MSiA 400 - Everything Starts With Data
Final Project: Predicting Profit for Dillard's.

TEAM 12: Alejandra Lelo de Larrea, Xin Shu, Yi Chen, Yiqing Cheng

Fall 2022

Contents

1 Executive Summary

Dillard's Inc. is one of the largest fashion retailers in the US; and thus, profit forecasting is a very important aspect of the business that allows the company to make accurate and timely long term investment decisions. Currently, Dillard's uses a SARIMAX model to estimate its monthly profits; however, due to the importance of this task for the success of the business, it is interested in developing a Machine Learning (ML) solution to get better forecasts. We have been hired by Dillard's to develop a ML model that will increase the accuracy of its current profit forecast. To this end, we use data provided aggregated it to the national level and augmented with macroeconomic indicators to train four different ML models: SARIMAX (baseline), Facebook Prophet, Lasso Regression and XGBoost. To improve our forecast, we also aggregated data to the state level and train both the Lasso Regression and the XGBoost models. Lasso is the model that best performs for this task, explaining 63% of the variability in the validation set for the national level data and 87% of the variability in the validation set for the state level data. To test our models "as in production" we forecast the profit of the last month in the sample, August 2005, and find that around 91%(66%) of the variability can be explained by the model for state level (national) aggregates.

2 About Dillard's[1]

Dillard's Inc. is an American department store chain founded in 1938 b William T. Dillard. It is one of the largest fashion retailers in the US, offering apparel, cosmetic and home selections from national and exclusive brand sources. Dillard's chain increased over the years through the acquisition of many local chains. Up to 2020, Dillard's had approximately 285 stores in 29 different states, having the majority of the stores in Texas and Florida, although lacking of presence in some states in the north of the country (see Figure 15 in Appendix A). The firm also has an online store and around 28 clearance stores. The Dillard's family retains control of the company, however some stocks are publicly traded on the NY Stock Exchange.

3 Importance of Forecasting Profits

Forecasting profits is a very important aspect of any business, and Dillard's is no exception. Having an estimate of profits will give the company a more reliable estimate of its future cash flow, and as a consequence, help Dillard's make informed decisions regarding to its long term investments (Fairfield et al., 2009). Up to today, Dillard's uses a SARIMAX model to forecast their profits, and is interested in investing on a machine learning solution that helps them forecast their profits in an accurate way.[2]

The impact of having accurate profit forecasts can be measured in several aspects of the business. For example, the company will be able to answer questions like: how much money can we invest in new inventory?, will we have enough cash flow so that we can acquire smaller retailers and expand the business?, are we going to have enough profits to remodel old stores?, can we afford a new marketing campaign tailored for a specific population sector?, will we able to purchase rights for a brand exclusivity? Moreover, a timely and accurate forecast of profits is crucial so that the company is able to adjust its strategy on time. For instance, if at the end of this quarter Dillard's discovers that the company will face a negative profit next quarter, the company will have time to reduce some costs and/or implement investment strategies.

To build a Data Science solution for Dillard's business problem, we followed the CRISP-DM methodology Wirth and Hipp (2000). Our goal is to use national aggregations of the daily transactions data provided by the company to forecast daily profits. We solve this problem by estimating time series models, such as SARIMAX and FB-Prophet, as well as using Lasso Regression and XGBoost algorithms. As a second step, we agregate data at a state level to use a blocking design and improve our estimations. The following sections elaborate on the data cleaning, analysis and feature engineering (section 4), modeling (section 5), results (section 6) and ROI analysis for this project (section 7).

[1] Information retrieved from Dillard's website and from Wikipedia on December 1st, 2022. [2] This is an assumption of the project. We estimate the SARIMAX as part of our models and use it as baseline model. See Section 5 for more details.

4 Data

The data provided by the company spans from August 1st, 2004 to August 27th, 2005. It includes each transaction at each Dillard's store, as well as information about every product, the cost and retail prices of every product in every store, as well as information on the location of every store. It should be noted that the data provided did not correspond one to one to the schema. See appendix B for details on data cleaning. We believe that part of the trend in profits should be explained by the macroeconomic conditions of the country. Therefore, we enriched the data set with macroeconomic aggregators to include them as control variables. Specifically, we included a measure of inflation, unemployment and the Fed Funds Target rate for the sample period.[3].

4.1 Exploratory Data Analysis

We performed a first exploratory data analysis (EDA) with the raw data set to get a better understanding of the information provided and to detect possible problems with the features of interest. Appendix C shows some highlights of the EDA and the complete EDA for the raw data set can be consulted in the Github repository for this project. Unfortunately, we were not provided with a SKU dictionary, thus giving a deeper interpretation of the data is difficult. For example, we just know that product 4628597 is the most sold but we can not know what type of article it is or if the top 10 articles sold are from the same type. Some relevant findings for the rest of our analysis include: i) every row corresponds to only one sold item (i.e. quantity = 1 for ever row), this is relevant when calculating the profit of the products; ii) the total amount in transaction (variable amt) needs to be treated carefully because it might be repeated for several observations; iii) there is an "original price" and also a "retail price" for every product, we assume that original price is the suggested price of the article and that retail price is the actual price paid by the customer in the transaction, thus some articles were sold at discount; iv) we removed those rows corresponding to returns instead of purchases (8.3% of the data), v) the time series plot of the total number of products sold each day has some expected peaks (like dates around Thanksgiving), but we also encountered an unexpected peak on February 26th, 2005 (see figure 4) which, after researching, we found corresponds to the "President's day sale" one of the days of the year with the highest discounts.

4.2 Feature Engineering

For every observation in the transactions table, we calculate the profit for each product as $\pi_{kt} = p_{kt} - c_{kt}$ where p_{kt} is the retail price for product k at day t, and then aggregate data at a daily level.[4] Figure 13 in Appendix D shows the corresponding daily time series of Dillard's profit through the sample period. We also created variables to account for the number of stores selling each item each day, the number of different products sold, the average quantity sold per transaction, maximum, minimum and average prices and costs, average cost per transaction, discounts, as well as the number of different vendors, departments, cities and states involved. Moreover, since we are using time series analysis, we created several lags of our features and for the response. Our data has daily frequency and exhibits some seasonality. We believe that this seasonality could be due for the day of the week (Friday, Saturday and Sundays might have more sales than the rest of the week), weekly, biweekly or monthly. Thus, we calculate lags 1, 2, 3, 4, 5, 6, 7, 14, 21, 28. Lagging the variables will also prevent data leakage problems in the moment of estimation. To forecast the total profits for Dillard's, time-related features, such as year, quarter, and week of the year, are generated to capture the time effects for general machine learning models. In addition, we standardized the dataset to ensure unbiaseness estimation from the regression models.

5 Modeling

To evaluate model performance, data are split into three parts: training, validation, and testing. Data from August 1st 2004 to June 1st 2005 is used for training the different models. It is worth mentioning that when training the model, we use time series cross-validation to avoid data leakage, this methods validates the current model performance in fixed time intervals. As the validation set we use data between June 1st and July 31st 2005, this set is used to select the best model among each type. Finally, the best model of each type is retrained

[3] From the FRED economic data set we select the "Sticky Price Consumer Price Index less Food and Energy" as a measure of inflation, the unemployment and the Fed Funds Effective Rate [4] It is worth noting that we would ideally want to estimate monthly profits. However, our sample data consists of only 13 months, thus monthly aggregates will give us a small training sample and, probably, unreliable predictions. Instead we decided to model daily data to account for more variability. Monthly profit predictions can be achieved by adding daily profits.

with data spanning from August 1st, 2004 to July 31st, 2005 and use the period August 1st - August 27th, 2005 for testing and selecting the final model. Figure 14 in Appendix E shows the response variable split in the three aforementioned periods.

