Sales Forecasting of Product X

By Xiaoxuan Ma

Contents

Business Question Overview 2
Data Cleaning & Preprocessing 3
1. **Missing Value** 3
2. **Abnormal situation** 3
3. **Inventory** 3
4. **Outliers** 3
5. **Convert daily data to monthly data** 4
6. **monthly data overview** 5
Methodologies 6
1. **ARIMA** 6
2. **OLS Regression** 8
3. **PCA** 9

Business Question Overview

The **'Sales Info'** contains the daily data of 2015-01-02-2017-08-27.
Attributes are **Year Calendar Date (YYYYMMDD), WeekNo, Year_Week, $Price, $Sales Amount, # Sold Unit, # Inventory Unit**.

Product X has a very long sales period, and its price is constant. Therefore, I guess it's a classic product, like all white Air Force 1.

We need to forecast the sales performance of product X in the upcoming months, which means we need to forecast **$Sales Amount** and **# Sold Unit**. Generally, they are highly correlated. So, our dependent variable is **$Sales Amount** or **# Sold Unit**.

The data of $Price all are 114, which means it's constant. So $Price is not useful for our purpose. But I found the average sales price (**$Sales Amount/# Sold Unit**) is not constant. This maybe caused by discount or other sales promotions. We can consider using it as an independent variable, if we can decide or forecast it.

The data also shows inconsistent inventory accounting cycle and missing replenishment information. This need data processing. With some assumptions, like no replenishment or known replenishment, we can use historical inventory data, like the previous period inventory as an independent variable.

Overall, I found I don't have any attribute or factor of the original sales data to be used as an independent variable. Therefore, there is no need to think about some machine learning methods like random forest. The first method should be tried is time series analysis, otherwise, maybe we can create some implied variables and combine with regression model to forecast.

First of all, we need to clean and preprocessing data to have a better view.

Data Cleaning & Preprocessing

1. Missing Value

First, I check if there is any NaN in the data.

```
Year                          0
Calendar Date (YYYYMMDD)      0
WeekNo                        0
Year_Week                     0
$Price                        0
$ Sales Amount                0
# Sold Unit                   0
# Inventory Unit              0
```

There isn't.

2. Abnormal situation

There are situations that $ Sales Amount>0, at the same time # Sold Unit=0, which don't make sense, so I deleted the related rows.

3. Inventory

We don't have the first day's inventory. I used the first available inventory data minus the sum of previous #Sold Unit as the first day's inventory. Then use the inventory of the day before minus the current #Sold Unit as current inventory, if original inventory equals zero. By this way, I got the continuous inventory data.

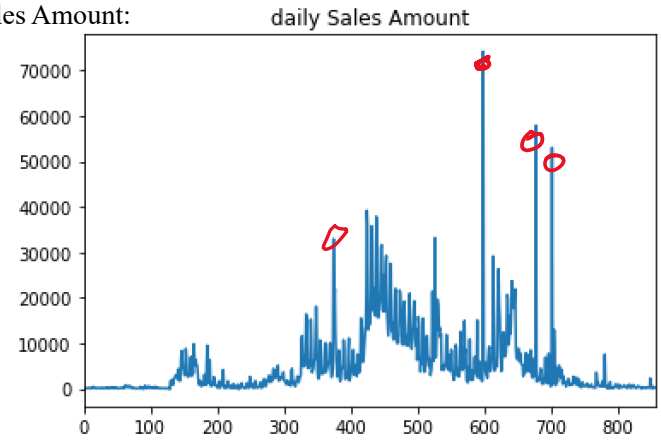
We don't want inventory is less than zero. Check inventory < 0:

Index	index	Year	lar Date (YYYYM	WeekNo	Year_Week	$Price	$ Sales Amount	# Sold Unit	# Inventory Unit
130	135	2015	20150820	34	2015_34	114.143	1716.71	32	-6
131	136	2015	20150828	35	2015_35	114.143	1651.65	15	-21
132	137	2015	20150829	35	2015_35	114.143	799.001	7	-28

I found 3 rows with inventory < 0. I will convert daily data to monthly data, so, I leave them now.

