | next\_dm\_msp | 下期DM上的会员售价 |
| 15 | psp\_nsp\_ratio | 当前DM：PSP / NSP |
| 16 | high\_temp\_month | 给定城市的平均每月高温（3年） |
| 17 | low\_temp\_month | 特定城市的月平均低温（3年） |
| 18 | curr\_psp\_start\_dow | PSP开始的日期编号（星期日= 0，星期一= 1等） |
| 19 | curr\_psp\_end\_dow | PSP结束的日期编号（星期日= 0，星期一= 1等） |
| 20 | curr\_psp\_days | 当前PSP的持续时间（以天为单位） |
| 21 | curr\_psp\_start\_week\_count | 当前DM开始以来的周数 |
| 22 | curr\_psp\_end\_week\_count | 当前DM结束前的周数 |
| 23 | next\_dm\_start\_week\_count | 直到下一个DM开始的周数 |
| 24 | next\_dm\_days | 以天为单位的下一个DM的持续时间 |
| 25 | last\_year\_festival\_ticket\_count | 如果目前有一个节日=>我们去年在同一个节日上的票数 |
| 26 | last\_year\_sales\_qty | 销售数量过去52周 |
| 27 | current\_dm\_busday | DM中的工作日数量 |
| 28 | current\_dm\_weekend\_days | DM中的周末天数 |
| 29 | next\_dm\_busday | 下一个DM中的工作日数 |
| 30 | next\_dm\_weekend\_days | 下一个DM的周末天数 |
| 31 | ind\_coupon\_typeid\_dd | 优惠券类型id独热码 |
| 32 | ind\_coupon\_typeid\_np | 优惠券类型id独热码 |
| 33 | ind\_coupon\_typeid\_ac | 优惠券类型id独热码 |
| 34 | ind\_coupon\_typeid\_cc | 优惠券类型id独热码 |
| 35 | ind\_coupon\_typecode\_cp | 优惠券类型code独热码 |
| 36 | ind\_coupon\_typecode\_mpm | 优惠券类型code独热码 |
| 37 | ind\_coupon\_typecode\_mp | 优惠券类型code独热码 |
| 38 | ind\_coupon\_typecode\_other | 优惠券类型code独热码 |
| 39 | ndv\_coupon\_activity\_type\_id | 此商品/商店/周的不同不同活动类型ID的数量 |
| 40 | ndv\_coupon\_typecode | 此商品/商店/周的不同不同类型代码的数量 |
| 41 | ndv\_coupon\_count1 | 此商品/商店/周的不同不同count1代码的数量 |
| 42 | ind\_coupon\_count1\_27 | 优惠券“count1”类型独热码 |
| 43 | ind\_coupon\_count1\_02\_25 | 优惠券“count1”类型独热码 |
| 44 | ind\_coupon\_count1\_60 | 优惠券“count1”类型独热码 |
| 45 | ind\_coupon\_count1\_4 | 优惠券“count1”类型独热码 |
| 46 | ind\_coupon\_count1\_1 | 优惠券“count1”类型独热码 |
| 47 | ind\_coupon\_count1\_58 | 优惠券“count1”类型独热码 |
| 48 | ind\_coupon\_count1\_11 | 优惠券“count1”类型独热码 |
| 49 | ind\_coupon\_count1\_16\_34 | 优惠券“count1”类型独热码 |
| 50 | ind\_coupon\_count1\_56 | 优惠券“count1”类型独热码 |
| 51 | ind\_coupon\_count1\_37 | 优惠券“count1”类型独热码 |
| 52 | ind\_coupon\_count1\_44 | 优惠券“count1”类型独热码 |
| 53 | ind\_coupon\_count1\_9 | 优惠券“count1”类型独热码 |
| 54 | ind\_coupon\_count1\_10 | 优惠券“count1”类型独热码 |
| 55 | ind\_coupon\_count1\_other | 如果该商品/商店/周的任何其他计数1代码 |

### 模型训练的逻辑

#### 模型预处理

预处理从datalake下载的数据:

检查数据质量，

在需要时转换数据类型,

创建虚拟特征,

创建DM功能期间的天数

将数据集保存到pickle文件中

在启动脚本之前要更改的元素：

folder = '83.new\_bp/'

big\_table = 'forecast\_sprint3\_v11\_flag\_sprint4.csv'

sql\_table = 'vartefact.forecast\_sprint3\_v11\_flag\_sprint4'

target\_value = 'sales\_qty\_sum'

dataset\_name = 'dataset\_1307'

输入：

包含数据集的文件夹

数据集的名称

datalake中相应表的名称

目标值的列名

输出数据表的名称

输出：

folder / Dataset\_part1.pkl：预处理数据集的前半部分

folder / Dataset\_par2.pkl：预处理数据集的后半部分

方法：

阅读以前在服务器上下载的csv

为每列分配适当的类型

创建当前dm周末日作为功能

按类型拆分列：

识别：不是功能，而是识别给定行的方法（item\_id，项目名称等...）

平面功能：我们事先知道或不随时间变化的静态功能（周数，一周中的假日天数，去年的销售数量......）

时间特征：随着时间的推移而变化的动态功能我们事先并不知道（4周销售平均值，竞争对手的产品......）

检查拆分中的错误（重复的名称，缺少的名称等）

将功能保存在csv文件的“folder +'features /'”中

加载保存的文件

过滤数据集中的负销售额

将数据集分成两部分（由于pickle的大小问题）

将数据集保存在文件夹中的2个pickle中

#### 模型训练

训练模型并对过去进行预测，以检查结果的质量或预测未来。

在启动脚本之前要更改的元素：

folder = '83.new\_bp/'

data\_set1 = 'dataset\_1307dataset\_1307\_part1.pkl'

data\_set2 = 'dataset\_1307dataset\_1307\_part2.pkl'

date\_stop\_train = '2019-01-13'

