Machine Learning Engineer Nanodegree

Capstone

Agnes Yan

Udacity

30, June 2019

I. Definition

Project Overview

Rossmann is Germany's second-largest drug store chain and it operates over 3,000 drug stores in 7 European countries. The aim of this project is to forecast Rossmann's sales by leveraging Rossmann's store data and machine learning algorithm.

There are multiple advantages to solve this time series analysis based business forecasting problem. Firstly, from the store management aspect, store manager could utilize data analysis results to make the decision on what kind of drug do they need to prepare monthly or seasonally. Secondly, from financial perspective, by forecasting sales in advance, medical store manager could also reduce medicine waste and enhance their business profits. Nonetheless, mining data could also gain insights which drive strategies for customer acquisition, retention, and optimization of best loyalty offerings (David, J. 2016).

The main datasets used in this project are from Kaggle official website, the training data is from "train.csv" and "store.csv" two datasets. The " train.csv "dataset contains over 1 million sales data for 1115 Rossmann drug stores all over the country. The "store.csv "dataset records store related features like store type, assortment and competition stores related information.

Problem Statement

This project is a supervised regression problem. Our goal is to use given features, build an appropriate model and forecast 6 weeks sales in advance.

So, the expected problem to be solved is:

Predict the daily sales for up to six weeks in advance.

Sales --- Y-hat is the Predicted Value in this model, other features like promo, customer number etc. should be the independent variable X which helps us to find the relationship

between $\hat{Y}$ and X.

$$\hat{Y} = AX + b$$

Before taking any further analysis, I just briefly visualized the datasets to gain some insights of the basic relations between different features and sales.

Then, according to Hans-On Machine Leaning with Scikit- Learn & TensorFlow (Aurélien Géron, 2017, 59), I followed 9 steps in data prepossessing part:

(1) Firstly, examine the whole dataset, discovering original dataset distributions, missing values, outliers and incorrect data.
(2) Do deep exploratory data analysis to check feature relations one by one.
(3) Do feature engineer according to EDA result in the previous step.
(4) Deal with categorical data by taking use of one hot encoder.
(5) Normalize numerical features if they are not normal distributed.
(6) Split data to train and validation datasets by utilizing time series split.
(7) Implement XGBOOST model and using cross validation to train XGBoost model.
(8) Tuning model according to RMSPE result.
(9) Forecast test dataset and submit the result on Kaggle.

Metrics

The evaluation metrics for this project is Root Mean Square Percentage Error (RMSPE).

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

As we can see from above equation, RMSPE is the square root of MSPE. For it is a relative error which is calculated based on the percentage error, RMSPE has the advantage of scale independence comparing to RMSE, .

In this equation, y_i denotes the sales of a single store on a single day and $\hat{y}_i$ denotes the corresponding prediction. Any day and store with 0 sales is ignored in scoring in this project.(Kaggle)

II. Analysis

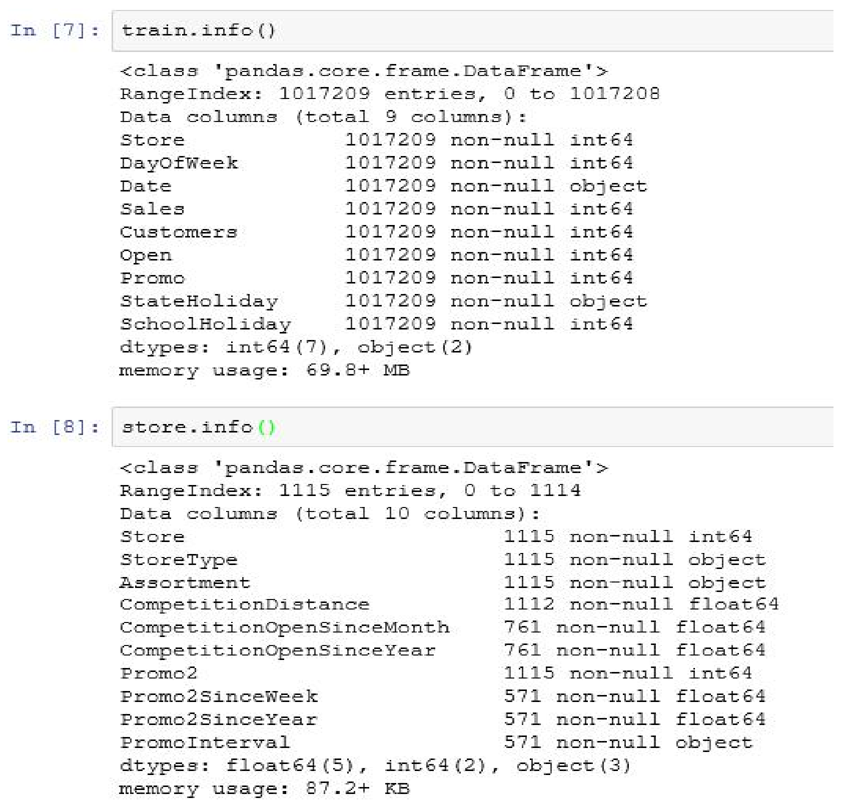
Data Exploration

• Dataset Overview

According to the Kaggle competition page, the dataset of Rossmann's store ("train.csv" and "store.csv") contain 15 features which are:

[1]Id - an Id that represents a (Store, Date) duple within the test set
[2]Store - a unique Id for each store
[3]Sales - the turnover for any given day (this is what you are predicting)
[4] Customers - the number of customers on a given day
[5] Open - an indicator for whether the store was open: 0 = closed, 1 = open
[6] 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
[7] SchoolHoliday - indicates if the (Store, Date) was affected by the closure of public schools
[8] StoreType - differentiates between 4 different store models: a, b, c, d
[9] Assortment - describes an assortment level: a = basic, b = extra, c = extended
[10] CompetitionDistance - distance in meters to the nearest competitor store
[11] CompetitionOpenSince[Month/Year] - gives the approximate year and month of the time the nearest competitor was opened
[12] Promo - indicates whether a store is running a promo on that day
[13] Promo2 - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
[14] Promo2Since[Year/Week] - describes the year and calendar week when the store started participating in Promo2
[15] PromoInterval - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

The basic data type and statistical details can be observed from Table 1 & Table 2 :



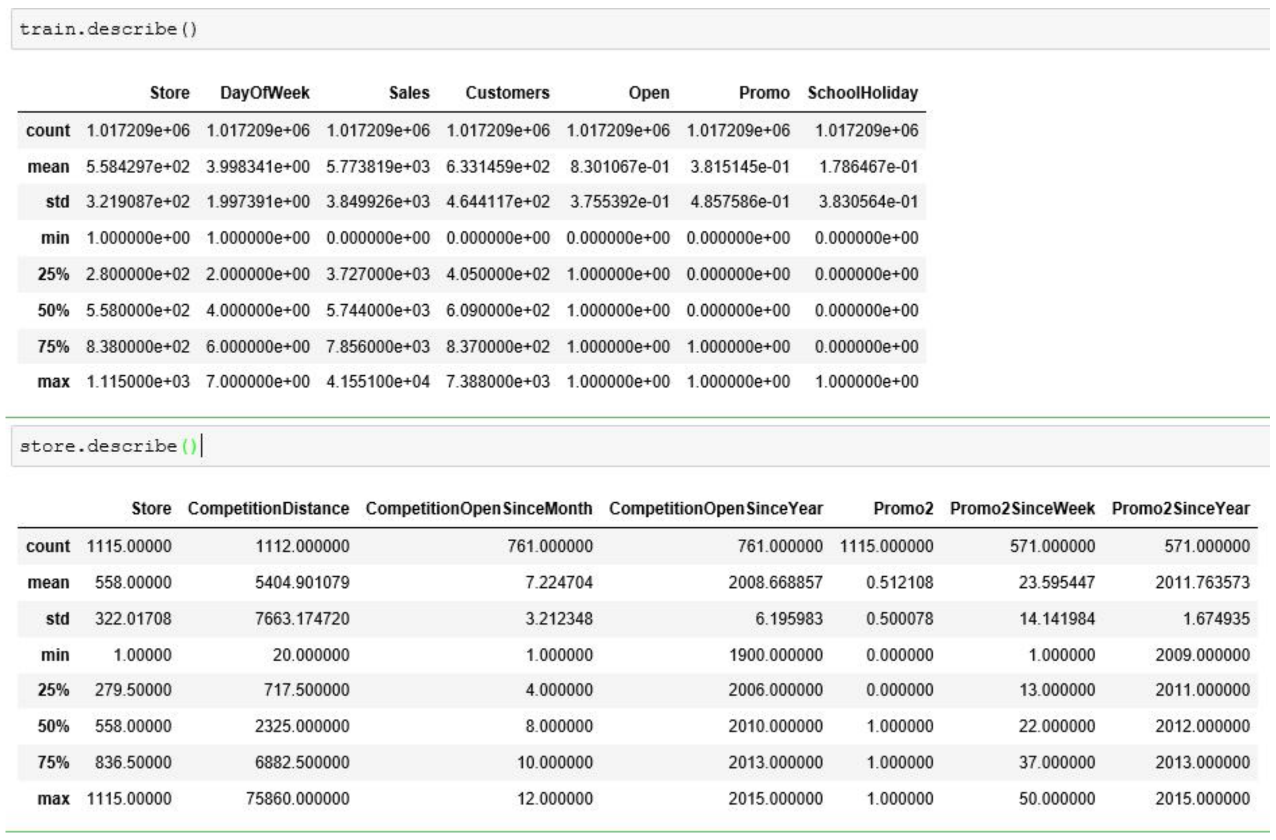
```
In [7]: train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 columns):
Store            1017209 non-null int64
DayOfWeek        1017209 non-null int64
Date             1017209 non-null object
Sales            1017209 non-null int64
Customers        1017209 non-null int64
Open             1017209 non-null int64
Promo            1017209 non-null int64
StateHoliday     1017209 non-null object
SchoolHoliday    1017209 non-null int64
dtypes: int64(7), object(2)
memory usage: 69.8+ MB
```

```
In [8]: store.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
Store                        1115 non-null int64
StoreType                    1115 non-null object
Assortment                   1115 non-null object
CompetitionDistance          1112 non-null float64
CompetitionOpenSinceMonth    761 non-null float64
CompetitionOpenSinceYear     761 non-null float64
Promo2                       1115 non-null int64
Promo2SinceWeek              571 non-null float64
Promo2SinceYear              571 non-null float64
PromoInterval                571 non-null object
dtypes: float64(5), int64(2), object(3)
memory usage: 87.2+ KB
```

Table 1 Basic Data Type, Null Value of Train and Store Dataset



```
train.describe()
```

	Store	DayOfWeek	Sales	Customers	Open	Promo	SchoolHoliday
count	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06
mean	5.584297e+02	3.998341e+00	5.773819e+03	6.331459e+02	8.301067e-01	3.815145e-01	1.786467e-01
std	3.219087e+02	1.997391e+00	3.849926e+03	4.644117e+02	3.755392e-01	4.857586e-01	3.830564e-01
min	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.800000e+02	2.000000e+00	3.727000e+03	4.050000e+02	1.000000e+00	0.000000e+00	0.000000e+00
50%	5.580000e+02	4.000000e+00	5.744000e+03	6.090000e+02	1.000000e+00	0.000000e+00	0.000000e+00
75%	8.380000e+02	6.000000e+00	7.856000e+03	8.370000e+02	1.000000e+00	1.000000e+00	0.000000e+00
max	1.115000e+03	7.000000e+00	4.155100e+04	7.388000e+03	1.000000e+00	1.000000e+00	1.000000e+00

```
store.describe()
```

	Store	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear
count	1115.00000	1112.000000	761.000000	761.000000	1115.000000	571.000000	571.000000
mean	558.00000	5404.901079	7.224704	2008.668857	0.512108	23.595447	2011.763573
std	322.01708	7663.174720	3.212348	6.195983	0.500078	14.141984	1.674935
min	1.00000	20.000000	1.000000	1900.000000	0.000000	1.000000	2009.000000
25%	279.50000	717.500000	4.000000	2006.000000	0.000000	13.000000	2011.000000
50%	558.00000	2325.000000	8.000000	2010.000000	1.000000	22.000000	2012.000000
75%	836.50000	6882.500000	10.000000	2013.000000	1.000000	37.000000	2013.000000
max	1115.00000	75860.000000	12.000000	2015.000000	1.000000	50.000000	2015.000000

Table 2 Statistical Distribution of Numerical Features for Train and Store Datasets

As we can see from the statistic description above, sales as a numerical number, is the predicted variable in this project.

