
Developing an opinionated sales forecast.

Team 26

OCTOBER, 2022



TABLE OF CONTENT

Developing an opinionated sales forecast	2
Abstract	2
1.0 Introduction	2
2.0 The Process	3
2.1 Exploratory Data Analysis	4
Observation	5
Observation	5
Observation	6
2.2 Feature Engineering & Preprocessing	6
2.2.1 Lags	7
2.2.2 Rolling-Median	7
2.2.3 Label-Encoding	7
2.3 Modeling	7
2.4 Implementing Quantile regression in LightGBM	8
Observation	9
3.0 Deployment	9
4.0 Conclusion	9
5.0 References	10



Developing an opinionated sales forecast

Mercy Gichuhi, Olusoji Onigbinde, Daniel Odukoya, Lawson Iduku

Abstract

This project was carried out to develop a sales forecasting solution for department stores. Using data obtained from Walmart, the team selected the LightGBM (**Light Gradient Boosting Machine**) model to train the data. After the data was trained, we stored it in a pickle format and deployed the trained model data (in C.S.V. format) on a web based streamlit app built around the **final pipeline**. This application returns the forecasted sales in tabular format based on the State, Store and Category selected. The model selected was able to accurately forecast daily sales for the next 28 days. Using this solution will help stores accurately predict future daily sales and optimize the inventory of stores. It will reduce the incidence of items 'out of stock' thereby improving customer loyalty & profitability of department stores.

1.0 Introduction

Department stores like Walmart have uncountable products and money transactions every day. Because of their rapid transaction rates, keeping a balance between inventory and customer is most important, hence the need for sales forecasting. Sales forecasting involves predicting sales based on the amount of items people will purchase given the product features and the conditions of the sale (Armstrong, J. S. - 1999). Over the years a lot of research has gone into developing solutions that will help companies accurately forecast demand and sales. In Dalrymple's (1975) survey of marketing executives in American companies, 93% said that sales forecasting was "one of the most critical" or "a very important" aspect of their company's success. Furthermore, formal marketing plans are often supported by forecasts (Dalrymple 1987). A solution that can accurately forecast daily sales per item in a store will help businesses stock their stores with the required level of goods to meet up with customer demand, therefore ensuring business profitability and sustainability.

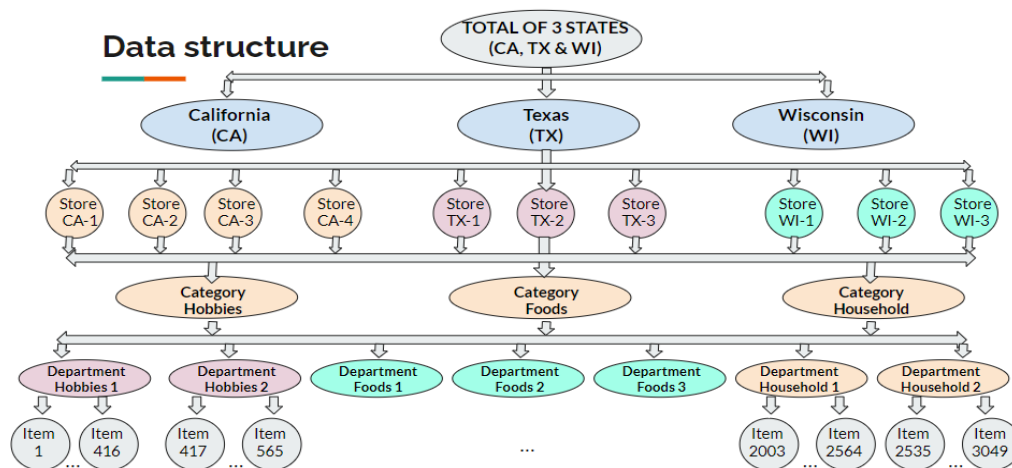
2.0 The Process

Using hierarchical sales data from Walmart, the world's largest company by revenue, we will forecast daily sales for the next 28 days. The data covers Stores in three American States (i.e. California, Texas, and Wisconsin) and includes items, departments, product categories and store details. In addition, it has variables such as price, promotions, day of the week, and special events. Together, this robust dataset was used to improve forecasting accuracy.

