Prescribing Product Allocation for Walmart

Daniel Chung | Zach Wayne

1. Problem

Product allocation is neither inconsequential nor trivial. Knowing how much of a given product to stock at a given store on a given week is what allows retailers to meet demand and make revenue, but it relies on a delicate balance. Stocking too little product accrues costs in unmet demand, while stocking too much product accrues unnecessary labor costs. Having an objective system to prescribe product allocation, therefore, is a serious asset to a retailer, and that is precisely what we intend to solve using prescriptive machine learning.

Here, the retailer is Walmart, and the product is toothpaste. Our goal is to develop a prescriptive system that outputs how many units of toothpaste Walmart needs to stock at a given store on a given week given predicted demand, minimizing instances of unmet demand and unnecessary labor costs. The better our prescriptions, the more Walmart saves, and the more value we create for the company, its stakeholders, and the Walmart consumer at large.

2. Data

2.1 Description of Variables

Our data consists of 6,435 weekly sales reports from 45 unique Walmart stores, recorded from May 2010 through October 2012, along with external data including temperature, fuel price, and CPI associated with each week. This dataset was found on Kaggle and is linked in the appendix of this report.[1] Absent from the original dataset is the number of toothpaste products sold per week, as are the labor costs associated with stocking the toothpaste. The original data was highly aggregated, meaning that carving out a well-scoped problem was by no means a trivial task. To do so, therefore, we operated on a set of simplifying assumptions.

The first assumption is that each Walmart store only sells one product (toothpaste), for $3.00 per unit. This means that for a weekly sale volume of $X, $X/3 units of toothpaste must have been sold. It is unrealistic that a Walmart would sell only one product, of course, but this can be generalized since our formulation takes in one product and the weekly sales generated by it. To extend our model to two products, therefore, one would simply take the first product and weekly sales generated by it to prescribe how much of it to stock at the beginning of a week, taking the second product and weekly sales generated by it to prescribe how much of that product to stock as well. Our model is modular, which is why we can develop it on the simplest retail case of one product for a single store.

The second assumption is that each Walmart store pays a cost of $19/hr to each employee and that each employee stocks 100 units of toothpaste per hour. This wage comes from a statement on Walmart's own website about the compensation for its associates.[2] Assuming toothpaste is stocked by one employee at a time, this means the cost of stocking a single unit of toothpaste is $19/100, or $0.19.

The third assumption is that the cost of unmet demand is $2 per unit of toothpaste. We decided to model the cost of unmet demand as less than the price of a unit of toothpaste because a product that is already on the shelf does not need to be restocked, meaning it won't accrue further labor costs. At the same time, however, there was precious floor space taken up by a product that wasn't in demand, so the cost of unmet demand cannot be 0 either. Thus, we landed on a heuristic value of $2.

Variable	Type	Description
Store	Integer	Unique store number
Date	String	End day for the week reported
Weekly_Sales	Float	Total sales from a given store in a given week (USD)
Holiday_Flag	Int	1 if the week has a holiday, 0 otherwise
Temperature	Float	Temperature on day of sale (F)
Fuel_Price	Float	Cost of fuel around store region (USD)
CPI	Float	Prevailing consumer price index
Unemployment	Float	Prevailing unemployment rate

Figure 1: Description of the Original Walmart Dataset

2.2 EDA

To probe for early signs of important features, we plotted the relationship between various features and weekly_sales, the variable that informs stocking quantity prescriptions.

