M5 Forecasting Competition:

M5 Forecasting - Accuracy

Parin Kittipongdaja
M.Sc. CSIS : Data Science

Presentation for Applying Data Scientist Position @ CORALINE

About me

Experienced Data Scientist Project:
Computer Vision, Information Extraction,
Cleansing data, Web application.

Education:

M.Sc. CSIS : Data Science, NIDA

M.Sc. CEB: Data Science for Healthcare,
Mahidol University

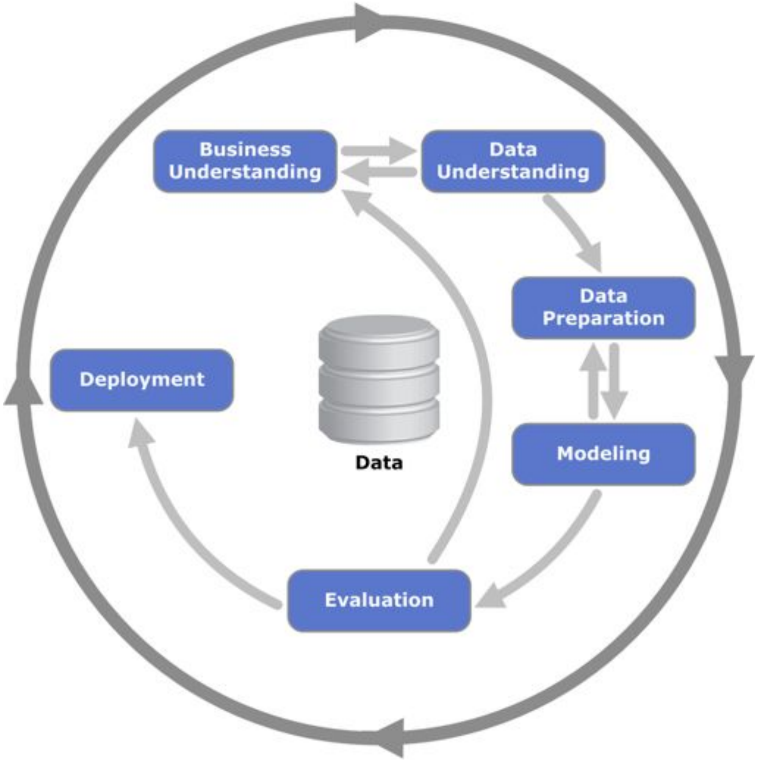
B.Sc. : Pharmacy, Chulalongkorn University

Introduction

This presentation is for demonstrating my data science skill following the CRISP-DM process by using Kaggle problem: M5 Forecasting - Accuracy to estimate the unit sales of Walmart retail goods.

The objective of the M5 forecasting competition is to advance the theory and practice of forecasting by identifying the method(s) that provide the most accurate point forecasts for each of the 42,840 time series of the competition.

CRISP-DM (CRoss Industry Standard Process for Data Mining)

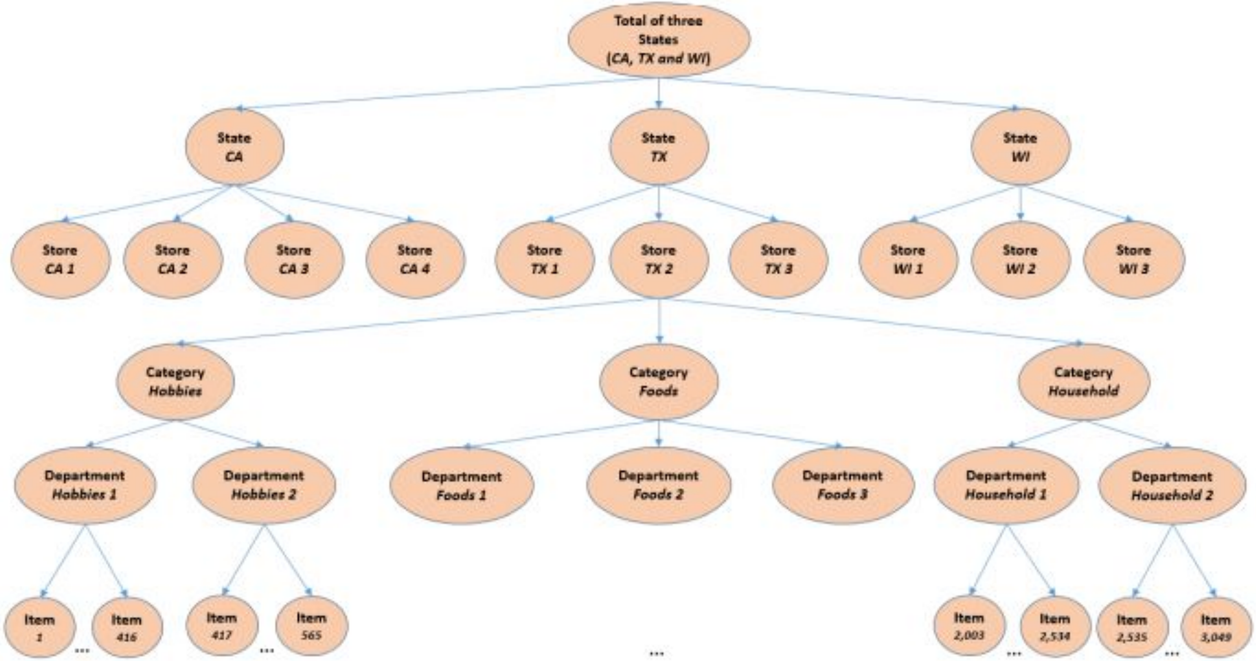


The CRISP-DM process or methodology of CRISP-DM is described in these six major steps. It is the framework that most widely-used analytics model.

