LiBerTY – Store Sales Forcast with Machine Learning

Suhas Anand Balagar* Hardy Leung† Loukya Tammineni‡ Xichang Yu§

November 2022

Abstract

LiBerTY is an ensemble regression engine to predict the store sales given past sales figures. It employed a variety of robust and general data analysis and machine learning techniques to achieve good result within a short amount of time, competitive to existing best-known result while being robust and generally applicable to other problems.

Introduction[1]

The ability to predict the sales of a variety of stores is highly sought out in supply chain logistics, as it finds applications in increasing customer satisfaction and reducing food waste. We are proposing use of multiple Supervised Learning methods to predict the sales of stores based on time series dataset of Corporación Favorita, an Ecuador based grocery retailer. Ecuador is a country whose economy is strongly dependent on the oil and fluctuates with the price of oil.

Our work focused on a dataset from an ongoing Kaggle competition [1], "Store Sales – Time Series Forecasting". The dataset includes multiple csv sheets of time series data. We will try to evaluate the different aspects that might impact the sales in a store like Holiday seasons, Oil prices and historical sales data across all stores.

We plan on using different Supervised Learning models to predict the prices and then evaluate which of those models give out the best results. Supervised learning approach works best in this case, as we have huge time series data of both input and the output parameters mentioned above. We applied several transformation and optimization to improve the data quality in preparation for the optimization. To seek the best store sales prediction, We have evaluated several models including linear regression, Gradient Boost (XGBoost), Light GBM, Random Forest, Support Vector Regression (SVR), and Long Short-Term Memory (LSTM). We found XGBoost to be the best- performing individual method, and focused on hyper-parameter tuning via grid search. We further employed ensemble prediction to further improve our results, achieving a notable RMSLE score of 0.425 within our compressed project time-frame.

Related Work[2]

The store-sales prediction problem can be directly formulated as a multivariate multiple time-series regression problems. Prior to the modern age of machine learning, Auto-Regressive Moving Average (ARMA) was one of the most well-known technique, first proposed by [2] in his Ph.D thesis, and later popularized by Box and Jenkins [3], according to Wikipedia [4]. ARMA provides a succinct description of a (weakly) stationary stochastic process in terms of two variables, one for the auto-regression (AR), and the second for the moving average (MA). ARIMA, where "I" stands for Integrated and "S" stands for seasonal, are variants of ARMA appropriate for cases when data show evidence of non-stationarity, such as a long-term upward trend, and seasonality.

*San José State University, `suhasAB@github`
†San José State University, `ksleung@github`
‡San José State University, `LoukyaTammineni@github`
§San José State University, `Codyyu36@github`
[1] Suhas's section

[2] Hardy's section, and others

Random forest, first proposed by Ho [5] in 2008, is a well-established technique that relies on an ensemble learning method for classification or regression based on a collection of decision trees at training time. For regression tasks, usually the average prediction of the individual trees is taken, thereby correcting the tendency of individual trees to overfit to their training set.

XGBoost [6], first released in 2016, is a popular open-source software library which provides a gradient boosting decision tree (GBDT) framework for a plethora of program languages.

LightGBM [7] is a gradient boosting decision tree (GBDT) developed by Microsoft with the explicit goal of speeding up the training process of conventional GBDT by up to 20X while achieving almost the same accuracy. The huge runtime efficiency makes it a popular GBDT technique in recent years.

- Support Vector Machine(SVM) /Support Vector Regression (SVR)
- Long Short-Term Memory (LSTM)

Data Preparation[3]

Our main dataset contained the daily sales figures of 54 stores of Corporación Favorita, from January 1st, 2013 to August 15th, 2017, except Christmas Days when the stores were all closed. We are also given the locations (city and state) of each store. There are a total of 33 different product families, from `Automobile` to `Seafood`. Note that not all stores sell all products. Moreover, there were some stores that opened only after the data collection has begun, and sometimes the sales figures of certain stores were missing over a few months. All stores in the dataset were still in business as of August 15th, 2017, After properly dealing with missing data, Christmas, and stores that were yet to open, we would have a dataset made of (#days) $\times$ (#stores) $\times$ (#families) $= 1688 \times 54 \times 33 = 3008016$ numbers. We were also given the daily oil price over the same period of time (shown in Fig-1), as well as the dates of the regional or national holidays. These information could have a tremendous impact on the accuracy of our prediction, and hence must be considered.

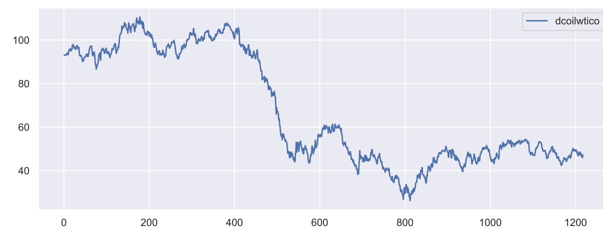

Fig-1: Oil price over time

Our job is to predict the sales figures for each store between August 16th, 2017 and August 31st, 2017, inclusive.

Due to the large amount of missing data in certain stores, or certain product families within a store, we could decide to treat those days as zero sales, in which the seasonality and trend could be severely compromised if we were to use the entire range of data. Figure 2 shows what the aggregate sales look like. It would confuse and possibly severely degrade the quality of our regressors if they were to be trained on such undesirable data.

