Wal-Mart Sales Prediction using XGBoost Algorithm

CAI Zhuoheng	20659895
HAN Yan	20639584
LI Lingpan	20652108
LI Yiling	20660284

Abstract

Wal-Mart Department Store is a worldwide chain company in the United States, which is the largest company in the world in terms of turnover. Therefore, we chose the project of predicting Wal-Mart sales in the kaggle competition.

Our project starts from the analysis on the data, which include time series data consisting of 3049 items sold in 10 stores in 3 states and the explanary variables that could give us the information of calendar, price, quantity and item-specific attribute, like which state the store belongs to. The total length of each time series is 1969, but we could only get 1913 (1969-28*2) observations of sales amount. Since we are going to predict 28-day ahead future values of sales amount, we separate the last 28 days of training dataset as validation set.

Based on the information of calendar, price, etc, we use some simple feature enginering to generate numeric features which can be regarded as the parameter in XGBoost model (eXtreme Gradient Boosting) to predict 28-day ahead sales amount time series data.

Keywords: Wal-Mart sales, feature enginering, XGBoost Model (eXtreme Gradient Boosting)

Introduction

In this article, we use XGBoost model (eXtreme Gradient Boosting), introduced by Chen (2015) to predict 28-day ahead sales amount time series data. Xgboost (eXtreme Gradient Boosting) is a tool for massively parallel boosted tree. The Xgboost algorithm evolves from the following algorithms: Bagging to Boosting, Boosting to Gradient boosting (GB), Gradient boosting (GB) and Decision Tree (DT) to Gradient Boosting Decision Tree (GBDT), Gradient Boosting

Decision Tree (GBDT) evolved to eXtreme Gradient Boosting (Xgboost).

The concept of Bagging is to randomly extract from the training data (after pumping back, called bootstrap), create multiple sets of training samples, train multiple sets of classifiers (you can set how many classifiers you want), the weight of each classifier Unanimity finally obtains the final result through Majority vote.

The concept of Boosting is to establish M models (such as classification) for a piece of data. The generally selected model is relatively simple. It is called a weak classifier (weak learner). Each classification will increase the weight of the last error data and then proceed Classification, so that the final classifier can get better results on both test data and training data.

Gradient Boosting is a Boosting method. The main concept is to let the loss function continue to decline every time the model is built, which means that the model is constantly improving. Gradient).

Decision Tree is carried out using the CART (Classification and Regression Trees) algorithm, and the steps performed are the following two steps: On one hand, decision tree generation: The process of constructing a binary decision tree recursively. A decision tree is generated based on the training data set. The generated decision tree should be as large as possible. Classification problems can use GINI, double or ordered double; regression problems can use least squares deviation (LSD) or least absolute deviation (LAD). On the other hand, decision tree pruning: Use the verification data set to prun the generated tree and select the optimal subtree. At this time, the minimum loss function is used as the standard for pruning.

GBDT is a combination of the two cores of GB + DT (whether it is a classification or regression problem). The core of GBDT lies in that each tree learns the residual of the sum of all previous tree conclusions. This residual is a cumulative amount that can be obtained after adding the predicted value.

Xgboost is one of the GBDTs and can also be applied to classification and regression problems. Xgboost has the following advantages: First, traditional GBDT uses CART as the base classifier, xgboost also supports linear classifiers, at this time xgboost is equivalent to logistic regression (classification problem) or linear regression with L1 and L2 regularization (Regression problem). Second, the traditional GBDT uses only the first derivative information when optimizing, and xgboost performs the second-order Taylor expansion of the loss function, and uses the first and second derivatives to improve the loss function. In addition, xgboost adds regular terms to the loss function to control the complexity of the model. The regular term contains the number of leaf nodes of the tree and the sum of squares of L2 of the score output on each leaf node. From the perspective of Bias-variance tradeoff, regularization reduces the model variance, makes the learned model simpler, and prevents overfitting. This is also a feature of xgboost over traditional GBDT.