However, the independent variables in this project have 4 types:

Data type	Feature Name	Dataset

Numerical	Sales, Customers,
	Open(Binary)
	Promo(Binary)
	Promo2(Binary)
	CompetitaionDistance
		train
		test
		store

Categorical	StateHoliday, SchoolHoliday, StoreType, Assortment	train, test, store
Datetime	Date, DayofWeek Promo2SinceWeek Promo2SinceYear PromoInterval CompetitionSinceMonth CompetitionSinceYear	train, test, store

Table 3 Different Feature Types of Datasets

• *Dataset Examination*

In order to examine the numerical, categorical and datetime variables thoroughly, I explored three datasets - train, store, test one by one from four aspects ----- (a.) Null. (b.) Duplicated data. (c.) Incorrect data.(d.) Distribution, the findings are showed below in Table 4:

Dataset Name	Null	Duplicated	Incorrect data
train	1. 180 stores do not have datafrom 1st July 2014 - 31st Dec 2014	none	1. Typing error for Stateholiday. 0 and '0' 2. There are 54 store records which open but with 0 sales.
store	1. Competitiondistance (3 missing values) 2. Competitionopensincemonth(354 missing values) 3. Competitionopensinceyear (354 missing) 4. Promo2sinceweek (454 missing) 5. Promo2sinceyear (454 missing) 6. Promointerval (454 missing)	none	none

test	1. Open (11 missing values)	none	none

Table 4 Null, Duplicated and Incorrect Data info of datasets

Comparing with train and test datasets, some other findings are discovered as listed below:

1. There is no sales and customers features in test dataset
2. There are 856 unique stores in test dataset, and 1115 unique stores in train dataset.
3. StateHoliday feature in training dataset has 4 different values '0','a','b','c';
   However, in test dataset, StateHoliday feature only contains value '0' and 'a'.
4. The distribution of SchooolHoliday is different in train and test dataset, 18% of record is during school holiday period in train dataset, however this figure in test dataset is about 45%.
5. The distribution of sales is right skewed as we can see from figure 1.

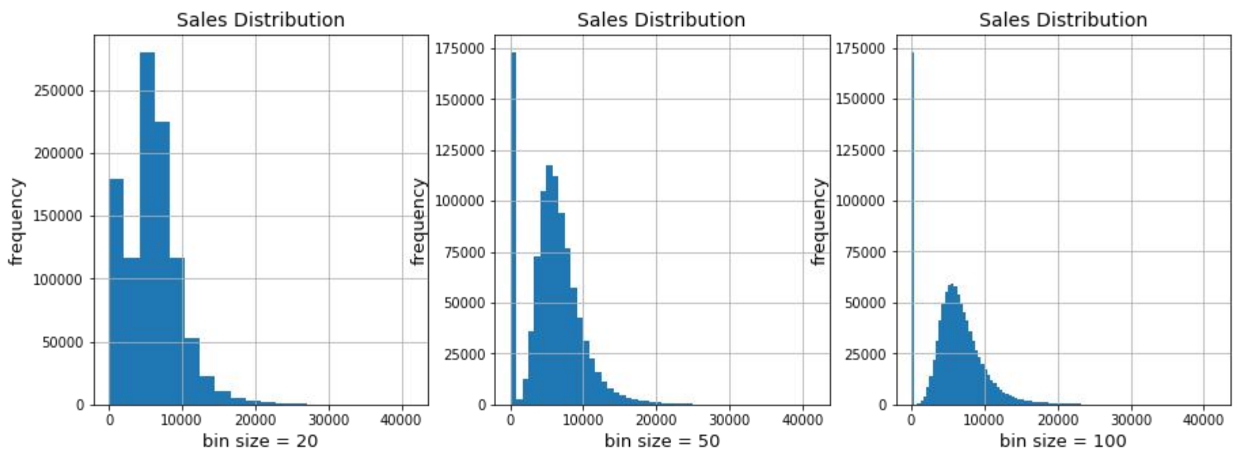
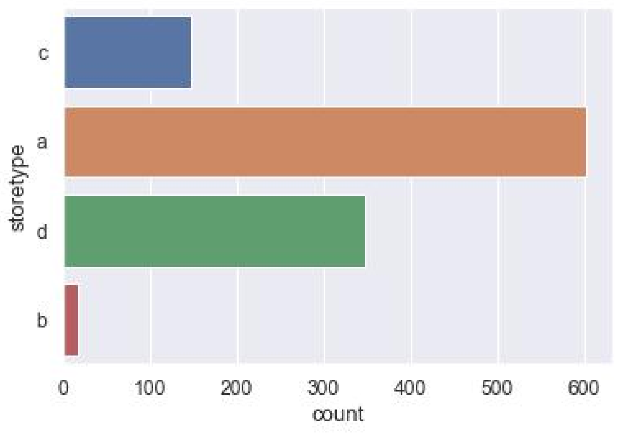


Figure 1. Sales Distribution in Train Dataset

6. When examine the feature distribution of store dataset, interesting finding include : (i). the number of store type 'a' is almost the sum of store type 'b', 'c', 'd'. (ii). the number of assortment 'a' is slightly bigger than assortment 'c', as we can see from figure 2 &3.



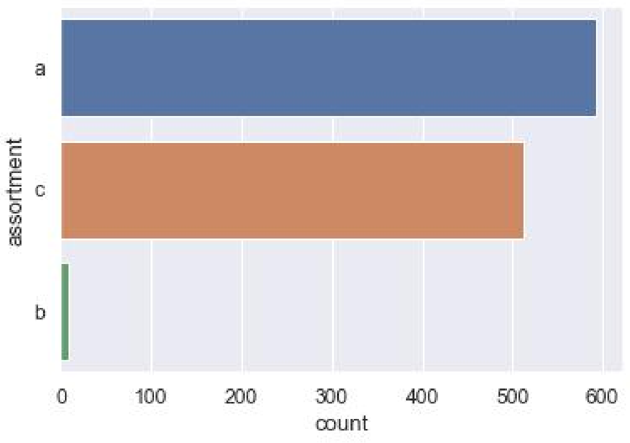
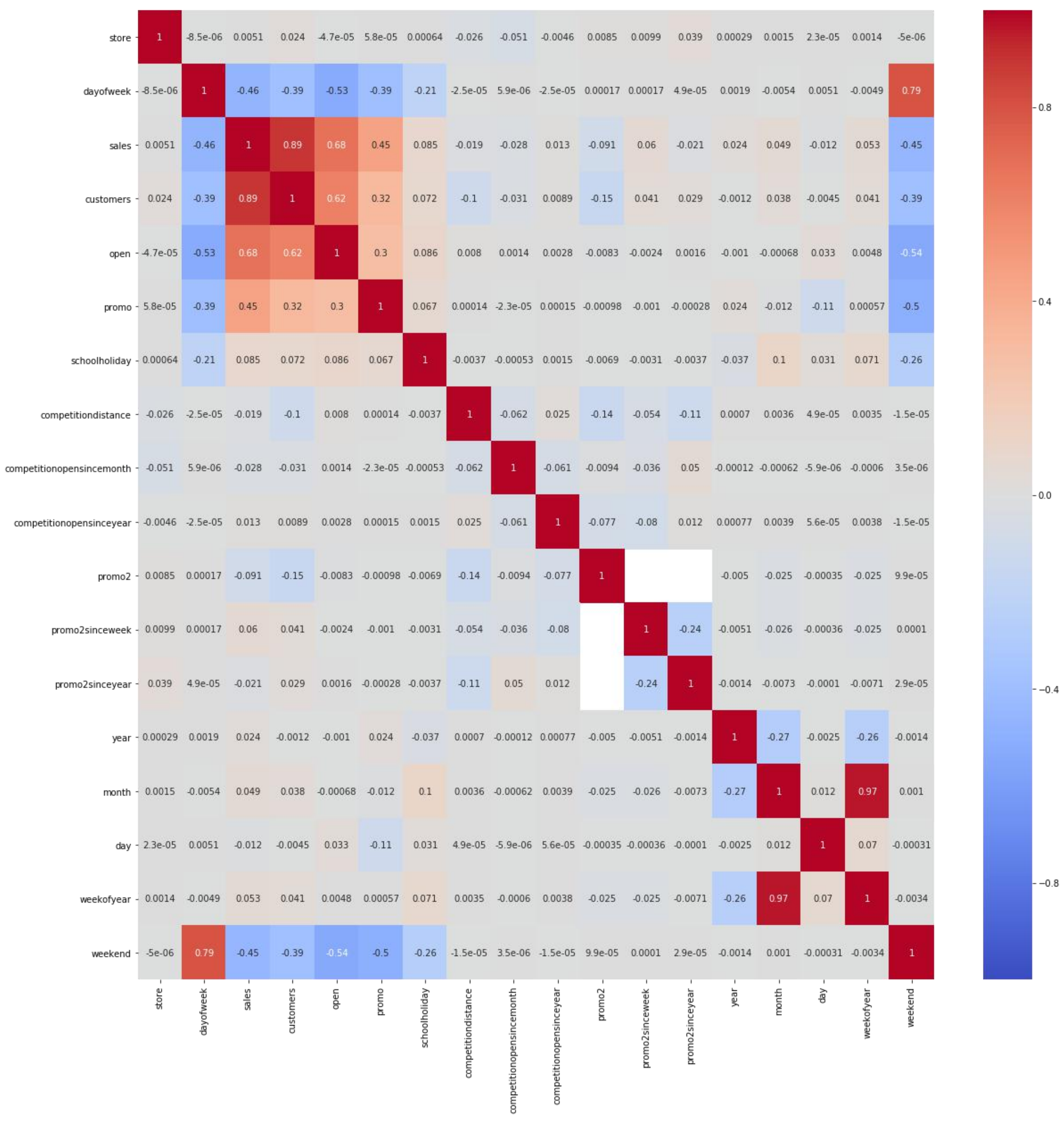


Figure 2. Count of Store Types in Store Dataset ***Figure 3. Count of Assortment Types in Store Dataset***

Exploration Visualization

• *Numerical Feature Correlation*

As we can see from the figure 4 - the heatmap of numerical variables, variable day of week, customers, open, promo and weekend have relatively stronger linear relations with sales.