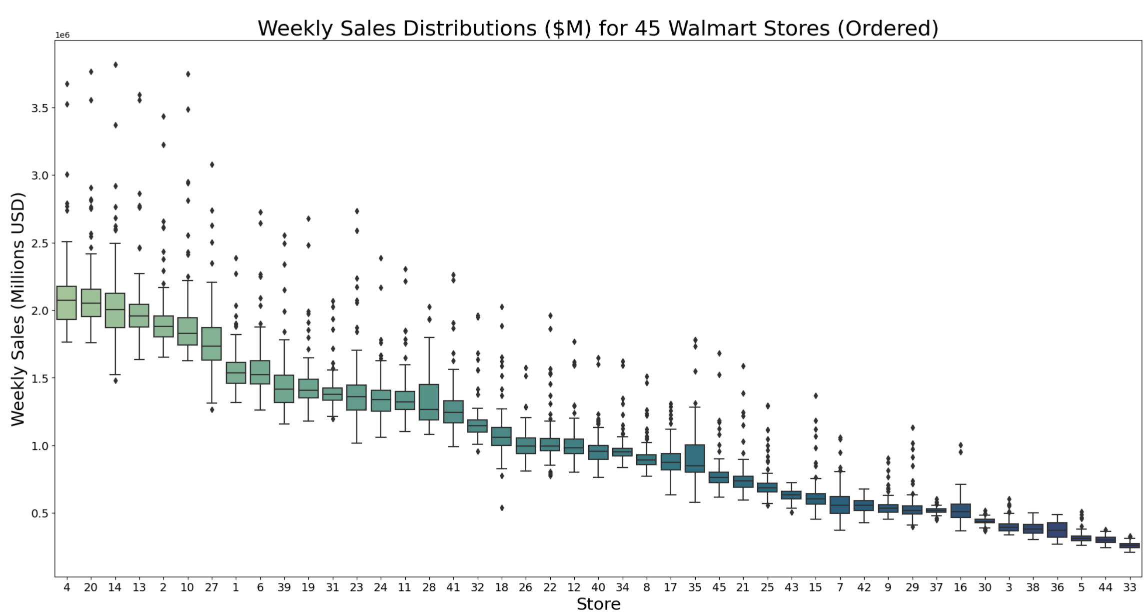

Figure 2: average weekly sales per store, in descending order of average weekly sale volume.

As seen in figure 2, the clearest discrepancies in average weekly sales were between different stores, which was our first indication that store mattered significantly in prescribing how many units of toothpaste to stock. One would expect to stock far more units of toothpaste at store 4, for example, which has weekly sales north of $2M, than at store 33, which barely breaks $250K. This is a finding that our models later validated when we looked at feature importance.

3. Approach, Results, and Insights

3.1 Training Testing Split

For each model, we applied a train-testing split such that we use the first 23 months of data to prescribe how much toothpaste to stock given the predicted sales demand for that week. Because we split on week, both the training and testing sets have data from all 45 Walmart stores.

3.2 Cost Function

To calculate the cost of a prescription, we take into account not only the profit from selling the unit of toothpaste but also the costs associated with stocking that unit of product and stocking too much. This is key because we want to penalize taking up floor space with toothpaste that will not meet demand. With z = how many units of product we stock for the week (our prescription) and y = how many units we sell that week, our cost function is as follows:

$$c(z, y) = min(3z, y) - z(\frac{19}{100}) - 2min(3z - y, 0)$$

The first minimization means that we sell the minimum of what we put on the floor and demand, where the total value of the product we put on the floor is the z units of toothpaste times the price of $3. The second part is the price of stocking, where 19/100 is the cost to stock one toothpaste. The last term is the cost of the unsold product on the floor. This is $2 times either the difference between how much we stocked versus the demand, or 0, if we stocked less than demand. Together, these make up a function that can effectively evaluate how “good” a toothpaste stocking prescription is based on the costs and benefits of product allocation.

3.3 CART for Prescription

Next we fit a CART model from our data. For this we use IAI’s OptimalTreeRegressor with localsearch=false to replicate CART. We can use GridSearch to look for the best hyperparameters. A key insight is that initially, although the model had a low MSE, we learned that if the min bucket size gets too small, it hurts the prescriptive power because there are too few points in a leaf. Through GridSearch, we found max_depth = 10 and minbucket = 20 to be optimal. The R^2 for our CART model is .937.

Looking at the importance of each feature, we again see the prevalence of ‘Store’.

