MONOPRIX

SUP'COM
Higher School of Communication of Tunis

DESIGNING AND IMPLEMENTING A SALES FORECASTING SOLUTION

By:

Abid Mohaned

Gaidi Lamjed

Framed by:

Ben Nasr Hafedh

Ben Letaifa Asma

Ben Brahim Ali

TABLE OF CONTENTS

Problem

Obsolete Stock: One of the biggest drains on your profitability is obsolete or dead stock.

Customer Demand: Customers are looking for more value and convenience from their favorite brands.

Solution

For a **business** to operate efficiently, it needs some idea of what the **future** will look like. A **forecast** provides this look as a foundation upon which to plan.

Our solution leverages a famous ML model "XGBoost" to perform forecasting.

With a simple user interface to interact with the model and observe results.

Tools

HTML

CSS

JS

Flask

canvasJS

python

scikit learn

dmlc XGBoost

fast.ai

AI model

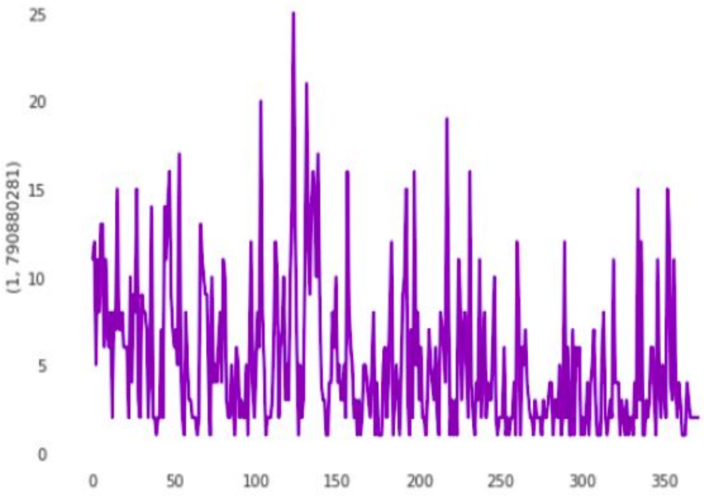
In order to find the right model that fits our data well and has the ability to generalize on new data, we cleaned and analyzed our dataset carefully looking for insights that can help us carry out our experiments with various types of models and data features.

AI model: Exploratory Data Analysis

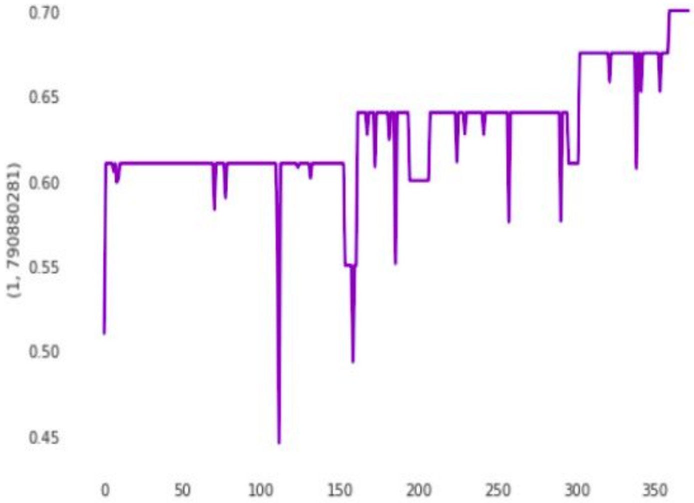
- This is a sample from our dataset containing dates, shop and item id to identify all products as well as item category and id struct to categorize all items. Also we have the price of the item and the amount of sold units per day.
- This dataset contains data for 4 years (2014-2018) representing 92 items in 5 different shops.

	A	B	C	D	E	F	G
1	Date	shop id	item id	m catego	id struct	Price	item cnt day
2	1/1/2014	2	425300160	1442	1442607080203	3.25	1
3	1/1/2014	2	431000630	1443	1443656130301	1.52	2
4	1/1/2014	2	437000050	1443	1443638020702	0.69	7
5	1/1/2014	2	437000130	1443	1443641010301	1.99	1
6	1/1/2014	2	554280020	1428	1428659090401	1.8	1
7							

AI model: Exploratory Data Analysis



Sales over time



price over time

AI model: Data cleaning & Preprocessing

- include for all shop,item pairs data points statiting 0 as sold units per day to force our model to account for the days that the product didn’t sell.
- Aggregate the price on average per month and the sales as the sum of the sold quantities per month.
- Adding “date_block_num” feature to play the role of the date since our models only accept numeric values.

