Springboard

Springboard Capstone Project II

Walmart sales forecasting

Part I: Annual sales forecasting with regression models

Qiwei Lu
OCT 2020

Table of Contents

1. Introduction

For small business owners, getting their products into Walmart is a triumph just like winning retail Olympic gold. Even for big companies, the sales to Walmart alone can make up 20% of the gross annual sales. Although you can see thousands of different products at Walmart, putting your products on one of the shelves can be a long way of application and negotiation.

Walmart operates under one of the most technologically advanced, and efficient, supply chain management systems in existence. For over 40 years, Walmart suppliers are responsible for managing inventory in Walmart's warehouses, which constituted a vendor-managed inventory system. The results are a smoother flow of inventory, less irregularities and availability of products on the shelves when requested by customers.

To become a supplier, the production company applying has to be able to have full control on how many products will be manufactured in the coming year so more stock will be shipped whenever the inventory hits the restock line in Walmart's vendor system.

This research is aimed for the potential and current suppliers of Walmart, who could use the information they have for the next year, such as price and selling store location to estimate the production volume in the future.

2. Methodology

2.1 Data Acquisition

The dataset is from 5th Makridakis forecasting competitions (M5) on Kaggle. The dataset contains 30,490 hierarchical time series data. The data were obtained in the 3 US states of California (CA), Texas (TX), and Wisconsin (WI). "Hierarchical" here means that data can be aggregated on different levels: item level, department level, product category level, and state level. The sales information is from Jan 2011 to June 2016. In addition to the sales numbers, there are also corresponding data on prices, promotions, and holidays.

The data comprises 3049 individual products from 3 categories and 7 departments, sold in 10 stores in 3 states. The hierarchical aggregation captures the combinations of these factors.

This research is the part I of a two-part project and will not work with the time element in the data. Instead of predicting daily sales for 28 days of each product, I will only use the same dataset to predict the annual sales volume for each product, which makes it a simple regression problem. I will add in the time element and use tools for time series data in part II.

2.2 Data Structure Overview

There are three csv files provided for the competition. They are all well-arranged and contain no missing data. The final data to be fitted in the models, however, has to be extracted from the three dataframes, based on our understanding of the data provided.

The first file ***"sales_train.csv"*** contains the historical daily unit sales data per product and store. Each row shows the daily sales for each item from each store, starting at the product-store level and being aggregated to that of product departments, product categories, stores, and three geographical areas: the States of California (CA), Texas (TX), and Wisconsin (WI).

The second file ***"sell_prices.csv"*** contains information about the price of the products sold per store and date. The price of the product for the given week/store. The price is provided per week (average across seven days). If not available, this means that the product was not sold during the examined week. Note that although prices are constant on a weekly basis, they may change through time (both training and test set).

The most important use of this dataframe is that it is the only place to know if an item is sold in the store or not. You cannot find it in the first file because the daily sales will only show 0, not indicating if there is no inventory or there is no one buying.

The third file ***"calendar.csv"*** contains information about the dates the products are sold. It acts as a bridge between the first file and the second file. In the first file, the date is shown as d_1, d_2, …, d_i, … d_1941: The number of units sold at day i, starting from 2011-01-29. But the time of the weekly price

in the second file is shown as 'wm_yr_wk'. You can transform the data structure in this calendar file. The third file also contains if a certain date is holiday and if SNAP is allowed in certain states, but for this research, since we are predicting on the annual basis. This information is not necessary.
To understand the structure of all the information given is very important. Luckily there is a clear explanation of what is going on each aggregation level from the M5 official competition guide.

Table 1. Number of M5 series per aggregation level

Level id	Aggregation Level	Number of series
1	Unit sales of all products, aggregated for all stores/states	1
2	Unit sales of all products, aggregated for each State	3
3	Unit sales of all products, aggregated for each store	10
4	Unit sales of all products, aggregated for each category	3
5	Unit sales of all products, aggregated for each department	7
6	Unit sales of all products, aggregated for each State and category	9
7	Unit sales of all products, aggregated for each State and department	21
8	Unit sales of all products, aggregated for each store and category	30
9	Unit sales of all products, aggregated for each store and department	70
10	Unit sales of product *x*, aggregated for all stores/states	3,049
11	Unit sales of product *x*, aggregated for each State	9,147