futur\_prediction = False

输入：

包含数据的文件夹

数据集的第1部分（来自预处理脚本）

数据集的第2部分（来自预处理脚本）

停止培训的日期（并预测其余部分）

是或否，取决于是对未来进行预测还是进行回测（输出文件的格式更改）

输出：

Forecast\_10w\_on\_the\_fututre\_error\_'date'.txt：错误的日志文件

Forecast\_10w\_on\_the\_fututre\_log\_'date'.txt：进度的日志文件

Skipped\_week\_forecast\_10.csv：由于缺少数据而跳过的项目/商店周的回顾

Top10\_feature\_forecast\_.csv：前10名按模型重要

results\_forecast\_10w\_.csv：模型的最终结果

model\_result /：包含所有模型的文件夹

方法：

加载pickle数据集

加载特征

按照销量降序对商品排序

准备文件以保存错误的功能/结果/日志

准备10周预测，循环所有商品：

创建字典以保存结果

仅选择当前项目

创建与要预测的周相对应的10列

如果数据集中缺少以下10周之一，则排除该行

在train / test中拆分数据集

用kfold和3个分裂训练模型

如果该项目记录少于3行（跨所有商店），则跳过该项目

用MAPE评估模型表现

训练置信区间模型

保存培训结果

使用10周模型进行预测

保存预测结果

在服务器上写入文件

## DM期间的销售预测模型

### 模型介绍

在DM期间，我们使用一个模型来预测项目/商店级别的整个DM期间的销售额。

训练范围：从2018-01-01开始，直到最后一个数据可用。

只要DM信息可用，就可以在将来预测多个DM。

### 模型训练使用的特征

|  |  |  |
| --- | --- | --- |
| **普通特征** | | |
| **序号** | **列名** | **描述** |
| 1 | current\_dm\_page\_no | 现在DM的页面代码 |
| 2 | current\_dm\_nsp | 现在DM上的平均售价 |
| 3 | current\_dm\_psp | 现在DM上的促销售价 |
| 4 | current\_dm\_psp\_nsp\_ratio | 现在DM上PSP和NSP相除的比率 |
| 5 | last\_year\_dm\_sales | 去年同月DM销售量 |
| 6 | last\_year\_dm\_psp\_nsp\_ratio | 去年同月DM上PSP和NSP相除的比率 |
| 7 | last\_year\_fam\_dm\_sales\_avg | 去年同月整个二级品类的平均销售量 |
| 8 | last\_5dm\_sales\_avg | 商品的最后5个DM的平均销售量 |
| 9 | fam\_last\_5dm\_sales\_avg | 整个二级品类最后五个DM的平均销售量 |
| 10 | current\_dm\_weekend\_days | DM期间的周末日期数量 |
| 11 | last\_year\_dm\_sales\_vrai\_exact | 基于Excel的去年销售DM的销售量 |
| 12 | vrai\_exact\_or\_lunar\_1m | 如果有农历一个月前的销售量 |
| 13 | vrai\_exact\_or\_lunar\_3m | 如果有农历三个月前的销售量 |
| 14 | current\_dm\_busday | DM期间的天数 |
| 15 | uplift\_value | 4w\_sales\_4w\_bef乘以uplift |
| 16 | uplift | 基于excel的 DM映射的去年DM的提升的比率 |
| 17 | 4w\_sales\_4w\_bef | 四个周之前的销售量 |
| 18 | coupon\_disc\_ratio\_avg\_max | 可用促销优惠券中的最大优惠 |
| 19 | coup\_disc\_ratio\_mech\_max | 所有可用促销机制中促销的最大优惠 |

|  |  |  |
| --- | --- | --- |
| **独热编码** | | |
| **序号** | **列名** | **描述** |
| 1 | store\_code |  |
| 2 | item\_seasonal |  |
| 3 | sub\_family\_code |  |
| 4 | current\_dm\_page\_strategy\_code |  |
| 5 | nl | DM属于全国还是区域性的 |
| 6 | current\_dm\_slot\_type\_code |  |
| 7 | festival\_type |  |

### 模型训练的逻辑

#### 模型预处理

预处理从datalake下载的数据：

检查数据质量，

在需要时转换数据类型，

创建虚拟特征，

创建DM功能期间的天数

将数据集保存到pickle文件中

在启动脚本之前要更改的元素：

folder = '83.new\_bp/'

big\_table = 'forecast\_sprint3\_v11\_flag\_sprint4.csv'

sql\_table = 'vartefact.forecast\_sprint3\_v11\_flag\_sprint4'

target\_value = 'sales\_qty\_sum'

dataset\_name = 'dataset\_1307'

输入：

包含数据集的文件夹

数据集的名称

datalake中相应表的名称

目标值的列名

输出数据表的名称

输出：

folder / Dataset\_part1.pkl：预处理数据集的前半部分

folder / Dataset\_par2.pkl：预处理数据集的后半部分

方法：

阅读以前在服务器上下载的csv

为每列分配适当的类型

创建当前dm周末日作为功能

按类型拆分列：

识别：不是功能，而是识别给定行的方法（item\_id，项目名称等...）

平面功能：我们事先知道或不随时间变化的静态功能（周数，一周中的假日天数，去年的销售数量......）

时间特征：随着时间的推移而变化的动态功能我们事先并不知道（4周销售平均值，竞争对手的产品......）

检查拆分中的错误（重复的名称，缺少的名称等）

将功能保存在csv文件的“folder +'features /'”中

加载保存的文件

过滤数据集中的负销售额

将数据集分成两部分（由于pickle的大小问题）

将数据集保存在文件夹中的2个pickle中

训练模型并对过去进行预测，以检查结果的质量。

在启动脚本之前要更改的元素：

**desc = 'promo\_97\_train\_until\_01-07-2019'**

**folder = '97.promo\_futureDMs/'**

**data\_name = 'dataset\_promo.pkl'**

**target\_value = 'dm\_sales\_qty'**

**date\_stop\_train = '2019-07-01'**

**learning\_rate = 0.3**

#### 模型训练

输入：

所需文件的名称

包含数据的文件夹，

数据集名称（来自预处理脚本）

预测列的名称

停止训练的日期（并预测其余部分）

可选：模型的学习率

输出：

model\_error.pkl2 - 标准差模型的pickle

model\_promo\_'running\_name'. pkl - 销售预测模型的pickle

Model\_promo\_'running\_name'.model - 使用XGBoost保存功能保存的模型

Hyperparameters\_'running\_name'.csv - 销售模型的超参数的csv文件

Promo\_sales\_order\_prediction\_by\_item\_store\_dm\_created\_on\_'date'.csv：具有置信区间的最终预测文件

prediction / forecast\_dump.pkl：预测值的pickle

方法：

训练模型

加载pickle数据集

加载功能

过滤周数为负销售数量

仅选择2018年之后的周用来训练（否则所有关于“去年”的所有特征在2018年之前都是空的）

在train / test中拆分数据集

使用k-fold和3个分裂训练预测模型

训练置信区间模型

用MAPE评价模型

训练置信区间模型

将预测模型保存为pickle

保存置信区间模型

保存超参数

创建预测：

加载模型

对可用数据进行预测

过滤否定预测

过滤极高的预测（超过预测的0.1％百分位数的2倍，可以通过Model类中的：check\_result方法修改以更改乘数和百分位数）

在“分析”文件夹中保存要素重要性

将预测结果保存为pickle

计算预测的置信区间

将最终结果以置信区间保存为服务器上的csv

# 需求计划的设计

## 需求计划介绍

开发销售需求计划系统。结合供应商，门店，和大仓订单参数，把销售预测转化成大仓和门店的订单。此项目模型为概念验证型模型。

系统应在给定范围内进行稳定的自动化运行，无需手动输入数据或者干预。

系统应提供如下产出

* 按日生成的门店和大仓订单文件。
* 按周生成的需求计划预测文件
* 所需的可测量KPI的测量结果

## 需求计划的架构

系统每次运行需要产生百万条以上的数据，逻辑较为复杂。系统需在每日8点30分前生成运算结果。数据池中数据最早在每日5点30分可以准备完毕。系统可用运行时间较少。鉴于以上两点和现有环境，系统基于Spark 批处理技术进行搭建。Spark由加州大学伯克利分校AMP实验室开发，是一个围绕速度、易用性和复杂分析构建的大数据处理框架。十分适合用来构建大型的、低延迟的数据应用程序。

系统包含以下几个组成

1. 数据载入部分

Spark 运行环境对Kudu表的支持并不理想。家福乐数据池中的Kudu表多以全量备份形式更新，难以获取过去某一时间点的数据状态。数据载入部分使用Impala运行环境获取数据池表当前快照，并写入需求计划自用表中，以便以后使用。