	A	B	C	D	E	F	G	H
1	Date	date_block_num	shop_id	item_id	item_category	id_struct	Price_agg	item_cnt_month
2	2014-01-01	0	1	283400170	2507	2507349030701	1.61	0
3	2014-02-01	1	1	283400170	2507	2507349030701	1.61	0
4	2014-03-01	2	1	283400170	2507	2507349030701	1.61	0
5	2014-04-01	3	1	283400170	2507	2507349030701	1.61	0
6	2014-05-01	4	1	283400170	2507	2507349030701	1.61	0
7	2014-06-01	5	1	283400170	2507	2507349030701	1.61	0
8	2014-07-01	6	1	283400170	2507	2507349030701	1.61	0
9	2014-08-01	7	1	283400170	2507	2507349030701	1.61	0
10	2014-09-01	8	1	283400170	2507	2507349030701	1.61	0
11	2014-10-01	9	1	283400170	2507	2507349030701	1.61	0
12	2014-11-01	10	1	283400170	2507	2507349030701	1.61	0
13	2014-12-01	11	1	283400170	2507	2507349030701	1.61	0
14	2015-01-01	12	1	283400170	2507	2507349030701	1.61	0
15	2015-02-01	13	1	283400170	2507	2507349030701	1.61	0
16	2015-03-01	14	1	283400170	2507	2507349030701	1.61	0

AI model: Benchmarking experiments

As an evaluation metric for our experiments we have used RMSE (Root Mean Squared Error) since our problem can be viewed as a regression supervised learning problem.

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

AI model: Modeling & feature engineering

The models that we tried can be categorized into :

- Tree-based models : RandomForest, Adaboost , CatBoost , XGBoost
- Time series models: Prophet , ARIMA
- Deep learning models : RNN , LSTM , FASTAI TabularLearner

AI model: Modeling & feature engineering

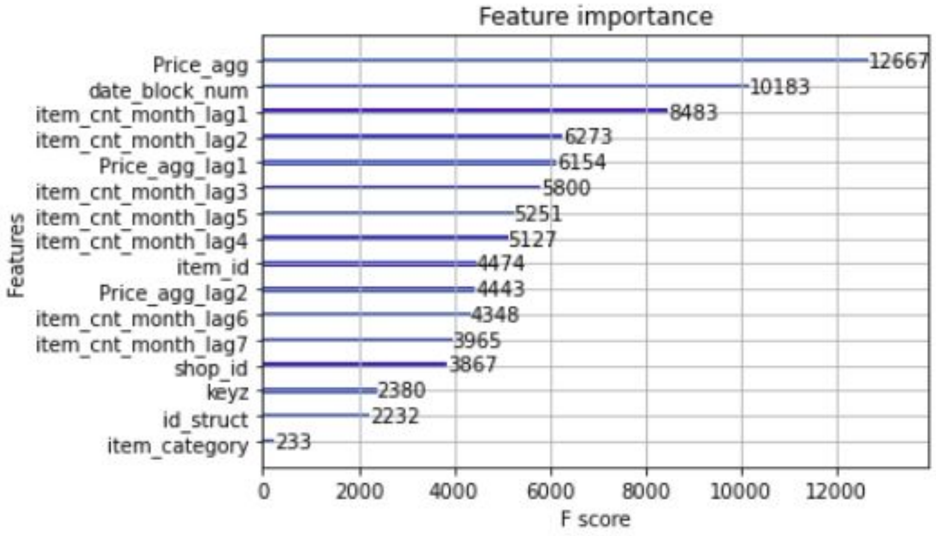
After all experiments with models and features, This was our best model:

In [9]:

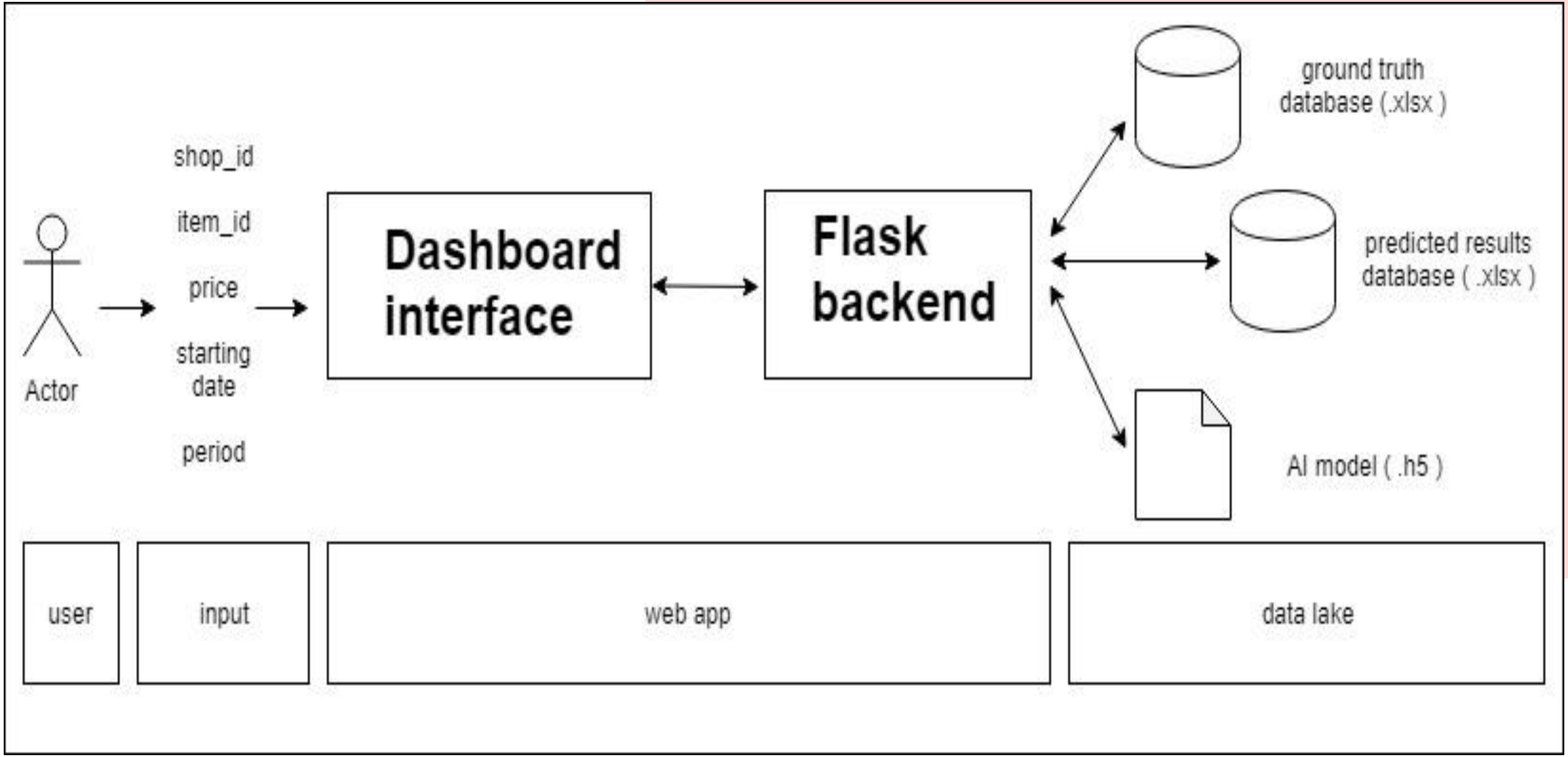
```
## model
param={'colsample_bytree': 0.8, 'subsample': 0.75, 'eta': 0.02, 'n_estimators': 1100, 'max_depth': 7, 'min_child_weight': 1}
model = XGBRegressor(**param)
```