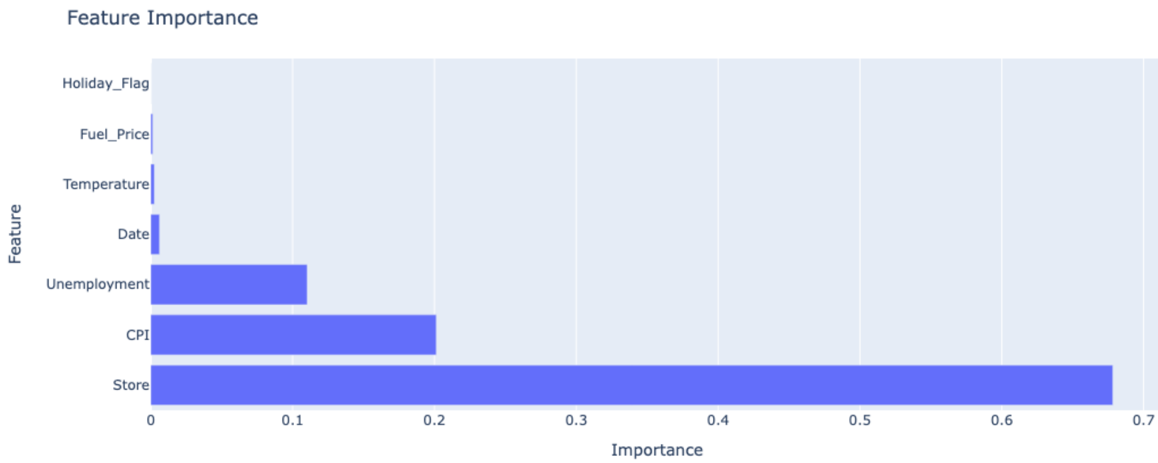

Figure 3: CART Feature Importances for Toothpaste Quantity Prescription

Store is by far the most predictive feature. This makes sense, as each store has its own space capacities and unique demand dictated by its region. The next most important factors are CPI and Unemployment, suggesting that the state of the economy has a nontrivial effect on people’s buying habits. The next step is creating our prescription framework. This is as follows:

$$argmin \sum_{R(x^i)=R(x)} c(z, y_i)$$

R(x^i) = R(x) are all the points in the leaf of the CART that our new point falls into. We loop over all new points to get the optimal z for each new row. Then we can plug each prescribed z back into the cost function, this time with y_i being the observed sales demand, to get what the actual profit is for each point given we stock z toothpastes. Below are the prescribed values (how much we stocked) vs the actual demand. The results are fairly accurate, but there are stretches of overshooting or undershooting.

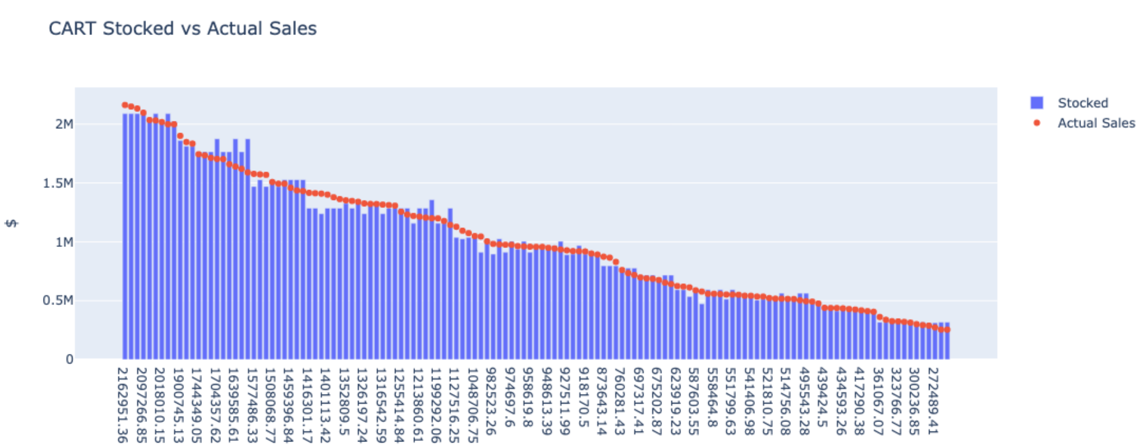

Figure 4: CART Stocked Vs Actual Sales

The average profit per row, which represents the weekly sales of a store, is $896,813.69.