1. 逻辑部分

此部分为系统核心。在数据载入后此部分运行订单逻辑，生成订单信息，生成需求计划。生成的信息会写入数据池表中。此部分有个分为两个子类，DM订单逻辑和日常订单逻辑。

DM订单逻辑较为简单，逻辑分支较少，使用Pyspark技术基于Spark运行环境开发。日常订单逻辑复杂，分支众多，使用Scala技术基于Spark运行环境开发。

1. 输出部分

此部分读取订单逻辑部分输出，将其按照规定的格式生成订单文件和需求计划文件，并放置于指定位置。此部分基于Python技术开发。使用openpyxl包以生成规定格式文件

1. 监控部分

此部分用以计算所需的可测量KPI。此部分使用Jupyter Notebook技术为基础进行开发。使用运行环境获取所需数据。

1. 运行部分

此部分用以实现系统的自动化运行。使用Airflow技术进行开发。

## 需求计划系统模块

系统包含如下模块

|  |  |  |
| --- | --- | --- |
| 模块归类 | 模块名 | 用途 |
| 运行部分 | show\_dag\_args | 记录当次运行输入参数，如运行日期 |
| 输入载入部分 | load\_data | 获取数据池表的快照 |
| 逻辑部分 | calculate\_service\_level | 计算服务水平 |
| 逻辑部分 | run\_dm\_order | 计算DM订单订量 |
| 逻辑部分 | run\_onstock\_store\_order | 计算存库单品门店订单订量 |
| 逻辑部分 | run\_xdock\_order | 计算越库单品门店订单订量 |
| 逻辑部分 | run\_dc\_order | 计算存库单品大仓订单订量 |
| 输出部分 | generate\_store\_order\_file | 生成门店订单文件 |
| 输出部分 | generate\_dc\_order\_file | 生成大仓订单文件 |
| 监控部分 | check\_store\_order | 检查门店订单生成情况 |
| 监控部分 | check\_dc\_order | 检查大仓订单生成情况 |
| 输出部分 | generate\_forecast\_file | 生成需求计划文件 |
| 监控部分 | run\_monitoring | 以计算所需的可测量KPI |
| 输出部分 | copy\_output | 将KPI测量结果放至指定位置 |

## 模块结构

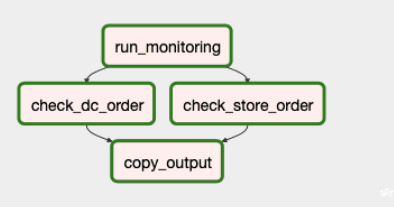
系统模块依据功能分为订单生成和系统监控两部分。两者无依靠关系，独立运行。

### 订单生成

图片包含 屏幕截图

描述已自动生成

### 系统监控



## 详细设计

### 数据库设计

#### 参数表

下列表为需求计划系统参数表。储存程序运行基本参数。表内容更新不频繁。

|  |  |
| --- | --- |
| vartefact.forecast\_store\_item\_details | 门店单品信息 |
| vartefact.forecast\_onstock\_order\_delivery\_mapping | 存库单品送货信息 |
| vartefact.forecast\_xdock\_order\_mapping | 越库单品送货信息 |
| vartefact.forecast\_dc\_item\_details | 大仓单品信息 |
| vartefact.forecast\_calendar | 系统日 |
| vartefact.forecast\_dc\_order\_delivery\_mapping | 大仓订货日 |
| vartefact.forecast\_stores\_delv\_time | 门店送货到店时间 |
| vartefact.v\_forecast\_inscope\_store\_item\_details | 门店可订货单品信息 |
| vartefact.v\_forecast\_inscope\_dc\_item\_details | 大仓可订货单品信息 |
| vartefact.v\_forecast\_daily\_onstock\_order\_items | 当日应在门店下订单的存库单品信息 |
| vartefact.v\_forecast\_daily\_xdock\_order\_items | 当日应在门店下订单的越库单品信息 |

#### 数据输入表

下列表为需求计划系统数据输入表。表中信息按日更新。

|  |  |
| --- | --- |
| vartefact.forecast\_nsa\_dm\_extract\_log | nsa.dm\_extract\_log 表的快照 |
| vartefact.forecast\_item\_code\_id\_stock | 门店每日库存 |
| vartefact.forecast\_p4cm\_store\_item | p4cm.store\_item表的每日快照 |
| vartefact.forecast\_lfms\_daily\_dcstock | 大仓每日库存 |
| vartefact.forecast\_dc\_latest\_sales | 大仓最新平均销量 |
| vartefact.t\_forecast\_daily\_sales\_prediction | 销售预测 |

#### 数据输出表

下列表为需求计划数据输出表，用以接收每次运行的产出。

|  |  |
| --- | --- |
| vartefact.forecast\_onstock\_orders | 存库单品门店订单 |
| vartefact.forecast\_xdock\_orders | 越库单品门店订单 |
| vartefact.forecast\_dc\_orders | 存库单品大仓订单 |
| vartefact.forecast\_onstock\_orders\_hist | 存库单品门店订单历史输出记录 |
| vartefact.forecast\_xdock\_orders\_hist | 越库单品门店订单历史输出记录 |
| vartefact.forecast\_dc\_orders\_hist | 存库单品大仓订单历史输出记录 |
| vartefact.forecast\_dm\_orders | DM单品门店订单 |
| vartefact.forecast\_dm\_dc\_orders | DM单品大仓订单 |
| vartefact.forecast\_store\_daily\_order\_files | 门店订单文件存档 |
| vartefact.forecast\_dc\_daily\_order\_files | 大仓订单文件存档 |

#### KPI监控表

下列表储存所需的可测量KPI的测量结果。表中信息按日更新。

|  |  |
| --- | --- |
| vartefact.foreacst\_store\_monitor | 门店库存状况监控 |
| vartefact.foreacst\_dc\_monitor | 大仓库存状况监控 |
| vartefact.forecast\_monitor\_store\_order\_diff | 门店订单执行情况监控 |
| vartefact.forecast\_monitor\_dc\_order\_diff | 大仓订单执行情况监控 |

#### 模拟运行表

下列表为模拟运行所需表。日常运行时这些表不会被使用。

|  |  |
| --- | --- |
| vartefact.forecast\_simulation\_dm\_orders | 模拟运行产生的DM单品门店订单 |
| vartefact.forecast\_simulation\_dm\_dc\_orders | 模拟运行产生的DM单品大仓订单 |
| vartefact.forecast\_simulation\_orders\_hist | 模拟运行产生的订单历史输出记录 |
| vartefact.forecast\_simulation\_result | 模拟运行历史输出记录 |
| vartefact.v\_forecast\_simulation\_lastest\_result | 模拟运行输出 |
| vartefact.v\_forecast\_simulation\_stock | 模拟运行产生的每日预计库存 |
| vartefact.v\_forecast\_simulation\_orders | 模拟运行产生的订单 |