We used two types of ML models for addressing the problem: time series and general ML models. The time series models include SARIMAX, the time series model of excellence to account for seasonality and including observed features, and the recently developed (2017) Facebook Prophet Model for time series. As for the general ML algorithms, we will use Lasso Regression to allow for feature selection and XGBoost.

After training the different models to forecast profits at a national level, we found that the best model could only account for around 60% of the variability in the response. Thus, as a way to account for more information in the data but keep the daily forecasts, the general ML models (Lasso and XGBoost) were also trained but with data aggregated daily at a state level. The following subsections summarize the estimation process for each algorithm and more details can be found in Appendix E.

5.1 SARIMAX

Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors (SARIMAX) is an updated version of the ARIMA model family which includes seasonality and exogenous variables. We used Auto ARIMA package in Python for the estimation. We set the stepwise as the search method over all possible models and seasonal orders to allow the algorithm to automatically select the best combination of parameters. We allow parameters to range from 1 to 5.[5] Auto ARIMA works by conducting differencing tests. We choose Augmented Dickey-Fuller(ADF) test as the test method and the algorithm will determine the order of differencing d, and then fit models within the defined ranges. Auto ARIMA takes into account the AIC and BIC values generated (as you can see in the code) to determine the best combination of parameters. For more information, consult the package instruction.

5.2 Facebook Prophet

FB Prophet was built to perform better for forecasting time series that show multiple seasonality. When building a Prophet model, we need to select the appropriate *seasonality* and *extra_regressors*. For the selection of the *seasonality* parameter, since we only have one-year of data, we added a weekly seasonality as opposed to a yearly seasonality to catch potential patterns during a week. We also added the holiday features supported by the Prophet model. This is important because we were analyzing retailers' profits which would be influenced by holiday events a lot, for example the Black Frida sale or the President's day sale. As for the *extra_regressors* parameters, we selected those features with the highest correlation to total profits. This include *minorigprice_1*, *ndept_1*, *totalretail_1*, *nvendors_1*, *totalcost_1*, *avgprofittrnsact_1*, *nsku_1*. We tune the model selecting from the following combinations of hyperparameters. *seasonality_prior_scale* to account for seasonality's flexibility. *holidays_prior_scale* to change the flexibility of holiday effects. Since the total profit has as increasing trend in the final part of the sample, thus we also chose the *changepoint_prior_scale*, which will allow the model to determine the flexibility of the trend and, in particular, how much the trend changes at the trend *changepoints*.

5.3 Lasso Regression

Lasso regression is a technique used for feature selection. It includes an extra regularization term in the loss function of linear regression to penalize for model complexity. This is, $Loss = Error(Y - \widehat{Y}) + \alpha \sum_1^n |w_i|$. After standardizing the data, the Lasso regression model is estimated using cv to automatically find the optimal regularization parameter. This model is used to forecast profits at both national and state levels.

5.4 XGBoost

XGboost is a variate gradient boosting model. It is an efficient, scalable, and accurate classic modeling method based on trees. For the profit forecasting at national level, XGBoost is tuned on *max_depth* (3,5,7,9), *min_child_weight* (1,4,6), *learning_rate* (0.1, 0.01, 0.001), and *n_estimators* (100,500,1000) to avoid overfitting. For the state level profit forecasting, XGBoost is tuned on *max_depth* (3,5,7,9), *min_child_weight* (1,4,6), and *learning_rate* (0.1, 0.01, 0.001).

[5] Stepwise is the suggested method according to the package manual.

6 Results

Table 1: Model Performance Comparison Measured by R^2

	National Level Data				State-Level Data	
	SARIMAX	**FB Prophet**	**Lasso**	**XGBoost**	**Lasso**	**XGBoost**
Validation	0.5716	0.5381	0.6341	0.1953	0.8796	0.8756
Test	0.626	0.3592	0.6626	—	0.9188	0.8639

After training and fine tuning the models described in section 5, we use the validation set to test the hyperparameter selection and compare the best model of each type. The first row in Table 1 shows the validation R^2. As can be observed, the Lasso regression model is the one that has the better performance for predicting Dillard's profits with national aggregated data, giving an R^2 for the validation set of 0.6341. When models are estimated using state-level data, Lasso and XGBoost perform better than national-level estimates. Lasso has R^2 of 0.89 and XGBoost has R^2 of 0.87. When adding state-level features model performance increases because we have a blocking design that reduces variability within each block. Moreover, as data are grouped by state level, there are more samples in the training set, which helps to improve model performance. Table F in Appendix F shows the top 5 important features when estimating the Lasso regression. As can be observed, both models do account for a weekly seasonality in profits and also in different seasonality for the average profit per transaction, the average number of vendors per store, and the original price of the article.

Lastly, we retrained the best models using the data from August 1st 2004 to July 31st 2005 to forecast the total profit of Dillard's during August 2005. Table 2 shows the estimated profit given by each model and its comparison with the observed value. Models trained with national aggregates underestimate the total profit, while models trained with state level aggregates over estimate Dillard's profit in August.

Table 2: Profit Forecast for August 2005

	Model	**Predicted Profit Aug. 2005**	**Over/under Estimation**
	Observed	$55,934,830.00	—
National Agg.	**SARIMAX**	$47,958,258.81	-$7,976,571.19
	Prophet	$47,268,340.64	-$8,666,489.36
	Lasso	$ 51,270,190.49	-$4,664,639.51
State Agg.	**Lasso**	$ 58,167,441.33	$2,232,611.33
	XGBoost	$ 56,414,563.06	$479,733.06

7 ROI Analysis

Table 3 shows the calculated ROI for our predictions in August 2005. It is calculated as $ROI = \frac{G-C}{C}$ where G are the total gains and C the total cost. We assume that Dillard's will use the estimated profits to invest in a marketing campaign. For this industry, every dollar spent in marketing has a return of 9%, which is higher than Dillard's IRR (7.35%). By implementing the predictions of our ML model, Dillard's will be able to invest $3,311,932.00 compared to their current forecasts. As a result, the expected gains for the company will be approximately $215,000 greater than if using the Lasso model compared to our baseline model.[9] Once subtracting the total cost of the project we obtain an ROI of 14.28% indicating that Dillard's will have a positive return if investing in developing our model. It should be noted that our ROI is subject to risks. For instance, our estimations depend on macroeconomic indicators, such as inflation, unemployment and target interest rate; therefore, economic and financial conditions might skew the real results.

Table 3: ROI Analysis for August 2005

Profit	$55,934,830.00
Baseline	$47,958,258.81
Model	$51,270,190.00
Extra Profit	$3,311,932.00
Market Interest Rate[6]	2.5%
IRR[7]	7.35%
Marketing Rate of Return[8]	9.00%
Total Gains	$215,276.00
Duration(Months)	5
FTE	3
Annual Salary	$15,000.00
Salary Payment	$187,500.00
Computing Hours	8760
Cloud Per Hour	$0.10
Total Cloud Cost	$876.00
Total Cost	$188,376.00
ROI	**14.28%**

[9] Gains are calculated as $G = I * (9\% - 2.5\%)$ accounting for the fact that the best substitute to investing in our proposal is will be investing in government bonds.

Additionally, there is a risk that the marketing campaign is not successful and returns out of this project are less than expected. Other projects into which invest profits should be considered before making a final decision.

8 Conclusions

Through this project we aim to forecast Dillard's total profits by building ML models that would increase the accuracy its current predictions and, as a consecuence, help Dillard's make better investment decisions for the long term. To this end, we aggregated the transactions data provided to the national level and augment the data set with macroeconomic variables to account for the economic and financial environment. We then trained SARIMAX, FB Prohet, Lasso and XGBoost models on data on national level data for the period August 1st 2004 - May 31st 2005 using time series cross-validation to select the best hyperparameters and comparing models with the R^2. We use the period June 1st, 2005- July 31st, 2005 as validation to select the best models. Even though the performance of the model was good, we decided to take advantage of more granular data aggregating daily transactions at a state level and train as Lasso and XGBoost models on this data. Having a blocking design increased the sampled size and helped reduce volatility in the data and, thus, to get better estimates.