AI model: Modeling & feature engineering

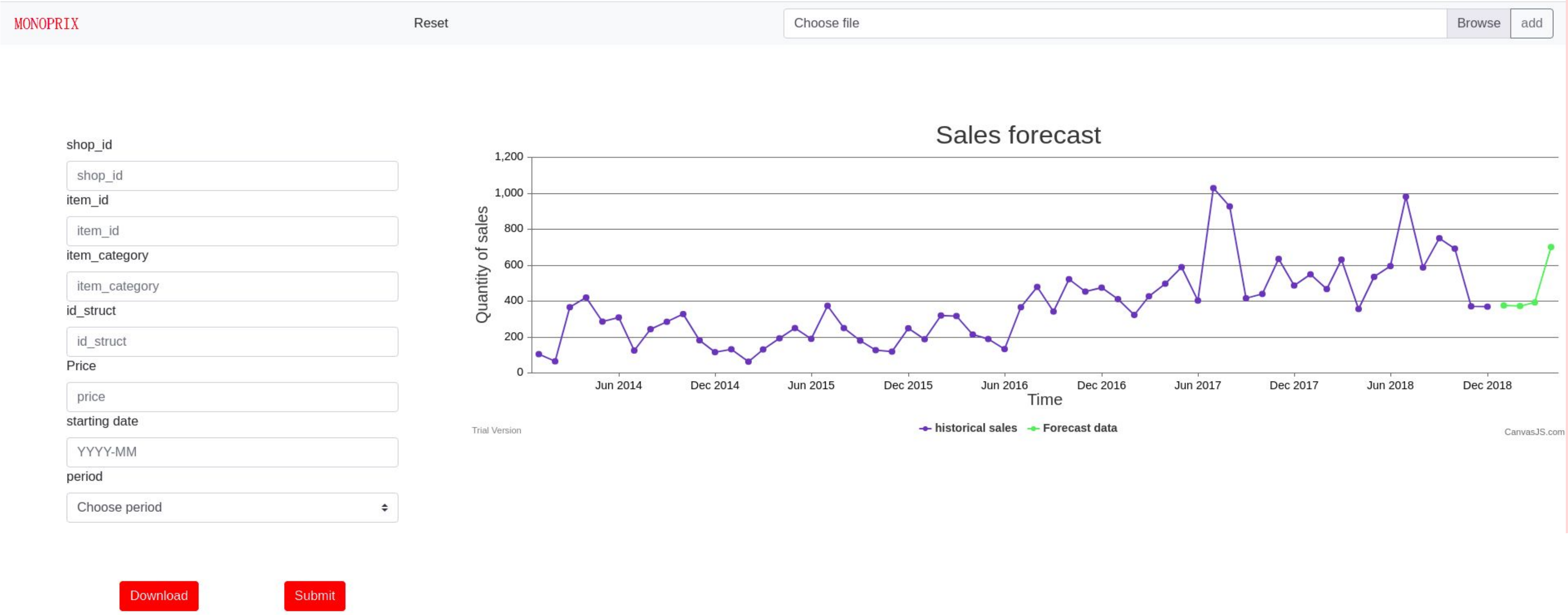
- Number of holidays per month
- 7 degrees of lag features for the sales counting
- 2 degrees of lag features for the price