Business Understanding

Walmart is the department store that having uncountable products and money transactions every day. Because of their rapid transaction rates keeping a balance between inventory and customer is most important. Therefore making an accurate sales prediction for different products becomes an essential need for stores to optimize profits.

Data Overview



Data Description

The data is hierarchical unit sales of various products sold in the USA, organized in the form of **grouped time series**. More specifically, the dataset involves the unit sales of **3,049 products**, classified in **3 product categories** (Hobbies, Foods, and Household) and **7 product departments**, in which the above-mentioned categories are disaggregated. The products are sold across **ten stores**, located in **three States** (CA, TX, and WI).

The historical data range from 2011-01-29 to 2016-06-19. Thus, the products have a (maximum) selling history of 1,941 days / 5.4 years

Data Files

1. **calendar.csv :** Contains information about the dates on which the products are sold.
2. **sales_train_validation.csv :** Contains the historical daily unit sales data per product and store [d_1 to d_1913].
3. **sell_prices.csv :** Contains information about the price of the products sold per store and date.
4. **sales_train_evaluation.csv :** Includes sales [d_1 to d_1941].
5. **sample_submission.csv :** The correct format for submissions.

1. Import Data

1. Import Data

calendar.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1969 entries, 0 to 1968
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   date          1969 non-null   object
 1   wm_yr_wk      1969 non-null   int64
 2   weekday       1969 non-null   object
 3   wday          1969 non-null   int64
 4   month         1969 non-null   int64
 5   year          1969 non-null   int64
 6   d             1969 non-null   object
 7   event_name_1  162 non-null    object
 8   event_type_1  162 non-null    object
 9   event_name_2  5 non-null      object
 10  event_type_2  5 non-null      object
 11  snap_CA       1969 non-null   int64
 12  snap_TX       1969 non-null   int64
 13  snap_WI       1969 non-null   int64
dtypes: int64(7), object(7)
memory usage: 215.5+ KB
```

sales_train_evaluation.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30490 entries, 0 to 30489
Columns: 1947 entries, id to d_1941
dtypes: int64(1941), object(6)
memory usage: 452.9+ MB
```

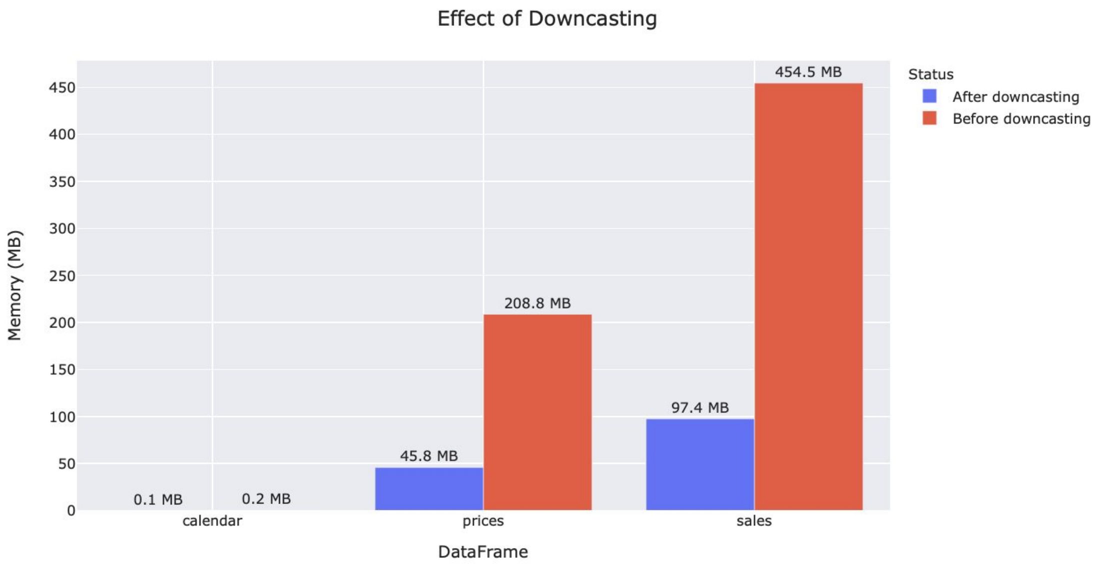
sell_prices.csv

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6841121 entries, 0 to 6841120
Data columns (total 4 columns):
 #   Column      Dtype
---  ------      -----
 0   store_id    object
 1   item_id     object
 2   wm_yr_wk    int64
 3   sell_price  float64
dtypes: float64(1), int64(1), object(2)
memory usage: 208.8+ MB
```

Data in sales_train_evaluation contains 30,490 rows with 1,947 columns and most of them is int64 data type which consume huge memory.

2. Preparing Data

2. Preparing Data: Downcasting



Downcasting the dataframes to reduce the amount of storage used by them and also to execute the operations performed on them more faster.

In figure, we can save more than 80% of memory in each dataframes.

2. Preparing Data: Melting the data

- To make analysis of data in table easier we can reshape the data into a more computer-friendly by converting from wide to long format.
- Store number of item sold in ‘sold’ column name
- Drop row that contains null value in ‘sold’ column.
- Join two tables into sales dataframe by using price data from prices dataframe and days data from calendar dataset.

Data Information

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60034810 entries, 0 to 60034809
Data columns (total 22 columns):
 #   Column         Dtype
---  ------         -----
 0   id             category
 1   item_id        category
 2   dept_id        category
 3   cat_id         category
 4   store_id       category
 5   state_id       category
 6   d              object
 7   sold           int16
 8   date           datetime64[ns]
 9   wm_yr_wk       int16
 10  weekday        category
 11  wday           int8
 12  month          int8
 13  year           int16
 14  event_name_1   category
 15  event_type_1   category
 16  event_name_2   category
 17  event_type_2   category
 18  snap_CA        int8
 19  snap_TX        int8
 20  snap_WI        int8
 21  sell_price     float16
dtypes: category(11), datetime64[ns](1), float16(1), int16(3), int8(5), object(1)
memory usage: 2.8+ GB
```

