Store Sales - Time Series Forecasting

Ira Deshmukh
Department of Computer Science
University of Southern California
ideshmuk@usc.edu

Prem Tibadiya
Department of Computer Science
University of Southern California
tibadiya@usc.edu

Swathy Dakshinamoorthy
Department of Computer Science
University of Southern California
swathyda@usc.edu

Ryan Luu
Department of Computer Science
University of Southern California
rluu@usc.edu

Abstract

Sales forecasting can play a major role in the success of an organization. Accurate sales forecasts allow organizations to make informed decisions while setting organizational goals, hiring, and budgeting among other profit impacting factors. If goods are not readily available or goods availability is more than demand overall profit can be compromised. As a result, sales forecasting for goods can be significant to ensure loss is minimized. Additionally, the problem becomes more complex as retailers add new locations with unique needs, new products, ever transitioning seasonal tastes, and unpredictable product marketing. In this analysis, we tried to develop a forecasting model using machine learning algorithms to improve the accurately forecasts product sales. We have tried to build models that can help us to capture the trends. Due to the hierarchical structure of our data, it was difficult to directly use this data and perform simple time-series based forecasting. So, we aggregated this data and performed feature engineering to make it suitable for feeding it to our models. We treated this problem as a time-series based as well as a static supervised learning problem.

1 Introduction

This is a Kaggle Competition called "Store Sales - Time Series Forecasting" where the task is to predict stocking of products to better ensure grocery stores please customers by having just enough of the right products at the right time. Sales forecasting is the process of estimating future revenue by predicting the amount of product or services a sales unit (which can be an individual salesperson, a sales team, or a company) will sell in the next week, month, quarter, or year.In this project, we are trying to forecast product sales based on the items, stores, transaction and other dependent variables like holidays and oil prices.

1.1 Dataset

The dataset chosen is a Kaggle competition dataset hosted by Store Sales - Time Series Forecasting. Currently we are predicting the sales of just Grocery I items, but this can be further extended to other classes of groceries and beauty products which you might find a super market. The link to our dataset is `https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data`.

Preprint. Under review.

The dataset has the following files and properties:

- Train.csv: Consists of train data with unit sales per day.
- Stores.csv: Consists of all the stores, their location and their individual store numbers.
- Holidays_Events.csv: Consists of the holidays and events metadata.
- Oils.csv: Consists of Daily oil prices.
- Test.csv: Consists of test data with same features as the train data except for the sales per day.

1.2 Exploratory Analysis

We are predicting the sales for the grocery items by using a custom regressor using decision trees, random forests, ensemble methods along with ridge regression and support vector machine. They have been combined together to give the best possible prediction. The features for this project were selected after exploring the effect of the provided data on store sales.

The following plots were obtained:

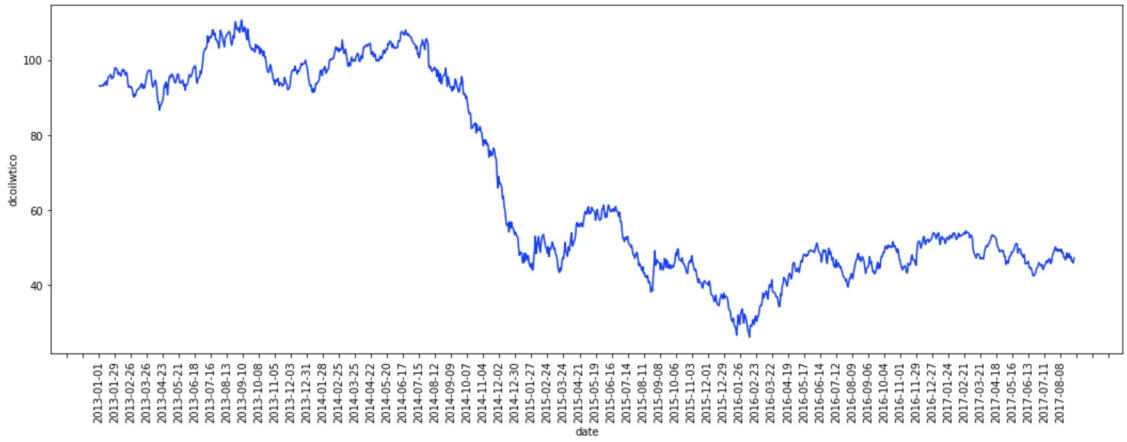


Figure 1: Oil prices over the complete date range.