3.4 SAA Baseline

With this framework, we can create baseline models for this problem to compare to CART. The first baseline we implemented was sample average approximation (SAA), which makes the prescription which minimizes the average cost for all instances in the training data according to the cost function. We chose SAA as a baseline because it does not utilize any feature data to make a smarter prescription. Rather, it takes historical weekly sales and makes a prescription based on those alone, ignoring the potential value of the additional information like store number, weather, CPI, etc. that exist in the Walmart dataset. This is only a method that we would use if we lacked access to that kind of additional data.

$$\hat{z}_N^{\text{SAA}} \in \arg\min_{z \in \mathcal{Z}} \frac{1}{N} \sum_{i=1}^{N} c(z; y^i)$$

In the figure below, we can see that maximizing over the average of all points severely underprojects the high performing stores. This matters because any time that the amount of product stocked is lower than actual demand, there comes a cost with the lost sales that could have occurred. In practice this is inevitable, but the baseline is exceptionally poor because in many cases it prescribes quantities of toothpaste to stock that are greater than one million dollars below actual toothpaste demand on a given week. This is millions of dollars in unmet sales potential over time.

Further, there are many instances where SAA over-prescribes, sometimes by as much as 100K units of toothpaste, which in our model comes out to $200K in unmet demand costs. Product allocation is nontrivial, and as a quick baseline model shows, the stakes of maintaining this balance are in the order of serious money.

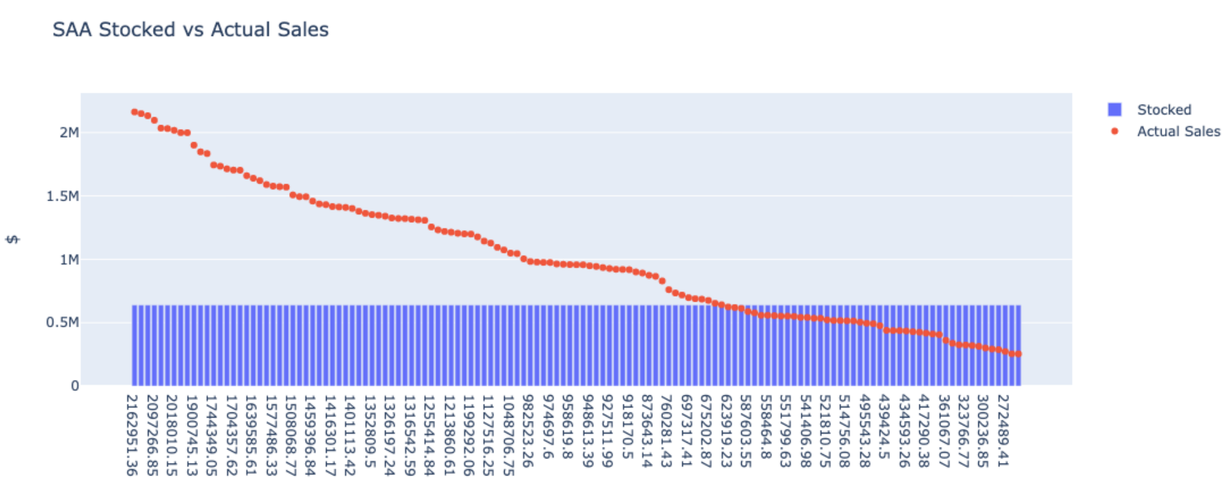

Figure 5: SAA Stocked Vs. Actual Sales

The average profit per row is $537,652.92. This means our CART model outperforms SAA by 67%.

3.5 Regress and Compare Baseline

The next baseline we will check is regress and compare. Here we use our CART model to directly predict sales, rather than optimizing over a leaf. The prescription is the number of toothpaste units that matches the regression prediction for sales at that store during that week.