12	Unit sales of product *x*, aggregated for each store	30,490
Total		42,840

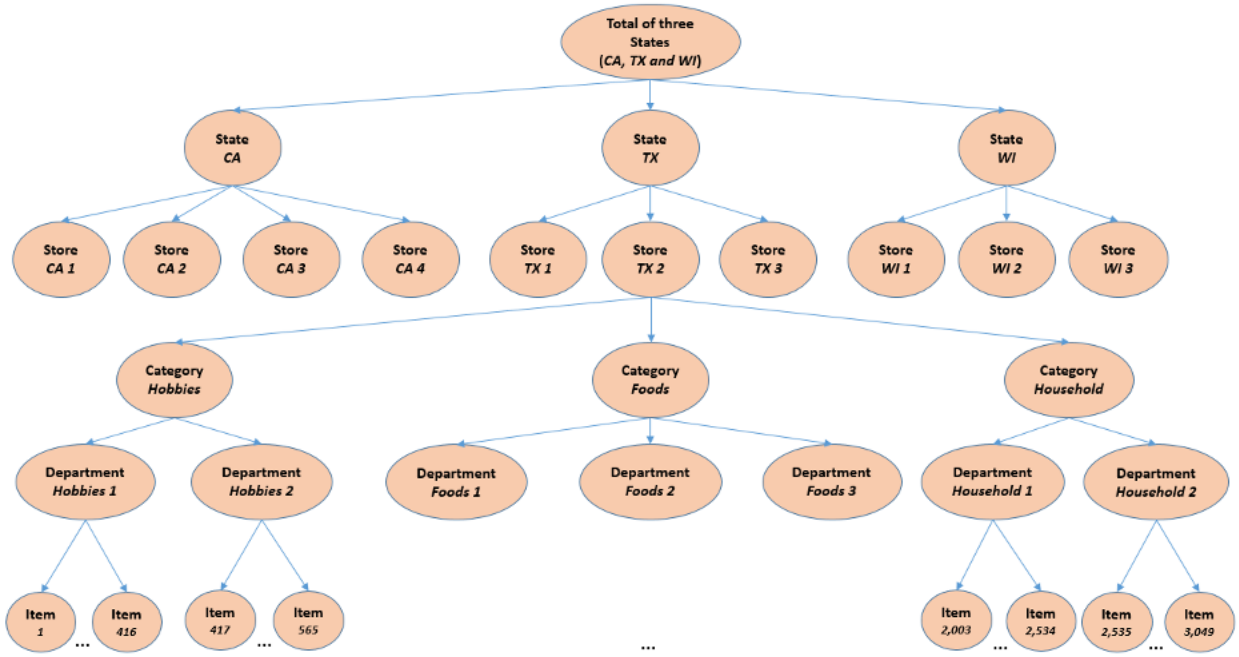


Chart 1 Data structure on different levels

Table and chart are from M5 competition official guide

2.3 Assumptions

After taking a quick look at the data, we realized there is limited explanatory information for the data given and we have to make some reasonable assumptions to make the full use of the data given and make the models more practical for use.

A. About SNAP: The products come from three categories: food, hobbies and household. In the calendar dataset, weather SNAP is allowed to use has been indicated on each day. By rules, the SNAP cards cannot be used on hobbies or household items. Foods like prepared foods fit for immediate consumption (pizza in the food court) or hot foods are not allowed either. For the convenience of the research, we will assume the

products under the food category are all qualified to be purchased by SNAP cards.

B. About sale of '0': when you see a sale of '0' in the data for this research, there are two scenarios: one is that there is inventory but no customer bought it for that day; the other is that the item is not sold in that store. These two scenarios can be distinguished by finding the price given for that item on that day. If a price is given, then sale of '0' means scenario number one. If not, scenario number two.

   However, in this non-perfect world, sometimes the product is sold in the store but there is no inventory or the last pieces have been damaged. Or there can be chances that there is no item on the shelf, but some inventory lies in the store warehouse. Because Walmart is usually understaffed, restocking the shelf can take some time and rarely will any customer ask the staff to find items in the warehouse. If this is the case, then there will be a price given because the item is indeed sold in the store but the customer cannot buy it because of the inventory problem. For the convenience of this research, I will assume that these cases won't happen and that if a customer wants to buy a product at a store where the item is sold, there should always be inventory.

C. About when the price is set: There is limited information on how Walmart suppliers determine their prices. We will assume that the prices are set on the previous year end and the suppliers know if their products will be sold at each store by that time as well. Making this assumption makes sure that suppliers can use this research to fit in the planned price for the next year along with other information and get the prediction on the following year's sale number.

2.4 Modeling Approach

There are 5.4 years of data provided and we will work with the year 1-5. Information will be gathered from all the files: sales, price and calendar. After data wrangling, the final dataframe we will use is a mix of continuous data and categorical data. Categorical data will be hot coded and several

models will be used to explore the relationship between the predictors and the target:

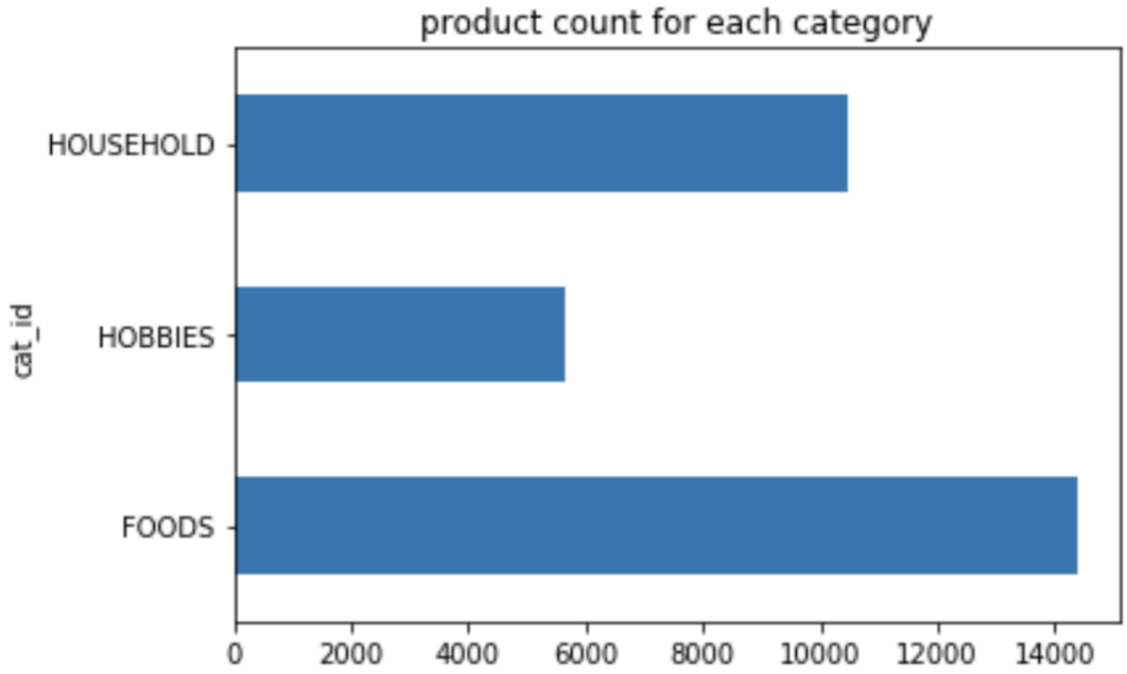
3. Data Exploration

After we are aware of what information we have on hand, we will do some exploratory data analysis on each file and see the trend and pattern on the product-level, store level and state level. We know our goal is to predict the annual sale of each item across all the stores in all the states so we would be careful when picking our features to assemble our training data.