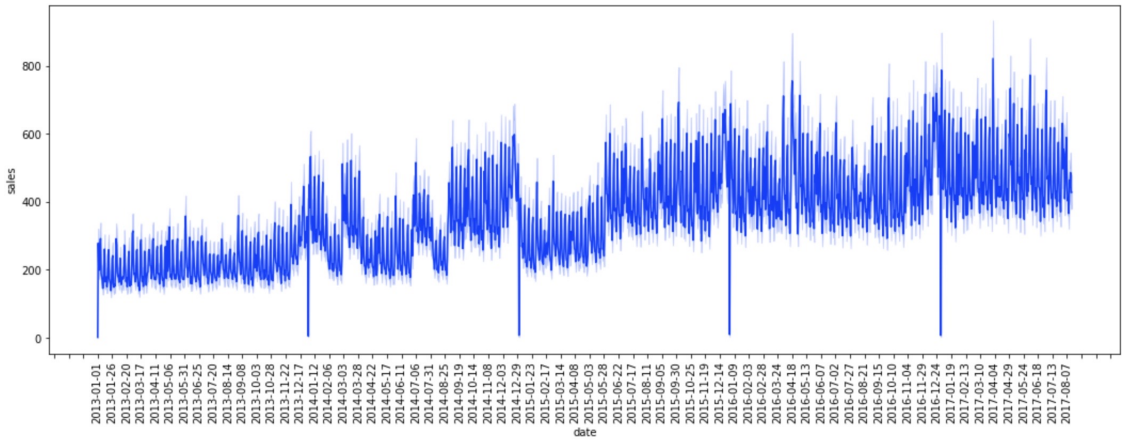


Figure 2: Sales over the complete date range

We further found the count for each type of holiday and store.