	store	dayofweek	sales	customers	open	promo	schoolholiday	competitiondistance	competitionopensincemonth	competitionopensinceyear	promo2	promo2sinceweek	promo2sinceyear	year	month	day	weekofyear	weekend
store	1	-8.5e-06	0.0051	0.024	-4.7e-05	5.8e-05	0.00064	-0.026	-0.051	-0.0046	0.0085	0.0099	0.039	0.00029	0.0015	2.3e-05	0.0014	-5e-06
dayofweek	-8.5e-06	1	-0.46	-0.39	-0.53	-0.39	-0.21	-2.5e-05	5.9e-06	-2.5e-05	0.00017	0.00017	4.9e-05	0.0019	-0.0054	0.0051	-0.0049	0.79
sales	0.0051	-0.46	1	0.89	0.68	0.45	0.085	-0.019	-0.028	0.013	-0.091	0.06	-0.021	0.024	0.049	-0.012	0.053	-0.45
customers	0.024	-0.39	0.89	1	0.62	0.32	0.072	-0.1	-0.031	0.0089	-0.15	0.041	0.029	-0.0012	0.038	-0.0045	0.041	-0.39
open	-4.7e-05	-0.53	0.68	0.62	1	0.3	0.086	0.008	0.0014	0.0028	-0.0083	-0.0024	0.0016	-0.001	-0.00068	0.033	0.0048	-0.54
promo	5.8e-05	-0.39	0.45	0.32	0.3	1	0.067	0.00014	-2.3e-05	0.00015	-0.00098	-0.001	-0.00028	0.024	-0.012	-0.11	0.00057	-0.5
schoolholiday	0.00064	-0.21	0.085	0.072	0.086	0.067	1	-0.0037	-0.00053	0.0015	-0.0069	-0.0031	-0.0037	-0.037	0.1	0.031	0.071	-0.26
competitiondistance	-0.026	-2.5e-05	-0.019	-0.1	0.008	0.00014	-0.0037	1	-0.062	0.025	-0.14	-0.054	-0.11	0.0007	0.0036	4.9e-05	0.0035	-1.5e-05
competitionopensincemonth	-0.051	5.9e-06	-0.028	-0.031	0.0014	-2.3e-05	-0.00053	-0.062	1	-0.061	-0.0094	-0.036	0.05	-0.00012	-0.00062	-5.9e-06	-0.0006	3.5e-06
competitionopensinceyear	-0.0046	-2.5e-05	0.013	0.0089	0.0028	0.00015	0.0015	0.025	-0.061	1	-0.077	-0.08	0.012	0.00077	0.0039	5.6e-05	0.0038	-1.5e-05
promo2	0.0085	0.00017	-0.091	-0.15	-0.0083	-0.00098	-0.0069	-0.14	-0.0094	-0.077	1			-0.005	-0.025	-0.00035	-0.025	9.9e-05
promo2sinceweek	0.0099	0.00017	0.06	0.041	-0.0024	-0.001	-0.0031	-0.054	-0.036	-0.08		1	-0.24	-0.0051	-0.026	-0.00036	-0.025	0.0001
promo2sinceyear	0.039	4.9e-05	-0.021	0.029	0.0016	-0.00028	-0.0037	-0.11	0.05	0.012		-0.24	1	-0.0014	-0.0073	-0.0001	-0.0071	2.9e-05
year	0.00029	0.0019	0.024	-0.0012	-0.001	0.024	-0.037	0.0007	-0.00012	0.00077	-0.005	-0.0051	-0.0014	1	-0.27	-0.0025	-0.26	-0.0014
month	0.0015	-0.0054	0.049	0.038	-0.00068	-0.012	0.1	0.0036	-0.00062	0.0039	-0.025	-0.026	-0.0073	-0.27	1	0.012	0.97	0.001
day	2.3e-05	0.0051	-0.012	-0.0045	0.033	-0.11	0.031	4.9e-05	-5.9e-06	5.6e-05	-0.00035	-0.00036	-0.0001	-0.0025	0.012	1	0.07	-0.00031
weekofyear	0.0014	-0.0049	0.053	0.041	0.0048	0.00057	0.071	0.0035	-0.0006	0.0038	-0.025	-0.025	-0.0071	-0.26	0.97	0.07	1	-0.0034
weekend	-5e-06	0.79	-0.45	-0.39	-0.54	-0.5	-0.26	-1.5e-05	3.5e-06	-1.5e-05	9.9e-05	0.0001	2.9e-05	-0.0014	0.001	-0.00031	-0.0034	1

Figure 4. Correlation Matrix Shows Numerical Variable Correlation Coefficient

- Customers vs Sales

The significant linear relation between customers and sales can also be observed from figure 5

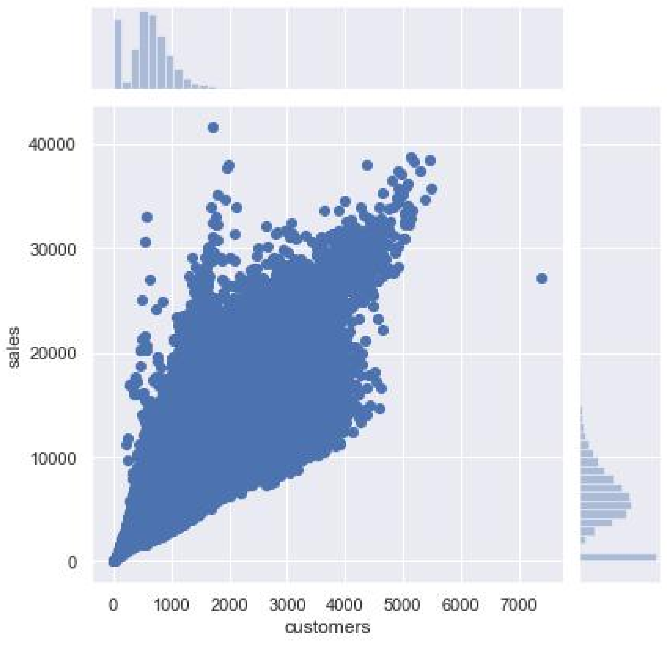


Figure 5 Seaborn Jointplot of Sales and Customers

- Promo Promo2 vs Sales

The box plot between promo and sales (figure 6) also implied the total sales boost when stores are in promotion. Figure 8 which showed sales data in promo (marked with cross) separately with stores are not in promo (marked with dot), revealed that promo drive higher sales and more customers to store.

However the as we can see from figure 4, there seems no strong relation between promo2 and sales.

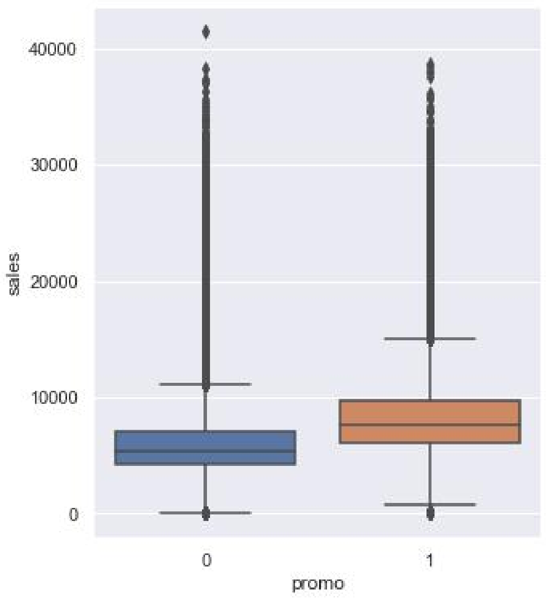


Figure 6 Boxplot between Sales and Promo

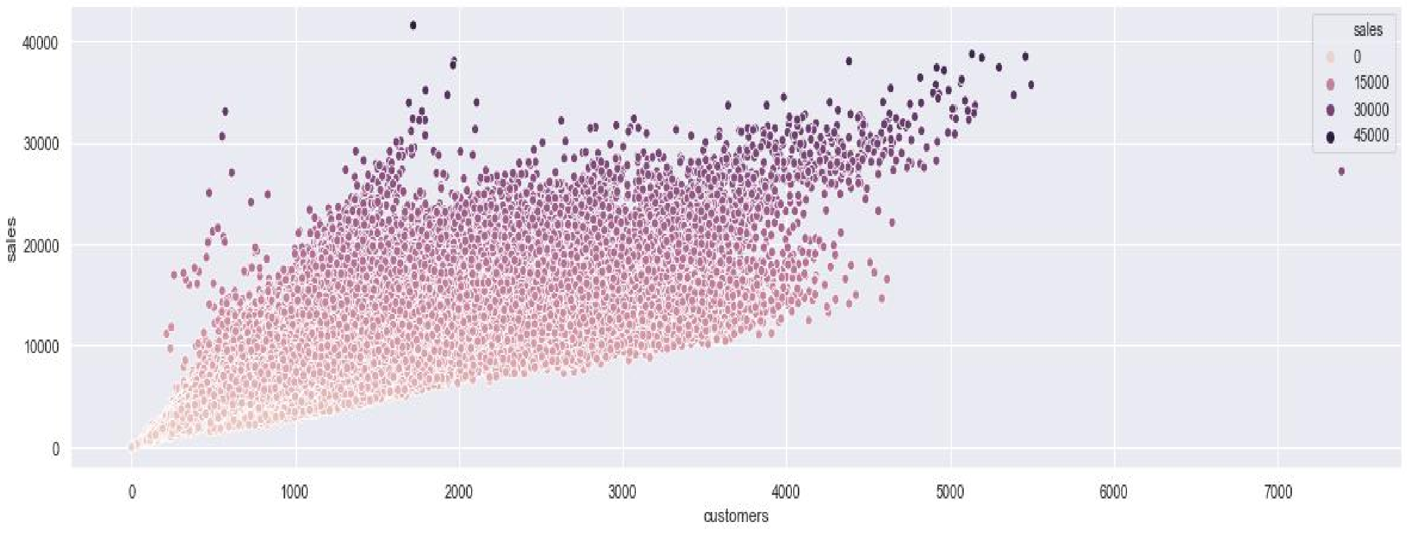


Figure 7 Scatter plot between Sales and Customers

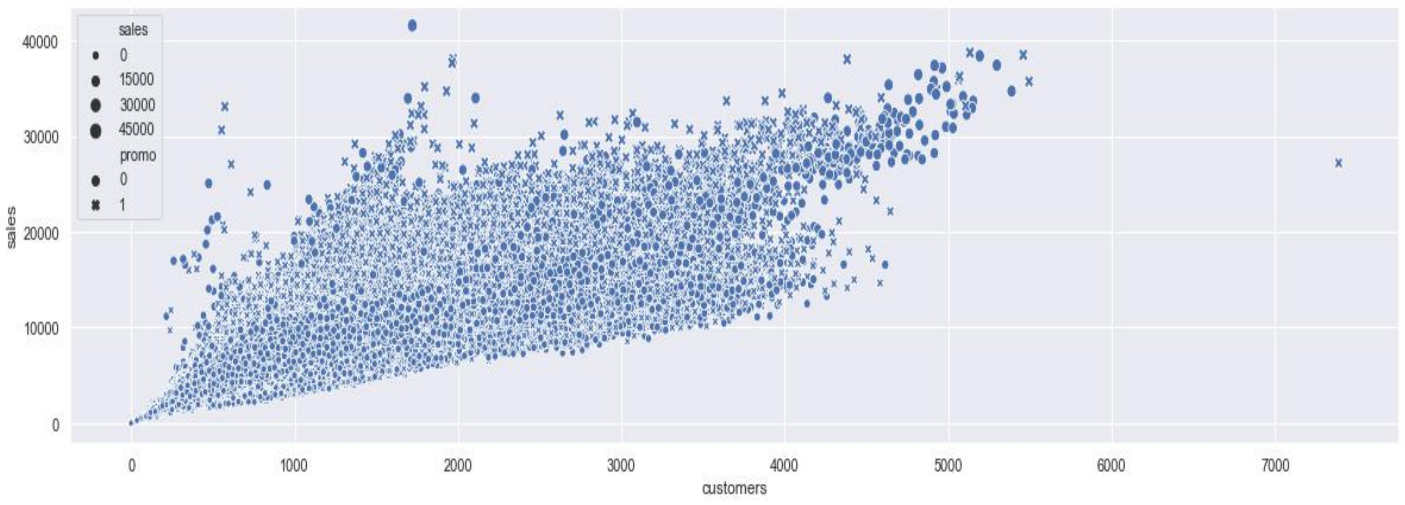


Figure 8 Scatter plot between Sales and Customers Differentiated by Promo

• Categorical Feature vs Sales

- StateHoliday vs Sales

As we can see from figure 9 & 10, sales boosted during state holiday compared with non state holiday. Moreover, the trend in figure 10 also showed sales in state holiday has been gradually increasing from 2013 to 2015.