Fig 2.0.1 - Table explaining the procedures

ANALYSIS	DETAILS OF PROCEDURE
Data Ingestion	Loading the raw data (i.e. our .csv files) into a Jupyter notebook making use of the Pandas library.
Data Quality Testing	A data verification process done with programming code verifying data types of fields, length of characters, formats, and whether the values falls within an acceptable range.
Exploratory Data Analysis	Analyzing the Data via Histograms, bar charts, pie charts etc. to further understand our data.
Feature Engineering	Manipulation of our data set to improve machine learning model training, better performance and greater accuracy. Effective feature engineering is based on sound knowledge of the business problem and the available data sources.
Developing our Model	The best performing model was the LightGBM Model amongst a list of time series models. This was used to forecast the sales for the 28 days.
Confidence Intervals	Predicting future sales per item per store (using LightGBM Model) with uncertainty estimates for the next 28 days, i.e. per day predict the 50%, 67%, 95% and 99% confidence intervals.

Fig 2.0.2 - A general overview of our data structure



2.1 Exploratory Data Analysis

To begin with the building of our solution to the business problem indicated above the team began first with carrying out an Exploratory Data Analysis on the dataset provided after ingesting data into our Jupyter notebook. See below key insights garnered from our EDA.

Fig 2.1.1 – A plot showing price range per product category in each state.

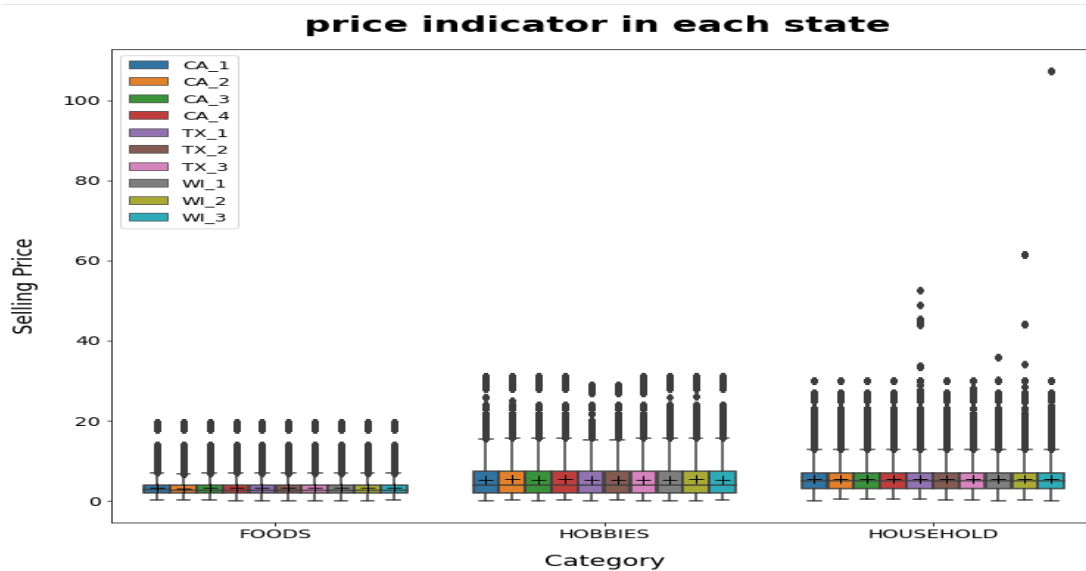
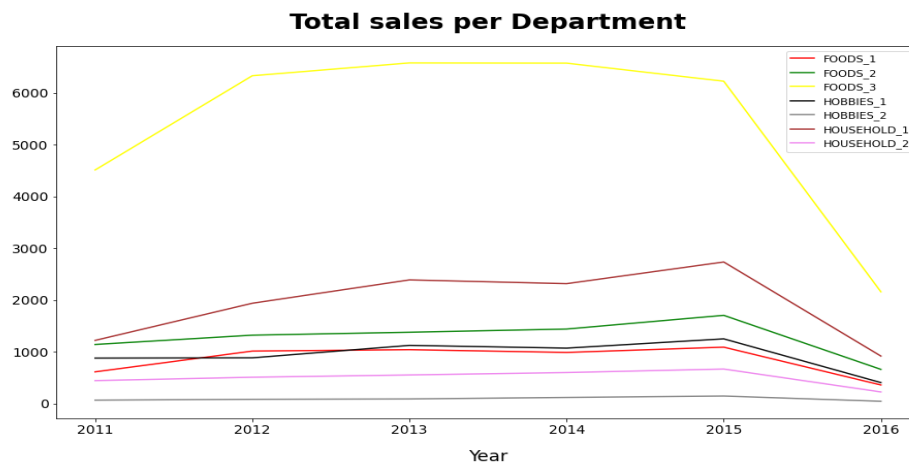