## DM订单模块设计

开始

载入运行参数

读取DM相关信息

剔除已处理过的DM

保留距运行日最近的DM

计算DM门店订单日

读取DM销售预测

读取DM前后日常销售预测

计算DM门店订单

储存DM门店订单

计算DM大仓订单

储存DM大仓订单

计算DM大仓订单日

结束

## 常规订单模块设计

开始

载入运行参数

读取当日为订单日的的单品

读取单品信息和未来订单日

剔除不可订货单品

读取库存

读取已生成的DM订单信息

读取在途订单信息

读取过去生成的订单计划

是否为大仓订单

读取未来门店向大仓订货计划

读取销售预测

否

是

计算订单量

储存订单

结束

# 使用说明

## 销售预测与需求计划项目文件夹内容介绍

|-- README.md（目录说明文档）

|-- config（和设定相关数据）

| |-- input\_config\_data（输入设定数据）

| `-- setup\_config\_data（安装设定数据）

|-- data\_modeling（模型训练和预测的代码）

| |-- dm\_sales（处于DM期间的商品模型训练与预测的代码）

| |-- normal\_sales（常规商品模型训练与预测的代码）

|-- data\_preperation（数据处理的代码文件夹）

| |-- data\_aggregation（数据处理的具体代码）

|-- demand\_forecast\_kpi\_measurement（需求计划的KPI衡量代码）

| |-- 0\_generate\_report.sh

| |-- Difference\_order\_sales

| |-- Queries

| |-- Results\_Checking

| `-- a001-auto-monitor-checks.ipynb

|-- demand\_planning（需求计划代码）

| |-- order\_parameter\_history

| |-- order\_parameters

| |-- order\_template

| `-- source\_code

|-- documents（项目说明文档）

| |-- 功能要求 Fuctional Requirements

| |-- 家乐福销售预测与需求计划项目说明文档.docx

| |-- 每次会议的会议记录和工作计划

| |-- 访谈报告 Interviews reports

| |-- 需求计划文档

| |-- 需求计划程序API文档

`-- workflow\_integration（任务执行代码）

|-- forecast\_dm\_flow.py（DM期间商品模型训练与预测的Airflow代码）

|-- forecast\_monitoring.py

|-- forecast\_orderflow.py

|-- forecast\_regular\_flow.py（常规商品模型训练与预测的Airflow代码）

|-- forecast\_simulation.py

|-- jar

|-- monitor\_checks.ipynb

|-- py\_scripts

`-- sql

## 执行与设定方法

（注意：需先完成Airflow的配置，配置方式请查阅8. 安装和部署方法。）

### 开启自动化执行

可以通过两种方式开启系统自动化执行。

#### 通过Airflow UI管理界面

Airflow UI管理界面，选择的DAGs

A screenshot of a social media post

Description automatically generated

DAG 名称左侧图式代表自动化执行是否开始。 On 代表自动化执行启动。 Off 代表自动化执行停止。单击图标可进行状态切换。如下图所示，DAG的按钮从OFF改成ON。

A screenshot of a cell phone

Description automatically generated

#### 通过命令行

在Linux运行环境中执行下列命令

|  |  |  |
| --- | --- | --- |
| 命令 | 说明 | 示例 |
| airflow pause {DAG名称} | 开启自动化执行 | airflow pause forecast\_orderflow |
| airflow unpause {DAG名称} | 关闭自动化执行 | airflow unpause forecast\_orderflow |

### 代码中的参数设定

#### 常规商品销量预测的代码参数设定

需设定参数的文件位于于workflow\_integration/forecast\_regular\_flow.py（50到72行）

如图所示，default\_args和DAG的设定请参考Airflow的文档（<http://airflow.apache.org/tutorial.html>）

Config 字典介绍：

Config[‘database’] 为表创建其中的数据库名称。

Config[‘parent\_path’] 为项目文件夹存放位置。

Config[‘config\_data\_path’] 为代码执行所需文件在项目文件夹中的位置。

Config[‘starting\_date’] 为处理的数据的起始日期。



#### DM期间商品销量预测的代码参数设定

需设定参数的文件位于于workflow\_integration/forecast\_dm\_flow.py（66到87行）

如图所示，default\_args和DAG的设定请参考Airflow的文档（<http://airflow.apache.org/tutorial.html>）

Config 字典介绍：

Config[‘database’] 为表创建其中的数据库名称。

Config[‘parent\_path’] 为项目文件夹存放位置。

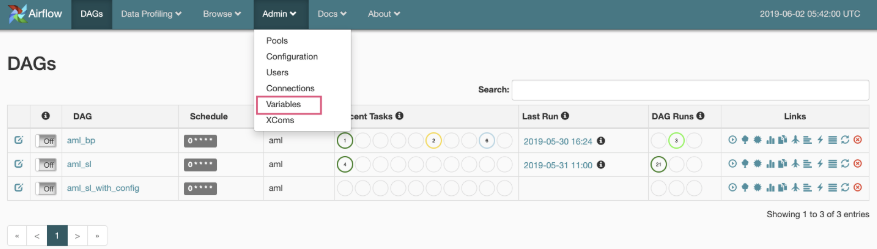
Config[‘config\_data\_path’] 为代码执行所需文件在项目文件夹中的位置。

Config[‘starting\_date’] 为处理的数据的起始日期。

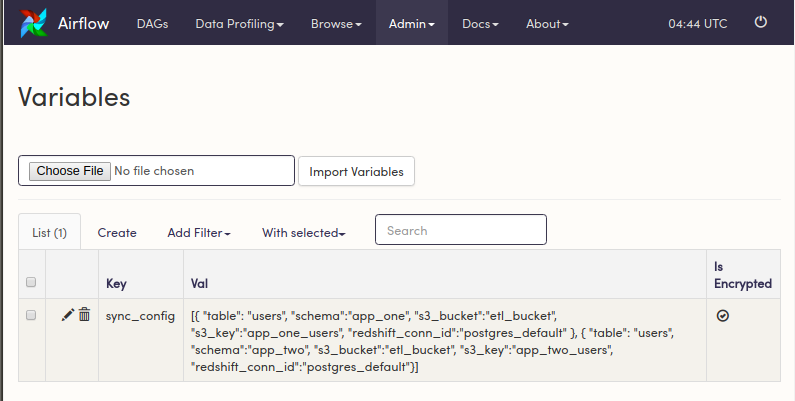


#### 需求计划的代码设定

打开浏览器，进入Airflow UI管控界面。点击Admin。在显示的下拉列表中选择Variable。



在Variables界面中，点击参数进行修改



## 运维方法

### 检查运行状态

前往Airflow安装目录，确保下列两个pid格式文件存在。

|  |  |
| --- | --- |
| airflow-webserver.pid | Airflow UI 管理界面的进程pid |
| airflow-scheduler.pid | Airflow scheduler 调度器的进程pid |

使用ps命令查看运行中的进程，确保上述文件中的pid存在与查询结果。如pid在结果中不存在，需要删除pid文件并重启对应的airflow服务。

检查目录中的err文件。正常运行状态下err文件应该为空。如不为空需打开文件，查看当前问题。

### 检查运行结果

需求计划项目每次运行结果都会写入表vartefact.forecast\_script\_runs 中。表包含如下信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段英文名称 | 字段中文名称 | 类型 | 长度 | 允许空 | 缺省值 | 说明 |
| 1 | insert\_time | 写入时间 | TIMESTAMP |  |  |  | 写入时间 |
| 2 | run\_date | 运行日期 | STRING |  |  |  | 运行输入日期 |
| 3 | run\_status | 运行状态 | STRING |  |  |  | 运行结果 |
| 4 | script\_name | 脚本名称 | STRING |  |  |  | 运行模块名称 |
| 5 | script\_type | 脚本类型 | STRING |  |  |  | 运行类型，如模拟运算 |
| 6 | script\_parameter | 脚本参数 | STRING |  |  |  | 运行输入参数 |
| 7 | output | 输出 | STRING |  |  |  | 模块运行输出，如订单数量 |
| 8 | info | 信息 | STRING |  |  |  | 模块运行相关信息，如运行时长 |
| 9 | error | 错误 | STRING |  |  |  | 模块运行产生的错误 |