3.1 Analysis on "***sales_train.csv***"

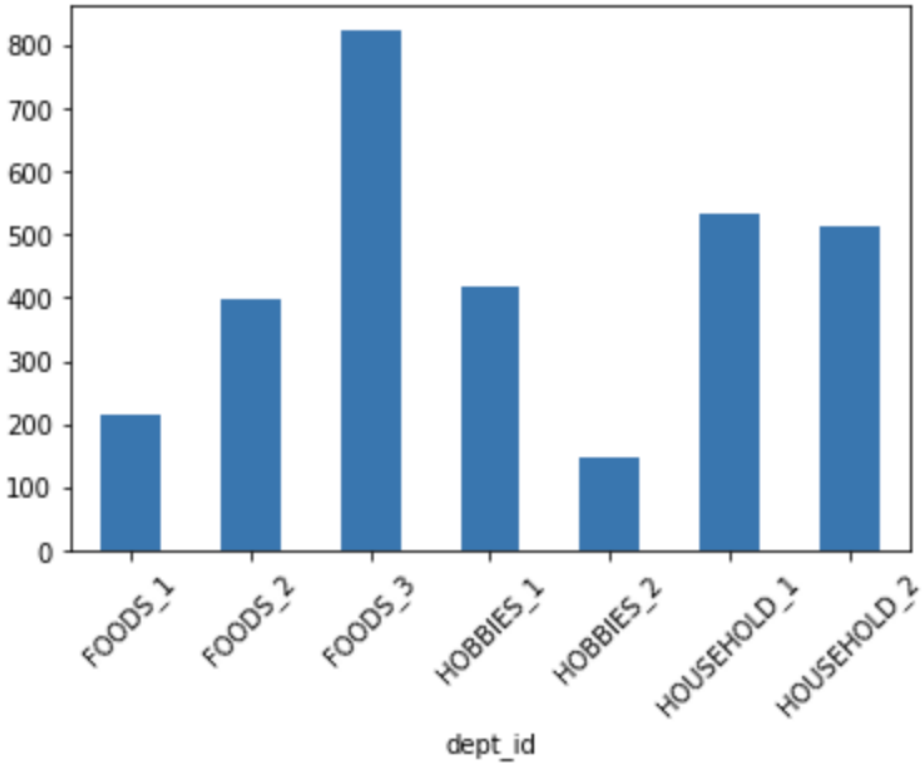
Frist, we start with "sales_train.csv" since it contains most of the important information. The file contains 30490 rows for 3049 unique items. There are rows for each of the 10 stores respectively. That means, for each same item, it has 10 rows of records of its own. There are 3 categories of products and it can be seen clearly that the number of food product items is roughly the total of those of household products and hobbies products.

Chart 2. Unique product count on category level



Within each category, each department does not always contribute the same either. FOODS_3 and HOBBIES_1 dominate in their own category while HOUSEHOLD_1 and HOUSEHOLD_2 have basically equal shares. We can see that the category and department information is two must-have categorical features in the training data and we should definitely include them.

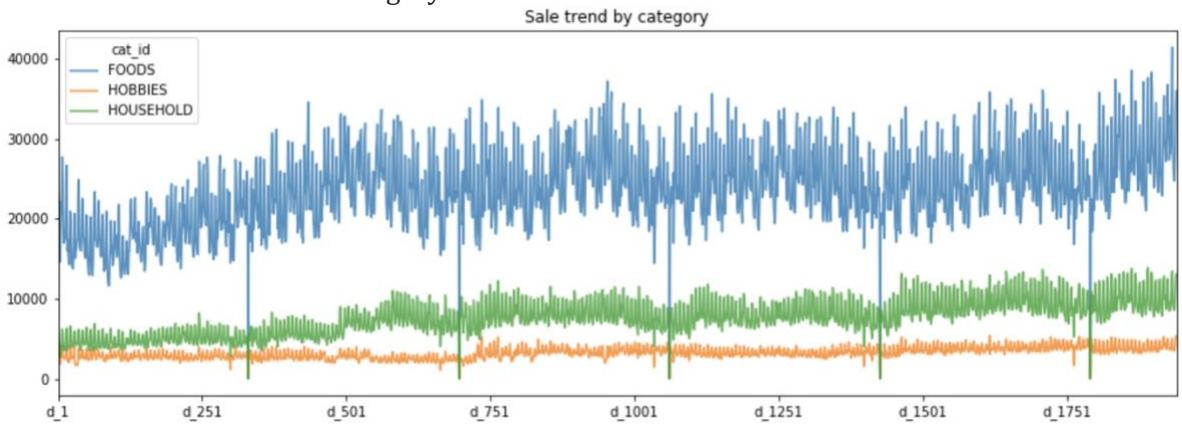
Chart 3. Unique product counts on department level



Having the first features in pocket, we will take a deeper look at the sales number for each category over time. We can see clearly from the chart that the main sales by number comes from FOODS and HOUSEHOLD and then HOBBIES, which is not surprising given FOODS has the most item numbers. There is definitely some kind of seasonal element in each category with FOODS having the most seasonal ups and downs and HOBBIES items a little hard to observe an obvious seasonal fluctuation. FOODS has the most variance as well, whereas HOBBIES got more variance near day_550 and HOUSEHOLD maintained the same. The sales over the 5.5 years went a little bit up for FOODS and HOUSEHOLD but not so much for HOBBIES. That is a good sign that the annual sales prediction will be more meaningful when there is no steep or sudden increase in sales over the years.