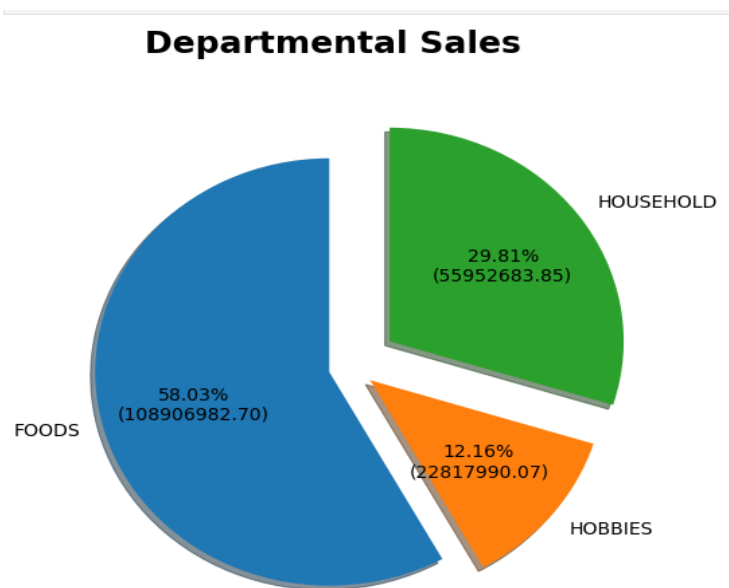
Fig 2.1.2 – A line plot showing sales for each department from 2011 to 2016



Observation

- Sales of FOOD_3 experience the highest sales amongst other departments.
- Sales of HOUSEHOLD_1 is the second highest sold out department.
- Sales of FOOD_2 is the third highest in the ranking of departments.
- Sales of HOBBIES_1 is the fourth highest in the ranking of departments.
- Sales of FOOD_3 is the fifth in the ranking of departments.
- Sales of HOUSEHOLD_2 is the sixth in the ranking of departments.
- Sales of HOBBIES_2 barely sells in the ranking of departments.
- Sales within 2015-2016 experiences a fall across the year.

