# Machine Learning Engineer Nanodegree

## Capstone Proposal

## 毕业项目-开题报告 UDA学员 李欣

**2019.2.13 农历（己亥年）猪年 北京**

### Domain Background

这是一个来自于Kaggle的真实竞赛项目，项目要求预测Rossmann商店的日常销售。Rossmann是一个在7个欧洲国家经营着3,000多家商店的连锁企业。目前，Rossmann商店经理的任务是预先提前六周预测他们的日常销售。商店销售受许多因素的影响，包括促销，竞争，学校和州假日，季节性和地方性。成千上万的个体经理根据他们独特的情况预测销售额，结果的准确性可能会有很大差异。

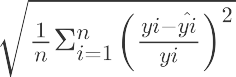
Kaggle的比赛要求：预测遍布德国各地的1,115家商店未来6周的销售额。可靠的销售预测可以使商店经理能够创建有效的员工时间表，从而提高生产力和动力。通过帮助Rossmann创建一个强大的预测模型，您将帮助商店经理专注于对他们最重要的事情：他们的客户和他们的团队！

之所以选择这个项目，是因为该项目与我的实际工作中亟待解决的问题同出一辙。

### Problem Statement

该项目的具体任务是在1,115家Rossmann商店的历史销售数据中预测测试集的“Sales”列，这属于回归类的需求。可以通过构建一个有监督学习类的模型来减少预测值（ŷ）与实际值（y）之间的误差来解决项目需求。

使用Kaggle的RMSPE 函数来验证真实的销售数据与预测数据的差异性，主要是先利用特征工程从杂乱的数据中遴选出有效特征，再使用回归模型训练数据集，最后使用Rmspe函数验证训练结果。RMSPE 函数：



**Datasets and Inputs**

一、数据探索

根据Kaggle提供的数据，现有1,115家Rossmann商店的历史(2013.01.01-2015.07.31)销售数据（如下表，训练集：train.csv；测试集:test.csv；商店的特征集:store.csv）。 测试集的“Sales”列就是实际值（y）。 数据集中的某些商店暂时关闭以进行翻新。由于指标中yi不能为零，因此在数据清理过程中将排除实际销售额为零的数据。

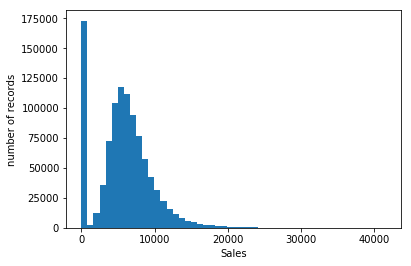
train.csv包含9个特征，共 1017209 个样本；test.csv有8个特征， 41088 个样本，测试集比训练集少Sales字段（Sales字段即预测目标）和Customers特征同时增加Id字段（表示记录编号）。Store.csv有10个特征，1115条记录。训练集没有缺失值， 测试集的部分字段有缺失值，需要填充。对于异常值需要转换数据类型。

|  |  |  |
| --- | --- | --- |
| 数据集 | 内 容 | 数 量 |
| train.csv | 销售⽇、销售额、顾客数、促销、营业、国家假期、学校假期 | 1017209 |
| test.csv | 销售⽇、顾客数、促销、营业、国家假期、学校假期 | 41088 |
| store.csv | 商店的特征信息，包含：商店类型，等级，竞争者信息，促销活动 | 1115 |

|  |  |
| --- | --- |
| 特征名 | 含义 |
| Store | 商店编号，1-1115 |
| Sales | 日销售额 |
| Customers | 日顾客数 |
| Open | 是否营业，0关门/1开门 |
| StateHoliday | 州假日，a=全部假日/b=复活节/c=圣诞节 |
| SchoolHoliday | 学校假日是否开门，0关门/1开门 |
| StoreType | 商店类型 ，a,b,c,d |
| Assortment | 商店评级 ，a,b,c |
| CompetitionDistance | 与最近的竞争者之间的距离 |
| CompetitionOpenSinceYear | 最近的竞争者开业的年份 |
| CompetitionOpenSinceMonth | 最近的竞争者开业的月份 |
| Promo | 当天是否促销，0否/1是 |
| Promo2 | 商店是否参与长期促销，0否/1是 |
| Promo2SinceYear | 商店参与长期促销开始的年份 |
| Promo2SinceWeek | 商店参与长期促销开始的日历周 |
| PromoInterval | 长期促销的月份，Feb/May/Aug/Nov |