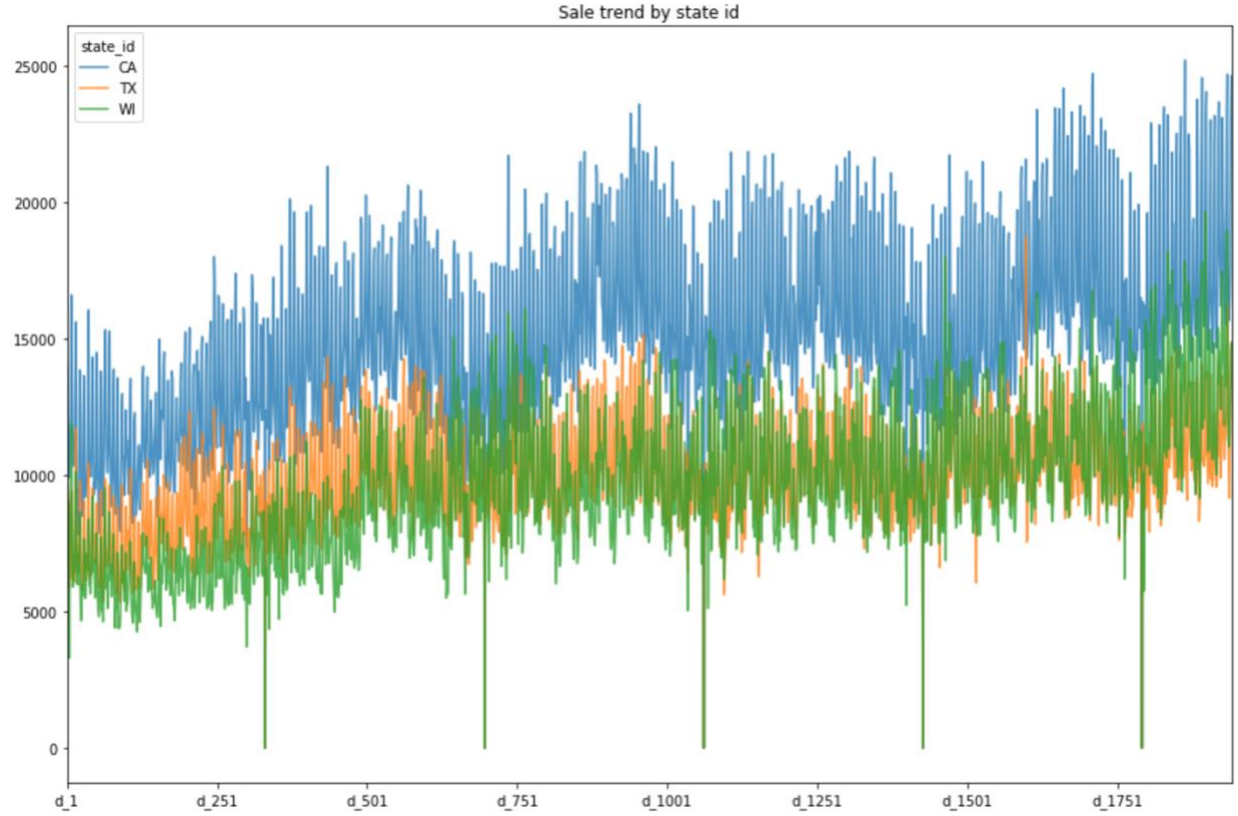
While no one can ignore the big drop for the sales of each category to almost zero, one can intuitively know what is going on without checking on the dates. Without any surprise, the five big drops were all on Christmas! On Christmas, even 24/7 operating Walmart will close their doors to customers. What is even more interesting is that the sales of these 5 days are not completely zero! For the 5 Christmas days, 72 FOODS_3 products, 4 FOODS_1 products,1 FOODS_2 product and 1 HOUSEHOLD_2 product were bought! I have tried to navigate the reason that these items were bought when the store was closing and why FOODS_3 was the particular popular thing on Christmas. I was guessing either there are some errors in the records or the product may be bought from the Walmart fuel station or from the fast food chain inside the store. Unfortunately, there is limited information online to prove the hypothesis. In the real world, it is worth the time to consult the system personnel to clear the doubt. There is also a relatively subtle right before Christmas every year and those happened on Thanksgiving where most people stayed at home rather than do last-minute shopping despite that Walmart was open.

Chart 4. Sales trend for each category



Having in mind that CA, TX and WI are three very different states in the US, I'm curious to see the sales compared geographically. It is apparent that the CA has the biggest sales. WI was a little lower than WI but it caught up by the end of 5.5 years. In this case, the location indeed influences the sales. For each item, the days annually it was sold in different stores make a difference and since the sales are basically consistent, previous sales in each store will help to predict current annual sales.