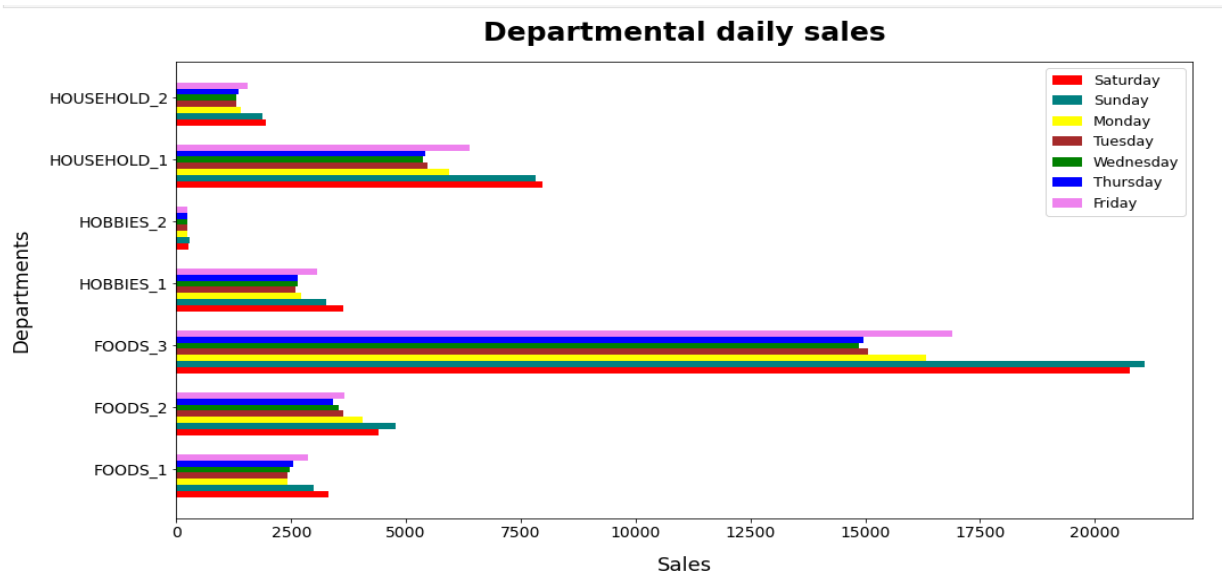
Fig 2.1.3 – A pie chart showing overall profit per product category



Observation

- Food items generated more income meaning it sold out more than other departments.
- Household items generated half of what food items made in income meaning it is the second on the list.
- Hobbies is the least when compared with Food and Household in sale/income yet it did 12% of the entire sales.

Fig 2.1.4 – A bar chart showing sales across departments for each day of the week



Observation

- Saturday and Sunday have the highest sales.

2.2 Feature Engineering & Preprocessing


To prepare the data for the modelling stage we carried out some feature engineering and preprocessing of the dataset.

The following feature engineering was done on our dataset.

- Replaced 'NaN' values by 'no_event' on event column.
- Added feature 'is_weekend' on the event column indicating if that day falls on a weekend or not.
- Adding feature 'month_day' which tells day of the month (1 to 31 depending on month).
- Adding feature 'month_week_number' indicating week of the month (1 to 5 depending on month).
- Adding feature 'events_per_day' indicating number of events on particular day (examples of events include super bowl, thanksgiving etc.).

We also performed down casting on the data to reduce the memory usage by the data set. This reduced the memory usage of our dataset by **55%**.

We performed a melt & merge process on the data set to make analysis of the data easier, melt() function is useful to massage a Data Frame into a format where one or more columns are identifier variables while all other columns considered measured variables are unpivoted to the row axis leaving just two non-identifier columns i.e. variable and value.



2.2.1 Lags

- Lag features are the classical way that time series forecasting problems are transformed into supervised learning problems.
- Lag is expressed in a time unit & corresponds to the amount of data history we allow the model to use when making the prediction.
- Here we have applied Lags on 'demand' column.
- The maximum Lags taken is 70 days
- We replaced 'NaN' in 'lags' features with 0

2.2.2 Rolling-Median

- Rolling is a very useful operation for time series data.
- Rolling means creating a rolling window with a specified size and perform calculations on data in this window which of course rolls through data.
- Here we applied Rolling-Median on 'demand' column.
- The maximum Window size taken is 42.
- We replaced 'NaN' in 'rolling_ median' features with 0.