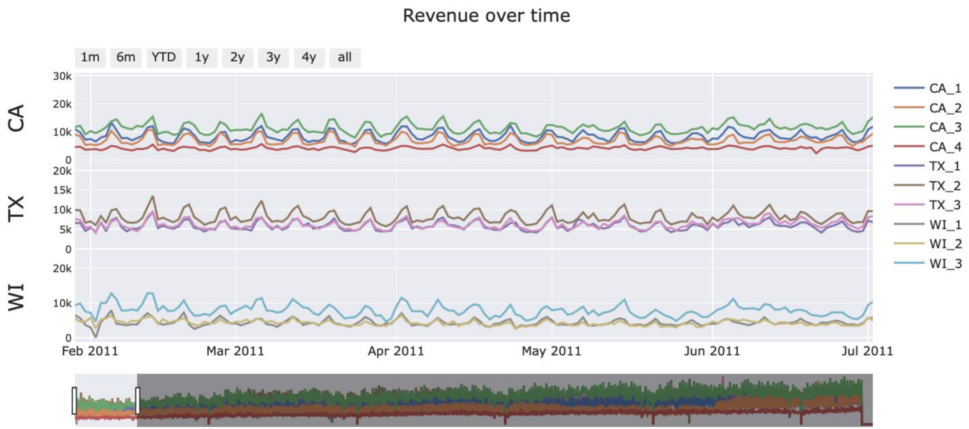
```
1 df.isna().sum()
```

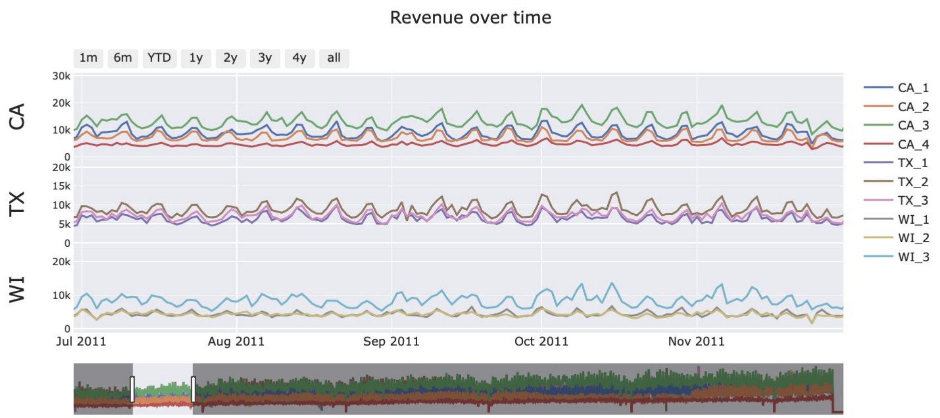
```
id                     0
item_id                0
dept_id                0
cat_id                 0
store_id               0
state_id               0
d                      0
sold                   0
date                   0
wm_yr_wk               0
weekday                0
wday                   0
month                  0
year                   0
event_name_1    55095430
event_type_1    55095430
event_name_2    59882360
event_type_2    59882360
snap_CA                0
snap_TX                0
snap_WI                0
sell_price      12299413
dtype: int64
```