1. Data description

1.1 Background

On one hand, we are given the grouped unit sales data, starting at the product-state level: the States of California (CA), Texas (TX), and Wisconsin (WI). And each state has different number of stores. Specifically, there are four stores in the CA state, and both the TX and WI states have three stores. In addition, every store has three categories: Hobbies, Foods, and Households. Among three categories, both hobbies and household have two departments in their categories while food categoty has three departments. And each department has many items, the number of items in each department is different. On the other hand, besides the time series data, we are given the explanatory variables such as sell prices, promotions, days of the week, and special events (e.g. Super Bowl, Valentine's Day, and Orthodox Easter) that typically affect unit sales and could improve forecasting accuracy.

1.2 Data Charateristics

There are many datasets as mentioned above, and we use three of them to plot the sell price during different periods.

Datasets:
calendar.csv contains information about product sales dates.
sales_train_validation.csv contains historical daily unit sales data for each product and store
sell_prices.csv contains information about the prices of products sold at each store and date.

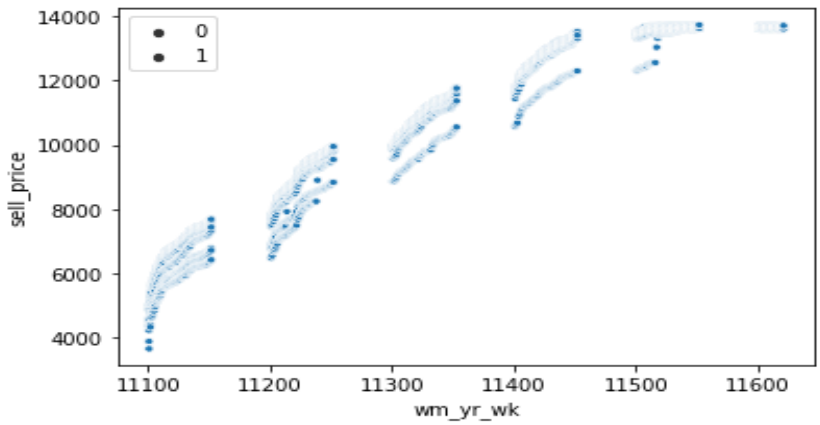

Figure 1: Sell price during different periods

Figure 1gives sell price during different periods, which shows two apparent features. On one hand, the trend of sell price is increasing. On the other hand, some datapoints for sell price are missing in some particular time points, which may appear in some holidays.

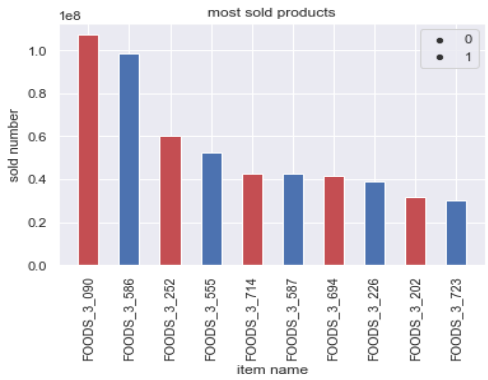

Figure 2: The top 10 best sold products

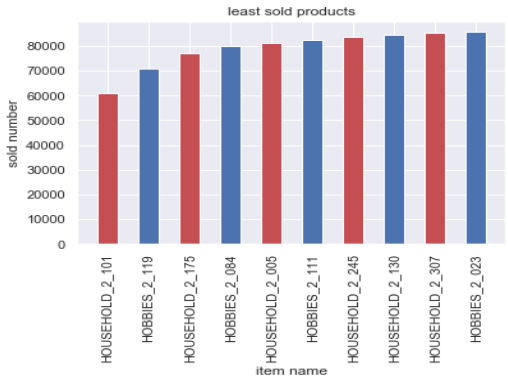

Figure 3: The least 10 sold products

Figure 2 and figure 3 are the top 10 best sold products and the least 10 sold products respectively. It can be easily found that all the listed products in figure 2 come from food department, and almost half of the least 10 sold products come from household department while the remaining half come from hobbies department.