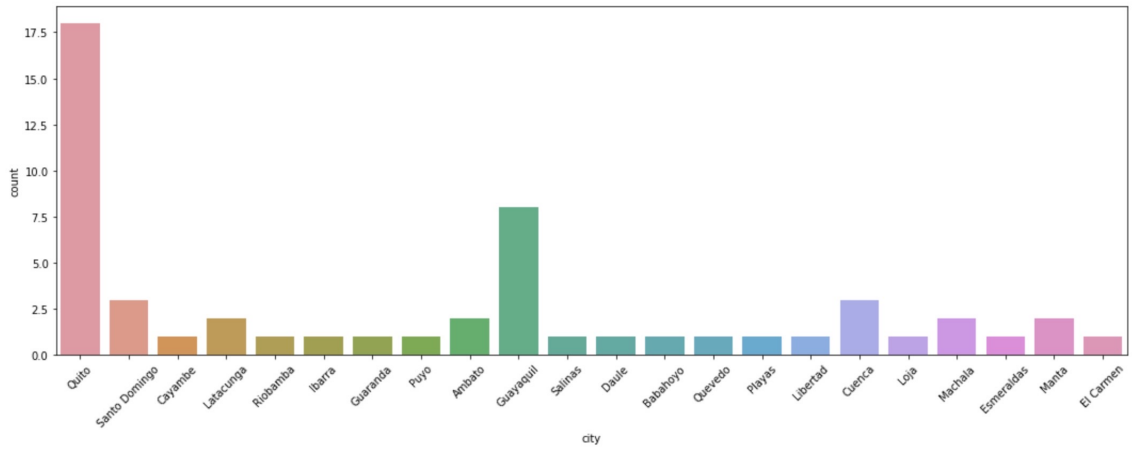


Figure 3: Count of city

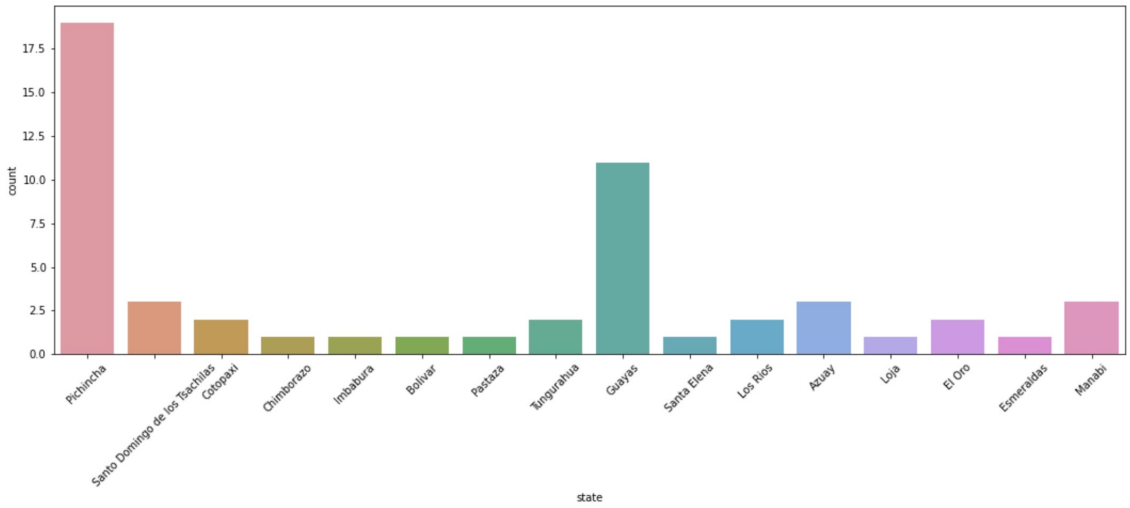
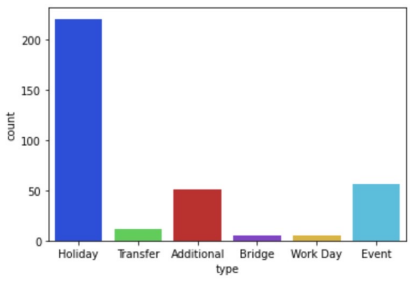
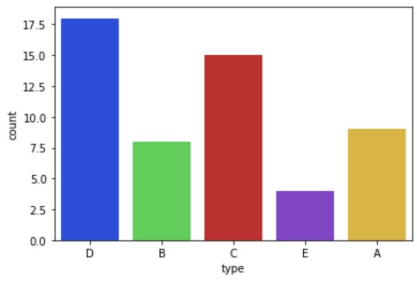


Figure 4: Count of state



(a) Count for each type of holiday



(b) Count for each type of store

Figure 5: Count of holiday and store

These plots helped us analyze the data that we would be utilizing and an idea on what is important for pre-precessing.

2 Solution

2.1 Feature Engineeering

To obtain a good performance on our model, we used multiple features from various datasets given in the competition. This was done based on the exploratory analysis we did on each of the datasets. Each of the features used were processed before the training stage. We generated the final dataset by appending store_nbr, family, date, sales from the train.csv dataset. Next, we used the date column to generate a Fourier trend series using a python package. Further, we used the dcoilwtico column to generate a new feature called ma_oil which is the rolling average of dcoilwtico. We then used this rolling mean to generate three lag features with a shift of 1, 2, and 3 respectively. This was appended to the main dataset along with one-hot coding depending on holiday type and day of the week. After obtaining the final dataset, we applied linear regression on this dataset and found the RMSLE for each family of items. The following results were seen:

Family	**RMSLE**
AUTOMOTIVE	0.259202
BABY CARE	0.066660
BEAUTY	0.267450
BEVERAGES	0.199187
BOOKS	0.026701
BREAD/BAKERY	0.125449
CELEBRATION	0.295910
CLEANING	0.204513
DAIRY	0.136196
DELI	0.108830
EGGS	0.147672
FROZEN FOODS	0.145027
GROCERY I	0.210306
GROCERY II	0.347753
HARDWARE	0.273930
HOME AND KITCHEN I	0.259483
HOME AND KITCHEN II	0.219104
HOME APPLIANCES	0.154737
HOME CARE	0.122103
LADIESWEAR	0.259483
LAWN AND GARDEN	0.216374
LINGERIE	0.400216
LIQUOR,WINE,BEER	0.612719
MAGAZINES	0.254086
MEATS	0.123141
PERSONAL CARE	0.137507
PET SUPPLIES	0.210464
PLAYERS AND ELECTRONICS	0.223845
POULTRY	0.122109
PREPARED FOODS	0.126234
PRODUCE	0.184626
SCHOOL AND OFFICE SUPPLIES	1.394414
SEAFOOD	0.253164