As is shown in 1, on the national level, Lasso performs the best for forecasting Dillard's profit with an R^2 of 0.6341, followed by SARIMAX with an R^2 of 0.5716. On the state level, Lasso remains the best model with an R^2 of 0.8796 on the validation set. As for the prediction of total profits in August 2005, the model that best performs is Lasso for state-level data. Models trained on nationwide aggregated underestimate the observed profit and models trained on statewide aggregated data overestimate this measure. It should be noted that, when forecasting profits it is better to have a more conservative estimate (i.e. underestimation). For instance, when overestimating profits the company could engage in long term financial commitments based on its expected cash flow that will eventually not be able to afford impacting Dillard's ability to sustain competitiveness and production plans. Although the R^2 of models on nationwide data, based on the results for the test set, it is our advise that the Lasso model trained with national level data be the one used when estimating profits.

As for the ROI analysis, we estimate a rate of return on the investment of this project of 14.28% based on our lasso model compared to the baseline model (SARIMAX). This is solid evidence that Dillard's will have a positive return from implementing data driven solutions for the business.

9 Future Developement

As there is an unusual upward trend in the last few months of our data, we tried to include three macroeconomic features to improve the prediction performance. However, in some models such as SARIMAX, the macroeconomic features did not improve the model performance significantly. In future analysis, we could include more external features that help account for variability in the data. We can also try to obtain the stock data of Dillard's and some measures of risk as well as inflation on producers prices. Also some data on Dillard's inventories could help improve estimations by taking into account inventories management. since we have a limitation of small number of samples, time series models don't perform as well as it should be. It could be helpful to obtain more history on transactions to improve model performance and account for yearly patterns on the data as well as the impact of holidays or special sales dates.

Bibliography

Fairfield, P. M., Kitching, K. A., and Tang, V. W. (2009). Are special items informative about future profit margins? *Review of Accounting Studies*, 14(2):204–236.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–39. Manchester.

[7] The Market Interest Rate is the average Federal Funds Rate between 2004 and 2005 28 [8] The Internal Rate of Return for Dillard's is estimated from Dillard's Weighted Average Cost of Capital since a retailer's cost of capital is used as the required rate of return.
[9] The Required Rate of Return of marketing strategy is estimated from Nielson's Maximize the Return on your Advertising Spend.

A About Dillard's

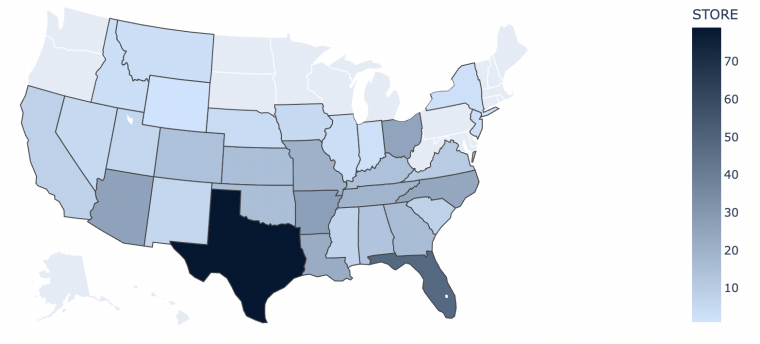

Figure 1: Number of Dillard's stores by state (2004-2005). Own calculations with information of the data set.

B Data Cleaning

The data provided included 5 tables with information on transactions (TRNSACT), characteristics of every SKU (SKUINFO) and the associated departments (DEPTINFO), cost and retail price of every product (SKSTINFO) as well as information of every Dillard's store (STRINFO). See figure 2

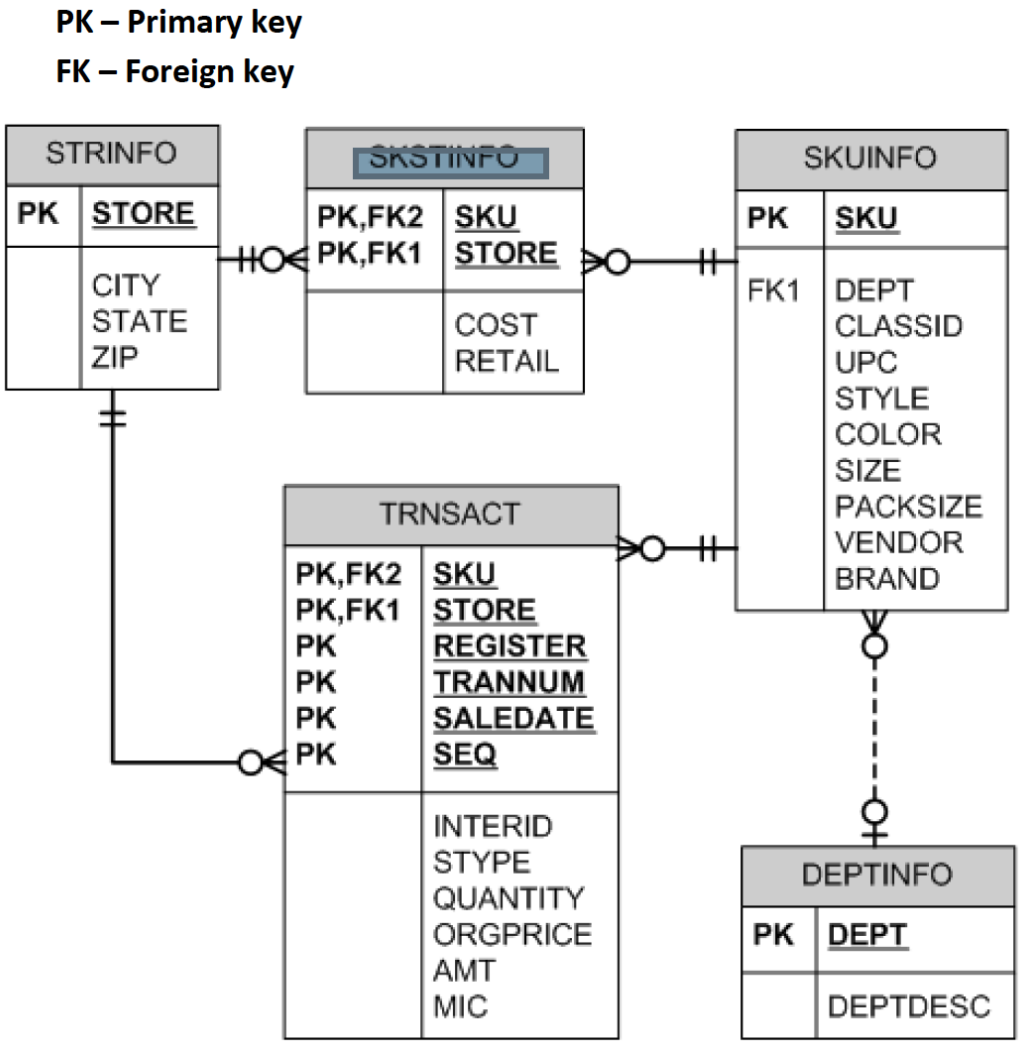

Figure 2: Number of Products Sold

However, the data on the files did not correspond one to one to the schema. For instance, all columns contained an additional column filled with zeros or ones. We assume this was a mistake in the data and that it had no interpretability and ignore it for the rest of the analysis.

We also found that the TRNSACT table had columns in a different order than the schema. Even though the data provided did not have headers, based on some datatypes of the dataset we were able to figure out that the order of the columns was different to the schema and did the best to our knowledge to match the columns to the corresponding feature. However we must recognize that it is possible that some columns are mistakenly labeled. Also there were two columns with the exact same values for every entry expect for 13 observations. We decided to remove the duplicated column and removed the duplicated observations.