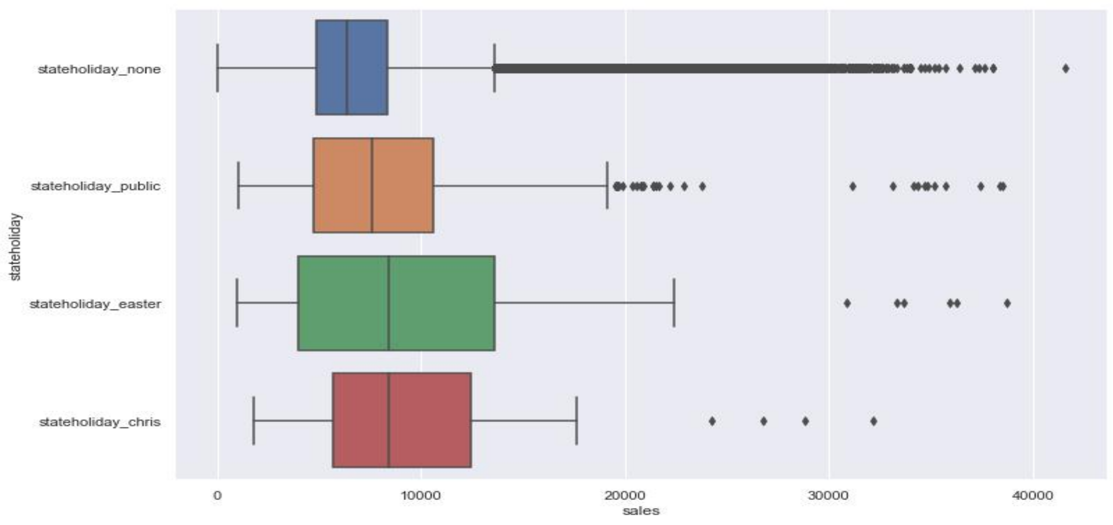


Figure 9 Box Plot of State Holiday vs Sales

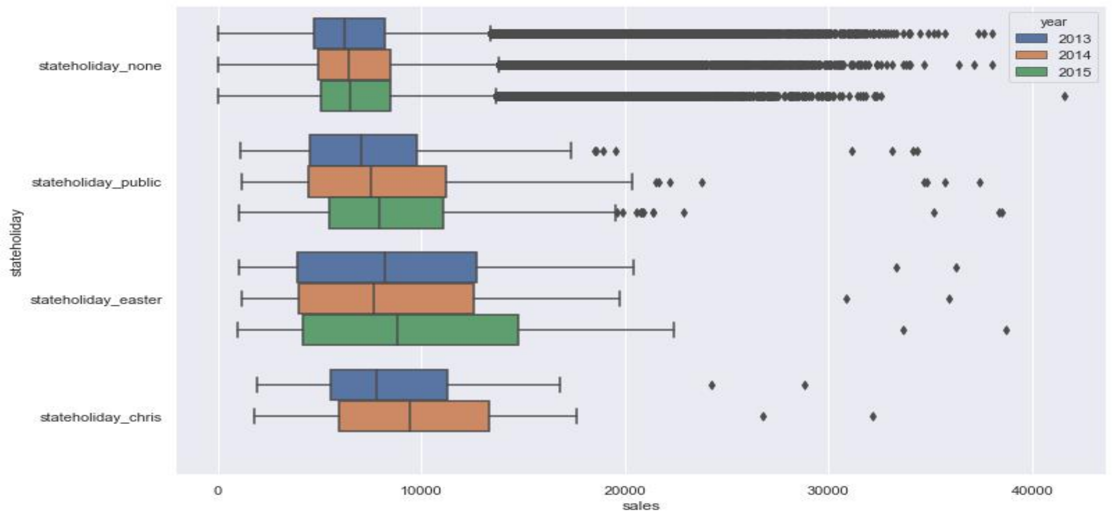


Figure 10 Box Plot of State Holiday vs Sales from 2013 - 2015

- StoreType vs Sales

How about the relations between different store type a, b, c, d and sales ?

Let's look at figure 11, the scatter plots between customers and sales clearly showed four different store type tends to have divers characteristics.

Store type b which is marked with red point shows they have the largest customer flow. However the sales per customer is lower than the other three store types (see figure 14 - *Box Plot of State Holiday vs Sales per Customer for Four Store Types*). Store type b is also the only store type opened in Public holiday, East holiday and Christmas.

However Store type d on contrast, has the least customer flow but with the highest sales per customer (see figure 14 - *Box Plot of State Holiday vs Sales per Customer for Four Store Types*). Maybe the location of store type d is in some wealthier community.

We can tell the difference between store type a and c in figure 12, which implies store type a have the most store numbers, higher amount of sales than store type c. We can also observe from Figure 13 & 14 that, the amount of sales per customers is similar between store type a and c. However, people tends to spend more in store type c during public holiday compared with store type a.

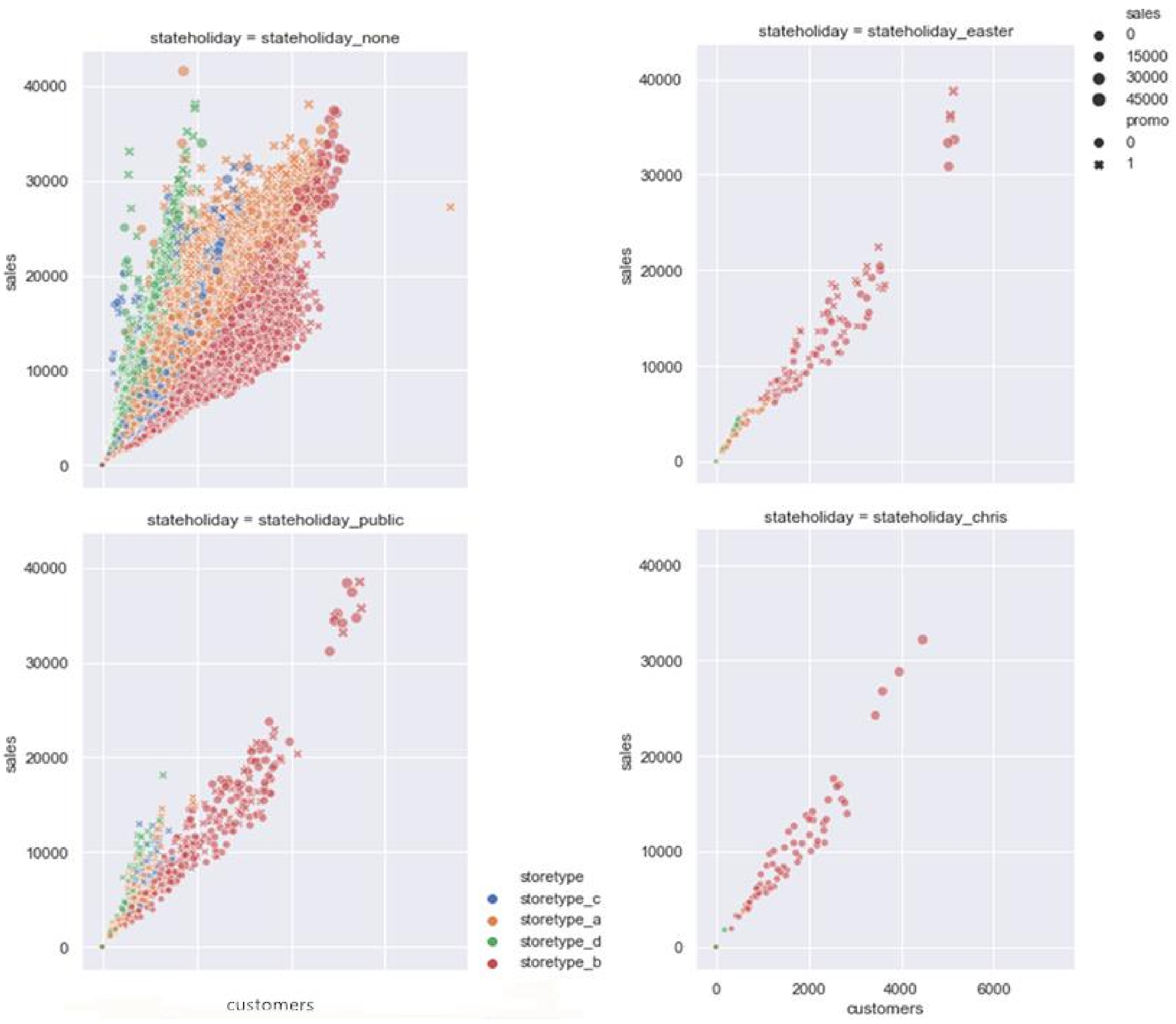


Figure 11 Scatter Plot of Sales and Customers
Differentiated by State Holiday, Store Type and Promo

