

# Data Mining Final Project

--- Rossmann Store Sales Predict

Yonghao Duan  
Yankun Gao



1. Learn about the project and data
2. Exploratory data analysis
3. Data preprocessing
4. Feature engineering
5. Modeling\*
6. Result



# 1. Learn about the project and data

Description: The project is about the prediction of store sales of Rossmann, which is Germany's second-largest drug store chain, with over 3,790 stores in Europe. The project is about prediction of daily sales of each Rossmann store in Germany. The sale time of prediction will be up to six weeks in advance. A robust prediction model will greatly help the store managers to create effective staff schedules that increase productivity and motivation.

Reasons we choose the project:

Firstly, It is a real-world problem, and analyzing a real-world case is always exciting. Secondly, this is a huge data set, and it contains lots of features. Dealing with all kinds of features can help us to learn better how to make predictions of store sales. Thirdly, since this project is a competition we got online, our result can be compared with the results from other excellent data scientists which will urge us to work harder to get a better prediction score.



# 1. Learn about the project and data

```
train.shape
test.shape
store.shape
```

```
len(train.Date.value_counts()) # 942 days store sales for train data and 48 days for test data
(train.Date.value_counts().index.min()) # time start for train data
(train.Date.value_counts().index.max()) # time end for train data
(test.Date.value_counts().index.min()) # time start for test data
(test.Date.value_counts().index.max()) # time end for test data
```

executed in 361ms, finished 21:15:28 2018-11-20

(1017209, 9)

942

'2013-01-01'

(41088, 8)

'2015-07-31'

'2015-08-01'

(1115, 10)

'2015-08-01'

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1
3	4	5	2015-07-31	13995	1498	1	1	0	1
4	5	5	2015-07-31	4822	559	1	1	0	1

Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear
1	c	a	1270.0	9.0	2008.0	0	NaN	NaN
2	a	a	570.0	11.0	2007.0	1	13.0	2010.0
3	a	a	14130.0	12.0	2006.0	1	14.0	2011.0
4	c	c	620.0	9.0	2009.0	0	NaN	NaN
5	a	a	29910.0	4.0	2015.0	0	NaN	NaN



# 1. Learn about the project and data

Feature list: ['Id', 'Store', 'DayOfWeek', 'Date', 'Sales', 'Customers', 'Open', 'PromCompetitionDistance', 'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear', 'Promo2', 'Promo2SinceWeek', 'Promo2', 'StateHoliday', 'SchoolHoliday', 'StoreType', 'Assortment', '2SinceYear', 'PromoInterval']

- **Id** - an Id that represents a (Store, Date) tuple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (this is what you are predicting)
- **Customers** - the number of customers on a given day
- **Open** - an indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indicates if the (Store, Date) was affected by the closure of public schools
- **StoreType** - differentiates between 4 different store models: a, b, c, d
- **Assortment** - describes an assortment level: a = basic, b = extra, c = extended
- **CompetitionDistance** - distance in meters to the nearest competitor store
- **CompetitionOpenSince[Month/Year]** - gives the approximate year and month of the time the nearest competitor was opened
- **Promo** - indicates whether a store is running a promo on that day
- **Promo2** - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
- **Promo2Since[Year/Week]** - describes the year and calendar week when the store started participating in Promo2



## 2. Exploratory data analysis

By using *matplotlib* and *seaborn* packages, the relationships between label and features will be displayed by using different data visualization functions. More deep understanding for features and label will be discovered by using exploratory data analysis.

The relationships between Sales(label) and features will be visualized. The features contain DayOfWeek, Promo, StateHoliday, SchoolHoliday, StoreType, Assortment, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear.

Shown on Jupyter notebook



### 3. Data preprocessing

It is a very clean data and we do not do too much work.

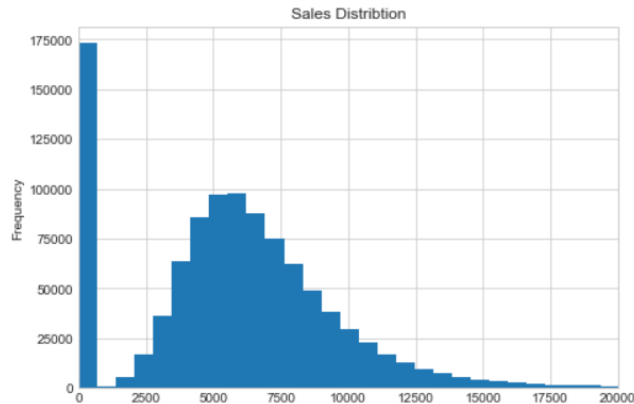
The only missing values shown right.

Attributes	No. of Missing Values
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	544
Open	11



## 4. Feature engineering

### 4.1 Use log-transformation for label *sales*



### 4.2 Split feature *date* into more features

Split it into features *Year, Month, Day, WeeofYear and Season*

### 4.3 Transform values of discrete features into dummy variables

model can only deal with numbers not string





## 5. Modeling

### 5.1 Evaluation formula/objective formula

The evaluation formula of the result is the Root Mean Square Percentage Error (RMSPE).  $y_i$  denotes the sales of a single store on a single day and  $\hat{y}_i$  denotes the prediction values.