2.2.3 Label-Encoding

- Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form.
- Machine learning algorithms can then decide in a better way on how those labels must be operated.
- It is an important preprocessing step for the structured dataset in supervised learning
- We dropped all the categorical columns because we already added corresponding columns with label-encoding

2.3 Modeling

After feature engineering & pre-processing we proceeded to carry out modelling.

We divided the data into Train/Test/Validation

- Train: From d_1 to d_1885
- Validation: From d_1886 to d_1914
- Test: From d_1914 to d_1941

We tested 4 different models and ended up selecting the Light GBM model as our model of choice because it had the lowest R.M.S.E. (Root Mean Square Error) score.

We know that the lower the R.M.S.E. Score, the more accurate our model is.

We calculated the R.M.S.E with the formula below:

```
R.M.S.E. = np.sqrt(((pred-value)**2).mean())
```

Where :

np – Numpy (i.e. the numerical python library used to make mathematical calculations)

sqrt() – a method in numpy used in finding the square root.

pred – Predicted values.

value – Actual values.

mean() – a method that calculates the average.

Models RMSE Score Comparison

- Moving Average: 3.28
- Linear Regression: 1.88
- Decision Tree: 1.86
- LightGBM: 1.83

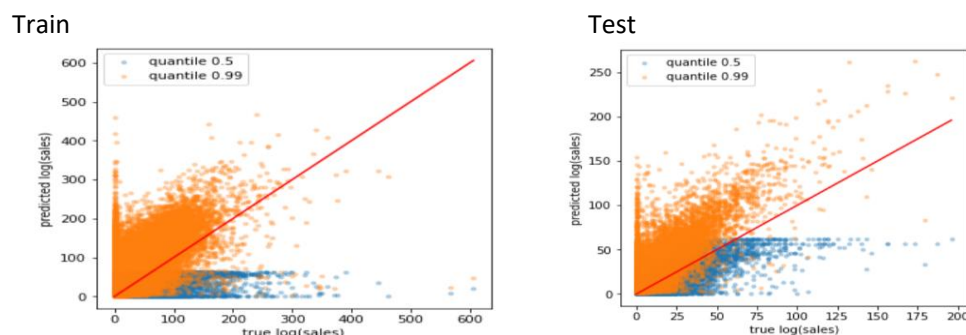
Light GBM is a gradient boosting framework that uses tree based learning algorithms, Light GBM is prefixed as 'Light' because of its high speed. Light GBM can handle the large size of data and takes lower memory to run.

2.4 Implementing Quantile regression in LightGBM

Quantile regression will allow us to create a model that estimates the values of the target variable (quantiles) at the given quantiles, in our case quantiles = (.5, .67, .95, .99). Knowing the values of the objective variable within these quantiles will allow us to estimate the distribution of the objective variable and as such we will be able to evaluate the uncertainty of the estimate.

We visualize our estimates within the highest and lowest quantiles.

Fig 2.4.1 – A distribution of our predictions using our train and test data



Observation

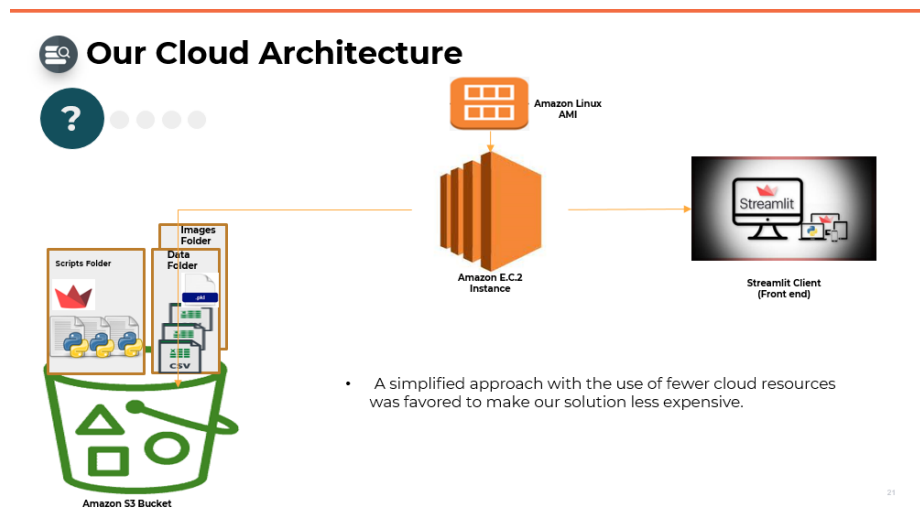
- We noticed, the larger the quantile, the larger the estimate. However, there are some cases where this is not the case which may mean there could be some error.