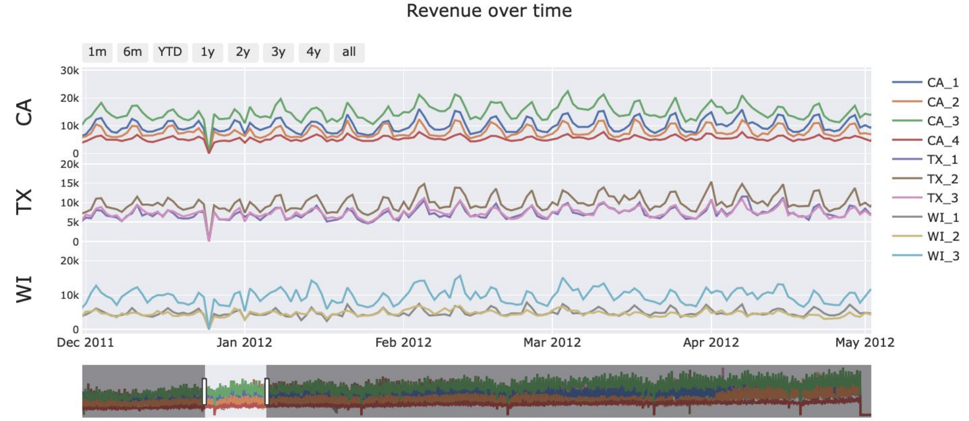
**The description is in speaker note.*

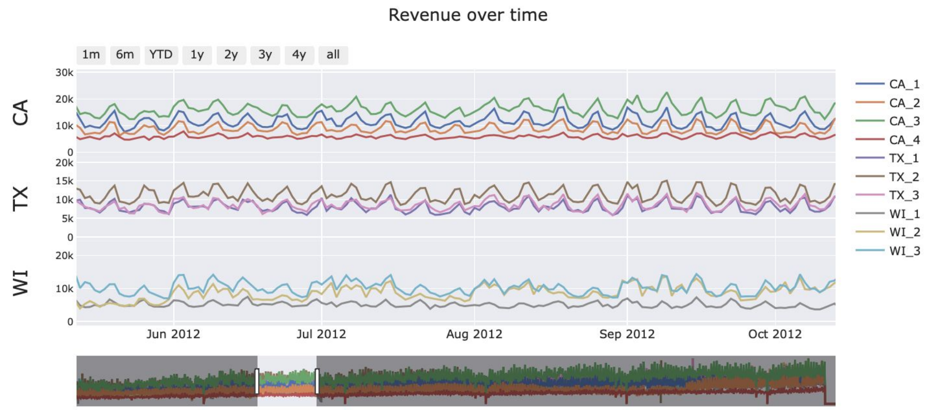
3. Exploratory Data Analysis

The description is in speaker note.