需求计划项目日志在如下文件夹中

|  |
| --- |
| /data/jupyter/Carrefour-China-Supply-Chain-Forecast/logs/forecast\_orderflow |

需求计划项目会在运行时生成下列日志文件

|  |  |
| --- | --- |
| 日志文件名 | 说明 |
| run\_parameter.log | 运行输入参数 |
| run\_dm\_order.log | DM 订单日志 |
| run\_onstock\_store\_order.log | 存库商品门店订单生成日志 |
| run\_xdock\_order.log | 越库商品门店订单生成日志 |
| run\_dc\_order.log | 大仓订单生成日志 |

Airflow也会在统一文件夹生成运行日志。运行日志中的重要内容均已同时写入上述日志。通常不需要查看Airflow运行日志

### 订单逻辑调试

订单逻辑调试可以重新计算给定单品的订货，并将计算过程和结果输出至给定的表。订单逻辑调试生成的订单信息不会写入订单表，只错调试使用。

使用如下模板编辑订单逻辑调整运行语句，

spark-submit --class 运行主程序 --master yarn --driver-memory 6G --num-executors 14 jar/carrefour\_forecast\_2.11-0.1.jar 订单日 其它参数 debug\_table=数据库名.表名 >> 日志文件

* 运行主程序

此选项有三个可选值，对应不同类型的订单计算

|  |  |
| --- | --- |
| carrefour.forecast.process.OnStockForecastProcess | 计算存库单品门店订单 |
| carrefour.forecast.process.XDockingForecastProcess | 计算越库单品订单 |
| carrefour.forecast.process.DcForecastProcess | 计算寸库单品大仓订单 |

* 订单日

订单日为yyyMMdd 格式订单日期，如20190902

* 其它参数

订单逻辑调试可使用其它参数来限制运行范围。用户可选择一个或多个下列参数。

|  |  |  |
| --- | --- | --- |
| 其它参数 | 参考格式 | 用途 |
| item\_id | item\_id=1380542 | 限制程序只运行此item\_id对应的单品 |
| sub\_id | sub\_id=1380542 | 限制程序只运行此sub\_id对应的单品 |
| store\_code | store\_code=101 | 限制程序只运行此store\_code对应的门店 |

在linux环境中进入下列文件夹，运行编辑好的语句

|  |  |
| --- | --- |
| /data/jupyter/Carrefour-China-Supply-Chain-Forecast/workflow\_integration | 需求计划程序根文件夹 |

### 示例

spark-submit --class "carrefour.forecast.process.XDockingForecastProcess" --master yarn --driver-memory 8G --num-executors 8 jar/carrefour\_forecast\_2.11-0.1.jar 20190906 cdebug\_table=temp.forecast\_xdock\_order\_debug >> xdock\_log.txt

运行20190906 日的越库单品订单，结果写入temp.forecast\_xdock\_order\_debug 表

spark-submit --class "carrefour.forecast.process.OnStockForecastProcess" --master yarn --driver-memory 6G --num-executors 14 jar/carrefour\_forecast\_2.11-0.1.jar 20190902 item\_id=1380542 store\_code=121 debug\_table=temp.forecast\_onstock\_order\_debug >> onstock\_log.txt

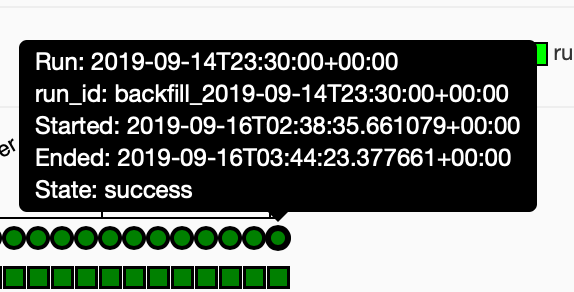
运行20190902 日的存库单品门店订单，结果写入temp. forecast\_onstock\_order\_debug 表。运行仅限于单品1380542，门店121

### 重新生成

打开浏览器，进入Airflow UI 管理界面。点击对应DAG进入DAG界面。进入后默认显示Tree View。如未显示点击Tree View。

图片包含 屏幕截图

描述已自动生成

讲鼠标移至圆形图标，即可查看DAG的每次运行。选择需要的运行日期。如需重新生成2019-09-15日的订单，需选择Run：2019-09-13T23:30:00+00:00的运行。

点击圆形图标来选择此次运行，在弹出的窗口中选择Clear。此操作会将运行状态重设为未执行。

图片包含 屏幕截图

描述已自动生成

在新窗口中选择OK！，完成重设。

Airflow只会自动启动最新的运行。对于过去日期的运行，还需要在linux环境中执行以下指令。

airflow backfill {DAG名} -s {开始日期，包括} -e {结束日期，不包括}

如需重新执行Run：2019-09-13T23:30:00+00:00的运行，命令为

airflow backfill forecast\_orderflow -s 20190913 -e 20190914

# 测试方法

## 测试环境

测试环境需要安装有如下系统

|  |  |
| --- | --- |
| 项目 | 版本号 |
| Airflow | 1.10.3 或更高版本 |
| Jupyter Lab | 0.35.4或更高版本 |
| Spark | 2.4.0 |
| openpyxl | 2.6.3或更高版本 |

测试将于家乐福数据池环境中执行。环境地址为<http://10.172.26.42:8081/lab>。测试输入为家乐福数据池中数据。测试输出将写入家乐福数据池temp数据库中。