3.0 Deployment

After preliminary work on our dataset, we moved the entire solution (Dataset, Jupyter Notebooks, Streamlit Application and all other resources associated with the project) to the cloud. For this an object storage (S3 bucket) was launched on A.W.S. and all resources were saved inside the bucket. Then an E.C.2 instance was spun up, mounted onto the S3 bucket and finally the Streamlit app was loaded onto the E.C.2 instance along with all required dependencies to showcase a minimum viable product (MVP) of our solution that clients can use.

Streamlit is an open source app framework in Python language that helps us create web apps for data science and machine learning in a short time.

Fig 3.0.1 – A distribution of our predictions using our train and test data



4.0 Conclusion

In conclusion, with the use of our solution clients would no longer have to struggle with the issue of items 'out of stock'. Customer loyalty & profitability of department stores will be improved.

5.0 References

Armstrong, J. Scott, Sales Forecasting (July 20, 2008). Available at SSRN: <https://ssrn.com/abstract=1164602> or <http://dx.doi.org/10.2139/ssrn.1164602>

Dalrymple, D. J. (1975) "Sales forecasting: Methods and accuracy," Business Horizons 18: 69- Dalrymple, D. J. (1987) "Sales forecasting practices: Results from a U.S. survey," International Journal of Forecasting 3: 379-91.

<https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/163684>

<https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/163216>

<https://www.analyticsvidhya.com/blog/2018/02/time-series-forecasting-methods/>

<https://www.kaggle.com/tarunpaparaju/m5-competition-eda-models>

<https://mofc.unic.ac.cy/m5-competition/>

<https://dipanshurana.medium.com/m5-forecasting-accuracy-1b5a10218fcf>

<https://stackoverflow.com/questions/21608228/conditional-replace-pandas>

[https://www.kite.com/python/answers/how-to-drop-rows-with-all-zeros-in-a-pandas-dataframe-in-python#:~:text=DataFrame%20as%20\(df%20!%3D,non%2Dzero%20values%20using%20pandas.](https://www.kite.com/python/answers/how-to-drop-rows-with-all-zeros-in-a-pandas-dataframe-in-python#:~:text=DataFrame%20as%20(df%20!%3D,non%2Dzero%20values%20using%20pandas.)

<https://www.kite.com/python/answers/how-to-get-the-indices-of-rows-in-a-pandas-dataframe-which-satisfy-a-given-condition-in-python>

<https://stackoverflow.com/questions/21608228/conditional-replace-pandas>

<https://stackoverflow.com/questions/14059094/i-want-to-multiply-two-columns-in-a-pandas-dataframe-and-add-the-result-into-a-n>

https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html

<https://stackoverflow.com/questions/60595374/typeerror-cannot-insert-an-item-into-a-categoricalindex-that-is-not-already-an>

<https://stackoverflow.com/questions/60595374/typeerror-cannot-insert-an-item-into-a-categoricalindex-that-is-not-already-an>



<https://stackoverflow.com/questions/6170246/how-do-i-use-matplotlib-autopct>