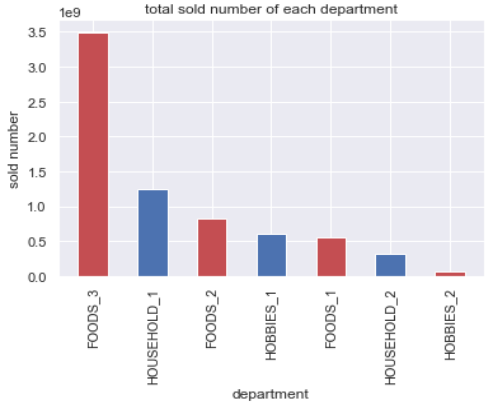

Figure 4: The sold number of each department

Figure 5: The sold number of each store

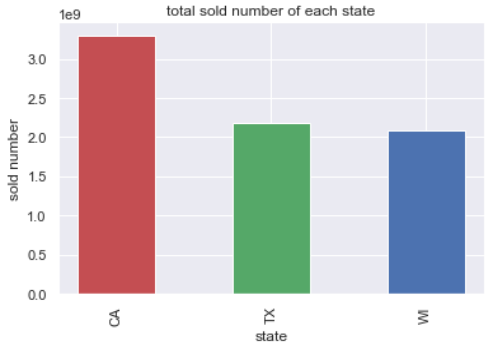

Figure 6: The sold number of each state

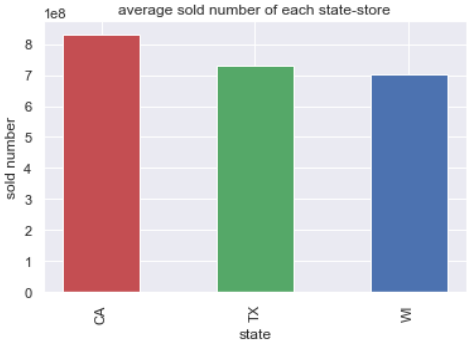

Figure 7: Average sold number of each state-store

Figure 4, figure 5 and figure 6 are the sold number of each department, each store and each state respectively. Figure 7 is the average sold number of each store in each state. From figure 4, we can find that that FOODS_3 department has an overwhelming sales number, while the HOBBIES_2 has the least total sold number. At the same time, figure 5 indicates that CA_3 store has the largest sold number. Figure 6 and 7 tell us that CA state accounts for the most total sold products number while TX state and WI state bear the similar sold number.

2. Sales Prediction

2.1 Model description

We use the XGBoost model, introduced by Chen (2015) to predict 28-day ahead sales amount time series data. The model of XGBoost encompasses K independent trees in an additive manner. For a given data set with n examples and m features,

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$$

in which $q: \mathbf{R}^{\dim(\mathbf{x})} \to T$. And in this paper(Chen, 2015), the author includes *regularized objective* in Loss function*:*

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \qquad \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

And our goal is to minimize the total loss.

2.2 XGBoost Sales Prediction

Due to the limit of computing power, we finally choose 1000 time series of 30490 time series to do the prediction. We adopt root mean square error (RMSE) as our loss gauge. In order to mine more information from existing data, we use some simple feature engineering techniques, which include three aspects. Firstly, label encoding, which is mainly used for item-specific features and calendar features. Secondly, mean encoding, using to generate average sales amount of items categorized by state, store, department, category and so on. Thirdly, trend encoding, which is a percentage change of price of past 28 days. The generated features are the follows:

	Features	Details
item specific	item_id	The id of the product
	dept_id	The id of the department
	cat_id	The id of the category
	store_id	The id of the store where the product is sold
	state_id	The State where the store is located
calendar	d	date
	sales_cnt_daily	daily sales amount
	wday	number of day in week
	month	month
	year	year
	event_type_1	type of event
	event_type_2	type of 2nd event (if exists)
	snap_CA	whether SNAP in California exists
	snap_TX	whether SNAP in Texas exists
	snap_WI	whether SNAP in Wisconsin exists
	event_num	total number of events
price	sell_price	weekly average price
	price_chg_pct	percentage price change
quantity	state_store_avg_sales_cnt_lag_28	28-day lagged average sales amount grouped by state and store
	store_cat_avg_sales_cnt_lag_28	28-day lagged average sales amount grouped by store and category
	cat_dept_avg_sales_cnt_lag_28	28-day lagged average sales amount grouped by category and department
	dept_item_avg_sales_cnt_lag_28	28-day lagged average sales amount grouped by department and item