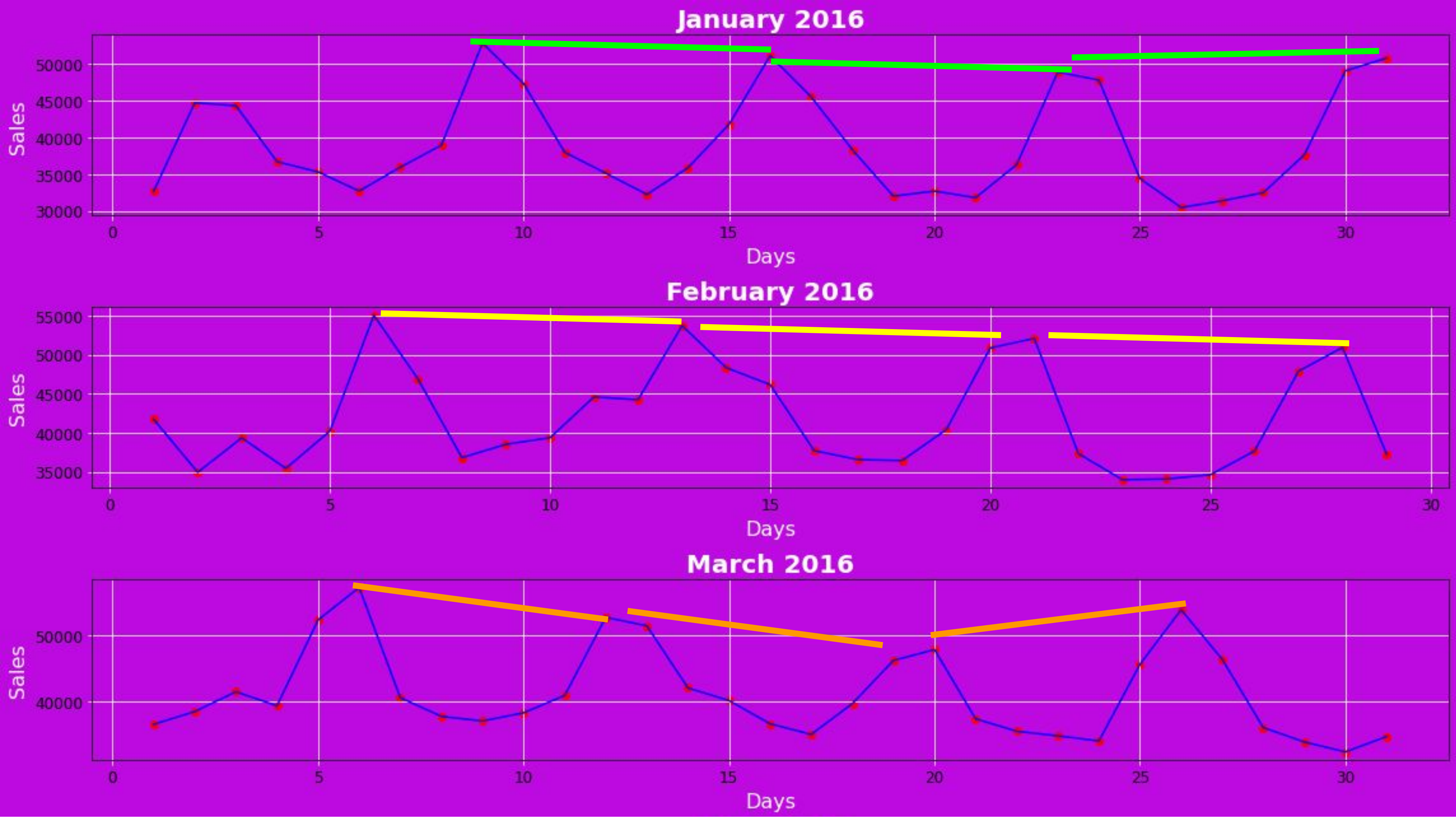




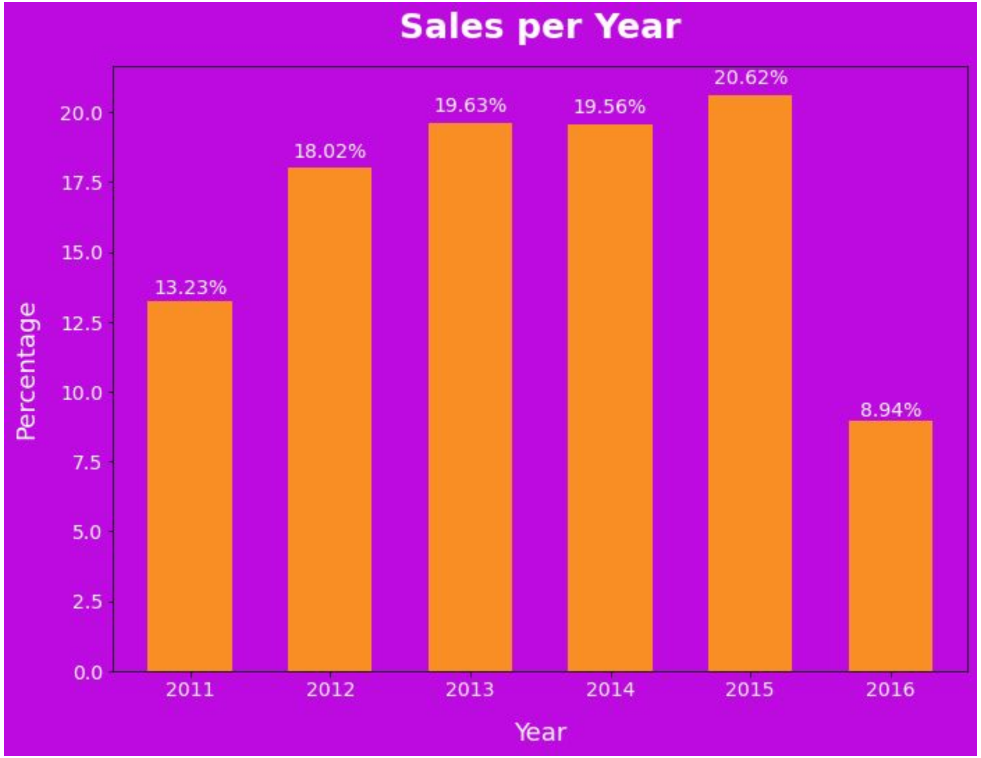




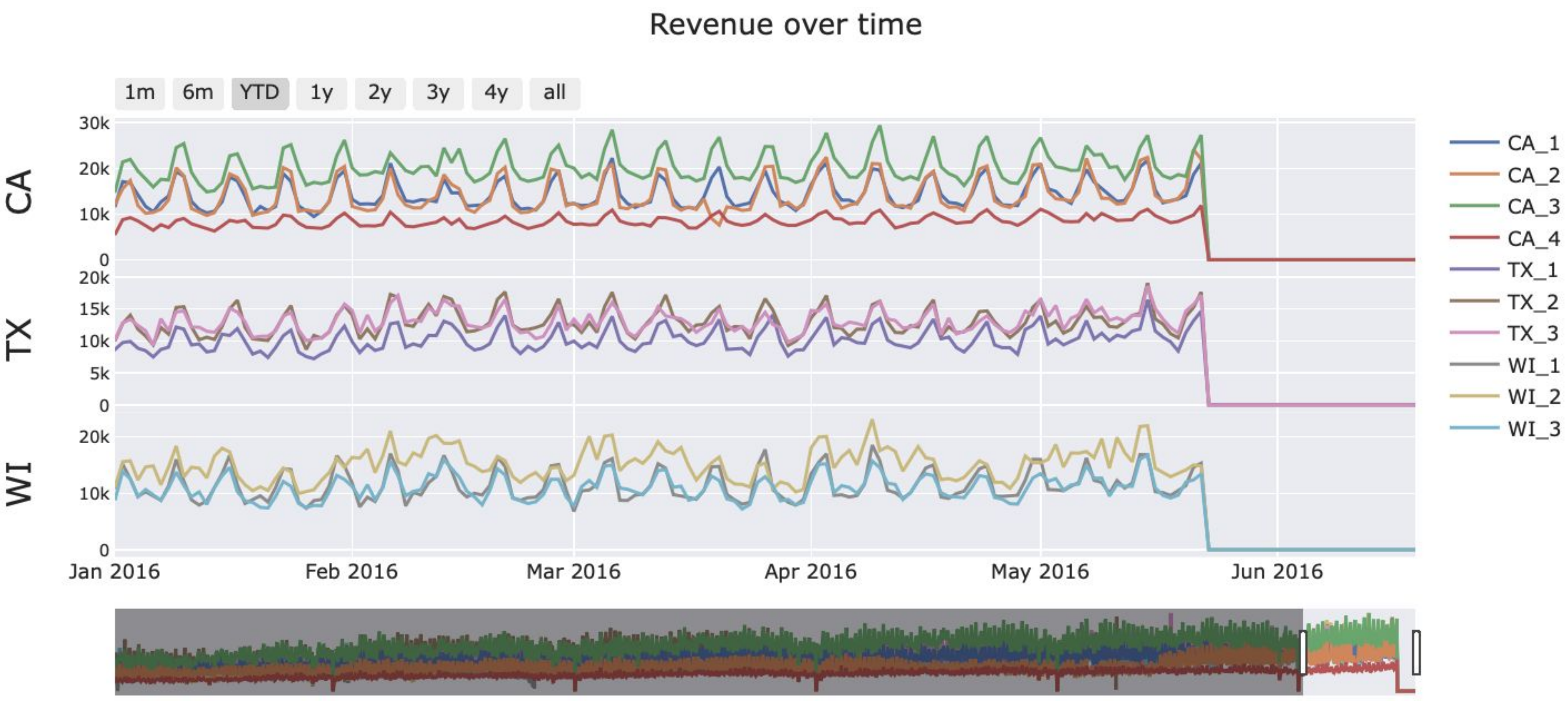
The description is in speaker note.