4. Outliers

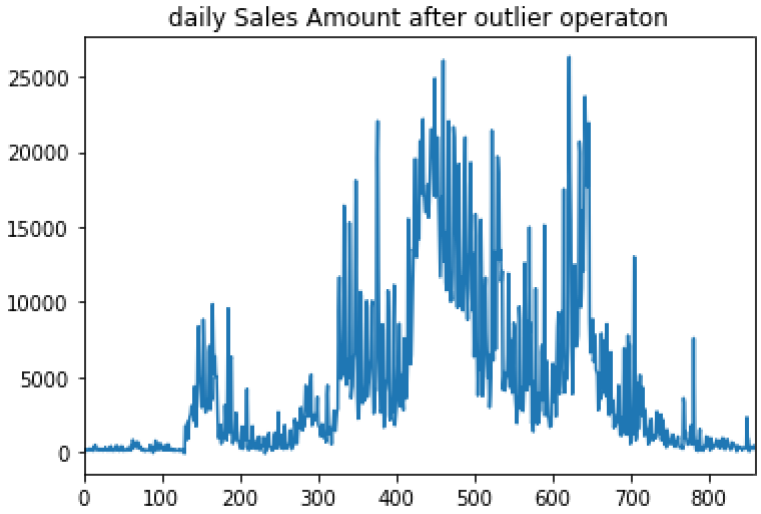
Plot the daily $Sales Amount:



I think these red dots areas contain outliers.

If $ Sales Amount is greater than mean of $ Sales Amount plus three multiple standard deviation, I regard it as an outlier, and replace it with previous date's $ Sales Amount.

After this operation, plot again:



Result is good, several outliers were removed, and didn't affect the overall trend.

5. Convert daily data to monthly data

We need to forecast the sales performance of product X in the upcoming months, due to having no known independent data, the longer the prediction period, the worse the effect. I decided to forecast the next three months' sales performance.

Using groupby() function to operate data. I found, for some months, number of days were less than the actual days. I used sales amount multiple thirty divided by data's number of days to adjust this.

And created average price by sales amount / sales units.

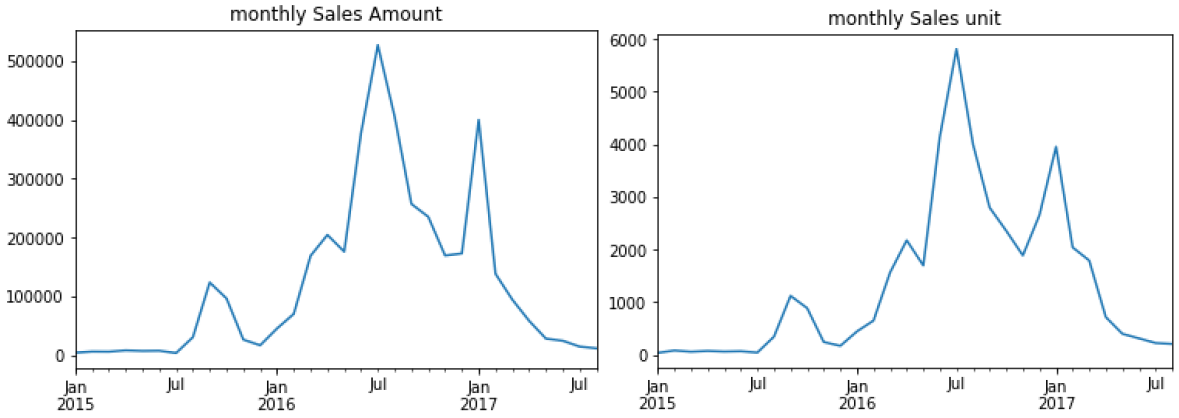
I create inventory related variables:
first_inv: inventory of the first day of this month
last_inv: inventory of the last day of this month
difference: last_inv - first_inv
last_month_end ; inventory of the last day of the previous month

6. monthly data overview

correlation coefficient matrix:

Index	Amount	Unit	avg price	first_inv	last_inv	difference	last_month_end
Amount	1	0.985485	0.23085	0.826751	0.866086	0.0800714	0.835145
Unit	0.985485	1	0.110624	0.854791	0.916282	0.121091	0.861402
avg price	0.23085	0.110624	1	-0.139167	-0.117566	0.038438	-0.134183
first_inv	0.826751	0.854791	-0.139167	1	0.851508	-0.263836	0.996295
last_inv	0.866086	0.916282	-0.117566	0.851508	1	0.281104	0.847162
difference	0.0800714	0.121091	0.038438	-0.263836	0.281104	1	-0.265051
last_month_end	0.835145	0.861402	-0.134183	0.996295	0.847162	-0.265051	1

Amount and Unit are highly correlated. Sales data and inventory data are also corelated.



From the above plots, we can find sales data are not stationary, and they don't have clear trend. We can find three spikes at Sep, Jul and Jan. But I won't say there is seasonal effect or pattern. The spikes may be caused by other effects.

Methodologies

1. ARIMA

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points. ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step can be applied one or more times to eliminate the non-stationarity.