Table 1: Generated features

As can be seen from the table, we have generated four categories of features. The first one is about items in different stores in different states, and there are a total of 5 features in item specific category. The second one is about calender, there are 11 features in category calender. Meantime, 2 features in category price, and 4 features in category quantity. What each feature represents is shown in the details column.

We use the generated features as mentioned above as input variables, using the XGBoost model to obtain the predicted sales values for the next 28 days.

id	F1	F2	F3	F4	F5	F6	F7	F8	...
HOBBIES_1_001_CA_1_validation	0.1793	0.2078	0.2078	0.2075	0.4401	0.3545	0.3054	0.1963	…
HOBBIES_1_001_CA_2_validation	0.5081	0.5366	0.5366	0.5363	0.6017	0.7603	0.5739	0.5100	…
HOBBIES_1_001_CA_3_validation	0.4199	0.4326	0.4326	0.4323	0.6543	0.6284	0.5346	0.4369	…
HOBBIES_1_001_CA_4_validation	0.2350	0.2476	0.2476	0.2473	0.2157	0.2671	0.2425	0.2487	…
HOBBIES_1_001_TX_1_validation	0.1362	0.1506	0.1506	0.1502	0.1824	0.2598	0.2449	0.1461	…
HOBBIES_1_001_TX_2_validation	0.0940	0.0865	0.0865	0.0882	0.1439	0.3552	0.2809	0.0971	…
HOBBIES_1_001_TX_3_validation	0.2095	0.2020	0.1943	0.1960	0.2254	0.3106	0.2364	0.2126	…
HOBBIES_1_001_WI_1_validation	0.2108	0.2234	0.2307	0.2324	0.3665	0.4610	0.4528	0.2285	…
HOBBIES_1_001_WI_2_validation	0.1327	0.1679	0.1698	0.1714	0.2125	0.1907	0.2111	0.1680	…
HOBBIES_1_001_WI_3_validation	0.1364	0.1490	0.1592	0.1609	0.2139	0.1884	0.2355	0.1542	…
HOBBIES_1_002_CA_1_validation	0.2024	0.2073	0.2073	0.2070	0.2677	0.3075	0.2707	0.2149	…
HOBBIES_1_002_CA_2_validation	0.3432	0.3481	0.3481	0.3478	0.4085	0.5252	0.3705	0.3407	…
HOBBIES_1_002_CA_3_validation	0.4834	0.4724	0.4724	0.4721	0.5223	0.6464	0.5743	0.5054	…
HOBBIES_1_002_CA_4_validation	0.1023	0.0914	0.0914	0.0911	0.1152	0.0914	0.0985	0.1115	…
HOBBIES_1_002_TX_1_validation	0.1467	0.1610	0.1610	0.1607	0.1947	0.2704	0.2883	0.1554	…
HOBBIES_1_002_TX_2_validation	0.1479	0.1404	0.1404	0.1421	0.2028	0.4213	0.3491	0.1477	…
HOBBIES_1_002_TX_3_validation	0.2093	0.2017	0.1940	0.1957	0.2341	0.3293	0.2571	0.2090	…
HOBBIES_1_002_WI_1_validation	0.4607	0.4700	0.4773	0.4790	0.5397	0.7235	0.6743	0.4773	…
HOBBIES_1_002_WI_2_validation	0.2022	0.2289	0.2308	0.2325	0.2733	0.3425	0.3650	0.2257	…
…	…	…	…	…	…	…	…	…	…

Table 2: Predicted value in 28 days

Table 2 shows the predicted value in 1912-1941 time series, because of page limitation, we only took screenshots of the forecast from the first day to the eighth day. In terms of accuracy, the RMSE of training and validation set are 1.71648 and 1.75823, respectively.