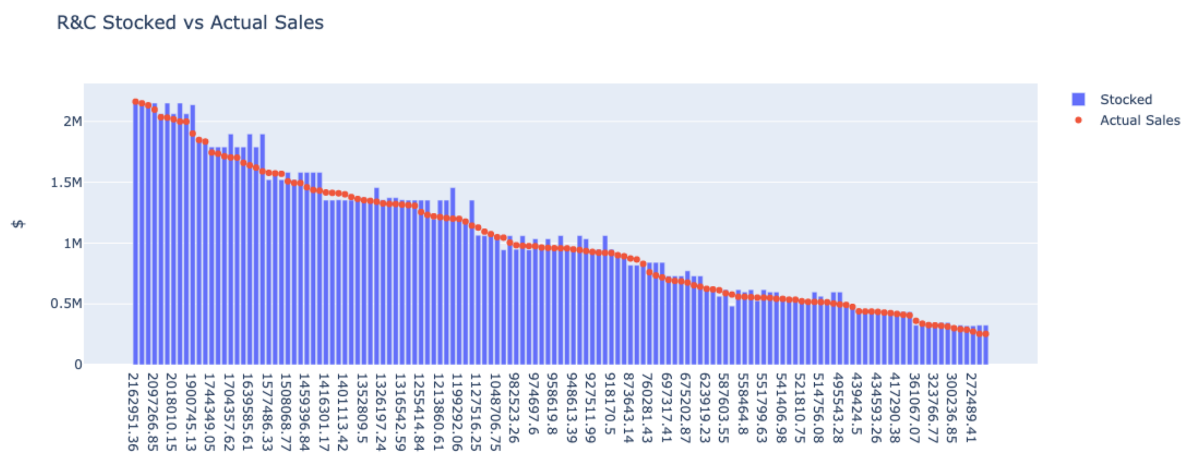

Figure 6: Regress and Compare Stocked Vs Actual Sales

Here we can see an increased accuracy from the previous baseline. However, regress and compare prescribes stocked values consistently over shoot the actual values, leading to excess cost from overstocking. This leads to an overall profit per row of $837,430.68. Regress and Compare worse performance proves the power of optimizing over multiple like points to make prescriptions.

3.6 Store Baseline

Given the high importance of Store, we investigated what the results would be if, instead of running a model, we simply optimized over the previous values from the store our new point is in. This gave $893,204.75 in profit on average per data point, the highest of any baseline. The results are below.

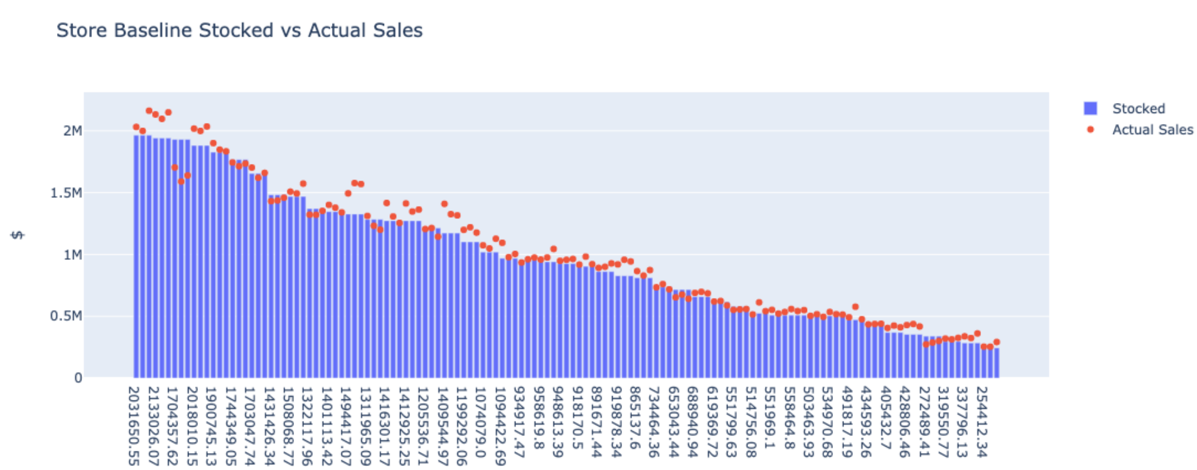

Figure 7: Store Baseline Stocked Vs Actual Sales

This graph is sorted by store and date, so one can see our stocked values are consistent across stores even as their sales fluctuate. We see that while actual sales increase over time for a store, the mode stays flat, decreasing prediction quality at higher volume stores, which have greater sales variability. The Store Baseline's performance is indicative that, while Store is most important, other factors still play a part in giving actual modeling techniques the edge.

3.7 Optimal Regression Trees

Lastly, we use IAI's Optimal Regression Trees to see if it can create a better performing model for prescribing. To do this, we use OptimalTreeRegressor with 'localsearch' and perform a GridSearch over our hyperparameters. The optimal ORT has depth 10, same as CART, but this time the minimum bucket size is 5. Overall the model achieves an R^2 value of .979. The prescription process that takes place is the same as with CART, namely that we optimize over all the points in the leaf that our new points fall into.