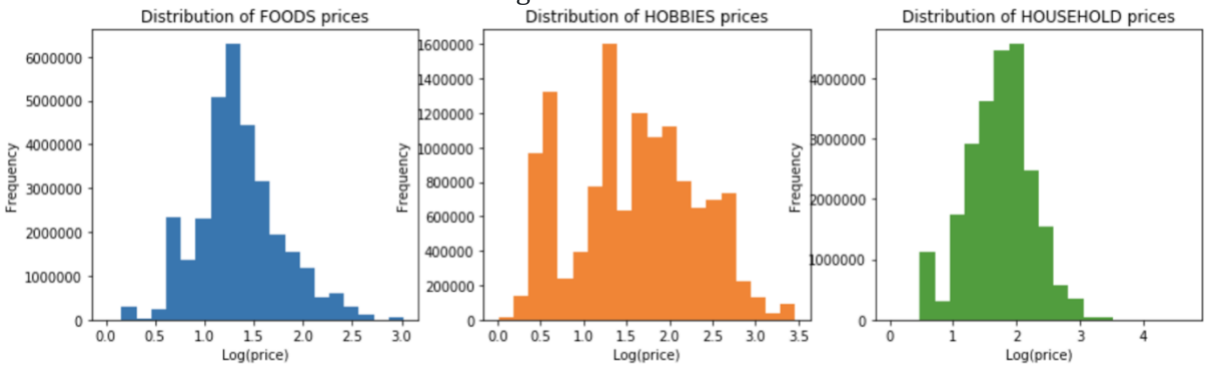
Chart 5. Sales trend for each state



3.2 Analysis on ***"sell_prices.csv"***

This file contains the item's price at one store in a certain week when that item is available for sale at that location. The price is calculated as a weekly average. If there is no price showing for certain weeks at one store, that item was not sold at that store. Thus, this file provides two very crucial feature sets. The prices of an item and how many days this item was sold at each store. We have to certainly contain the price information so let's take a look at the distribution of prices of each category. We can see from the histogram charts of the log(prices) that the prices of FOODS and HOUSEHOLDS are basically normally distributed. The distribution of HOBBIES products contains several local maximas. To better describe the prices, we will include annual average price, annual minimal price and annual maximum price for an item across the stores. We will also include previous year's price to describe the influence of previous year on the current year. As we have discussed in the assumption, we assumed that at the very beginning of the current year, we have already the full pricing plan for the current year, thus the feature we designed here will be aligned with this assumption.

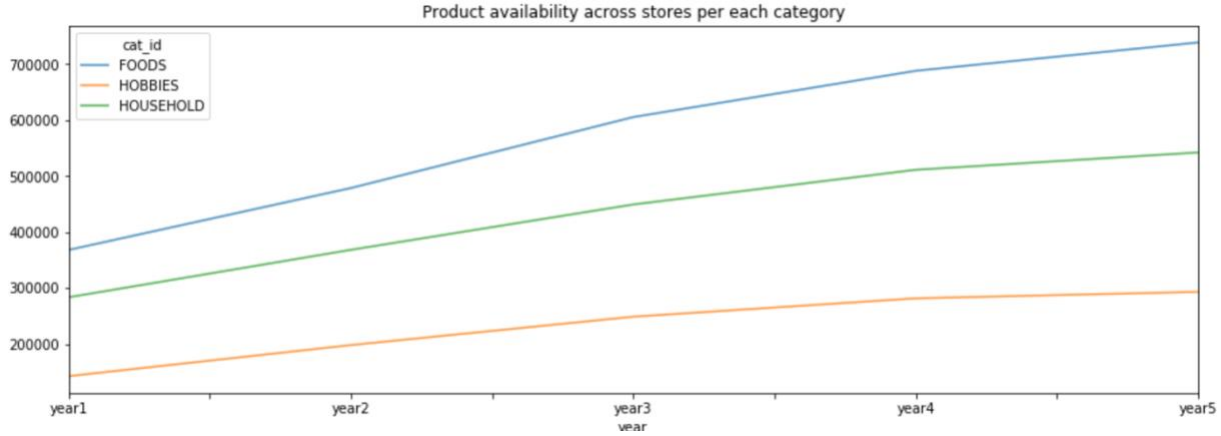
Chart 6. Price distribution in different categories



Another feature set as we have discussed above is the days that an item was sold in one store for. We have discussed the vagueness of 'sale of 0' in the assumptions. The analysis on this file will shed a light on the vagueness and provides more useful features for the models.

The chart below shows the available weekly average selling prices for each year for each category. We can see although the number of products we are researching on is 3049, the available prices are increasing each year, indicating the overall number of products coming into the market is higher than the products that got discontinued and removed from the shelves. We shall contain this information of price count each year for each store, aka. week number for an item available for sale in each store, as features in the training data. We should notice here however, that when we assemble the feature sets, we only include the certain information, rather than predictions and expectations that have not happened to make predictions. In this case, only the selling days in the previous year are certain and we cannot not use predicted selling days as features in the training data. Thus, we will include the previous selling days for each store and if the item will continue to be sold in this store (if the selling days for this year is over 0 than yes; if equals 0 then no).

Chart 7. Product availability across stores for different category



3.3 Analysis on *"calendar.csv"*

As we are conducting annual sales prediction, the daily information calendar provides is not that important as long as it remains constant across the years. For this project, it just acts like a bridge to link the day number to wm_yr_wk.

Since the first day of the observing period starts on Jan 29 rather than Jan 1, we cannot simply cut the last day of the year on Dec 31. There is a pattern in wm_yr_wk, however, after we count the days for each range. We can see that the first 5 period day counts are pretty close to 365 days. Thus, we can use wm_yr_wk to judge which year a record belongs to. This means, if wm_yr_wk is between 11100 and 11200, then the record is from year 1; if between 11200 and 11300, then the record is from year 2. This will make our work later much easier.

Table 2. Days count for different Wm_yr_wk range

Wm_yr_wk period	**Days in this period**
11100 - 11200	364
11200 - 11300	364
11300 - 11400	371
11400 - 11500	364
11500 - 11600	364
11600 - 11700	142

The calendar file also contains on a certain day, if SNAP is allowed to be used in the three states. Since the annual day counts allowed to use SNAP should be constant over the years, this information should not make a difference for annual sales prediction. As we have discussed in the assumptions, we cannot obtain the details of the FOODS products and we assume that as long as an item belongs to FOODS, we will assume it is eligible to be purchased by SNAP.