January 2016
Sales
50000
45000
40000
35000
30000
Days
0
5
10
15
20
25
30
February 2016
Sales
55000
50000
45000
40000
35000
Days
0
5
10
15
20
25
30
March 2016
Sales
50000
40000
Days
0
5
10
15
20
25
30

Total Sales per Year

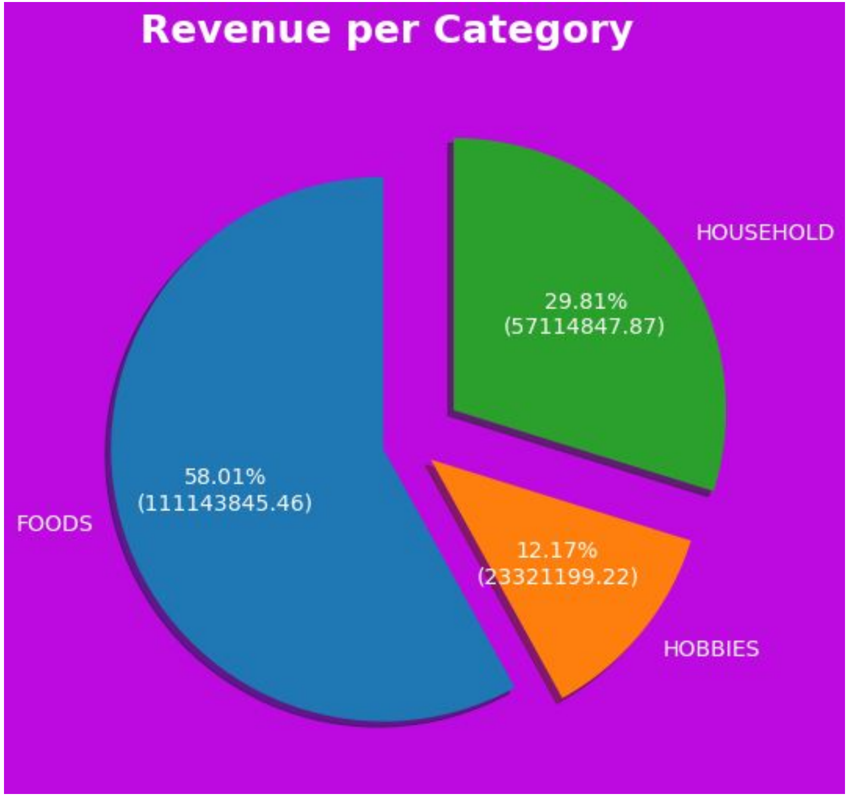


**The description is in speaker note.*



The description is in speaker note.

Revenue per Category



4. Feature Engineering

4. Feature Engineering

- Time Series data must be re-framed as a supervised learning dataset before we can start using machine learning algorithms.
- In this experiment, we apply many technique such as Label Encoding, Lags, Mean Encoding, Rolling Window Statistics, Expanding Window Statistics and Trends to create the feature for training model.

Label Encoding

Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form.ML algorithms can then decide in a better way on how those labels must be operated.

State (Nominal Scale)
Maharashtra
Tamil Nadu
Delhi
Karnataka
Gujarat
Uttar Pradesh

State (Label Encoding)
3
4
0
2
1
5

Lags

Lag is expressed in a time unit and corresponds to the amount of data history we allow the model to use when making the prediction.

Date	Value	$Value_{t-1}$	$Value_{t-2}$
1/1/2017	200	NA	NA
1/2/2017	220	200	NA
1/3/2017	215	220	200
1/4/2017	230	215	220
1/5/2017	235	230	215
1/6/2017	225	235	230
1/7/2017	220	225	235
1/8/2017	225	220	225
1/9/2017	240	225	220
1/10/2017	245	240	225

Mean Encoding

Feature Encoding - Target mean encoding

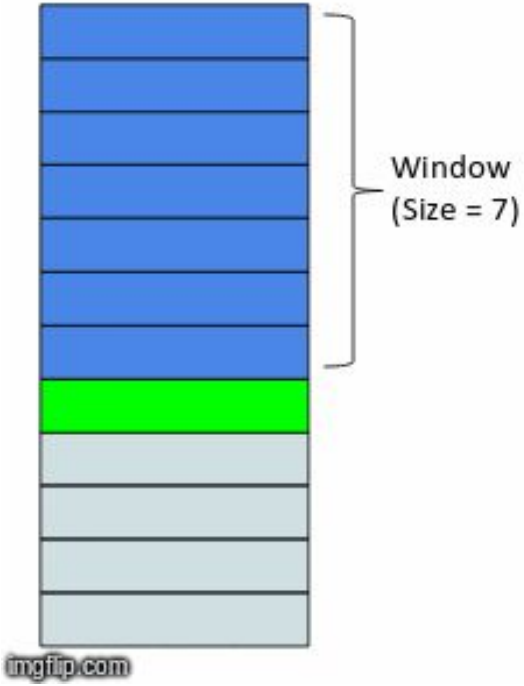
- Instead of dummy encoding of categorical variables and increasing the number of features we can encode each level as the mean of the response.