## 测试方案

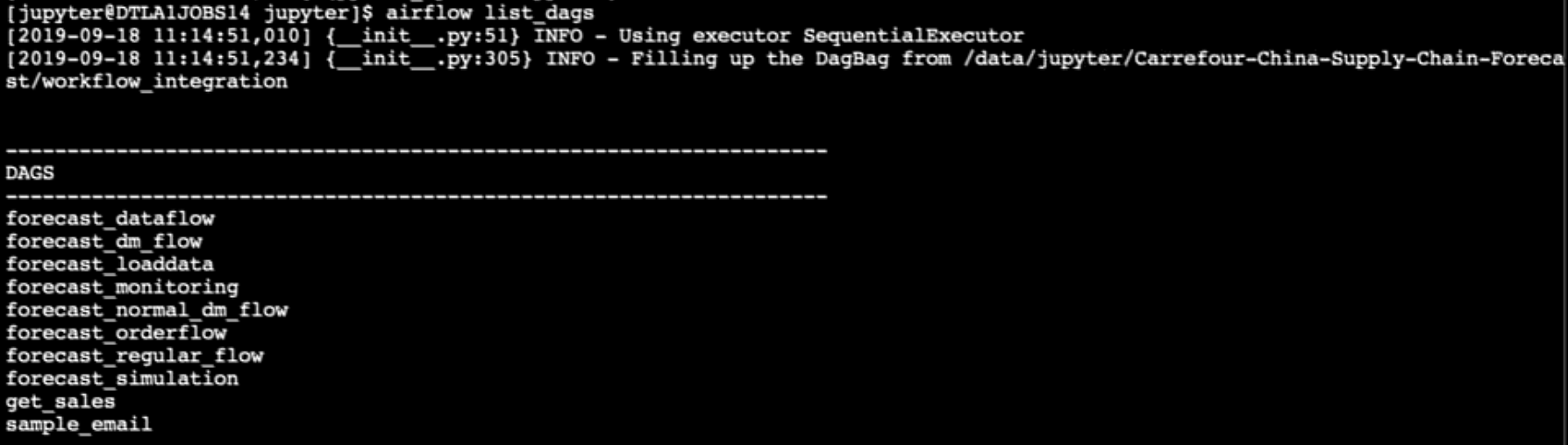
### 系统环境检查

测试方式依赖Airflow工具，

具体为在服务器命令行终端输入对应Airflow指令进行测试。

测试用例：

|  |  |  |
| --- | --- | --- |
| 命令 | 正确返回结果 | 错误对应 |
| airflow list\_dags | 如下图所示，DAGS列表中包含以下8个流程项：  1. forecast\_dataflow  2. forecast\_dm\_flow  3. forecast\_loaddata  4. forecast\_monitoring  5. forecast\_normal\_dm\_flow  6. forecast\_orderflow  7. forecast\_regular\_flow  8. forecast\_simulation | * 提示找不到Airflow，   则需要根据第8章节安装Airflow。   * 返回列表中，   缺少应该有的DAG，则没有正确部署，  则需要根据第8章节进行部署。 |



### 销售预测程序测试

目的：

测试销售预测程序的各个环节是否能够正确执行，包括数据处理以及模型训练和预测。

测试工具：

Airflow

测试环境设定以及注意事项：

测试销售预测前需将代码对应的数据库修改为‘temp’，具体修改方法可参考6.2.2章节中的参数设定。

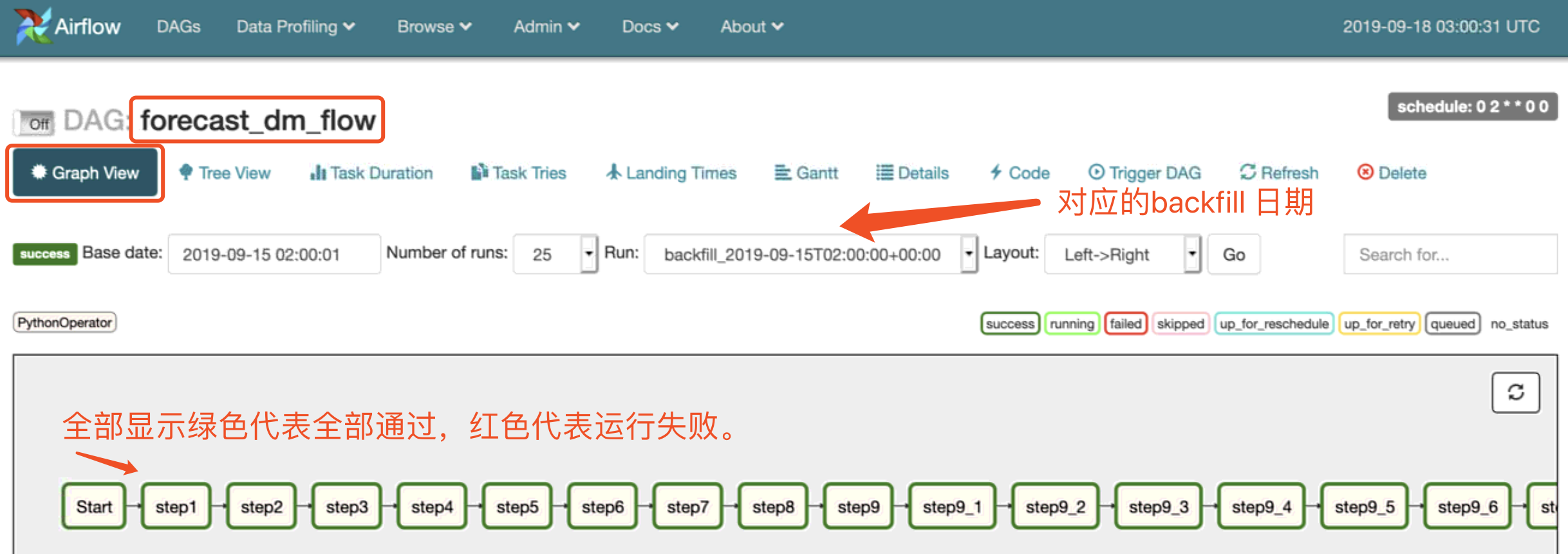
整体测试：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试对象 | 执行命令 | 正确的返回结果 | 错误对应 |
| 1 | 商品正常销售预测的数据处理以及模型训练 | airflow backfill forecast\_regular\_flow -s 20190902 -e 20190909 | 可在Airflow界面中查看。（如下图1所示。） | 可在出现错误的环节重的Log查看到对应的提示。 |
| 2 | 商品在DM期间的销售预测的数据处理以及模型训练 | airflow backfill forecast\_dm\_flow -s 20190902 -e 20190909 | 可在Airflow界面中查看。（如下图2所示。） | 可在出现错误的环节重的Log查看到对应的提示。 |