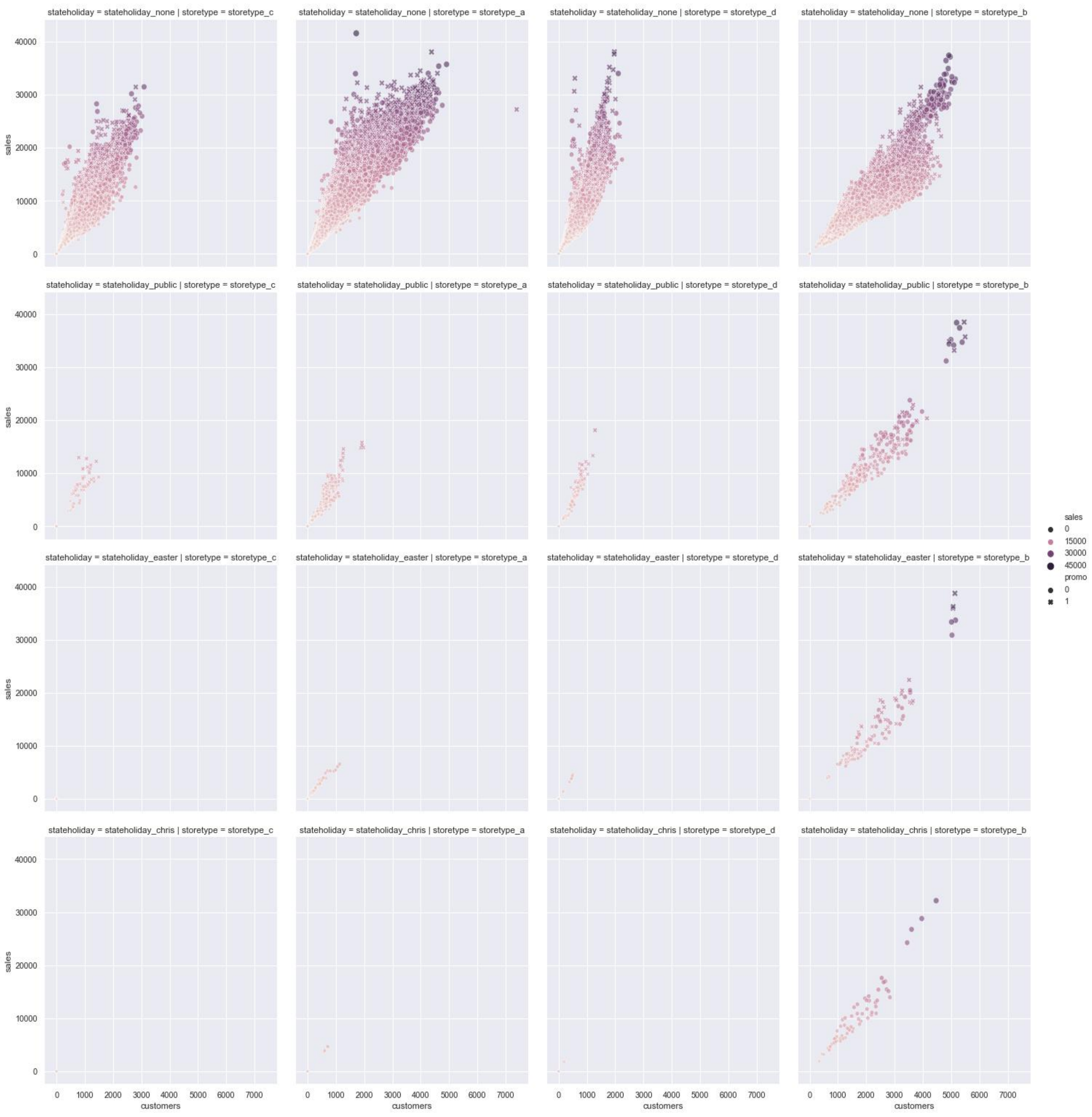


Figure 12 Scatter Plot between Sales and Customers

row = StateHoliday , column = StoreType

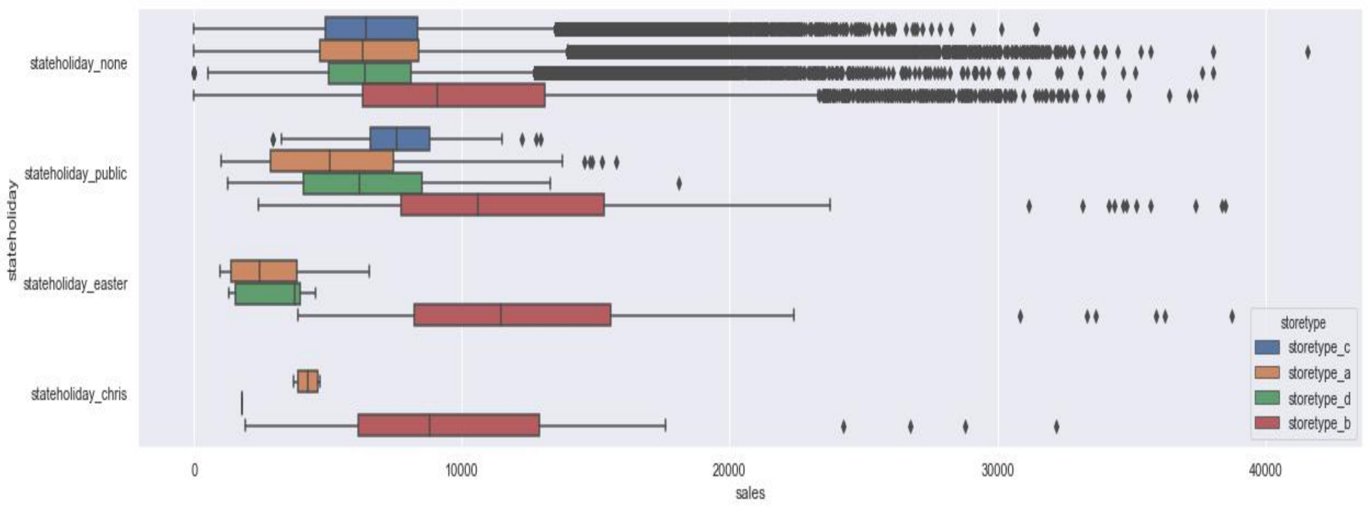


Figure 13 Box Plot of State Holiday vs Sales for Four Store Types

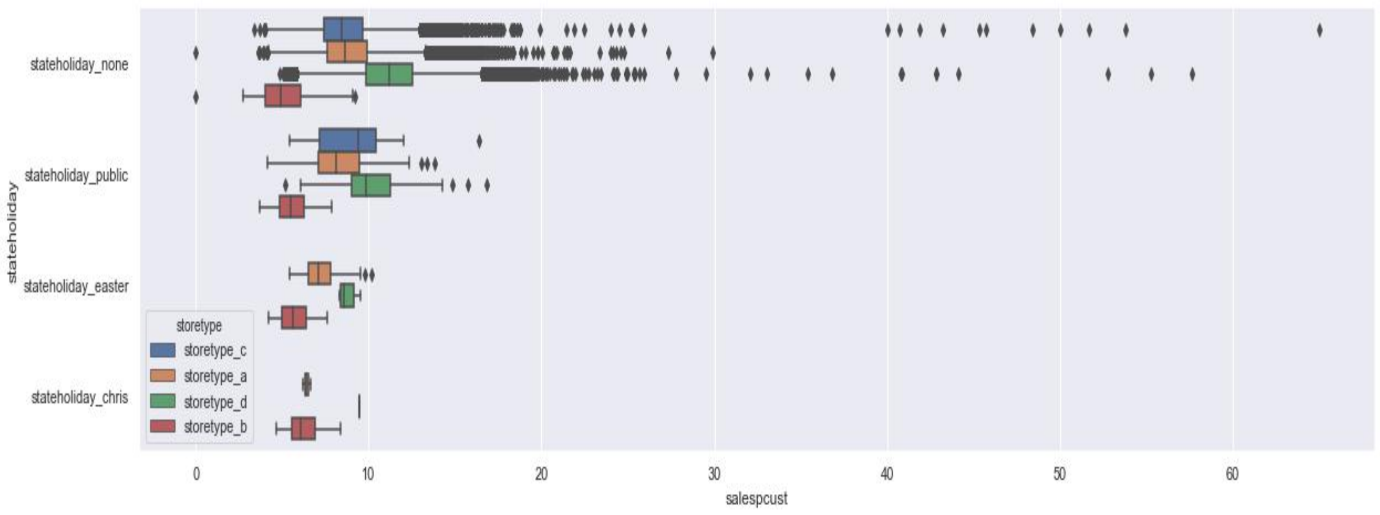


Figure 14 Box Plot of State Holiday vs Sales per Customer for Four Store Types

- SchoolHoliday vs Sales

As we can observe from figure 15 , there is no significant difference between sales in school holiday and sales in non school holiday. However, when we check the data by different state holiday, the sales of Easter holiday tends to increase significantly if it is non school holiday.

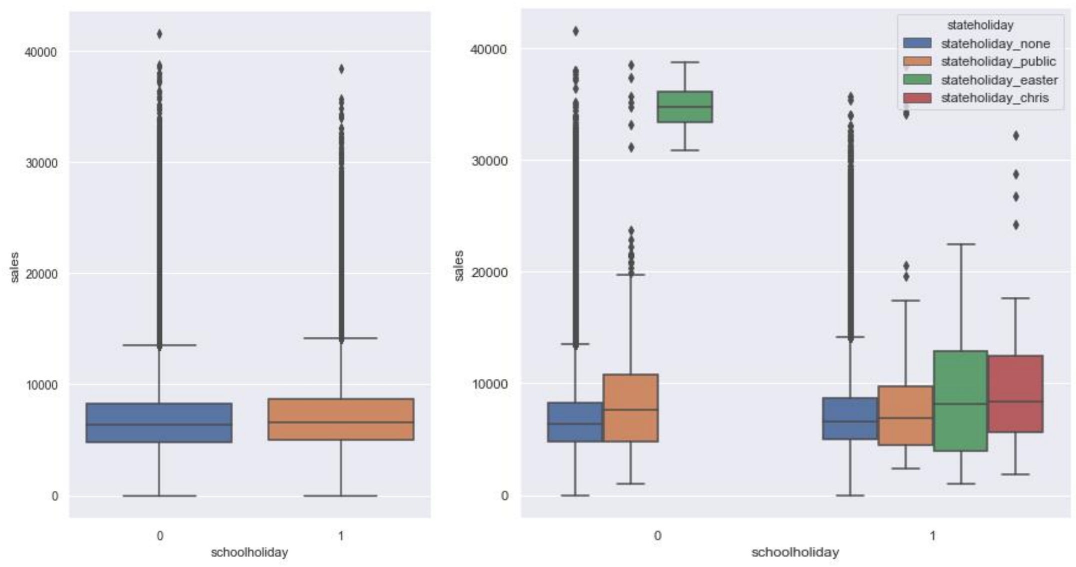


Figure 15 Box Plot of School Holiday vs Sales

- Assortment vs Sales

Figure 16 shows, only store type b has the assortment type extra, all other three store types only has basic and extended assortment. The sales of store type b with extended assortment is notably higher than basic assortment.

However, all other stores types with extended assortment tends to have slightly higher sales than basic assortment.

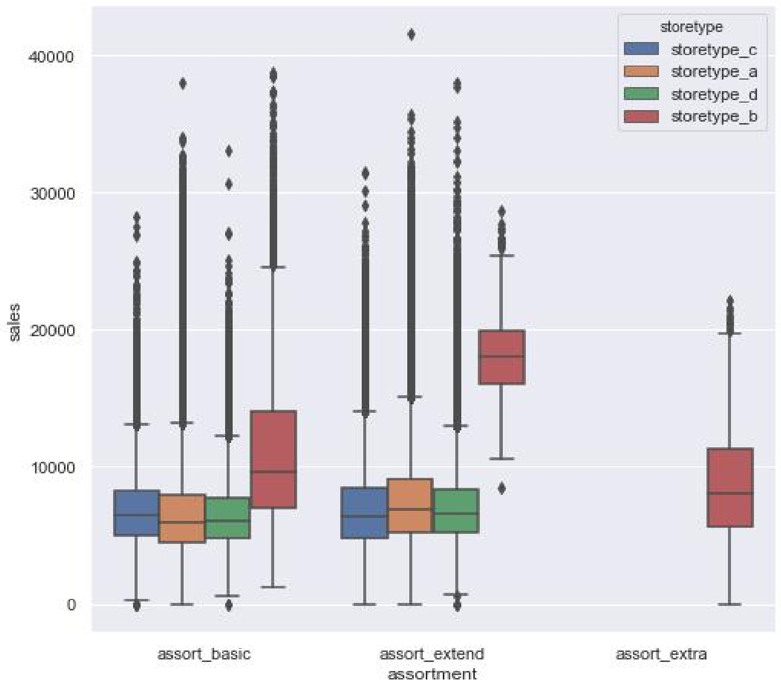


Figure 16 Box Plot of Assortment vs Sales by Different Store Types

• *Time Series Analysis vs Sales*

- Time Series Decomposition

When digging into time series analysis of training dataset, I randomly plotted 6 stores sales figure from January 2013 until July 2015. As we can see from figure 17, almost all stores sales skyrocketed in Christmas holiday.

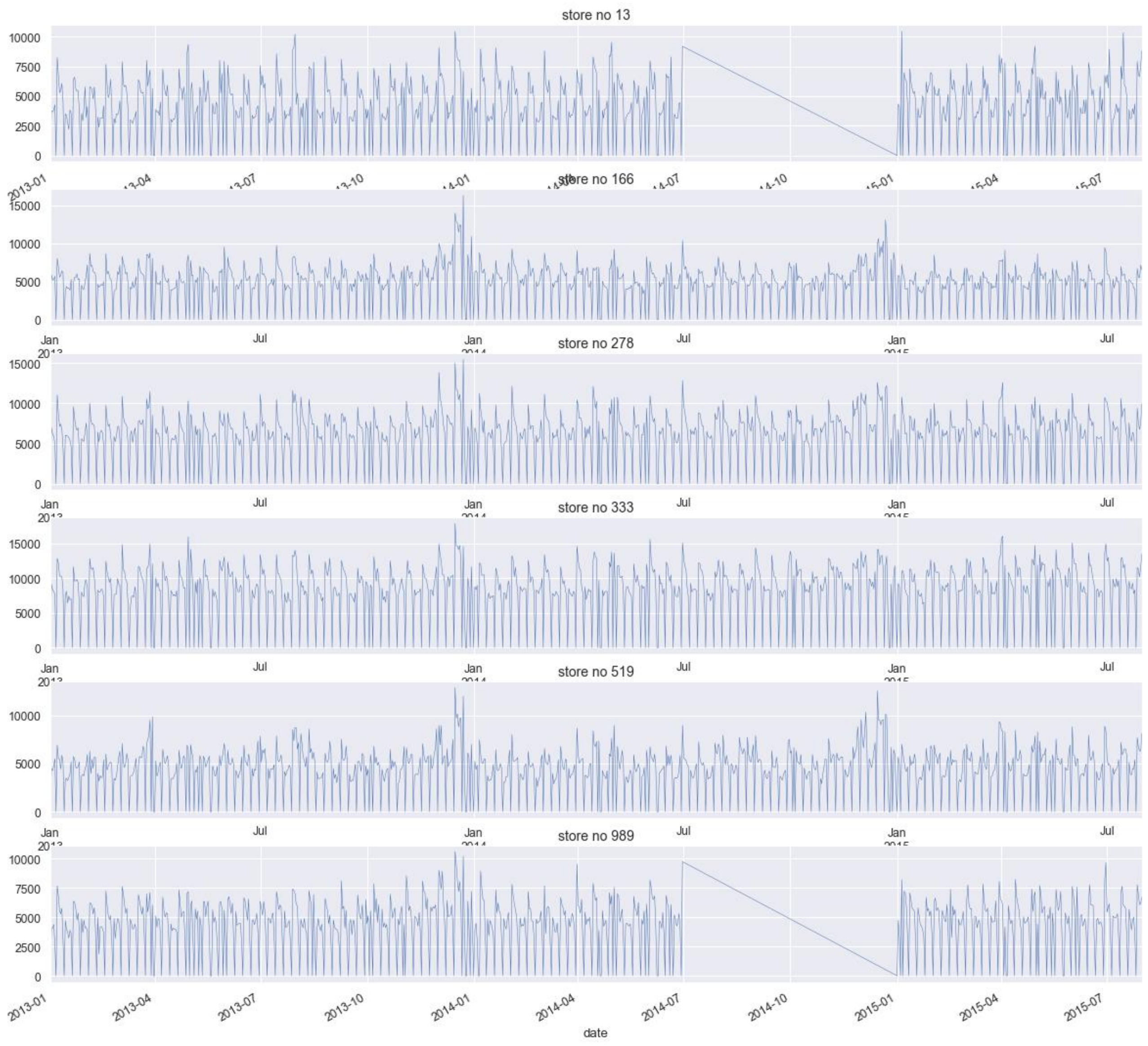


Figure 17 Time Series vs Sales Plot of Store No. 13,166,278,333,519,989

Then, I plotted 4 stores -- no.2, no. 259, no. 1, no15 which belong to four store type a, b, c, d respectively to check time series decomposition.

