预测 Rossmann 未来的销售额

Domain Background 背景

本项目是来自 Kaggle 的一个竞赛项目[1]。Rossmann 药妆店在欧洲的 7 个国家里有 3000 多家连锁店。影响药妆店的营业额的因素很多，比如打折，附近的竞争者，学校假期（寒暑假），国家假期（圣诞节），季节性和本地因素，药店规模和类型，药店装修歇业等等。这些因素都让每家药店的营业额各不相同[2]。

为了药妆店更好的运营，Rossmann 试图找到影响药店营业额的多种因素的潜在模型，从而更好地预测药店的营业额。在本项目中，Rossmann 提供了数据集提供了全德国 1115 家店的 1017209 条数据，时间周期是从 2013 年 1 月 1 日到 2015 年 7 月 31 日止，这 31 个月的数据。要预测的对这 1115 家店，从 2015 年 8 月 1 日到 2015 年 9 月 17 日共六周的营业额。

竞赛的评估标准则是：竞赛者提交的预测营业额的结果，都会对其进行 Root Mean Square Percentage Error (RMSPE)的计算[3]。计算公式如下：

$$RMSPE = \frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2$$

其中，y_i 代表了单店的单日营业额，$\hat{y}_i$ 则代表了相应的预测营业额。当 y_i =0，即单店单日营业额为零。

在计算 RMSPE 的过程中，这种数据会被自然忽略。

选取 RMSPE 作为评估度量的好处在于：它是完全独立于数据集的大小的，这样可以拿来比较评估不同规模的数据集。当然，也存在一个问题，那就是 RMSPE 永远为正，且没有上限，该值在实际应用中很容易形成 right-skewed 的情况[4]。

Problem Statement 问题陈述

在本项目中，Rossmann 提供的训练集中包含全德国 1115 家店的 1017209 条数据，时间周期是从 2013 年 1 月 1 日到 2015 年 7 月 31 日止，共 31 个月的数据。需要预测的是对这 1115 家店，从 2015 年 8 月 1 日到 2015 年 9 月 17 日共六周的营业额。

由于我的电脑配置较低，为了减少运行和优化的时长，我只选取数据集中 1115 家店中的前 400 家店进行预测分析。这样就把训练集的数据减少到 303023 条，测试集的数据减少到 14736 条，约占到原数据集的 30%。

于是本项目的任务目标就是：根据这 400 家店在过去 31 个月的历史经营数据，以及每家店的额外补充数据，如何尽量准确地预测 2015 年 8 月开始的六周时间内的单日单店营业额，从而让 RMSPE 越低越好。

Datasets and Inputs 数据集 & 输入值

本项目总共提供了四个数据文件：

- train.csv - 包含了营业额及顾客数的历史数据
- test.csv - 不包含营业额及顾客数的历史数据
- sample_submission.csv - 提交文件的正确模板文件
- store.csv - 每家店的补充信息

数据集中相关列的说明：

- Id – 只在 test set 中有的，对于每家店和营业日期的组合 ID
- Store – 每家店独有的 ID，train set 和 test set 中都有
- Sales – 单店营业额，也是我们的预测对象
- Customers – 单日顾客数
- Open – 是否营业，0 = 关店, 1 = 营业
- StateHoliday – 是否为国家公共假期。通常在公共假期，基本上所有店都会关门，除了极个别的情况。同时，学校会在所有公共假期和周末关闭。a = 公共假期, b = 复活节, c = 圣诞节, 0 = 非假期日
- SchoolHoliday – 学校假期，0 = 非假期, 1 = 房价
- StoreType – 有四种店的类型: a, b, c, d
- Assortment – 进一步对店进行分类: a = 基础店, b = 额外店, c = 延展店
- CompetitionDistance – 离最近的竞争者的距离，米
- CompetitionOpenSince[Month/Year] – 最近的竞争者的开店时间（月和年）
- Promo – 单店当日是否在进行促销活动，只在 train set 中
- Promo2 – 某些店进行的一些持续打折活动: 0 = 该店不参与促销, 1 = 该店参与促销
- Promo2Since[Year/Week] – 单店开始参与 Promo2 促销的开始年份和周数
- PromoInterval – 单店开始 Promo2 促销的月份间隔，数值型。比如 "Feb, May, Aug, Nov" 表示该店每逢 2 月，5 月，8 月和 11 月进行促销。

根据比赛靠前的参赛者的讨论和他们的意见，我获得了如下发现，而这些发现将会是我本项目中的重要输入依据：

1. **特征选择**：基本参赛者都投入大量的时间进行特征选择，第一名甚至生成了上百种特征，当然最后发现大部分的特征都不太有用。集合大家的讨论和意见，对建模最具有相关性的特征有 20 种左右[5]：'Store', 'DayOfWeek', 'Sales', 'Customers', 'Open', 'Promo', 'StateHoliday', 'SchoolHoliday', 'Year', 'Month', 'Day', 'WeekOfYear', 'SalePerCustomer', 'StoreType', 'Assortment', 'CompetitionDistance', 'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear', 'Promo2','Promo2SinceWeek', 'Promo2SinceYear', 'CompetitionOpen', 'PromoOpen'。因此在本项目中，我会直接在数据预处理和 EDA 的环节，直接构建这几项最相关的特征，用于后续的回归。