二、预测标签：Sales的可视化分析：

本项目中，模型的预测目的是商店的销售额，即训练集的Sales字段，因此有必要查看Sales的数据分布情况。如下图所示：



### Solution Statement

解决方案分为:1、数据分析；2、模型搭建；3、模型评估及调参三个阶段。  
 1、数据分析和清理

·为了获取数据中的有效特征，对数据进行可视化分析，获得数据分布特征，发现数据之间的内在联系并排序有效特征。

·数据预处理，独热编码、寻找并增减异常数据、补全缺失值、规范特征格式、修缮特征字段等。

2、模型搭建

采用XGBOOST建模，预测出 1115 家门店的未来销售额。

1. 模型评估及调参

### 使用GridSearchCV 方法或者贝叶斯优化来进行模型的参数调优，寻找最佳参数组合，以使模型绩效最大化。

### Benchmark Model & Evaluation Metrics

首先，搭建一个实现项目基本需求的基础模型，为之后的模型优化提供基准。

其次，使用LightGBM、Xgboost等算法训练数据，再依据Kaggle推荐的衡量方式：RMSPE来评估模型，最后选出最佳模型再对测试集的商店销售额进行预测。

最后，将预测成绩提交到Kaggle上参加竞赛，至少达到 leaderboard private 的 top 10%以内。实际对于测试集的Rmpse ，需要至少达到 0.11773分。

### Project Design

**第一阶段，数据分析与预处理**

1：对数据进行独热编码、寻找并增减异常数据、补全缺失值、规范特征格式、修缮特征字段等。

2：利用pandas、matplotlib 和 seaborn对数据进行探索性的可视化分析，深刻理解数据的内在联系。

3：对数据进行一些统计上的梳理，获取数据中的分布特征，并排序有效特征。

4：特征选定，将store与train和test数据集合并，并Drop掉冗余和无效特征。

**第二阶段，模型搭建**

选用模型（LightGBM、Xgboost）的默认参数进行训练。在第三阶段经过模型评估之后，再决定使用更适合的模型。

分别使用LightGBM、Xgboost对相同的特征集建立回归预测模型，默认参数表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **LightGBM默认参数** | | **Xgboost** | |
| **参数** | **值** | **参数** | **值** |
| Boosting\_type | Gbdt | Booster | Gbtree |
| Objective | regression | Objective | Reg:linear |
| Learning\_rate | 0.1 | Eta | 0.3 |
| Num\_leaves | 31 | Max\_depth | 6 |
| Max\_bin | 255 | Min\_child\_weight | 1 |
| Min\_data\_in\_leaf | 20 | Subsample | 1 |
| Bagging\_fraction | 1 | Colsample\_bytree | 1 |
| Bagging\_freq | 0 | Gamma | 0 |
| Feature\_fraction | 1 | Lambda | 1 |
| Lambda\_11 | 0 | Alpha | 0 |
| Lambda\_12 | 0 | Random\_state | 23 |
| Min\_split\_gain | 0 |  |  |

**第三阶段，模型评估及优化**

1：使用GridSearchCV 方法或者贝叶斯优化来进行模型的参数调优，寻找最佳参数组合，以使模型绩效最大化。

2：通过控制模型的复杂度、加强模型的鲁棒性来控制模型过拟合。

**第四阶段，结论与展望**

1：使用最后优胜的模型生成预测数据，提交到kaggle上，完成Kaggle项目竞赛。

2：对项目进行总结，分析特征工程、模型选择与调参过程中的理解。

3：对项目进行展望，记录不足与期望。

**参考文献**

1.Rossmann Stroe Sales, <https://www.kaggle.com/c/rossmann-stroe-sales>

2.LightGBM, <https://lightgbm.readthedocs.io/en/latest>

3.XGBoost, <https://xgboost.readthedocs.io/en/latest>

4. Rossmann 销售预测 Top1%, <https://blog.csdn.net/aicanghai_smile/article/details/80987666>

5. Model documentation 1st place <https://www.kaggle.com/c/rossmann-store-sales/discussion/18024>

6. entity-embedding-rossmann, <https://github.com/entron/entity-embedding-rossmann>