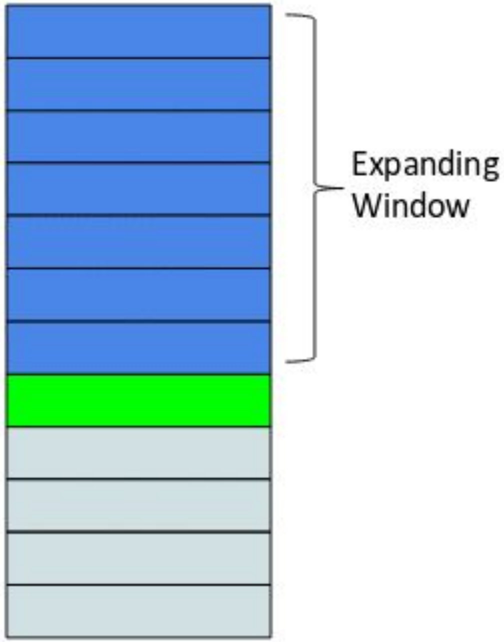
A	0.75 (3 out of 4)
B	0.66 (2 out of 3)
C	1.00 (2 out of 2)

Feature	Outcome	MeanEncode
A	1	0.75
A	0	0.75
A	1	0.75
A	1	0.75
B	1	0.66
B	1	0.66
B	0	0.66
C	1	1.00
C	1	1.00

H_2O.ai

Rolling Window Statistics & Expanding Window Statistics





4. Developing and Evaluation Model

Objective

To predict the products sales for the next 28 days which based on only the historical sales record (Previous studies on market sales prediction require a lot of extra information like customer and product analysis).

Performance Metrics

- Root Mean Squared Error: RMSE is the most widely used metric for regression tasks.
- This penalize large errors which mean one big error is enough to get a bad RMSE.
- RMSE is useful when large errors are undesired.

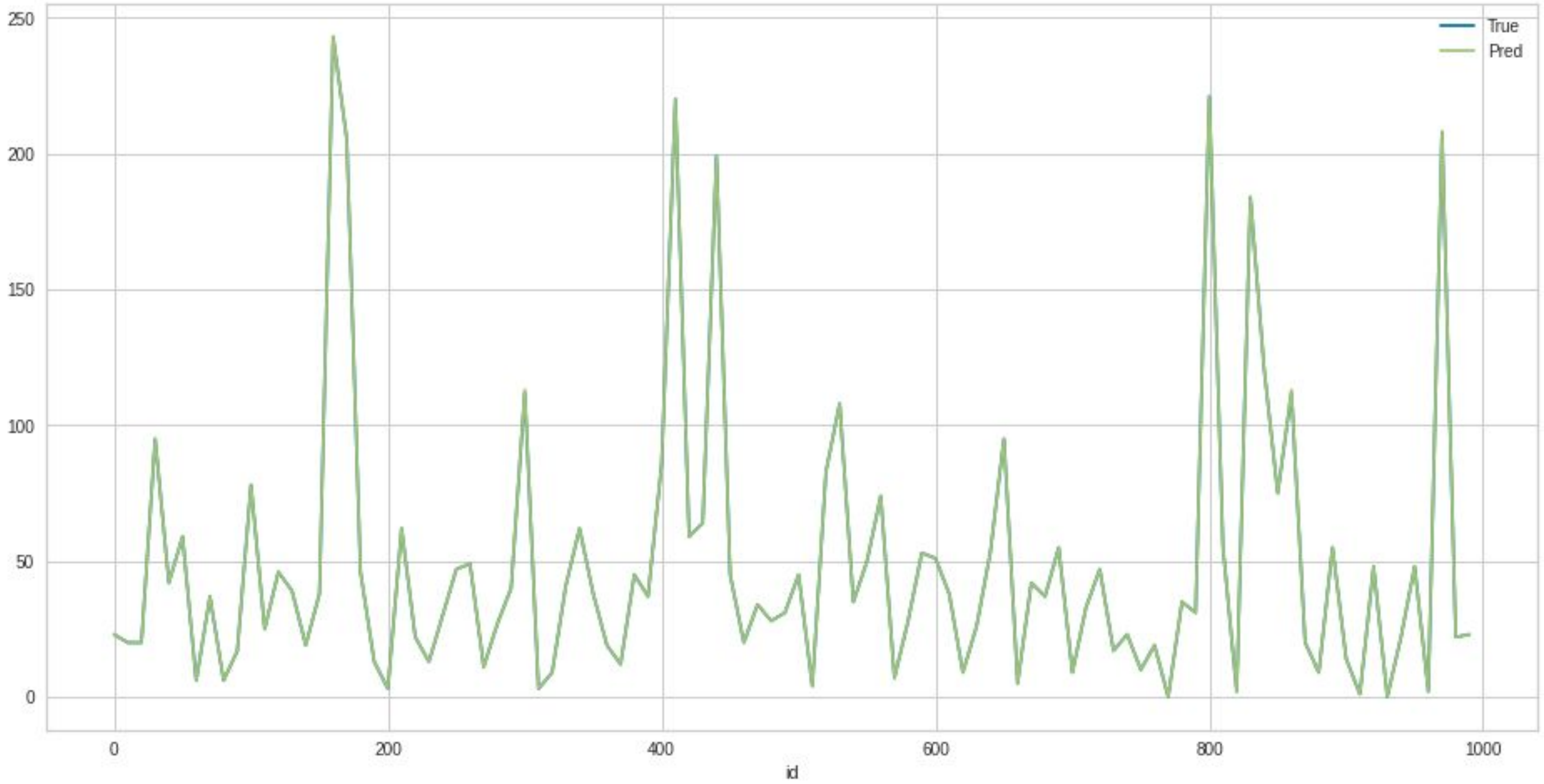
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lr	Linear Regression	0.0047	0.0002	0.0095	1.0000	0.0045	0.0026	2.0960
ridge	Ridge Regression	0.0001	0.0000	0.0006	1.0000	0.0001	0.0001	0.9220
lar	Least Angle Regression	0.0001	0.0000	0.0006	1.0000	0.0001	0.0001	0.9800
omp	Orthogonal Matching Pursuit	0.0001	0.0000	0.0006	1.0000	0.0001	0.0001	0.8780
br	Bayesian Ridge	0.0001	0.0000	0.0006	1.0000	0.0001	0.0001	2.5960
dt	Decision Tree Regressor	0.0017	0.0350	0.1788	0.9982	0.0034	0.0001	4.7720
rf	Random Forest Regressor	0.0014	0.0484	0.1794	0.9975	0.0025	0.0002	69.9400
et	Extra Trees Regressor	0.0034	0.0570	0.1908	0.9971	0.0038	0.0008	95.1360
par	Passive Aggressive Regressor	0.2097	0.1289	0.3577	0.9934	0.1569	0.1038	27.4840
gbr	Gradient Boosting Regressor	0.1316	0.1486	0.3766	0.9924	0.0742	0.1036	267.9960
en	Elastic Net	0.2428	0.3780	0.6148	0.9807	0.1415	0.1284	2.7460
lightgbm	Light Gradient Boosting Machine	0.0830	0.4175	0.6385	0.9787	0.0381	0.0498	7.5340
lasso	Lasso Regression	0.2572	0.4272	0.6536	0.9782	0.1400	0.1300	2.4260
knn	K Neighbors Regressor	0.3964	1.2977	1.1382	0.9337	0.2453	0.2506	15.2480
huber	Huber Regressor	0.6490	2.2273	1.4705	0.8863	0.3593	0.2667	38.3940
ada	AdaBoost Regressor	3.7714	16.7192	4.0037	0.1458	1.3268	1.9381	130.9520
llar	Lasso Least Angle Regression	2.0247	19.5721	4.4238	-0.0000	0.8586	0.5425	0.9720