As we can see from figure 18 -21, The trend of 4 stores has a similar pattern , there is a steady rise of sales during the past 30 months and most of the store sales reached peak during Christmas holiday.

There are also the seasonal change patterns which are clearly showed in their seasonal plots.

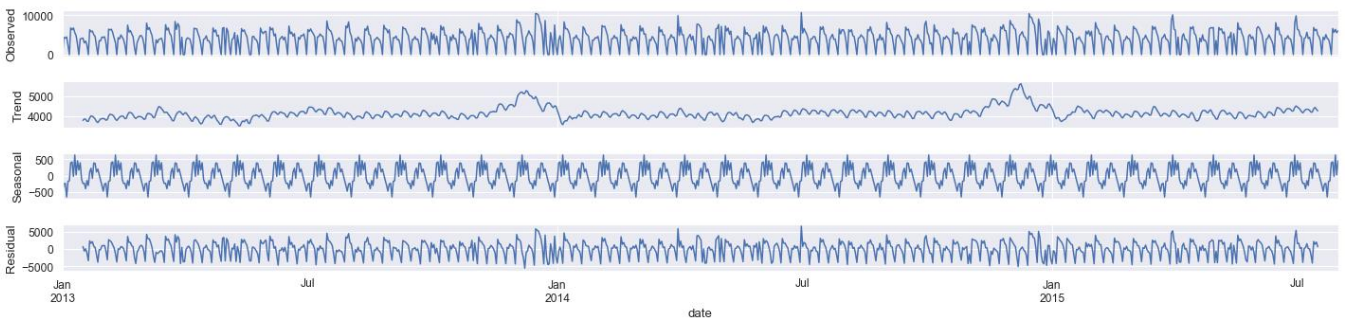


Figure 18 Time Series Decomposition of Store No. 2 (store type a)

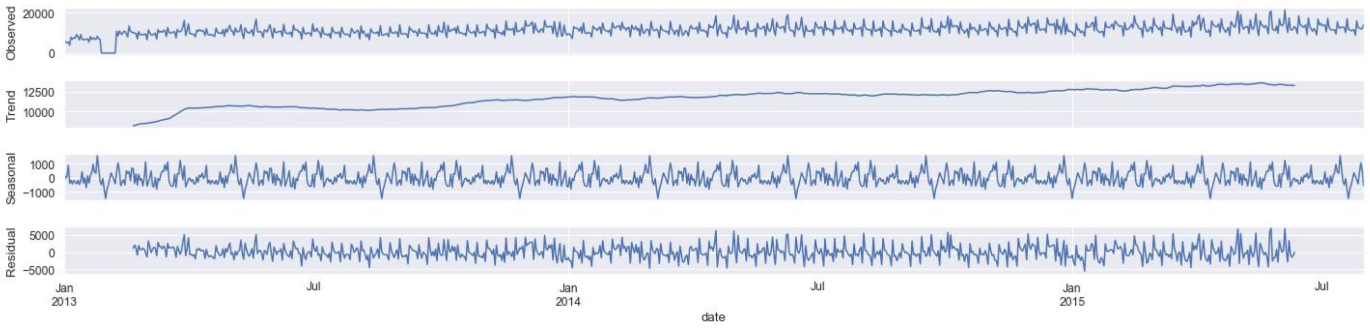


Figure 19 Time Series Decomposition of Store No. 259 (store type b)

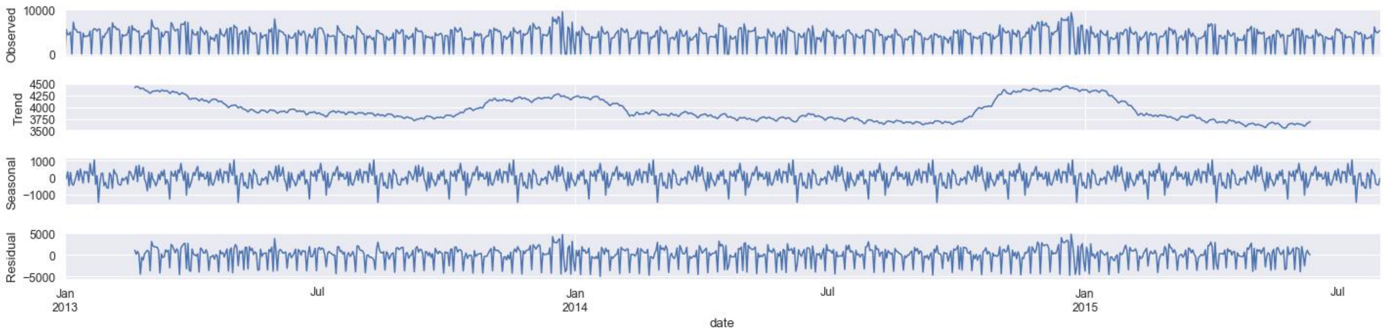


Figure 20 Time Series Decomposition of Store No. 1 (store type c)

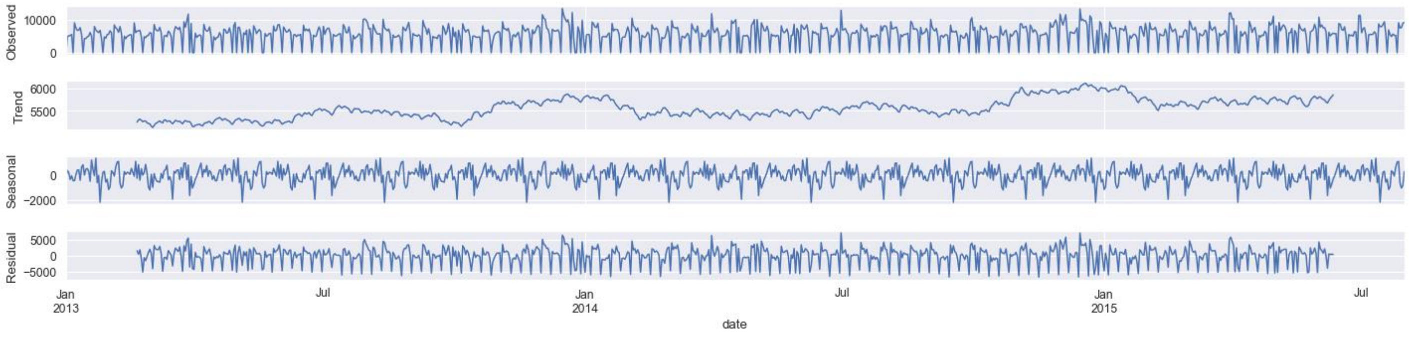


Figure 21 Time Series Decomposition of Store No. 15 (store type d)

- Time Series Heat map

In order to reveal the seasonal pattern more intelligently, I plotted the heat map of sales (mean sales per day) between day of week and week of year.

From the heat maps below (figure 22 -24) ,we could conclude:

(i) Most of stores closed on Sunday.

(ii) Unlike other store types, store type b (figure 24) is usually closed during Saturday and open on Sunday and store type b is usually open in state holiday, Easter holiday or Christmas holiday.

(iii) Monday usually get the highest sales among a week.

(iv) Saturday usually get the lowest sales among a week.

(v) There is a two weeks (14 days) sales cycle which can be observed clearly (figure 22,23) .

(vi) Christmas holiday (last two week of a year) as we analyzed above, is hottest sale season.

(vii) Four weeks before Christmas holiday tends to have higher sales than normal weeks.

(viii) Week 31st tends to be one of the highest sale week during mid year.

(ix) If the public holiday or Easter holiday is on Friday, the whole week tends to have higher sales than normal week.

(x) If there is a public holiday (or Easter holiday, Christmas holiday) on Monday, then the rest days of the same week may not perform very well.

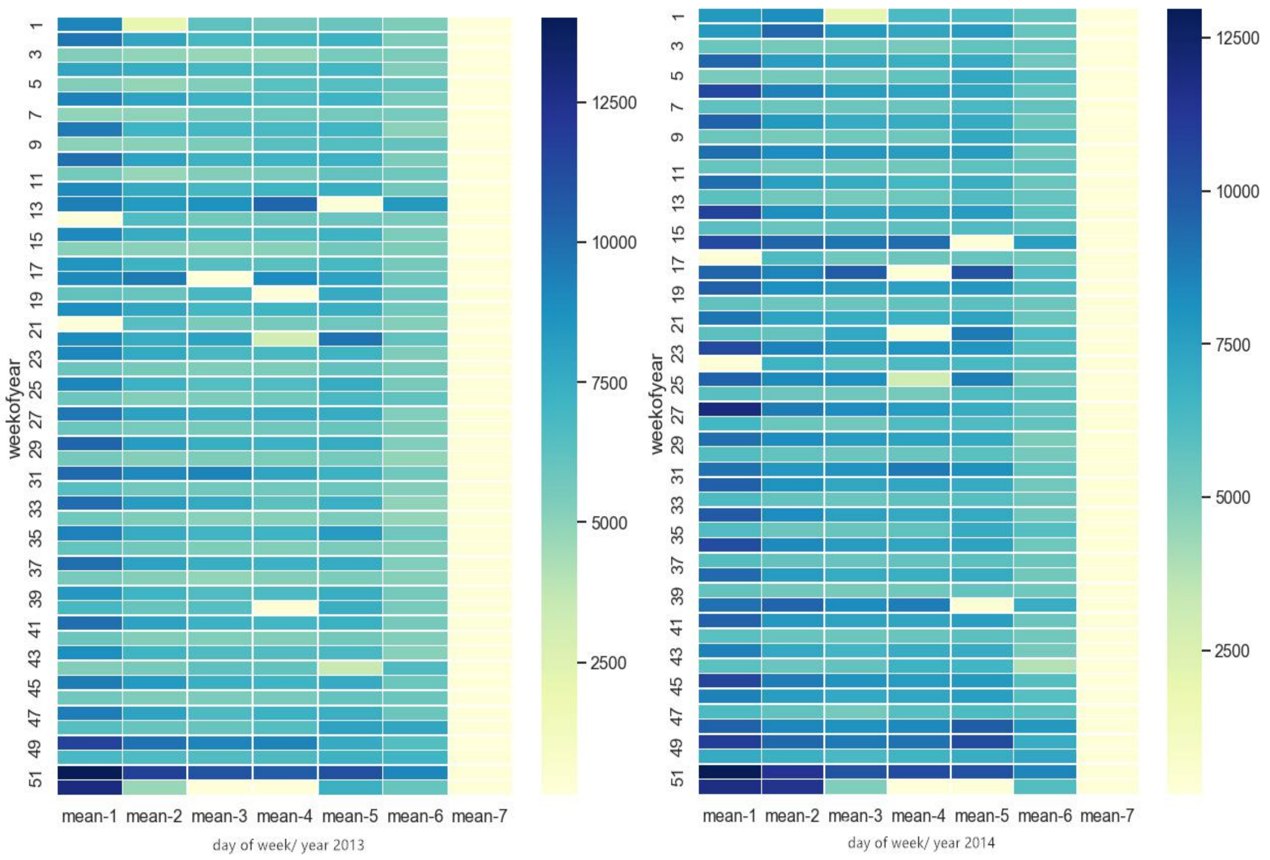
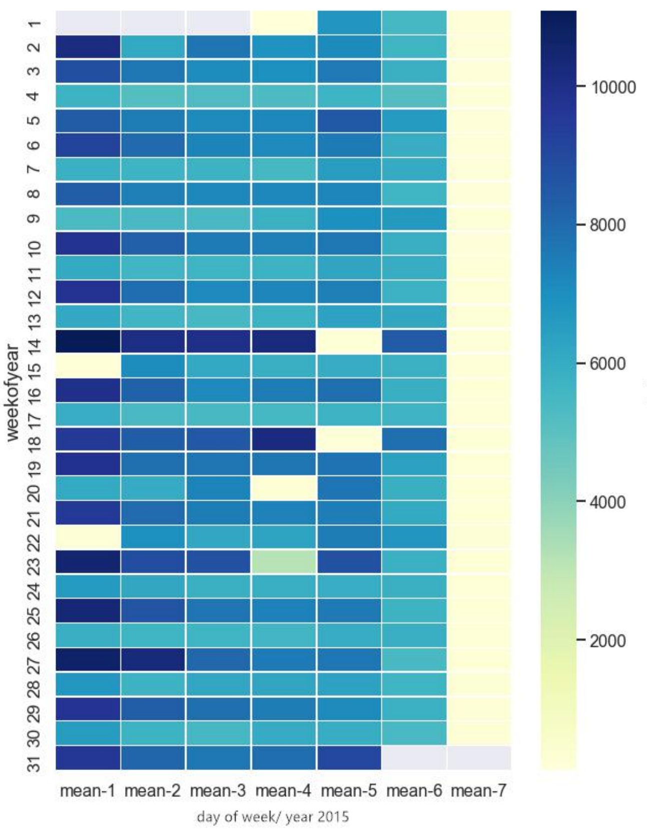