3.4 Feature Engineering and Feature Selection

After the analysis on the three files, we have gained a deeper understanding of the data we are dealing with and we have figured out the features for our training set to move forward to modeling in our next steps. The features should include:

I. current year' min, max and average price and previous year's min, avg, max price
II. previous year's total selling days for each store for the year (which is certain and happened already), and if in current years, the product will be continue (if the price count for current year is over 0 then yes, equals 0 then no)
III. previous year's sale by store_id and current year's sales number as target.
IV. department id and category id
V. if the item can be purchased by SNAP (we assume all the food item is eligible)

While all the features above seem to be helpful for making predictions, selecting the most relevant and important feature can help improve the performance of the models by eliminating the noises. I used Random Forest model in this process to filter out the weak predictors with the attribute feature_importances_ which ranks each feature by importance. I dropped all the features below the 0.5* median of all the importance value. After the selection, we keep the most important 32 features out of the 47 original features.

4. Performance Measure Metric Selection

Before starting the modeling session, we have to select proper metrics to compare difference models and serve as scoring in hyperparameter tuning section. The M5 competition has guidance on this matter: 'The accuracy of the point forecasts will be evaluated using the Root Mean Squared Scaled Error (RMSSE), which is a variant of the well-known Mean Absolute Scaled Error (MASE) proposed by Hyndman and Koehler (2006)[1]. The measure is calculated for each series as follows:'

$$\boldsymbol{RMSSE} = \sqrt{\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left(Y_t - \widehat{Y_t}\right)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(Y_t - Y_{t-1})^2}},$$

The main two reasons for choosing this metrics are:

A. The M5 series are characterized by intermittency, involving sporadic unit sales with lots of zeros. This means that absolute errors, which are optimized for the median, would assign lower scores (better performance) to forecasting methods that derive forecasts close to zero. However, the objective of M5 is to accurately forecast the average demand and for this reason, the accuracy measure used builds on squared errors, which are optimized for the mean.

B. The measure is scale independent, meaning that it can be effectively used to compare forecasts across series with different scales.

Since our goal is to predict the annual sales rather than daily sales, we will aggregate the daily sales numbers and the issue of sporadic unit sales with lots of zeros is not a concern here. In this case, we will just choose the typical metric for regression problem: root-mean-square error (RMSE) in this project. RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. It is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

[1] R. J. Hyndman & A. B. Koehler (2006). Another look at measures of forecast accuracy. International Journal of Forecasting 22(4), 679-688.

I also added another important score: R^2 (coefficient of determination) score on the side as reference for linear correlation measurement. R^2 is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

5. Model Selection and Training

I have chosen 9 regression models, including linear and non-linear models:

A. Linear Regressor
Linear regression is probably one of the most important and widely used regression techniques. It's among the simplest regression methods. One of its main advantages is the ease of interpreting results. The downside is that LinearRegressor model does not have a lot of parameters to tune or to reduce overfitting through regularization.

B. Ridge Regressor
Ridge Regression is a popular type of regularized linear regression that includes an L2 penalty, i.e. adds penalty equivalent to square of the magnitude of coefficients
This has the effect of shrinking the coefficients for those input variables that do not contribute much to the prediction task.

C. Lasso Regressor
Lasso regression is similar to Ridge Regression except that it performs L1 regularization, i.e. adds penalty equivalent to absolute value of the magnitude of coefficients

D. Support Vector Linear Regressor
Support Vector regression is a type of Support vector machine that supports linear and non-linear regression. Support Vector Machine is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables.

E. Support Vector Polynomial Regressor
Same as the Support Vector Linear Regression but we put a polynomial kernel as input.

F. Decision Tree Regressor
A decision tree is arriving at an estimate by asking a series of questions to the data, each question narrowing our possible values until the model get confident enough to make a single prediction. The order of the question as well as their content are being determined by the model. Decision trees regression normally use mean squared error (MSE) to decide to split a node in two or more sub-nodes.

G. Random Forest Regressor
Random Forest is a flexible, easy to use machine learning algorithm that produces great results most of the time with minimum time spent on hyper-parameter tuning.

H. Adaboost Regressor
Adaboost stands for Adaptive Boosting and it is widely used ensemble learning algorithm in machine learning. Weak learners are boosted by improving their weights and make them vote in creating a combined final model. It is based on decision tree boosted by AdaBoost.

I. Neural Networks
Neural networks (NN), also called artificial neural networks (ANN) are a subset of learning algorithms within the machine learning field that are loosely based on the concept of biological neural networks. We will build a very basic 3 layers model using Keras library in this project.

After confirming the list of the model choice, we will use the default parameters for all if the models to fit the training data, just shorten our list of models to be fine-tuned later. For the linear regression models, we are using here, some have the parameter to set if we want to fit a intercept or force the intercept to be 0 all the time. In our case, if all the variables are 0, for instance, the price, the days sold at stores, then surely there won't be any annual sales. So we set the 'fit_intercept=False' to have a better result.

Next, we need a benchmark to filter out the models that have obvious bad performance. The benchmark I use here is the Root Mean Square Error (RMSE) when we predict the target is just the same as the sum of previous year's sales number. The benchmark here after calculation is 2545.92.