图1



图2



分段测试用例：

一共分为63各环节(step)，除第一环节(step1)外，其他环节需要执行过一次它的上一环节方可进行测试。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 执行命令 | 正确的返回结果 | 错误对应 |
| 1 | airflow test forecast\_regular\_flow step1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 2 | airflow test forecast\_regular\_flow step2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 3 | airflow test forecast\_regular\_flow step3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 4 | airflow test forecast\_regular\_flow step4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 5 | airflow test forecast\_regular\_flow step5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 6 | airflow test forecast\_regular\_flow step6 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 7 | airflow test forecast\_regular\_flow step7 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 8 | airflow test forecast\_regular\_flow step8 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 9 | airflow test forecast\_regular\_flow step9 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 10 | airflow test forecast\_regular\_flow step9\_1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 11 | airflow test forecast\_regular\_flow step9\_2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 12 | airflow test forecast\_regular\_flow step9\_3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 13 | airflow test forecast\_regular\_flow step9\_4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 14 | airflow test forecast\_regular\_flow step9\_5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 15 | airflow test forecast\_regular\_flow step9\_6 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 16 | airflow test forecast\_regular\_flow step9\_7 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 17 | airflow test forecast\_regular\_flow step9\_8 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 18 | airflow test forecast\_regular\_flow step9\_9 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 19 | airflow test forecast\_regular\_flow step10 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 20 | airflow test forecast\_regular\_flow step11 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 21 | airflow test forecast\_regular\_flow step12 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 22 | airflow test forecast\_regular\_flow step12\_1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 23 | airflow test forecast\_regular\_flow step12\_2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 24 | airflow test forecast\_regular\_flow step12\_3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 25 | airflow test forecast\_regular\_flow step12\_4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 26 | airflow test forecast\_regular\_flow step12\_5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 27 | airflow test forecast\_regular\_flow step12\_6 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 28 | airflow test forecast\_regular\_flow step14 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 29 | airflow test forecast\_regular\_flow step15 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 30 | airflow test forecast\_regular\_flow step16 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 31 | airflow test forecast\_regular\_flow step17 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 32 | airflow test forecast\_regular\_flow step18 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 33 | airflow test forecast\_regular\_flow step19 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 34 | airflow test forecast\_regular\_flow step20 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 35 | airflow test forecast\_regular\_flow step21 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 36 | airflow test forecast\_regular\_flow step22 2019-09-0 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 37 | airflow test forecast\_regular\_flow step23 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 38 | airflow test forecast\_regular\_flow step24 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 39 | airflow test forecast\_regular\_flow step25 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 40 | airflow test forecast\_regular\_flow step26 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 41 | airflow test forecast\_regular\_flow step27\_model 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 42 | airflow test forecast\_regular\_flow step\_normal\_to\_day\_1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 43 | airflow test forecast\_regular\_flow step\_normal\_to\_day\_2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 44 | airflow test forecast\_regular\_flow step\_normal\_to\_day\_3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 45 | airflow test forecast\_regular\_flow step\_normal\_to\_day\_4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 46 | airflow test forecast\_regular\_flow step\_normal\_to\_day\_5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 47 | airflow test forecast\_dm\_flow step\_promo\_0 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 48 | airflow test forecast\_dm\_flow step\_promo\_1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 49 | airflow test forecast\_dm\_flow step\_promo\_2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 50 | airflow test forecast\_dm\_flow step\_promo\_3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 51 | airflow test forecast\_dm\_flow step\_promo\_4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 52 | airflow test forecast\_dm\_flow step\_promo\_5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 53 | airflow test forecast\_dm\_flow step\_promo\_6 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 54 | airflow test forecast\_dm\_flow step\_promo\_7 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 55 | airflow test forecast\_dm\_flow step\_promo\_8 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 56 | airflow test forecast\_dm\_flow step\_promo\_9 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 57 | airflow test forecast\_dm\_flow step\_promo\_10 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 58 | airflow test forecast\_dm\_flow step\_promo\_11\_model 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 59 | airflow test forecast\_dm\_flow step\_promo\_to\_day\_1 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 60 | airflow test forecast\_dm\_flow step\_promo\_to\_day\_2 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 61 | airflow test forecast\_dm\_flow step\_promo\_to\_day\_3 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 62 | airflow test forecast\_dm\_flow step\_promo\_to\_day\_4 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |
| 63 | airflow test forecast\_dm\_flow step\_promo\_to\_day\_5 2019-09-01 | INFO - Done. Returned value was: None | 程序中会给出对应的提示。 |

### 需求计划的测试

#### 订单生成测试

测试需求计划系统定单生成功能。验证订单可以正常生成，生成的订单文件格式与给定的模板相符合。此测试包含如下用例。

* 生成越库单品门店订单
* 生成存库单品门店定单
* 生成存库单品大仓订单
* 门店发往大仓的订单文件
* 大仓发往供应商的订单文件

#### 预测生成测试

测试需求计划系统预测生成功能。验证需求计划可以正常生成，生成的需求预测文件格式与给定的模板相符合。此测试包含如下用例。

* 发往三家供应商的需求计划文件

#### KPI生成测试

测试需求计划系统KPI测量结果生成功能。验证系统可以生成所需的可测量KPI的测量结果。此测试包含如下用例

* 生成KPI报告

#### 需求计划测试用例

|  |  |
| --- | --- |
| 用例编号 | 1 |
| 测试目的 | 生成越库单品门店订单 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  spark-submit --class "carrefour.forecast.process.XDockingForecastProcess" --master yarn --driver-memory 6G --num-executors 14 jar/carrefour\_forecast\_2.11-0.1.jar 20190902 item\_id=1307308 store\_code=134 debug\_table=temp. test\_forecast\_xdock\_order >> xdock\_log.txt |
| 期望结果 | 在家乐福数据池temp数据库中出现表 test\_forecast\_xdock\_order。表中出现越库单品1307308在门店134的订单数据 |

|  |  |
| --- | --- |
| 用例编号 | 2 |
| 测试目的 | 生成存库单品门店订单 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  spark-submit --class "carrefour.forecast.process.OnStockForecastProcess" --master yarn --driver-memory 6G --num-executors 14 jar/carrefour\_forecast\_2.11-0.1.jar 20190902 debug\_table=temp. test\_forecast\_onstock\_order >> onstock\_log.txt |
| 期望结果 | 在家乐福数据池temp数据库中出现表 test\_forecast\_onstock\_order。表中出现越库单品xxx在门店xxx的订单数据 |

|  |  |
| --- | --- |
| 用例编号 | 3 |
| 测试目的 | 生成存库单品大仓订单 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  spark-submit --class "carrefour.forecast.process. DcForecastProcess" --master yarn --driver-memory 6G --num-executors 14 jar/carrefour\_forecast\_2.11-0.1.jar 20190902 debug\_table=temp.test\_forecast\_dc\_order >> dc\_log.txt |
| 期望结果 | 在家乐福数据池temp数据库中出现表test\_forecast\_dc\_order。表中出现越库单品xxx在大仓的订单数据 |

|  |  |
| --- | --- |
| 用例编号 | 4 |
| 测试目的 | 门店发往大仓的订单文件 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  airflow test forecast\_orderflow generate\_store\_order\_file 20190831 |
| 期望结果 | 每日订单输出文件夹中出现文件Store\_Order\_20190902.xlsx。文件包含门店订单文件模板中出现的全部列 |

|  |  |
| --- | --- |
| 用例编号 | 5 |
| 测试目的 | 大仓发往供应商的订单文件 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  airflow test forecast\_orderflow generate\_dc\_order\_file 20190831 |
| 期望结果 | 每日订单输出文件夹中出现文件DC\_Order\_20190902.xlsx。文件包含大仓订单文件模板中出现的全部列 |

|  |  |
| --- | --- |
| 用例编号 | 6 |
| 测试目的 | 大仓发往供应商的订单文件 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  airflow test forecast\_orderflow generate\_forecast\_file 20190831 |
| 期望结果 | 需求计划输出文件夹出现下列文件。  Carrefour\_Order\_Forecast\_DC\_level\_693\_20190902.xlsx  Carrefour\_Order\_Forecast\_DC\_level\_002\_20190902.xlsx  Carrefour\_Order\_Forecast\_DC\_level\_700\_20190902.xlsx  文件包含需求计划文件模板中出现的全部列 |

|  |  |
| --- | --- |
| 用例编号 | 7 |
| 测试目的 | 生成KPI报告 |
| 操作步骤 | 在测试环境中进入需求计划程序根文件夹。执行如下命令  airflow test forecast\_monitoring run\_monitoring 20190902 |
| 期望结果 | 需求计划输出文件夹出现下列文件。 |

# 安装和部署方法

## 系统环境要求

本项目的运行需要如下系统环境

|  |  |
| --- | --- |
| 名称 | 版本号 |
| Airflow | 1.10.3 或更高版本 |
| Jupyter Lab | 0.35.4或更高版本 |
| Spark | 2.4.0 |
| openpyxl | 2.6.3或更高版本 |