This shows that the family ‘SCHOOL AND OFFICE SUPPLIES’ has a very unexpected high value. To counter this, we generated a flag for school season. This scaled back the ‘SCHOOL AND OFFICE SUPPLIES’ RMSLE.

2.2 Model

For this particular problem, we have analyzed the data as a regression problem. In order to forecasts the sales we have compared different regression models like Linear Regression, Ridge Regression, Decision Tree, and Support Vector Machine. After comparing multiple regression models, we implemented a combined custom regressor using the solutions posted by KDJ2020(1) and BIZEN(2).

2.3 Improvising Score with other submissions

To improve the score for the competition,we decided to compare our submission obtained with few others submissions that were found in the store_sales_submissions. This idea was borrowed from the solution posted by MARK BABAYEV(3). As a starting point, we took the submissions he had obtained and compared our solution with his.

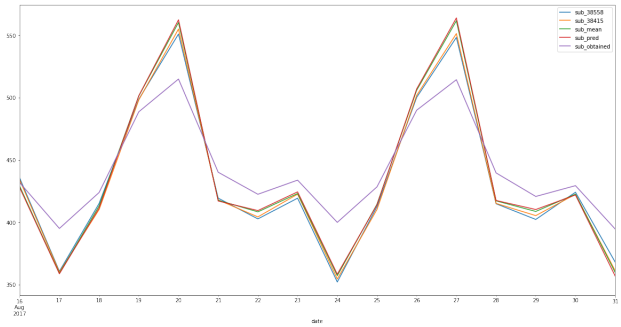


Figure 6: Comparing various submission

While his solution offered the mean of all the best solutions he had found, we decided to take mean of the solution we had obtained with his best solution, as ours were in between the best and worst solution. This also led to same result and score improvisation, as it gave the balance on the predictions where the best solution went wrong.

3 Evaluation Metrics

The evaluation metrics that is used in this competition to compare the predictions obtained by our team is Root Mean Squared Logarithmic Error (RMSLE). RMSLE is an extension of the typical Mean Square Error (MSE) and is helpful when data has large amounts of potential deviation.

The RMSLE is calculated as:

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

Where:

- n - is the total number of instances
- $\hat{y}_i$ - is the predicted value of the target for instance (i)
- y_i - is the actual value of the target for instance (i)
- log - is the natural logarithm

We tested various models on the train data and evaluated each of these models based on the evaluation metrics used in the competition.

The following results were obtained:

Model	RMSLE
Linear Regression	0.6942716503901292
Ridge Regression	0.7029830588882293
Support Vector Machine	0.7314329599162032
Decision Tree	0.7347383872229882
Random Forest Regressor	0.5213460644302836
Extra Trees Regressor	1.8341056718880664e-08
Custom Regressor	0.6855432492746178

References

[1] KDJ2020. October 29th 2021. Store Sales - Times Series Forecasting. Retrieved December 9th 2022 from `https://www.kaggle.com/code/dkomyagin/simple-ts-ridge-rf`

[2] BIZEN. December 3rd 2021. Store Sales - Times Series Forecasting. Retrieved December 9th 2022 from `https://www.kaggle.com/code/hiro5299834/store-sales-ridge-voting-bagging-et-bagging-rf/log`

[3] Mark Babayev. Octover 2022. Store Sales - Times Series Forecasting Retrieved December 9th 2022 from `https://www.kaggle.com/code/markbquant/stacking-upgini-darts`

[4] Carlo Lepelaars. July 2022. Understanding the metric: RMSLE. Retrieved December 12th 2022 from `https://www.kaggle.com/code/carlolepelaars/understanding-the-metric-rmsle/notebook`