As for the SKUINFO table, we found that the raw data was very dirty. The table was supposed to have 11 features, however we found that several observations had more than 13 columns. We used bash to extract those observations that had the wrong number of features and manually analyzed which was the problem. In most cases, the problem arrised from the company name or from the colors features. Company names with a comma as part of the name we causing and extra column and, for the colors, some products included different colors and those were separated by commas, causing an extra column in the csv files. We also find extra columns with no information. We used bash to replace the extra columns in company name and colors, removed the extra empty entries and fixed the remaining few errors by hand.

The following shows a description of the common errors in the database:

Table 4: Errors in SKUINFO table

Original value	Replacement
BB CO, I	BB CO I
F-50, LL	F-50 LL
WEMCO, RE	WEMCO RE
WEMCO,RE	WEMCO RE
MARAN, I	MARAN I
SUN B, L	SUN B L
AHA, INC	AHA INC
TROO,LLC	TROO LLC
YSL,ARRO	YSL ARRO
VANS, IN	VANS IN
NVY,CHTI	NVY-CHTI
RVB=RED, RY	RVB RED RY
RED, NAVY	RED NAVY

Once the data was clean, we created SQL tables in Postgress to store the information of the database in the MSiA servers.

C Exploratory Data Analysis

This appendix shows some of the most interesting findings of the intial EDA. You can get the complete EDA from the Github repository for this project.