Now, we set up the K-fold cross validation to get the performance metrics on the 9 models. The result is shown below:

Table 3. Cross validation results

	Linear	Ridge	Lasso	SVM Linear	SVM Polynomial	Decision Tree	Random Forest	AdaBoost	Neural Networks
rmse_ave	2132.75	2132.38	2131.70	2935.50	6523.93	2571.51	1999.71	4691.14	2018.95
rmse_std	1655.92	1656.22	1657.56	1434.29	1086.02	1688.46	1872.53	1364.84	1716.49
R2	0.93	0.93	0.93	0.90	0.60	0.91	0.93	0.78	0.93

It is clear that all the linear regressors did equally well in the cross validation. Neural Networks got the best performance. Random Forest yielded a nice result. SVM Polynomial and AdaBoost are surely off the table and Decision Tree performed a slightly worse than the benchmark. To keep the final model choice that cover diverse of errors, I will keep the 3 models of the original 9: Lasso, Random Forest and Neural Networks. We drop the Decision Tree, SVM Linear, SVM Polynomial and AdaBoost Regressors. We keep only Lasso in the Linear regressor group because they perform equally well and Lasso shows a better score.

We will then move forward to the fine-tuning the hyperparameters using GridsearchCV and RandomizedSearchCV. After we find the best parameter for the 3 models, we are ready for comparing the generalization error using test set.

6. Modeling Result and Application to the Real World

The final round of competition among models will be settled with the held-out test set. After we compare the prediction made by each model and the test set targets, we get the get the RMSE and R^2 score as follows:

Table 4. Prediction on test set results

	Lasso	Random Forest	Neural Networks
RMSE	2279.70	2810.62	7463.27
R^2	0.94	0.91	0.34

The results of R^2 is aligned with the results of RMSE. It is surprising that Neural Networks performed much worse than the training set result. The reason should be that Neural Networks itself is too complex for this problem. Simpler model like Lasso can do a better job than the two-layer Neural Networks. The complexity may cause overfitting and that's why the model has a much worse score in test set.

Among the three models, Lasso is the winner. It is reasonable instinctively because we have seen that the gross annual sales for each category remain very constant. The benchmark where we just predict the target to be the same sales as previous year is only 2545.92, which means the real target is not too far from the previous sales number. Given all the information on prices and selling days, the Lasso model is able to make a better prediction.

Since the numerical columns are all scaled, we can also get the coefficient to compare the feature importance very easily. After matching all the coefficients with feature names, we found that the indication is very clear. As we can see from the two charts below, we compare the absolute value of coefficients across features.

For numerical features, all the 26 numerical features are kept after feature selection and we can see that the previous sale number have a big impact on the target. We could see this when we were calculating RSME benchmark, we see the previous year sales is very close to the current year's.

For categoric features, only 6 features left after the feature selection and it is kind of blurred what the indication the coefficients show. The important features are all around the department ids and if the product will continued to be sold, especially for the stores in CA.

It is very clear that when the suppliers make a decision, what matters the most it's what category of product they are trying to make a prediction on,

weather it will be sold for the current year, as well as the previous year's sales data.