## 安装和配置

### Airflow

1. 安装

参考下列airflow官方文档进行airflow的安装。

<http://airflow.apache.org/installation.html>

1. 配置
   * + - 1. 配置文件夹路径

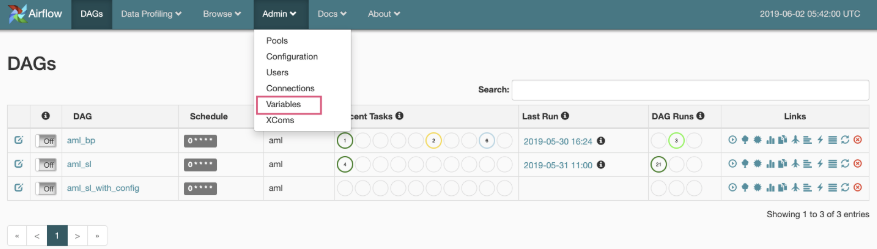
在Airflow安装目录找到airflow.cfg 文件，编辑文件内容。将dags\_folder 指向需求计划程序根文件夹(workflow\_integration文件夹)。 将base\_log\_folder 指向日志文件夹。保存修改内容。

完成修改后，参考下列官方文档启动Airflow

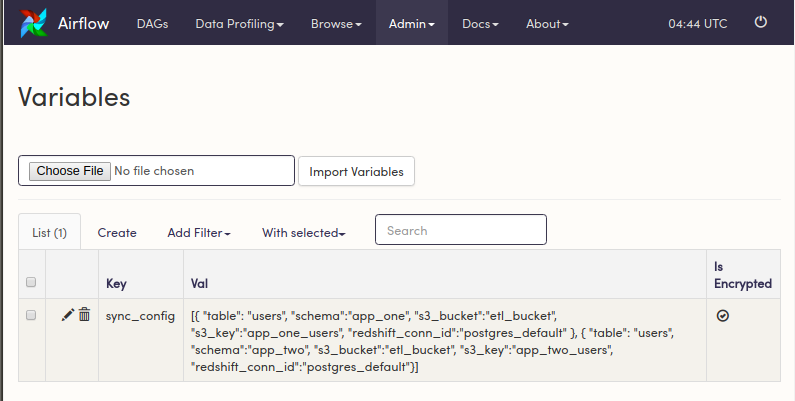
<http://airflow.apache.org/start.html>

* + - * 1. 配置Airflow参数

打开浏览器，进入Airflow UI管控界面。点击Admin。在显示的下拉列表中选择Variable。



在Variables界面中点击Choose File。 选中Airflow 参数配置文件airflow\_variables.json(在项目文件夹中的config文件夹中的setup\_config\_data文件夹里)。 点击Import Variables完成参数配置导入。



### Python包

1. 安装

在项目文件夹中的config文件夹中的setup\_config\_data文件夹里找到‘requirements.txt’并执行以下命令，进行安装。

” pip3 install -r requirements.txt”

## 初始化订单参数

将下列文件放置于同一文件夹

|  |  |
| --- | --- |
| load\_store\_parameters.ipynb | 载入或更新门店订单参数的软件 |
| load\_dc\_parameters.ipynb | 载入或更新大仓订单参数的软件 |
| prepare\_calendar\_table.ipynb | 载入或更新订单日历的软件 |
| East Parameter 20190906.xlsx | 最新门店订单参数 |
| East 3 Supps DC Item list 20190830.xlsx | 最新大仓订单参数 |
| store\_delivery\_time.xlsx | 门店订单到货时间 |

在Jupyter Lab中打开上面的 ipynb 文件。点击Run，在下拉列表中选择Run All Cells 以执行文件中的全部语句。

请确保如下表在数据库中可见

|  |  |
| --- | --- |
| vartefact.forecast\_store\_item\_details | 门店单品信息 |
| vartefact.forecast\_onstock\_order\_delivery\_mapping | 存库单品送货信息 |
| vartefact.forecast\_xdock\_order\_mapping | 越库单品送货信息 |
| vartefact.forecast\_dc\_item\_details | 大仓单品信息 |
| vartefact.forecast\_calendar | 系统日 |
| vartefact.forecast\_dc\_order\_delivery\_mapping | 大仓订货日 |
| vartefact.forecast\_stores\_delv\_time | 门店送货到店时间 |

## 初始化数据库表

打开脚本文件create table.sql，将其中的SQL语句拷贝至Impala运行环境。运行语句完成表的创建。

请确保如下表在数据库中可见

|  |  |
| --- | --- |
| vartefact.forecast\_script\_runs | 系统运行记录 |
| vartefact.forecast\_nsa\_dm\_extract\_log | nsa.dm\_extract\_log 表的快照 |
| vartefact.forecast\_item\_code\_id\_stock | 门店每日库存 |
| vartefact.forecast\_p4cm\_store\_item | p4cm.store\_item表的每日快照 |
| vartefact.forecast\_lfms\_daily\_dcstock | 大仓每日库存 |
| vartefact.forecast\_dc\_latest\_sales | 大仓最新平均销量 |
| vartefact.forecast\_onstock\_orders | 存库单品门店订单 |
| vartefact.forecast\_xdock\_orders | 越库单品门店订单 |
| vartefact.forecast\_dc\_orders | 存库单品大仓订单 |
| vartefact.forecast\_onstock\_orders\_hist | 存库单品门店订单历史输出记录 |
| vartefact.forecast\_xdock\_orders\_hist | 越库单品门店订单历史输出记录 |
| vartefact.forecast\_dc\_orders\_hist | 存库单品大仓订单历史输出记录 |
| vartefact.forecast\_dm\_orders | DM单品门店订单 |
| vartefact.forecast\_dm\_dc\_orders | DM单品大仓订单 |
| vartefact.t\_forecast\_daily\_sales\_prediction | 销售预测 |
| vartefact.forecast\_store\_daily\_order\_files | 门店订单文件存档 |
| vartefact.forecast\_dc\_daily\_order\_files | 大仓定单文件存档 |
| vartefact.foreacst\_store\_monitor | 门店库存状况监控 |
| vartefact.foreacst\_dc\_monitor | 大仓库存状况监控 |
| vartefact.forecast\_monitor\_store\_order\_diff | 门店订单执行情况监控 |
| vartefact.forecast\_monitor\_dc\_order\_diff | 大仓订单执行情况监控 |
| vartefact.v\_forecast\_inscope\_store\_item\_details | 门店可订货单品信息 |
| vartefact.v\_forecast\_inscope\_dc\_item\_details | 大仓可订货单品信息 |
| vartefact.forecast\_simulation\_dm\_orders | 模拟运行产生的DM单品门店订单 |
| vartefact.forecast\_simulation\_dm\_dc\_orders | 模拟运行产生的DM单品大仓订单 |
| vartefact.forecast\_simulation\_orders\_hist | 模拟运行产生的订单历史输出记录 |
| vartefact.forecast\_simulation\_result | 模拟运行历史输出记录 |
| vartefact.v\_forecast\_simulation\_lastest\_result | 模拟运行输出 |
| vartefact.v\_forecast\_simulation\_stock | 模拟运行产生的每日预计库存 |
| vartefact.v\_forecast\_simulation\_orders | 模拟运行产生的订单 |
| vartefact.v\_forecast\_daily\_onstock\_order\_items | 当日应在门店下订单的存库单品信息 |
| vartefact.v\_forecast\_daily\_xdock\_order\_items | 当日应在门店下订单的越库单品信息 |
| vartefact.v\_forecast\_weekly\_dc\_order\_forecast | 存库单品的供应商订货量预测 |
| vartefact.v\_forecast\_weekly\_xdock\_order\_forecast | 越库单品的供应商订货量预测 |