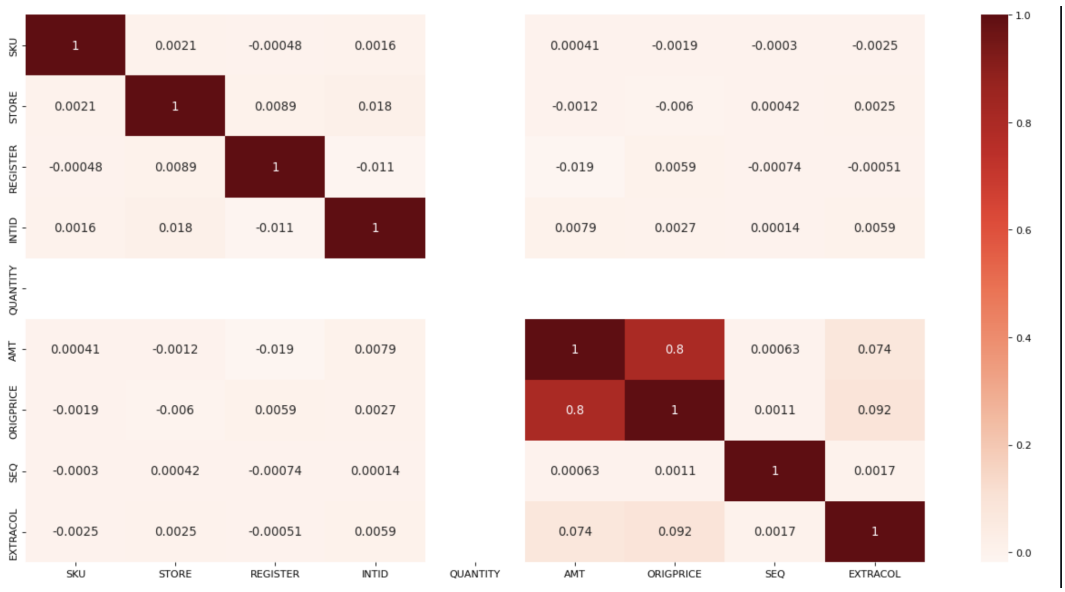

Figure 3: Feature correlations for transactions table

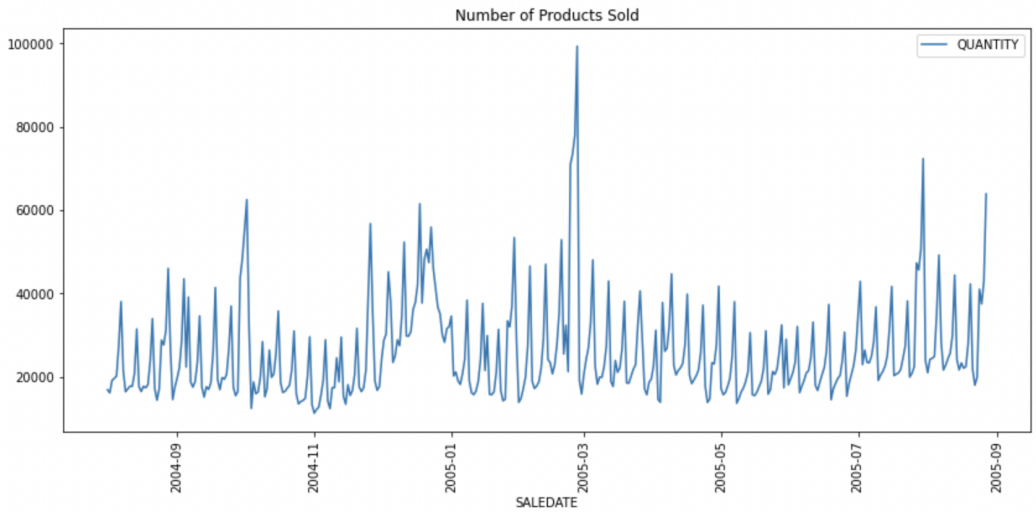

Figure 4: Number of Products Sold

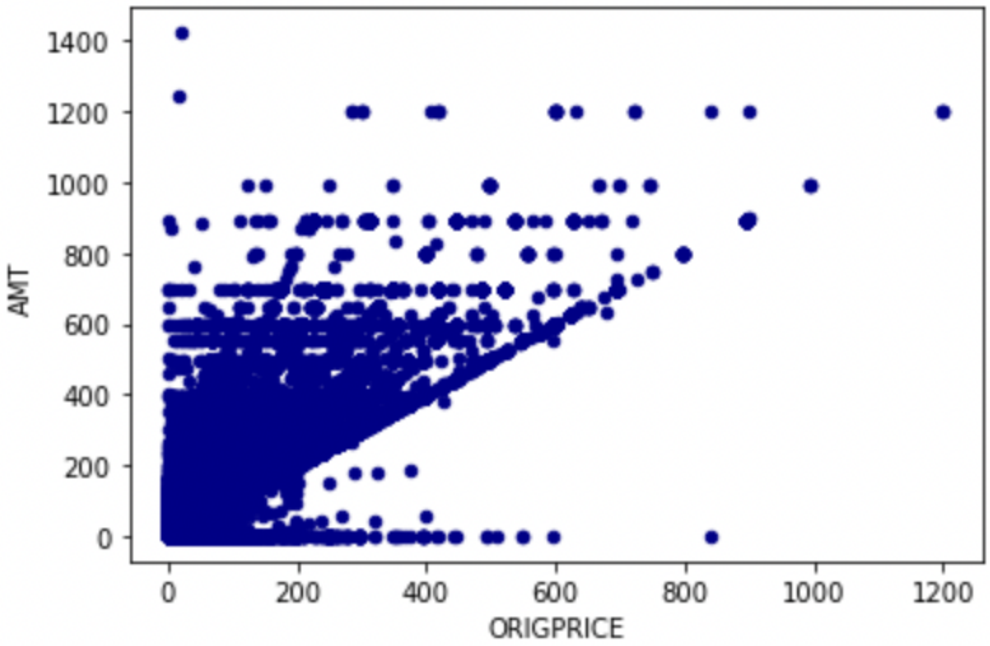

Figure 5: Total Amount vs Original Price

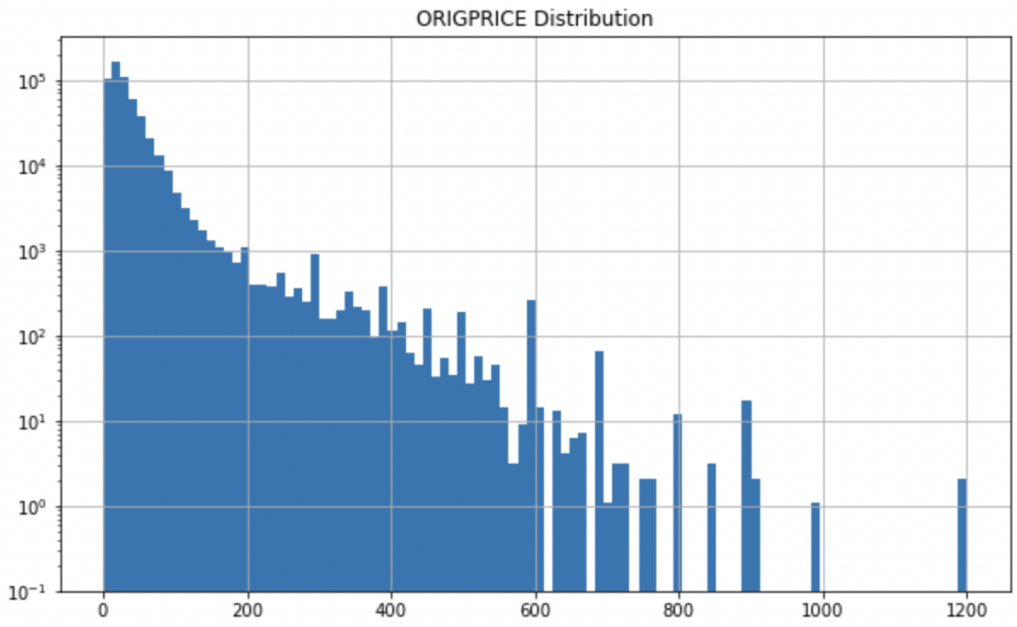

Figure 6: Original Price Distribution

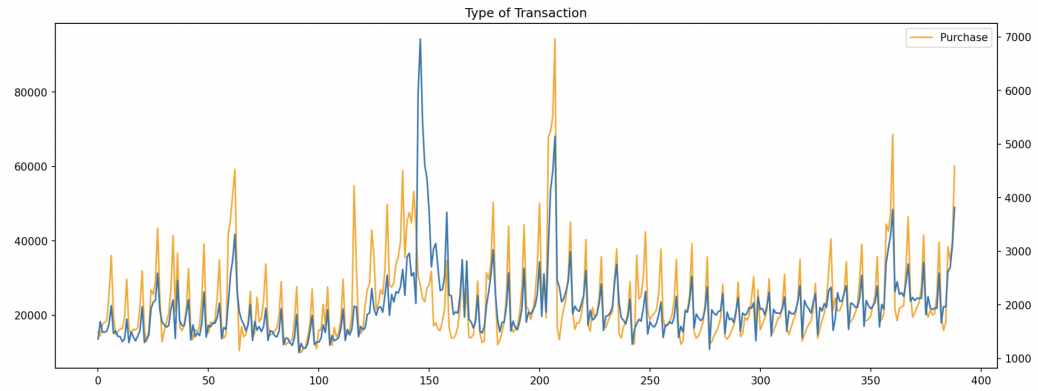

Figure 7: Transaction Type

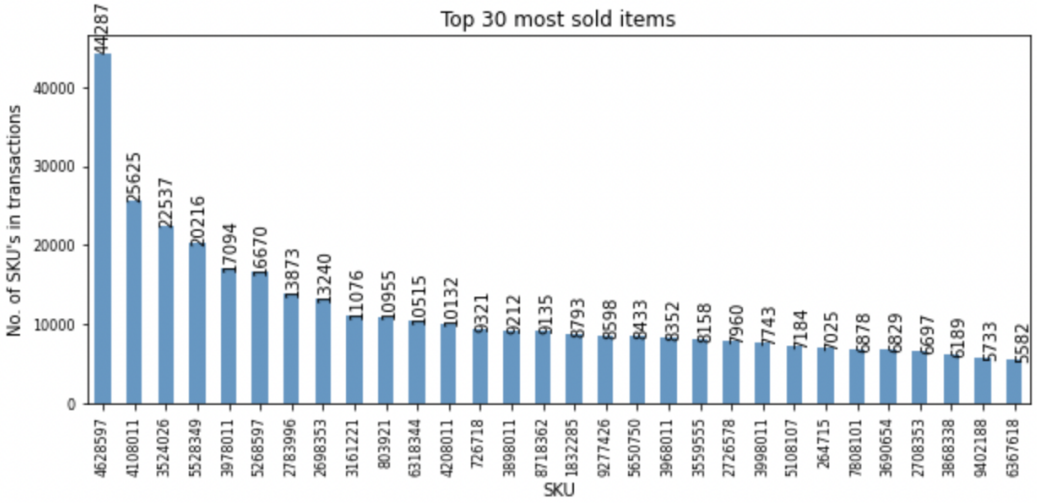

Figure 8: Top 30 articles sold

Figure 9: Wordclouds for colors of products

Figure 10: Sizes of articles sold

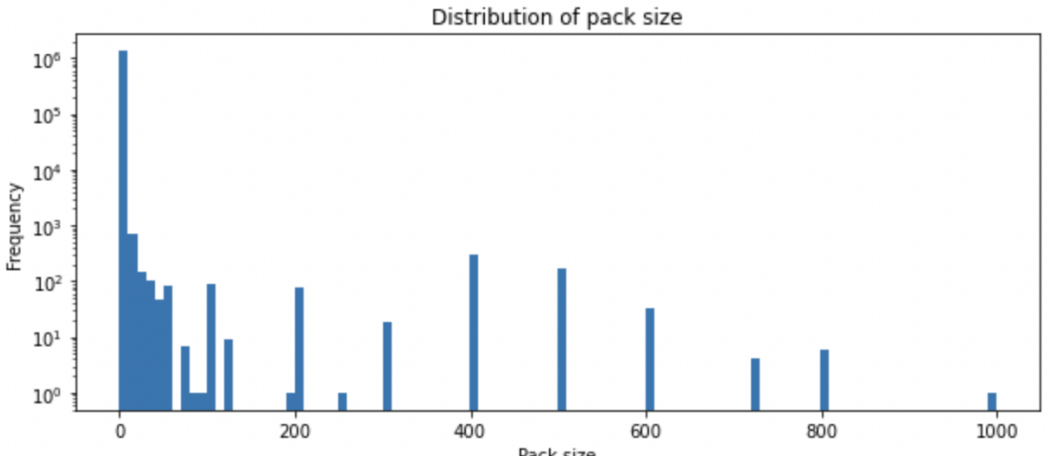

Figure 11: Packsize distribution

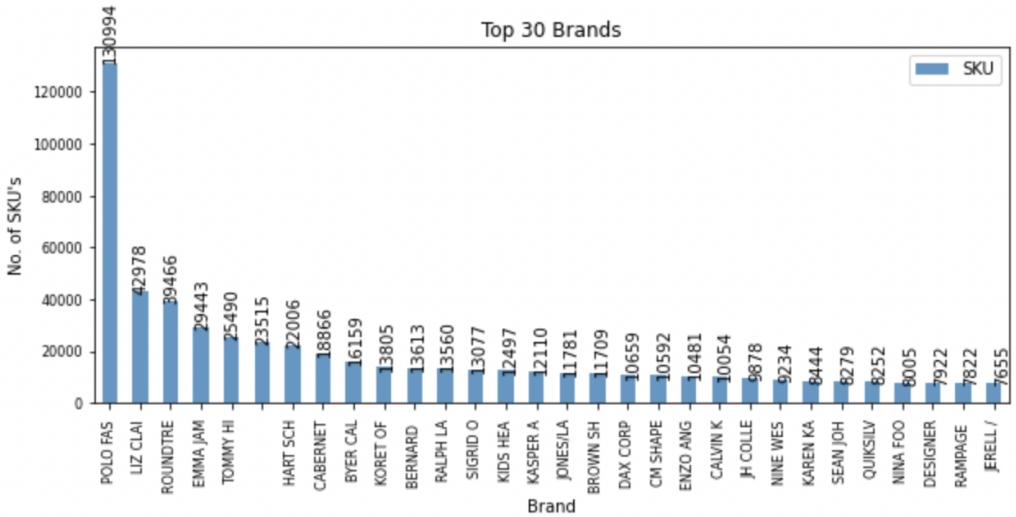

Figure 12: Top 30 Brands sold in Dillard's

D Feature Engineering

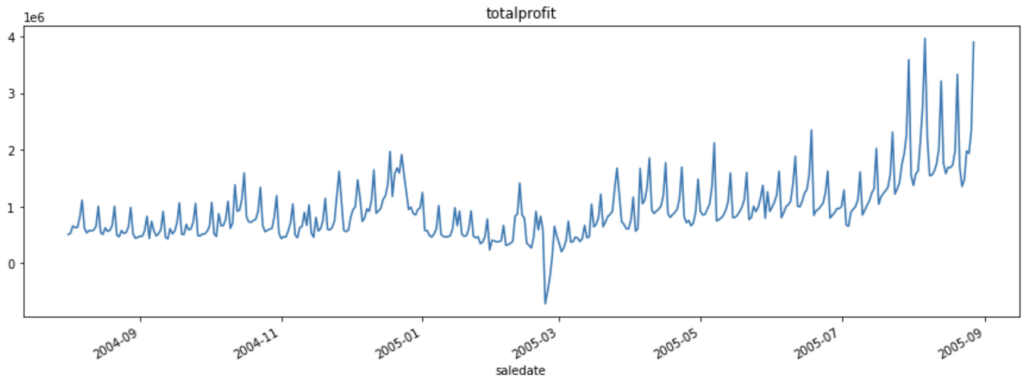

Figure 13: Evolution of profits through sample period.

E Modeling

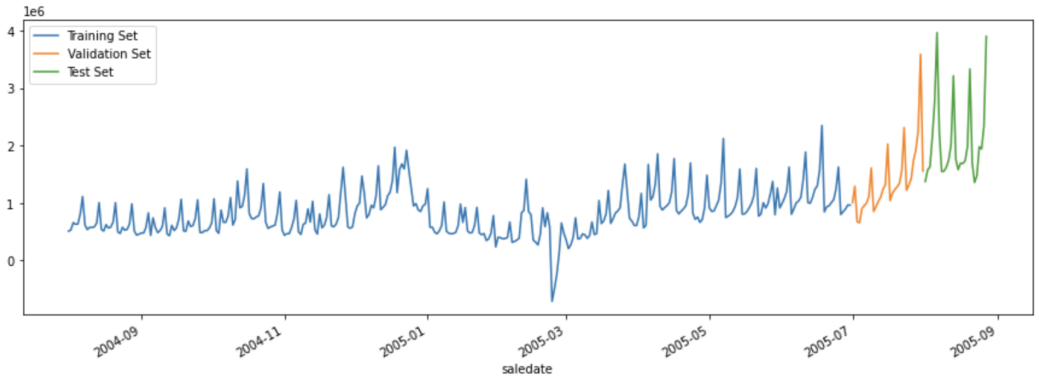

Figure 14: Response variable split in train, validation and test.

E.1 SARIMAX

SARIMAX is an extension of ARIMA class of models. SARIMAX models allow for differencing data by seasonal frequency as well as by non-seasonal differencing. The model also takes exogenous variables into account. SARIMAX has 7 parameters, which are p(the non-seasonal autoregressive order), d(non-seasonal differencing), q(non-seasonal moving average order), P(seasonal AR order), D(seasonal differencing), Q(seasonal MA order), and S(length of repeating seasonal pattern). The model is then represented as:

$$d_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{P} \phi_{nt-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t \tag{1}$$

For further details about SARIMAX models, see the articles about End-to-End Time Series Analysis and Forecasting and Time Series Forecasting with ARIMA, SARIMA and SARIMAX.