The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values. The purpose of each of these features is to make the model fit the data as well as possible.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where:

- **p**: The number of lag observations included in the model, also called the lag order.
- **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
- **q**: The size of the moving average window, also called the order of moving average.

As I mentioned at first, due to lack for factors, I first considered to use time series analysis, and we found sales data are non-stationarity, and non-seasonal. It seems ARIMA worth a shot.

Before applying any statistical model on a time series, we want to ensure it's stationary.
I used ADF test to test stationarity. The Augmented Dickey Fuller Test (ADF) is unit root test for stationarity. Unit roots can cause unpredictable results in time series analysis.

ADF Test results:

variable	P_value
sale amount	0.421935699866319
sale unit	0.43313645492524755
first-order difference of sale amount	2.617013351855824e-06
first-order difference of sale unit	8.367719432386857e-05

If p-value is greater than the threshold 0.05, we can conclude that the time series is not stationary. The first-order difference of sale amount & unit are stationary.

Then we need to test if they are white noise, by statsmodels.stats.diagnostic.acorr_ljungbox.
If a process is really white noise then it is not forecastable by definition, because its values at different times are statistically independent.

White noise test result:

variable	P_value
first-order difference of sale amount	0.86048601
first-order difference of sale unit	0.49178178

Both of them didn't pass white noise test, therefore, we can't use ARIMA model to directly fit sales data.

Think about other options, I mentioned about the implied variables, we may try average sales price (**$Sales Amount/# Sold Unit**) or the previous period's inventory (we have one known number). And then maybe use them to do regression.

ADF Test results:

variable	P_value
last_month_end	0.4284504000438478
monly_avg_price	0.9919523846920864
first-order difference of last_month_end	3.4977092268385004e-07
first-order difference of monly_avg_price	1.8092226472418445e-08

White noise test result:

variable	P_value
first-order difference of last_month_end	0.49348915
first-order difference of monly_avg_price	0.03848083

The average sales price passed!
Use It to fit ARIMA model, with min(BIC), p=0, q=1
Prediction result: next three months' average sales price=[62.37660715, 61.49550865, 60.61441015]

2. OLS Regression

Now, use the average sales price, and the inventory of the last day of the previous month as independent variables to fit the OLS Regression model with sale units as dependent variables.

Model result:

OLS Regression Results

Dep. Variable:	Unit	R-squared:	0.789
Model:	OLS	Adj. R-squared:	0.774
Method:	Least Squares	F-statistic:	52.30
Date:	Sun, 12 Jan 2020	Prob (F-statistic):	3.50e-10
Time:	12:17:19	Log-Likelihood:	-246.21
No. Observations:	31	AIC:	498.4
Df Residuals:	28	BIC:	502.7
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1796.5616	722.187	-2.488	0.019	-3275.895	-317.228
avg price	20.4473	7.678	2.663	0.013	4.721	36.174
inv	0.2547	0.025	10.136	0.000	0.203	0.306

Omnibus:	12.985	Durbin-Watson:	1.479
Prob(Omnibus):	0.002	Jarque-Bera (JB):	13.059
Skew:	1.245	Prob(JB):	0.00146
Kurtosis:	4.977	Cond. No.	4.19e+04

We can see that Adj. R-squared is not bad, and variables are statistically significant (P-value<0.05)

The inventory of the last day of the previous month is 2903,

next three months' average sales price = [62.37660715, 61.49550865, 60.61441015]

with the linear regression model above we can get:

next month's sale units prediction = 218

sale amount = 218*62.37660715

The inventory of the last day of the previous month = 2903-218

If we assume without replenishment, then the prediction results are:

			Prediction	
last_mouth_inventory	Average price		Unit	Amount
2903	62.37660715		218	13598.10036
2685	61.49550865		145	8916.848754
2540	60.61441015		90	5455.296914

If in the next two months, there will be two replenishment with 2000 units, then the prediction results are:

				Prediction(replenishment)	
last_mouth_replenishment	last_mouth_inventory	Average price		Unit	Amount
	2903	62.37660715		218	13598.10036
2000	4685	61.49550865		654	40218.06266
2000	6031	60.61441015		979	59341.50754

3. PCA

Although the result of OLS is not bad, it shows warning message:
[2] The condition number is large, 4.19e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
In statistics, multicollinearity is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy. This reduce the ability of prediction.
One way to reduce multicollinearity is to apply **principal component analysis** (**PCA**) to the independent variables.

After applying PCA, I tried two ways to ways to do the regression with and without constant value. In the first situation, it still shows a warning message:
[2] The condition number is large, 5.16e+03. This might indicate that there are strong multicollinearity or other numerical problems.

In the second situation, R-square is much smaller.
This may be caused by the small size of independent variables.