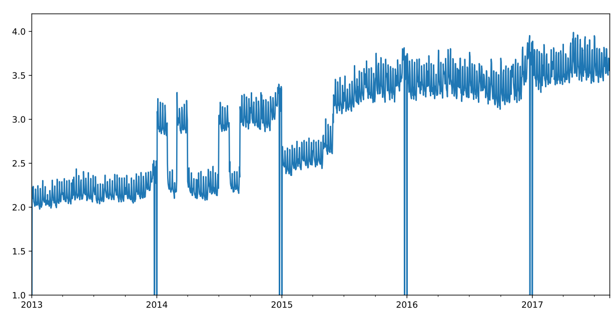

Fig-2: Aggregate sales price over time (original)

Alternatively, one can choose to ignore data that were too old. However, this approach is too conservative, because some stores may not have complete data until late, and we would be forced to trim the dataset to the tune of the worst offender. This approach seriously limit our datasize.

Instead, we propose to "inpaint" the sales figure using

[3] Hardy's section

patches of data from either a year ago or a year later (take the average if both are available). Note that this does create a theoretical possibility of a leakage problem since the missing data in the training set may be inpainted from dates in the validation set. We believe this issue is negligible, if at all. Figure 3 shows the aggregate sales over time after the inpainting. Note that the range of sales become much smoother.

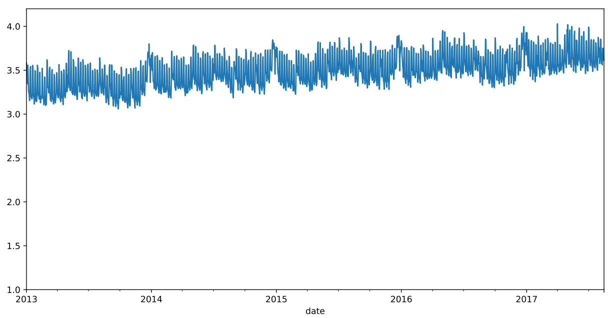

Fig-3: Aggregate sales price over time (inpainted)

In addition, we have performed a few standard data engineering techniques, including one-hot encoding of categorical attributes, standardization of certain attributes such as oil price, as well as generation of derived features such as `day-of-year`, `month-of-year`, `week-of-year`, `day-of-week`, `is-holiday`, and so on.

We have also checked for frequency distribution of data elements as part of data exploration. We have calculated and plotted correlation mapping to establish the correlations between different input and output parameters such as holiday events, oil prices, transactions per store type, dates of salary, natural calamities etc.

At this point, we can think of the training data as a list of observations X_i, where $1 \leq i \leq 3008016$. Each observation is keyed by [`store`, `family`], and is accompanied by a total of K features, $X_{i,1}, $X_{i,1}, $\ldots$, $X_{i,K}$. Note that we include among these features the most recent S (defaulted to 20) past sales, which we called lagging sales. This transformed and cleaned dataset is then passed to the regressors to obtain the prediction.

Experimental Setup[4]

Talk about the experimental setup, including how to

Part I[5]

- Linear regression
- XGBoost
- LightGBM
- Grid search technique to improve XGBoost performance
- CatBoost

Part II[6]

- LSTM, architecture, slightly different flow

Part III[7]

- Ensemble approach. You can quote this [8]
- Final result
- Visualization

Discussion[8]

Future Work[9]

- Note that we do not take into consideration the interaction between different stores and product families.
- Principle Component Analysis to reduce datasize

Conclusions[10]

In this work, we have presented LiBerTY, an ensemble regression engine that successfully predicts the store sales, which successfully employed a variety of data engineering and machine learning techniques.

[4] Hardy's section
[5] Loukya's section
[6] Cody's section
[7] Suhas's section
[8] TBD
[9] TBD
[10] Hardy

References

[1] "Store sales - time series forecasting," *Kaggle*. [Online]. Available: https://www.kaggle.com/competitions/store-sales-time-series-forecasting.

[2] P. Whittle, *Hypothesis testing in time series analysis*, vol. 4. Almqvist & Wiksells boktr., 1951.

[3] G. T. Wilson, "Time Series Analysis: Forecasting and Control, 5th Edition , by George E. P. Box , Gwilym M. Jenkins , Gregory C. Reinsel and Greta M. Ljung , 2015 . Published by John Wiley and Sons Inc. , Hoboken, N," *Journal of Time Series Analysis*, vol. 37, no. 5, pp. 709–711, 2016, [Online]. Available: https://ideas.repec.org/a/bla/jtsera/v37y2016i5p709-711.html.

[4] Wikipedia contributors, "Autoregressive–moving-average model — Wikipedia, the free encyclopedia." 2022, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Autoregressive%E2%80%93moving-average_model&oldid=1108230487.

[5] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, 1995, vol. 1, pp. 278–282 vol.1, doi: 10.1109/ICDAR.1995.598994.

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[7] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, vol. 30, [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[8] L. Rokach, "Ensemble-based classifiers," *Artificial intelligence review*, vol. 33, no. 1, pp. 1–39, 2010.