Figure 22 Time Series Heat Map (mean sales per day) of 2013, 2014



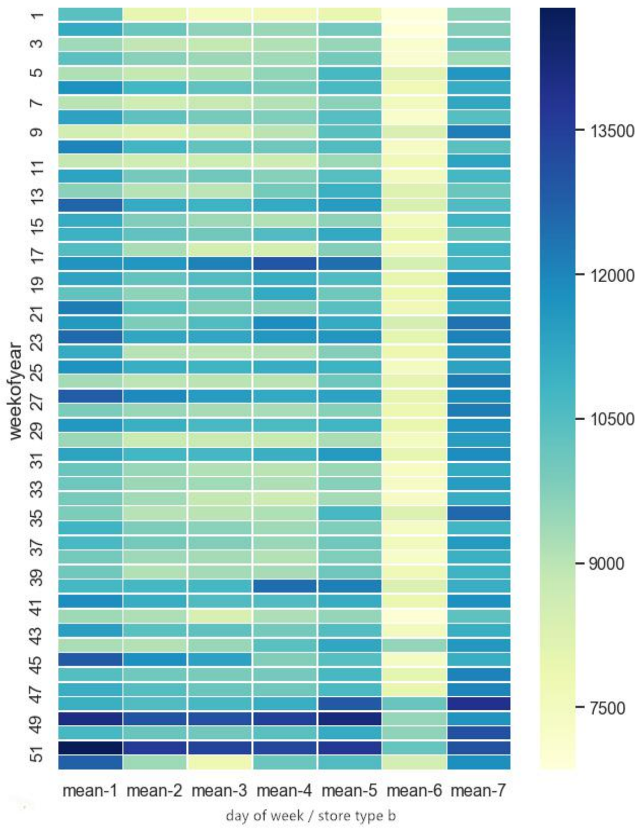


Figure 23 Time Series Heat Map
(mean sales per day) of 2015

Figure 24 Time Series Heat Map
(mean sales per day) of store type b

• ***EDA Conclusion***

(i) Numerical variable day of week, customers, open, promo and weekend has relatively stronger linear relationship with sales.
(ii) Categorical variable StoreType, StateHoliday tends to have stronger relationship with sales than Assortment , SchoolHoliday.
(iii) There is a two weeks (14 days) sales cycle which can be observed
(iv) Christmas holiday (last two week of a year) as we analyzed above, is hottest sale season.
(v) Four weeks before Christmas holiday tends to have higher sales than normal weeks.
(vi) Week 31st tends to be one of the highest sale week during mid year.
(vii) If the public holiday or Easter holiday is on Friday, the whole week tends to have higher sales than normal week. (This may be leaded by Rossmann store's sales strategy)
(viii) If there is a public holiday (or Easter holiday, Christmas holiday) on Monday, then the rest days of the same week may not perform very well.

Algorithms and Techniques

- **Algorithms:**

The main algorithm used in this project is XGBoost which is proved to be one of the most robust algorithm in most forecasting project (Synced,2017). XGBoost is also capable of dealing with missing values internally.

The objective function of XGBoost is different from other algorithms. One salient characteristic of objective functions is that they consist two parts: **training loss** and **regularization term** (Introduction to Boosted Trees, 2016):

$$\text{obj}(\theta) = L(\theta) + \Omega(\theta)$$

where L is the training loss function, and Ω is the regularization term. The training loss measures how predictive our model is with respect to the training data. A common choice of L can be the MES, RMSE or RMSPE.

The regularization term which can be easily be forgotten controls the complexity of the model, which helps us to avoid overfitting. In XGBoost, we define the complexity as (Introduction to Boosted Trees, 2016):

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

W is the vector of scores on leaves, q is a function assigning each data point to the corresponding leaf, and T is the number of leaves.

There three types of parameters in XGBoost algorithm (XGBoost parameters, 2016):

- **General parameters** relate to which booster we are using to do boosting, commonly tree or linear model.
- **Booster parameters** depend on which booster have chosen.
- **Learning task parameters** decide on the learning scenario. For example, regression tasks may use different parameters with ranking tasks.
- **Command line parameters** relate to behavior of CLI version of XGBoost.

The main hyper parameters are used in this project include:

Hyperparameter used	**Description**
"objective"	learning task parameter. [default=reg:squarederror] we use reg:linear for linear regression
"booster"	general parameter, indicate which booster to use. Can be gbtree or gblinear or dart. We use gbtree which means tree based models.

"eta"	[default=0.3] alias: learning rate, step size shrinkage used in update to prevents overfitting.
"max_depth"	range: [0,∞] [default = 6] Increasing this value will make the model more complex and more likely to overfit.
"subsample"	range: (0,1][default=1] subsample ratio of the training instances. Setting it to 0.9 means that XGBoost would randomly sample 90% training data prior to growing trees. This will prevent overfitting. subsampling will occur once in every boosting iteration.
"colsample_bytree"	colsample_bytree is the subsample ratio of columns when constructing each tree.
"silent"	[default=0] silent = 1, means silent mode, warning message is not displayed.
"seed"	seed [default=0] Random number seed.

Table 5 Hyper - parameters Used in this Project

- **Techniques:**

For this project is a time series forecasting task, to preventing from data leakage, I used time series split method from scikit-learn (Time Series Split, 2007) in cross validation (figure 25).

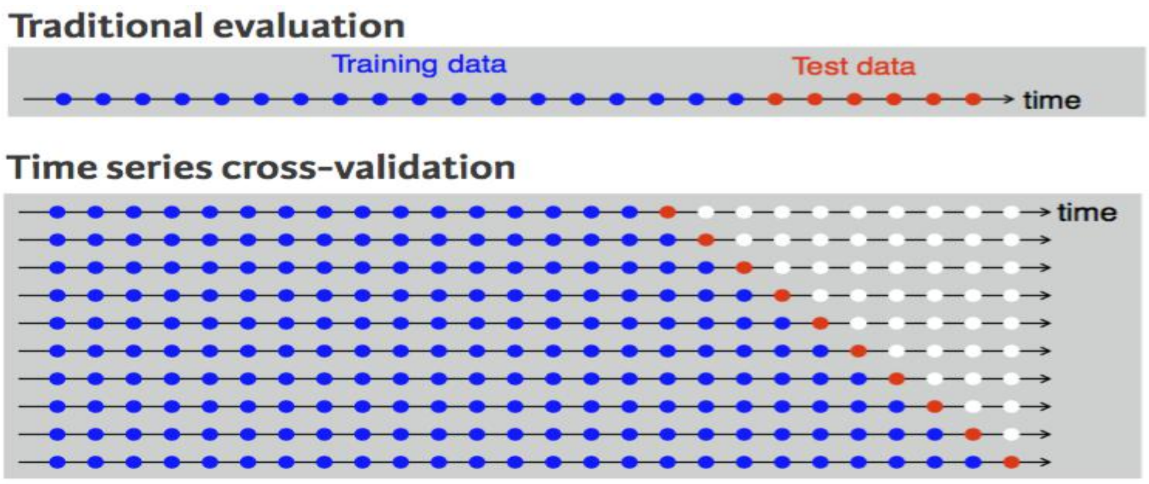


Figure 25 Traditional Evaluation Techniques vs Time Series Cross-Validation
(Rob.J & George.A, 2018)

Grid search method will be used to determine the most suitable hyper-parameter values in hyper parameter tuning part.

Benchmark

Default random forest regression model is selected as the benchmark model in this project. Random forests, also known as random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression problems. The model creates an entire forest of random uncorrelated decision trees to arrive at the best possible answer. (Raul Eulogio,2017)

I firstly split the training data to training and validation datasets (split manually first 70% of train data set as the training data, remaining 30% is the validation data). Then, I trained random forest model and checked the result for validation dataset. The RMSPE result of random forest model finally reached 0.2569.

III. Methodology

Data Preprocessing

Without relevant features, you can't train an accurate model, no matter how complex the machine learning algorithm is (Will Koehrsen . 2018). Therefore in regards to model optimization, the first and best choice is to do sound, thorough feature engineering.

Although XGBoost is able to deal with missing values internally, by logical analysis, I found some missing value in train, and store dataset can be replaced with more reasonable value. Therefore in the data preprocessing section, I dealt with missing value, incorrect value and generate relevant features according to findings of previous exploratory data analysis.

The main procedures include:

- **Data Cleaning and Preparing:**

  1. Train dataset - feature StateHoliday with typing error 0 and '0' - replace 0 with '0'.
  2. Train dataset - feature Open - set open to 0 when sales equal to 0.
  3. Test dataset - feature Open - fill 11 missing values with 1 according to Kaggle instruction.
  4. Store dataset - feature Competitiondistance (3 missing values) fill with mean.
  5. Store dataset - feature Competitionopensincemonth, Competitionopensinceyear (354 missing values) fill with mean

6. Store dataset - feature Promo2sinceweek, Promo2sinceyear,Promointerval (454 missing) - fill with 0 (since according to variable promo2, 454 stores do not have promo2)
7. Do one hot encoder to categorical variable
8. Use log function to normalize sales.
9. Reset train and test dataset index to a date ascending order. (originally is date descending).

- **Feature Generation:**

Feature generation is based on the results of EDA in part II:

Base Dataset	**New Feature Name**	**Description**
Store extended features	shopavesales	Shop average sales
	shopavesalespercust	Shop average sales per customer
	salesavgschoolholiday,	School holiday average sales per store
	shopsalesstateholiday	State holiday average sales per store
	shopsalespromo	Promo average sales per store
	shopavegsalesmon	Monday average sales per store
	shopavgsalestues	Tuesday average sales per store
	shopavgsaleswed	Wednesday average sales per store
	shopavgsalesthur	Thursday average sales per store
	shopavgsalesfri	Friday average sales per store
	shopavgsalessat	Saturday average sales per store
	shopavgsalessun	Sunday average sales per store
Open extended features	openonsunday	Open on Sunday
	openonsaturday	Open on Saturday
	openonschoolholiday	Open on School Holiday
	openonstateholidy_n	Open on non state holiday
	openonstateholidy_e,	Open on Easter holiday
	openonstateholidy_c	Open on Christmas holiday
	openonstateholidy_p	Open on public holiday
Timeseries generated features	oddweek	Odd or even week
	stateholidayweek	State holiday in this week
	stateholidayon_m	State holiday on Monday
	stateholidayon_f	State holiday on Friday
	lasttwoweeks	Last two weeks of a year

Table 6 Feature Generation of this Project

Implementation

Firstly, I defined evaluation metric - RMSPE in this project.

Secondly, I used one leave out cross validation method and random forest algorithm from scikit-learn to train the random forest benchmark model. The RMSPE result of random forest model is 0.2569.

Thirdly, I trained XGBoost model with the default parameter and one leave out cross validation method. XGBoost Algorithm reached a better RMSPE result at 0241966.

Then, I trained XGBoost model with the default parameter and 10 fold time series split cross validation. 10 fold time series split cross validation method improved the validation RMSPE result to 0.129836. However the model was overfitted for the training RMSPE is 0.172582.

Refinement

After implementing the XGBoost algorithm, I tried to tune XGBoost hyper - parameters by grid search cross validation method. However, it took me 440 minuets to just tune one parameter. Considering my hardware environment, gird search method is too time consuming and not as efficient as I assumed before.