For feature engineering, we first created lag 1, lag 2, lag 3, lag 4, lag 5, lag 6, lag 7, lag 14, lag 28, hour, day of week, quarter, month, year. Then, we normalized every x feature.

To select the exogenous variables, we calculated the correlation coefficients between each x features and the total

profit along one year. By selecting features that have correlation coefficients greater than 0.7, we decided to include the number of sku, total quantity, average quantity, total amount, average amount, total original price, average original price, total retail price, average retail price, total cost, average cost and average profit in our model.

	name	coef
321	totalprofit	1.000000
240	totalprofit_7	0.842549
248	avgprofittrnsact_6	0.784076
241	totalprofit_14	0.722531
234	totalprofit_1	0.702355
114	totalretail_7	0.701934
149	avgretailtrnsact_6	0.671608
115	totalretail_14	0.629148
239	totalprofit_6	0.607038
116	totalretail_28	0.597122

Figure 15: correlation coefficient between Xs and the total profit

We noticed that there is an upward trend in the model since July 2005 that can't be captured from the previous data, so we include macroeconomic features(the measure of inflation, unemployment and the fed funds target rate) into the model. However, macroeconomic features didn't help with the prediction, since the $\mathbb{R}^2$ dropped from 0.5755 to 0.5716.

In final SARIMAX model with macroeconomic features, we got the best parameters: p=5,q=0,d=2,P=2,Q=1,D=1 by running auto arima in python.

For the final results, we got $\mathbb{R}^2$ of 0.626, MAE of 309516.7039, MSE of 193542369379.6164 and RMSE of 439934.5058. The total prediction of profit for August 1 to August 27 (test dataset) is 47958258.81732269.

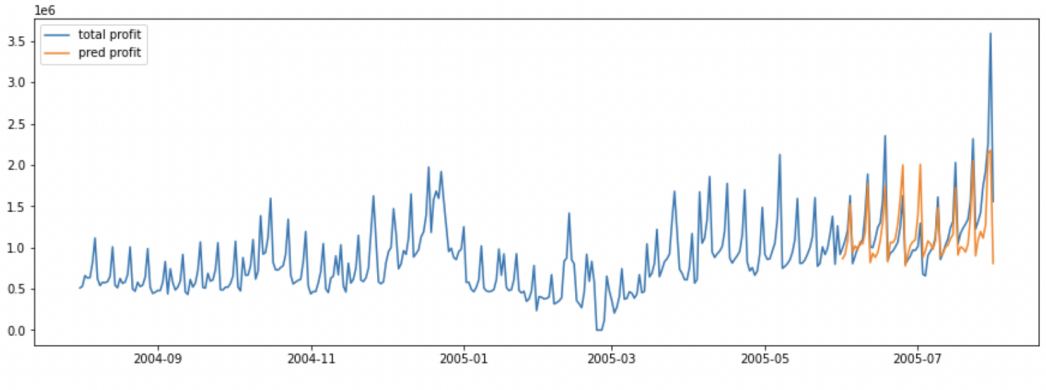

Figure 16: Total Profit Prediction from SARIMAX with macroeconomic features in JuneJuly 2005