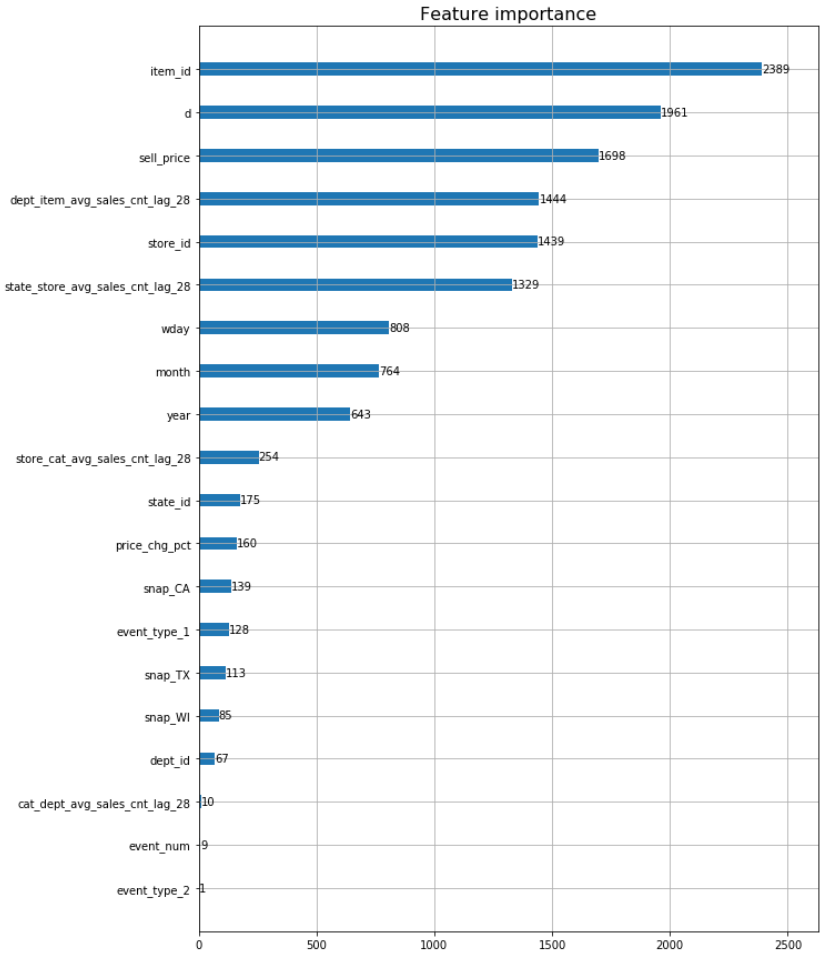

Figure 8: Feature importance

From the outcome of feature importance from the model. It can be seen that the top 6 features are: item_id, d, sell-price, dept_item_avg_sales_cnt_lag_28, store_id and store_cat_avg_sales_cnt_lag_28. To be specifically, firstly, item_id and store_id locates in item specific category, and they represent the id of the profuct and the id of the store where the product is sold respectively. Secondly, d represents date, and it is in the calender category. Thirdly, sell-price is the the weekly average price, locating in the price category. In addition, dept_item_avg_sales_cnt_lag_28, and store_cat_avg_sales_cnt_lag_28 are in quantity category, and they represents 28-day lagged average sales amount grouped by department and 28-day lagged average sales amount grouped by store and category.

Reference

Chen,T.,He,T.,Benesty,M.,Khotilovich,V.,&Tang,Y.(2015).Xgboost:extreme gradient boosting

Friedman, Jerome, Tibshirani, Robert, Hastie, & Trevor. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann. Statist.

Work Contribution

CAI Zhuoheng: Mainly responsible for code and write 1/3 report
HAN Yan: Presentation representative. Revise and audit the ppt slides.
LI Lingpan: Contribute to writing 1/3 report and make ppt slides.
LI Yiling: Mainly responsible for code and write 1/3 report

Link of presentation video

https://www.bilibili.com/video/BV1Tk4y1r774

Link of code

https://github.com/charlesczh/6010U-final