Choose The Best Model: Decision Tree Regressor

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.0020	0.0380	0.1948	0.9981	0.0028	0.0001
1	0.0015	0.0767	0.2770	0.9961	0.0032	0.0001
2	0.0018	0.0204	0.1429	0.9990	0.0040	0.0002
3	0.0015	0.0145	0.1205	0.9992	0.0035	0.0001
4	0.0017	0.0252	0.1588	0.9987	0.0037	0.0002
Mean	0.0017	0.0350	0.1788	0.9982	0.0034	0.0001
SD	0.0002	0.0223	0.0548	0.0011	0.0004	0.0000

Example chart of predict and actual value



Business Impact / Insights

- From slide 16 we can observe that days at last of every year have “Zero” sales.It might be because of Christmas day the store remains closed.
- From slide 19 sales growth around 5% per year which we can plan to stock more items 5% from last year.

Future plan

- เนื่องจาก time series เป็นหัวข้อเรื่องที่ใหม่สำหรับผม (คือ ไม่เคยลองเล่นกับข้อมูล time-series เลย) จึงทำให้การทำ assignment ในครั้งนี้ ใช้เวลาไปกับการปูพื้นฐาน ความรู้อยู่นาน
- สิ่งที่อยากจะทดลองเพิ่มคือ ทำไม พวก linear regression ถึงได้ค่า RMSE ต่ำสุด แล้วมัน overfit จริงไหม รวมถึง อยากจะลอง train บน full data ด้วย (จาก slide 31 ใช้เวลาในการ train กว่า 2 ชั่วโมง ซึ่งเป็นเพียงแค่ 10% ของ ข้อมูล)
- อยากจะ add feature เพิ่มในแง่ของ seasonality คือ ข้อมูล sold หรือ revenue ก็ตาม มีการขึ้นลง เป็นช่วงๆ ซึ่งอาจเป็นผลมาจาก weekend และ weekday โดยเราอาจจะเอาข้อมูลในวันเดียวกัน ของสัปดาห์ที่แล้ว เช่น ทำนายยอดขายวันจันทร์ ก็เอา ยอดขายของวันจันทร์ที่แล้ว มาคิดด้วย เป็นต้น
- Seasonality อาจจะมีอีกในระดับ Month หรือ year ซึ่งถ้ามี ก็สามารถนำมาสร้าง feature เพิ่มได้อีก
- นำ ผลลัพธ์ที่ได้ มา calibrate ใหม่ เพื่อให้ได้ค่าที่ใกล้เคียงกับ actual มากที่สุด โดยการหา ค่า Coefficient บางอย่างมาคูณกับผลลัพธ์ที่ได้จาก model อีกที

Reference

- https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/163916
- https://www.kaggle.com/anshuls235/time-series-forecasting-eda-fe-modelling

Thanks!