So I decided to tune the hyper parameters manually.

Firstly I lowered eta from 0.3 to 0.06 to prevent overfitting.

This strategy works. The val - RMSPE result is 0.131657 , while train RMSPE result is 0.157331. But this result still could not meet my expectation.

Then, I set eta equal to 0.03. Unfortunately, I got a worse RMSPE (0.133083) which proved the model is not able to learn very well at eta 0.03.

So I set eta back to 0.06 and change max_depth to 10, this combination achieved a better score at 0.127738.

I also tried a larger seed number(1440), which lead to a worse result (RMSPE 0.12802).

So, I kept eta equal to 0.06, max_depth equal to 10, seed equal to 1300.

Considering model may learn more thoroughly from more cv folds, I increased cross validation fold from 10 to 20, as we can see from table 7, the validation RMSPE got boosted to 0.119428.

Then, I tried 30 folds ,40 folds and 50 folds, RMSPE reached the highest score with 50 folds . Details are showed in the table below.

Number of fold in CV	Hyper - parameters tuning	Train RMSPE	Validation RMSPE	Public Score	Private Score
10 fold	eta = 0.3 max_depth =6 (default)	0.172582	0.129836	0.12522	0.13151
10 fold	eta = 0.06 max_depth = 6, seed =1300	0.157331	0.131657	0.12914	0.12894
10 fold	eta = 0.03, max_depth = 6, seed =1300	0.187064	0.133083	0.13100	0.13200
10 fold	eta = 0.06, max_depth = 10, seed =1300	0.086111	0.127738	0.12032	0.12366
10 fold	eta = 0.06, max_depth = 10, seed = 1440	0.089597	0.12802	0.12239	0,12530
20 fold	eta = 0.06, max_depth = 10, seed =1300	0.080762	0.119428	0.11209	0.11712
30 fold	eta = 0.06, max_depth = 10, seed =1300	0.078259	0.113478	0.11165	0.11693
40 fold	eta = 0.06, max_depth = 10, seed =1300	0.0835	0.099581	0.11089	0.11725
50 fold	eta = 0.06, max_depth = 10, seed =1300	0.080867	0.100032	0.10898	0.11474

Table 7 Hyper - parameters Tuning

IV. Result

Model Evaluation and Validation

I uploaded the predictions of every model to Kaggle leaderboard. As the figures showed in table 7, the final model performed well with private score 0.11474, public score 0.10898.

The final score proved the strategy of more than 10 folds time series split cross validation is effective and efficient. By doing multiple cross validation, the robustness of XGBoost model is also enhanced.

Justification

As we can see from table 8, the final XGBoost model performs better the the Random Forest baseline model. The validation RMSPE result has been enhanced from 0.2529 to 0.100032. As I stated before, Time Series Split method from Scikit - learn prevented model from data leakage. The method of multiple folds cross validation also improved the robustness of XGBoost model.

	Algorithm	RMSPE Result
Baseline Model	Random Forest Regressor	0.2529
Final Model	XGBoost Regressor 50 folds cv	0.100032

Table 8 Baseline Model vs XGBoost Model

V. Conclusion

Free- From Visualization

Feature engineer is one of the most crucial process for sales forecasting project. But how to judge the result of feature construction ? Is it beneficial to sales predicting ?

We can take a look at feature importance by taking use of Random Forest Regressor.

As we can see from figure 26, top important features in this project include state holiday average sales (statehoavesales), shop average sales (shopavesales), promo, promo and open (promopen), promotion average sales (promoavesales) etc. More than half of the important features are derived from feature engineer which may imply that the feature engineer strategy based on EDA of this project is practical and effective.

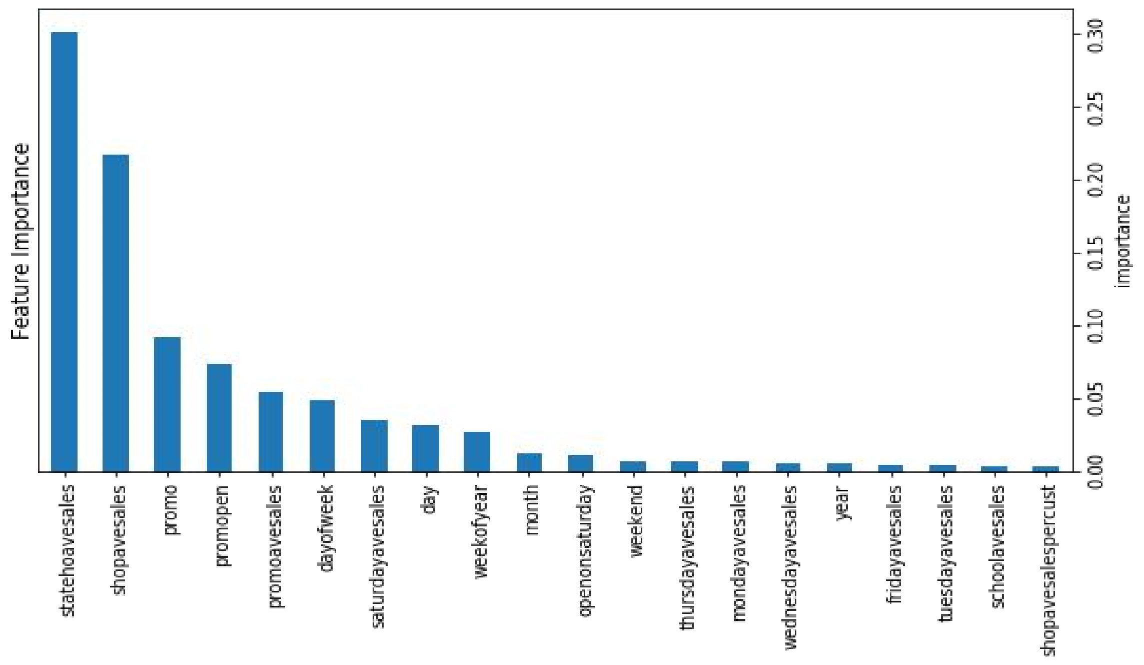


Figure 26 Top 20 Important Features for Random Forest Regressor

Reflection

- **Workflow:**

The data science team from Cdiscount France (Cdiscount Data Science, 2018) has summarized the whole pipeline (figure 27) for most of the machine learning projects. My project is mainly following the procedures they concluded below.

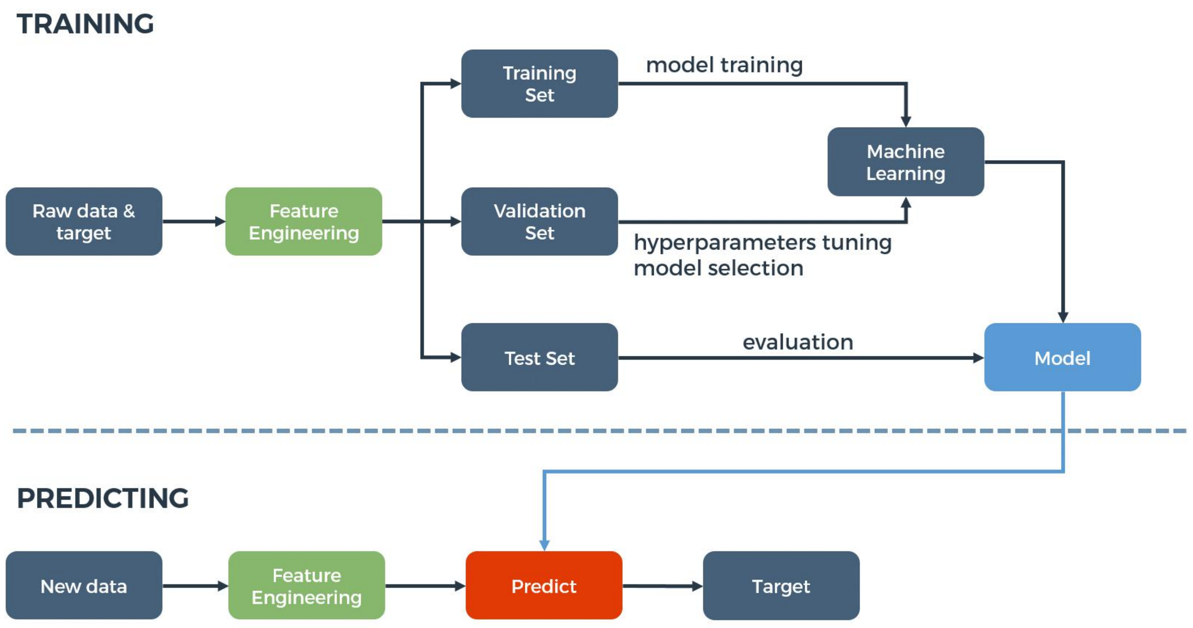


Figure 27 Workflow of Mahine Learning Project

- **Impressive Parts of the Project:**

The most impressive section in this project is data exploratory analysis. By taking use of powerful visualization tools (Matplotlib, Seaborn), we could see the underlying patterns between numerical, categorical features with target values. Especially by analyzing time series features, temporal autocorrelation between time series and target values is revealed clearly.

- **Challenge of the Project:**

The most challenging part of this project is hyper parameter tuning. Although grid search is guaranteed to find the optimal combination of hyperparamters, when come with the large number of hyper parameters (XGBoost algorithm has over 20 initial parameters), grid search can be very time consuming.

To tune hyperparameters manually can also be very tedious. One more drawback in model optimization part is tuning parameters manually may not be able to find the best combination of hyperparameters.

Improvement

For further improvements, from algorithm aspect, we could try to use Neural Network model and Lightgbm model in this project.

In the data preprocessing part, I only deleted outliers like stores open but without any sales. However the outliers of this dataset might be more complicated. It would be worthwhile to do further analysis and work out a better strategy to deal with outliers.

Finding out more relevant datasets (like weather data) to build more relevant features could also be a possible solution to model optimization.

I did not implement Grid Search method when tuning hyper - parameters for the limitation of my computer hardware,. I may try to apply this method by taking use of Google Cloud Platform in future.

If take the final model as the new baseline model. In order to improving the model performance, I may implement several powerful time series model like Neural Network model, ARIMA model, Lightgbm model , XGBoost model and taking use of ensemble method (Bagging, Boosting, Stacking) to develop an advanced robust system to achieve the best score.

.

References:
David J. Fong, PharmD (2016, Dec 19). Data Analytics: Key to Operating a Successful Pharmacy Business and Practice. [ONLINE] Retrieved May 13, 2019, from https://www.wolterskluwercdi.com/blog/data-analytics-key-operating-successful-pharmacy-business-and-practice/
Kaggle Rossmann Store Sales. [ONLINE] Retrieved May 30, 2019, from https://www.kaggle.com/c/rossmann-store-sales/overview/evaluation
Synced. (2017,Oct,23). Tree Boosting With XGBoost — Why Does XGBoost Win “Every” Machine Learning Competition? [ONLINE] Retrieved May 30, 2019, from: https://medium.com/syncedreview/tree-boosting-with-xgboost-why-does-xgboost-win-every-machine-learning-competition-ca8034c0b283.
Introduction to Boosted Trees, 2016 [ONLINE] Retrieved May 30, 2019, from: https://xgboost.readthedocs.io/en/latest/parameter.html
XGBoost parameters, 2016 [ONLINE] Retrieved May 30, 2019, from: https://xgboost.readthedocs.io/en/latest/parameter.html

Rob J Hyndman and George Athanasopoulos. (2018, April). Forecasting: Principles and Practice . Retrieved May 31, 2019, from https://otexts.com/fpp2/index.html
Scikit-learn. 2007. TimeSeriesSplit. [ONLINE] Retrieved May 30, 2019, from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html.
Raul Eulogio,2017. Introduction to Random Forests. [ONLINE] Retrieved May 31, 2019, from https://www.datascience.com/resources/notebooks/random-forest-intro
Will Koehrsen. (2018. Nov 11) Feature Engineering: What Powers Machine Learning. [ONLINE] Retrieved May 31, 2019, from https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d
Cdiscount Data Science (2018, May, 25).A brief overview of Automatic Machine Learning solutions (AutoML) [ONLINE] Retrieved May 30, 2019, from: https://hackernoon.com/a-brief-overview-of-automatic-machine-learning-solutions-automl-2826c7807a2a