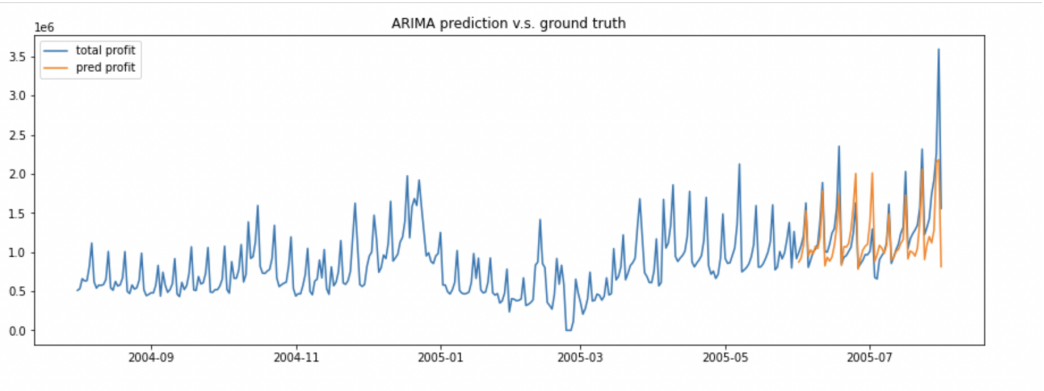

Figure 17: Total Profit Prediction from SARIMAX with non-macroeconomic features in June and July 2005

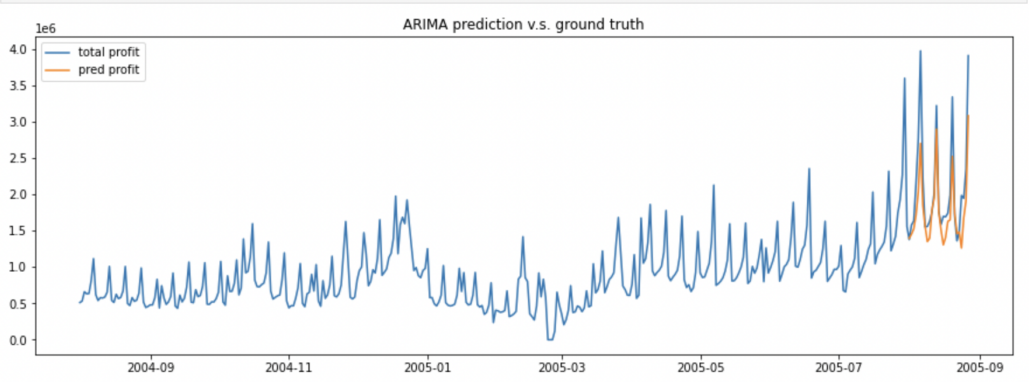

Figure 18: Total Profit Prediction from SARIMAX with macroeconomic features in Aug 2005

E.2 FB Prophet

Facebook Prophet is an open-source algorithm for generating time-series models that uses a few old ideas with some new twists. It is particularly good at modeling time series that have multiple seasonality. At its core is the sum of three functions of time plus an error term: growth(t), seasonality s(t), holidays h(t), and error ϵ_t :

$$y_t = g_t + s_t + h_t + \epsilon_t \tag{2}$$

For further details about FB Prophet model, see the articles about Prophet | Forecasting at scale. - Meta Open Source

In our project, we conducted a correlation analysis between total profits and multiple features first and put the features that had strong relationships with total profits (here we select those with the absolute value of correlation larger than 0.3) into the Prophet model as extra_regressors (which are minorigprice_1, ndept_1, totalretail_1, nvendors_1, totalcost_1, avgprofittrnsact_1, nsku_1). The model will automatically standardize our extra_regressors.

totalprofit	1.000000
totalprofit_1	0.704782
minorigprice_1	0.533894
ndept_1	0.526977
totalretail_1	0.512834
avgretail_1	0.407768
nvendors_1	0.400957
totalcost_1	0.386525
avgprofittrnsact_1	0.351141
nsku_1	0.322482

Figure 19: Features' Absolute Value of Correlation With Total Profits

After that, we tuned models with different flexibility of seasonality and change_points and finished our models with R-square of 0.4694.

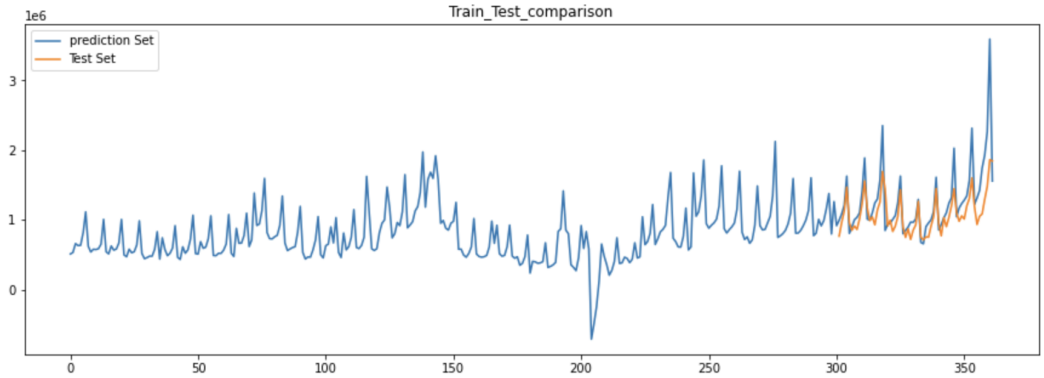

Figure 20: Predicted Trend for Prophet Model Without Macroeconomic Indicators

We can see that our Prophet model cannot catch the upward trend in the last week so we wonder if there is any macroeconomic indicator that will influence the trend of profits. So we added some macroeconomic indicators into the model and rebuilt the model with $\mathbb{R}^2$ of 0.5381, which is better.

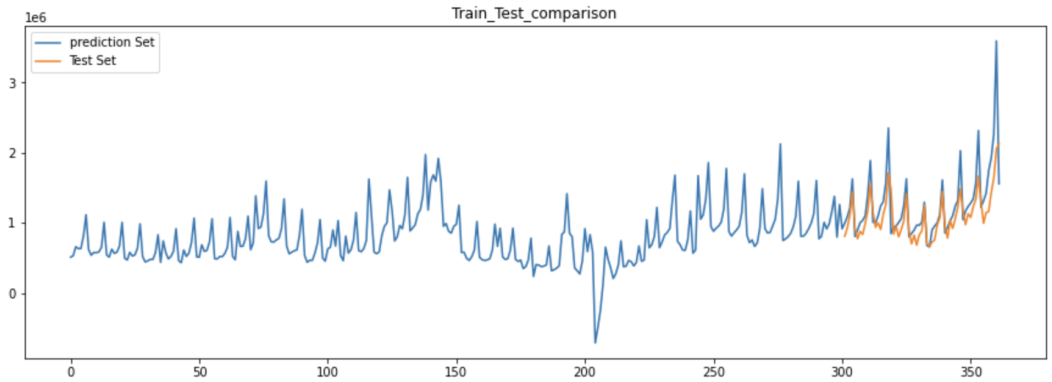

Figure 21: Predicted Trend for Prophet Model With Macroeconomic Indicators

E.3 Lasso Regression

Lasso regression is based on linear regression. It included an extra regularization term to penalize model complexity. The regularization term would force the insignificant features to 0 and drop them from the estimation.

We have used the Z-score normalized data to train the Lasso model. The Lasso Regression model use time series cross-validation automatically tune its alpha parameter which measures how strong is the regression.

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_{1}^{n} |w_i| \tag{3}$$

In our project, Lasso has the best performance for the prediction of nationwide total profits. It could due to the fact that we are dealing with time series data. It has an $\mathbb{R}^2$ of 0.6341.

The feature importance graph for Lasso is shown below (based on coefficient magnitude):

	Coef
totalprofit_1	139315.943075
avgprofittrnsact_6	118071.438457
maxprofit_5	37323.649390
maxcost_7	36456.571364
avgquantity_7	35030.333950
avgvendorsstore_7	34690.248612
nvendors_7	33439.493031
avgorigprice_1	29659.944625
totalprofit_7	28251.697840
maxdiscount_5	27713.867506

Figure 22: Feature Importance from Lasso prediction Model

We can see that lagging features of around 7 days ago are very important for predicting profit in the future. It suggests a weekly pattern for predicting future profits.