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



## 5. Modeling

### Benchmark 1: Linear Regression

<a href="#">benchmark_result.csv</a> 7 days ago by yonghao <a href="#">add submission details</a>	0.42897	0.41635	<input type="checkbox"/>
---	---------	---------	--------------------------

### Benchmark2: XGBoost without extra features and combo features

<a href="#">Benchmark2_XGBoost.csv</a> 7 days ago by yonghao <a href="#">add submission details</a>	0.12210	0.11320	<input type="checkbox"/>
---	---------	---------	--------------------------



## 5. Modeling

### Benchmark 3: XGBoost with more features

1.

Based on exploratory data analysis result, related features can be created by group by features *store*, *dayofweek*, *promo*, *year*.

2.

Add state location info for each store

3.

remove outlier based on median absolute deviation(MAD) for each store

[Test\\_eta0.0244fourAttributes.csv](#)

13 days ago by yonghao

[add submission details](#)

0.11386

0.10859



State: <https://www.kaggle.com/c/rossmann-store-sales/discussion/17048#96969>

MAD: <http://www.itl.nist.gov/div898/handbook/eda/section4/eda43.htm#lglewicz>



## 5. Modeling

Final Model: XGBoost with extra features and fine tune parameters

In order to get the best accuracy, we begin to tune hyperparameters of XGBoost to obtain a best result.

XGBoost is a gradient boosting algorithm and it is used for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models.

$$Obj = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

**Training loss**  
(RMSPE here)

**Complexity of the Trees**  
Regularization



## 5. Modeling

Regularization, which is used to penalty complexity of model refers several hyperparameters and we will tune several of them.

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

**Number of leaves**

**L2 norm of leaf scores**

Parameter	Values
booster	gbtree
eta	0.3
max_depth	13
cols_sample	0.9
subsample	0.7
train_test_split	2015-7-15
seed	42

Parameter description:

1. eta: learning rate to prevents overfitting
2. max\_depth: the max depth of the tree
3. cols\_sample: the percentage of features can be chosen
4. subsample: the percentage of samples can be chosen
5. seed: used to initialize parameters so that you can repeat your model result



## 5. Modeling

when max\_depth is 10, the valid loss is minimal

max_depth	eval-rmspe
6	0.095637
8	0.090162
10	0.088746
12	0.089783
14	0.091024

lower eta to obtain a better eval\_rmspe, eta=0.02 and 0.01 are good enough

eta	eval-rmspe
0.3	0.09041
0.1	0.086487
0.05	0.085552
0.02	0.084729
0.01	0.084853



## 5. Modeling

**subsample = 0.9 and colsample\_bytree = 0.7, eval\_rmsep is minimal, choose the two parameters**

---

subsample	colsample_bytree	eval-rmspe
0.7	0.7	0.09119
0.7	0.8	0.089963
0.7	0.9	0.089316
0.8	0.7	0.089537
0.8	0.8	0.089607
0.8	0.9	0.089136
0.9	0.7	0.088746
0.9	0.8	0.089186
0.9	0.9	0.089178



## 5. Modeling

*result of Valid\_RMSPE based on time split*

Time Split Point	RMSPE Val	RMSPE Train
2015-06-01	0.138664	0.113628
2015-07-01	0.115334	0.086579
2015-07-15	0.098203	0.069941


**Final Parameter Choice**

Parameter	Number
eta	0.02
max_depth	10
cols_sample	0.9
subsample	0.9
Time Split	2015-07-15





## 6. Result

<a href="#">benchmark_result.csv</a> 7 days ago by yonghao add submission details	0.42897	0.41635	<input type="checkbox"/>
<a href="#">Benchmark2_XGBoost.csv</a> 7 days ago by yonghao add submission details	0.12210	0.11320	<input type="checkbox"/>
<a href="#">Test_eta0.342fourAttributes.csv</a> 13 days ago by yonghao add submission details	0.11852	0.11296	<input type="checkbox"/>
<a href="#">Test_eta0.0244fourAttributes.csv</a> 13 days ago by yonghao add submission details	0.11386	0.10859	<input type="checkbox"/>
<a href="#">Test_eta0.0142fourAttributes.csv</a> 13 days ago by yonghao add submission details 	0.11331	0.10808	<input type="checkbox"/>
<a href="#">Stacking_20_models.csv</a> 7 days ago by yonghao add submission details	0.10878	0.10340	<input type="checkbox"/>

obtain 12th/3303 in private leaderboard



## References:

- [1]: Rossman Store Sales: <https://www.kaggle.com/c/rossmann-storesales>
- [2]: Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. 2016:785-794.
- [3]: Liaw A, Wiener M. Classification and Regression by randomForest[J]. R News, 2002, 23(23)
- [4]: Putting stores on the map: <https://www.kaggle.com/c/rossmann-store-sales/discussion/17048#96969>