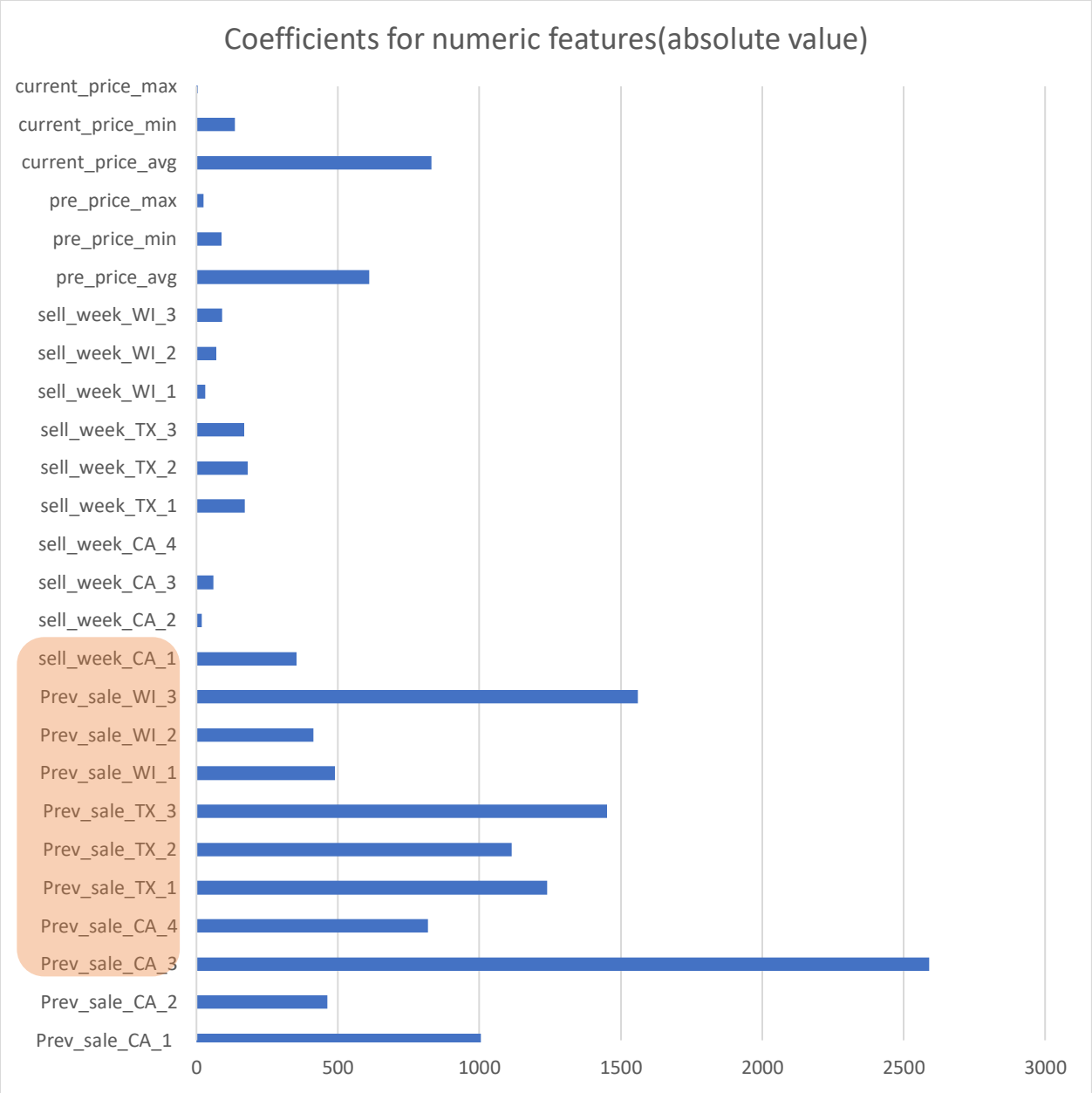


Chart 8. Feature importance for numerical features

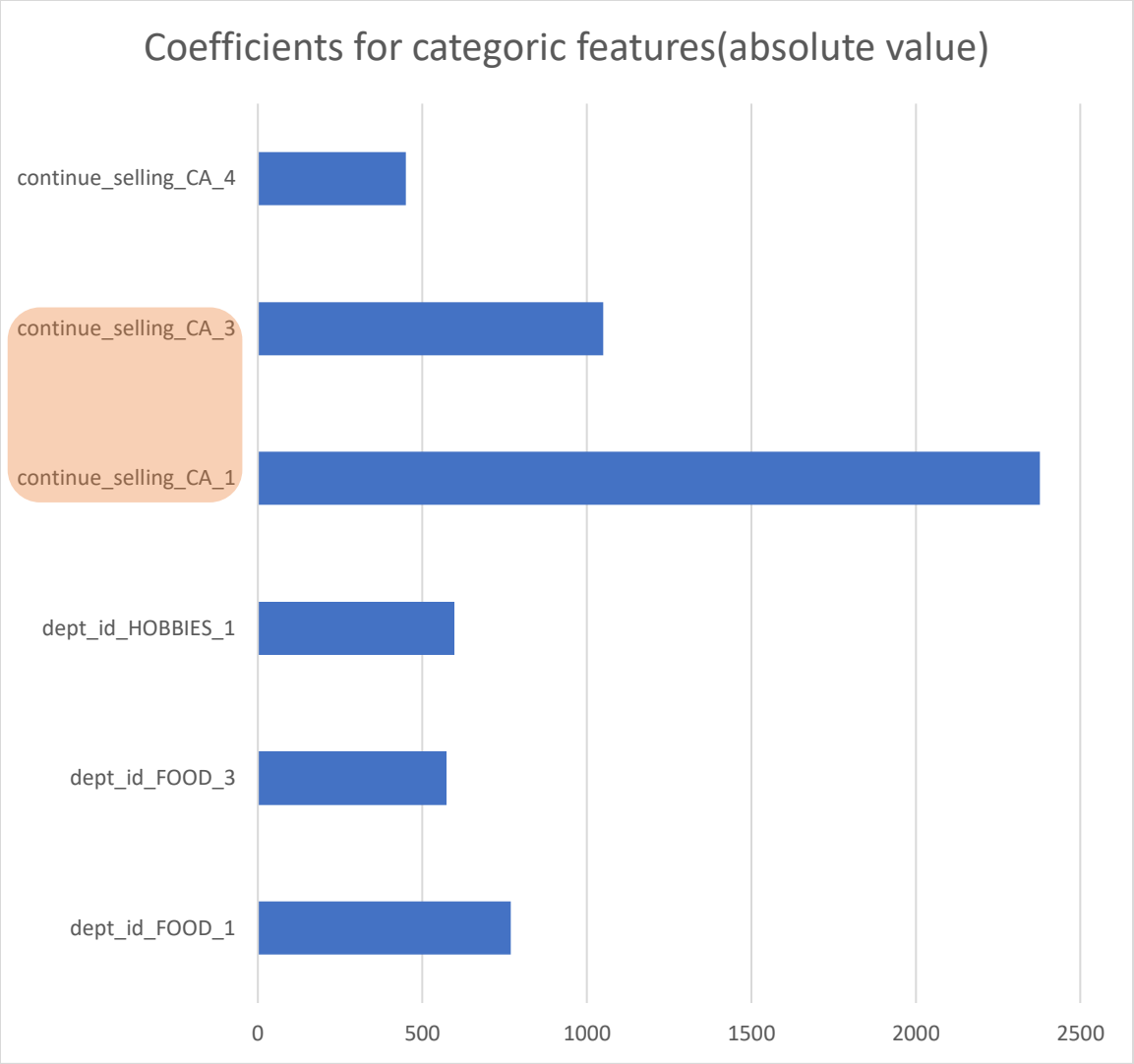


Chart 9. Feature importance for categoric features

7. Conclusion

In this research, we assembled a group of features from the original M5 Walmart Kaggle competition to serve our purpose to predict the annual sales for each item.

We have selected wide range of models and found Lasso Linear Regressor model to be best and it performed very well predicting the target. We also used the coefficient to rank the feature importance and found out that the previous sales number is an importance reference for current year sale's prediction. The category of the product also plays a critical role.

8. Reference

[1] https://www.analyticsvidhya.com/blog/2015/06/correlation-common-questions/

[2] https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/

[3] https://www.oxfordeagle.com/2016/12/24/what-time-does-walmart-close-tonight-christmas-eve-2016/#:~:text=Walmart%20closes%20at%206%20p.m.,usually%20closes%20on%20Christmas%20Eve.

[4] https://www.skubana.com/blog/walmart-leading-way

[5] https://www.kaggle.com/headsortails/back-to-predict-the-future-interactive-m5-eda

[6] https://www.kaggle.com/robikscube/m5-forecasting-starter-data-exploration