2. **算法选择**：排名靠前的参赛者大部分都用监督学习中的集成模型 XGB(Extreme Gradient Boosting) + 决策树的方法，即可得到令人满意的预测结果。XGB 的原理和算法见[6]。当然也有参赛者用了 CNN 的方法进行模拟。MLP 层数有 1 层，也有三层的。但是最终打分都不如 XGB 算法[5]。因此，在本项目中，也会采用 XGB+决策树的算法。

3. **算法包的选择**：关于 python 自带的 XGBoost (Extreme Gradient Boosting) 和 sklearn 的 GradientBoostRegressor 的比较，可以看出：XGB 包比 sklearn 在对决策树的处理上优于 sklearn 包，所以 XGB 的回归表现上也优于 sklearn，同时速度快，省内存[7]。同时，也有参赛者对 sklearn 包给出的算法结果不太满意[5]。因此在本项目的算法回归上，我会选用 python 的 xgboost 包.

4. **时间序列模型 ARIMA**：参赛者在讨论中时常提到 Time series analysis（TSA）时间序列分析，及 ARIMA（Autoregressive Integrated Moving Average Model）是一种常用的用来预测时间序列的统计学模型[8]。

在 r 和 python statsmodels 中都有 arima 的 model 可以使用。也有参赛者将 TSA 分析和 Xgb 回归结合的尝试[9]，也是对这种季节性的时间模型的一种补充。由于时间的原因，在本项目中，我不会探索这个模型，但是会考虑作为后续改进模型的一个方向。

Solution Statement 解决流程

结合上述的输入参数和选择，解决流程如下：

- 导入数据集
- 数据集预处理&特征构建
- 可视化&探索性数据处理 EDA&特征选择
- 训练模型
- 对训练模型的可视化及优化
- 对模型的讨论

Benchmark Model 基准模型

在 Kaggle 上的此次竞赛，最终吸引了 3303 名竞赛者提交了结果。其中第 330 名（10% percentile）的得分为 0.11773（即 RMSPE）。我会尽量在我的小数据集上跑出接近这样的分数。此为我设定的目标基准。

Evaluation Metrics 评估度量

如在背景中所说，竞赛的评估标准是 Root Mean Square Percentage Error (RMSPE)的计算[3]。计算公式如下：

$$RMSPE = \frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2$$

其中，y_i 代表了单店的单日营业额，$\hat{y}_i$ 则代表了相应的预测营业额。当 y_i =0，即单店单日营业额为零。在计算 RMSPE 的过程中，这种数据会被自然忽略。

选取 RMSPE 作为评估度量的好处在于：它是完全独立于数据集的大小的，这样可以拿来比较评估不同规模的数据集。即使我采用原数据集的一部分，也可以和参赛者的 RMSPE 一同进行比较。

Project Design 项目设计

项目流程如下:

- 导入数据集
- 数据集预处理&特征构建

→只从 1115 家店中选择前 400 家店进行训练和测试

→如上述，特征的构建遵循参赛者的讨论和总结，基本确定为以下 20 种左右的特征：'Store', 'DayOfWeek', 'Sales', 'Customers', 'Open', 'Promo', 'StateHoliday', 'SchoolHoliday', 'Year', 'Month', 'Day', 'WeekOfYear', 'SalePerCustomer', 'StoreType', 'Assortment', 'CompetitionDistance', 'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear', 'Promo2','Promo2SinceWeek', 'Promo2SinceYear', 'CompetitionOpen', 'PromoOpen'。当然，我也会在后续的 EDA 和可视化的过程中，继续探索其它特征。

- 可视化&探索性数据处理 EDA&特征选择

→ 通过对数据进行预处理，和可视化进行 EDA 分析，试图发现各特征的相关性

- 训练模型

→选用参赛者通用的 xgb + 决策树的算法，用 python 自带的 xgb 包进行模型训练。其中训练所需的模型参数，也会参照优胜参赛者的参数。

- 对训练模型的可视化及优化

→对训练模型进行可视化，根据训练的情况，进行模型优化（特征优化，参数优化等）

- 对模型的讨论

参考资料

[1] https://github.com/udacity/cn-machine-learning/tree/master/Rossmann_Store_Sales

[2] https://www.kaggle.com/c/rossmann-store-sales#description

[3] https://www.kaggle.com/c/rossmann-store-sales#evaluation

[4] https://cssd.ucr.edu/Papers/PDFs/MAPE-R%20EMPIRICAL%20V24%20Swanson%20Tayman%20Bryan.pdf

[5] https://www.kaggle.com/c/rossmann-store-sales/discussion/17896

[6] https://xgboost.readthedocs.io/en/latest/tutorials/model.html

[7] https://stats.stackexchange.com/questions/282459/xgboost-vs-python-sklearn-gradient-boosted-trees

[8] https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python

[9] https://www.kaggle.com/c/rossmann-store-sales/discussion/16930#97601

[10] https://www.kaggle.com/c/rossmann-store-sales/leaderboard