The three most important features are total profit_1, average profit per transact_6, and max profit_5.

The predicted trend is shown below:

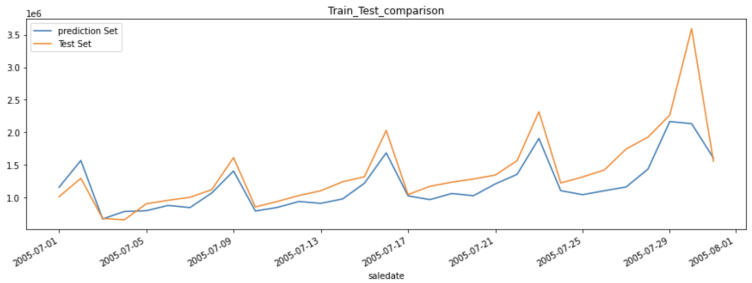

Figure 23: Prediction from Lasso$_predictionModel(NationLevel)$

When we try to predict profit on the state level, Lasso is tuned on the same parameters. The performance improved substantially. It can be that we are having a block design on State, so there is less variability for our samples. Also, as we are grouping data by states, we have more samples to fit the model, which also improves model performance. The final $\mathbb{R}^2$ value is 0.8927, which explains most of the variations in the data.

The feature importance graph for Lasso is shown below (based on coefficient magnitude):

	Coef
totalorigprice_1	21620.258195
totalorigprice_14	16617.522822
totalprofit_14	16313.137210
totalprofit_7	15851.896808
totalorigprice_5	15473.014703
totalorigprice_28	15234.005929
state_TX	12008.819553
totalorigprice_3	11671.749023
totalamt_14	11568.339931
totaldiscount_14	10153.512867

Figure 24: Feature Importance from Lasso prediction Model (State Level)

E.4 XGBoost

XGBoost is a variate gradient boosting model. It will build trees sequentially. Each tree would fit on previously predicted residuals and try to fix previous predictions. It is an efficient, scalable, and accurate model. In addition, XGBoost also implemented hardware optimization, parallel learning, and weighted quantile sketch to improve its efficiency.

For the nationwide profit forecasting, XGBoost is tuned on max_depth (3,5,7,9), min_child_weight (1,4,6), $learning\_rate$ (0.1, 0.01, 0.001), and $n\_estimators$ (100,500,1000) to avoid overfitting. The best-performing model has $\mathbb{R}^2$ 0.19, which does not explain much about the profit trend for Dillard. The best set of parameters found is:

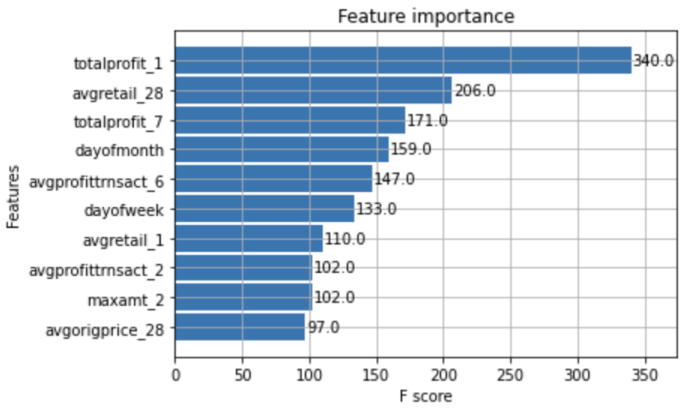

Figure 25: Feature Importance plot for XGBoost Model (Nation Level)

The predicted trend is shown below:

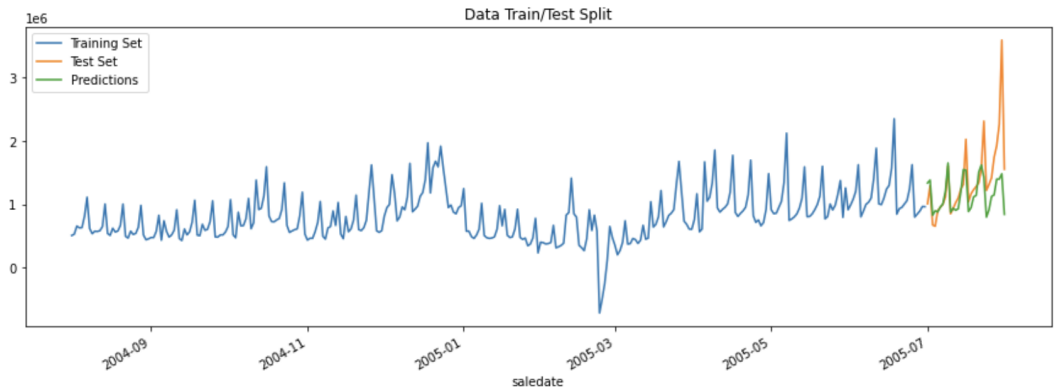

Figure 26: XGboost predict (Nation Level)

When we try to predict profit on the state level, XGBoost is tuned on the same parameters. However, the performance improved substantially. It can be that we are having a block design on State, so there is less variability for our samples. Also, as we are grouping data by states, we have more samples to fit the model, which also improves model performance.

For the state-level profit forecasting, XGBoost is tuned on max_depth (3,5,7,9), min_child_weight (1,4,6), and $learning\_rate$ (0.1, 0.01, 0.001).

The $\mathbb{R}^2$ value becomes 0.8787.

The variable importance is shown in the graph below:

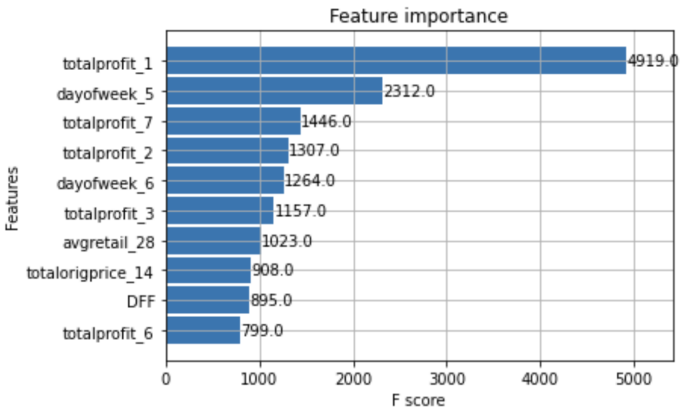

Figure 27: Prediction from XGBoost Model (State Level)

F Results

Table 5: Feature Importance for Lasso Regression Models: Top 5 features

National Level Data		**State Level Data**	
Features	**Coefficient**	**Features**	**Coefficient**
Total Profit$_{t-1}$	164473	Total Profit$_{t-7}$	20061
Av. profit per transaction$_{t-6}$	151969	Total Orig. Price$_{t-1}$	19336
Av.profit per transaction$_{t-14}$	47988	Total Orig. Price$_{t-28}$	15814
Total profit$_{t-7}$	42952	totalorigprice$_{t-5}$	14144
Av. vendors per store$_{t-7}$	40899	totalorigprice$_{t-14}$	12698

G ROI Analysis

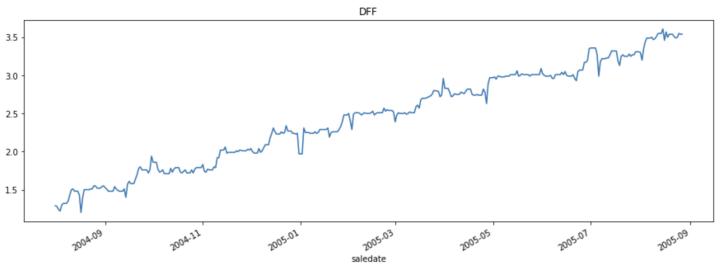

Figure 28: Federal Funds Rate Between 2004 And 2005