Architecture

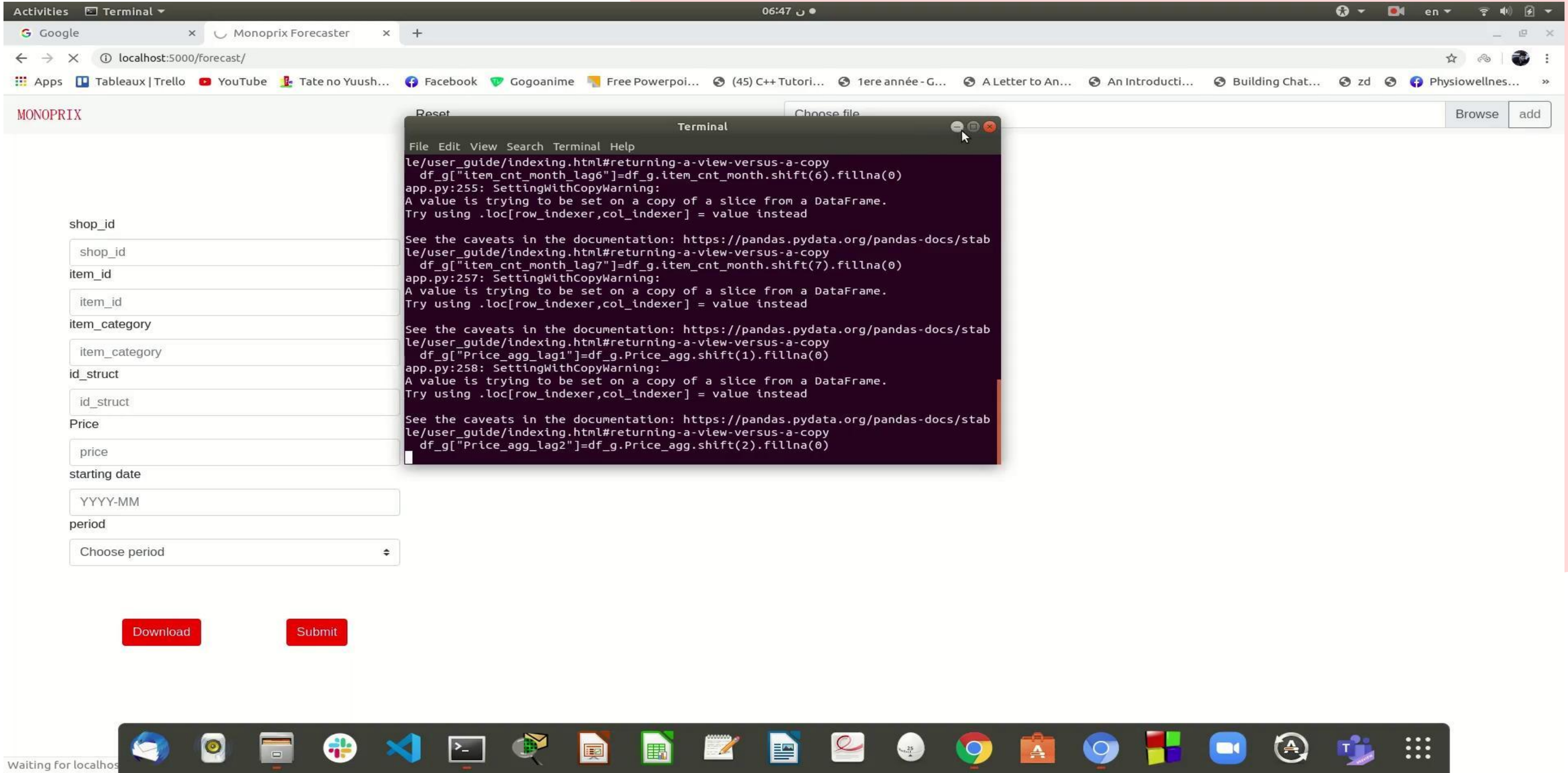


Interface



MONOPRIX
Reset
Choose file
Browse
add
shop_id
item_id
item_category
id_struct
Price
price
starting date
YYYY-MM
period
Choose period
Download
Submit
Sales forecast
Quantity of sales
1,200
1,000
800
600
400
200
0
Jun 2014
Dec 2014
Jun 2015
Dec 2015
Jun 2016
Dec 2016
Jun 2017
Dec 2017
Jun 2018
Dec 2018
Time
Trial Version
historical sales
Forecast data
CanvasJS.com

Demonstration



Conclusion

Despite the errors and the troubles we have encountered we consider our work to be satisfactory and we are proud of it but not to the point of considering it ideal. If the duration of the project was longer, we could have considered using different approaches like building an ARIMA model for each shop,item pair which can be more accurate but requires manual tuning and has scalability issues.

Conclusion

1000+
Lines of code

50+
Hours of Kaggle GPU resources

30+
Model Prototyping experiments

Github link:
https://github.com/mohaned-abid/DEMAND-FORECASTING-for-RETAIL-in-collaboration-with-MONOPRIX-

Thank you !

Any questions ?