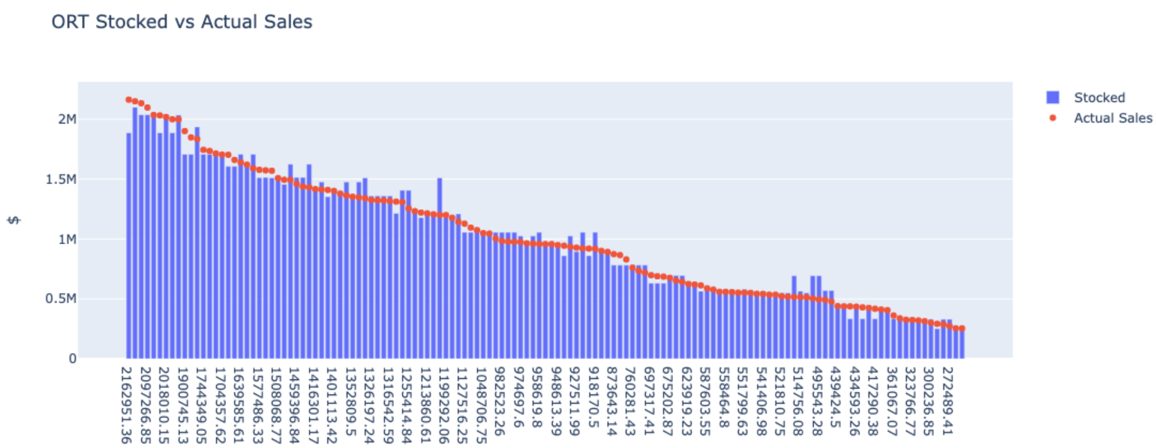

Figure 8: ORT Stocked Vs Actual Sales

Above is how much we stock versus the actual demand for each new point. Interestingly, with ORT there are some points that are farther from their actual than in CART or Regress & Compare, however, overall the model performs better. The average weekly sales of ORT is $915,831.93. This performance could be because of ORT's optimal approach. ORT performs split globally, so individual weeks may have worse performance but overall the model performs better. Overall, ORT has the strongest performance. The difference in weekly profit between ORT and CART is around $19,000 per week.

3.8 Optimal Regression Trees by Store

Following the success of ORTs and realizing the importance of store as a variable, we tried to see if training an optimal regression tree for each store would increase profit. but it only predicts an average weekly profit of $905,957.97. This is most likely because of a lack of data, and because our full model ORT is of depth 10, there already is sufficient splitting.

3.9 Final Results

A compact summary of results for our baseline models and prescriptive machine learning approaches is found below. We evaluated our models on the weekly profit for the average store that resulted from the prescribed product stocking solution, as well as on R^2 for appropriate models.

	SAA	Regress and Compare	Store Baseline	CART for Prescription	ORT for Prescription
Weekly Profit	$537,652.92	$837,430.68	$893,204.75	$896,813.69	$915.831.93
R^2	-	-	-	.937	.979

Figure 9: Prescriptive Model Results and Financial Impacts

4. Impact

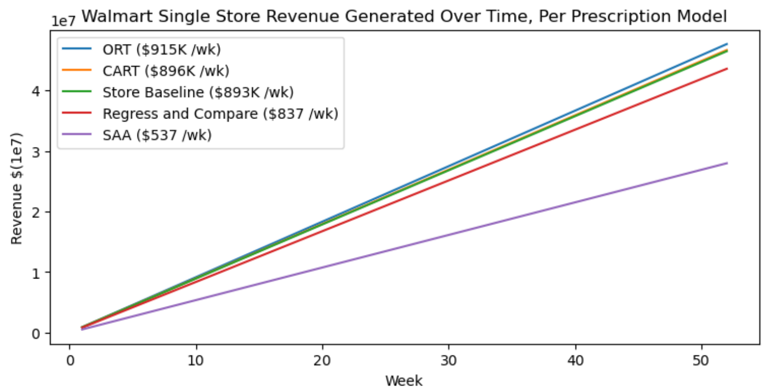

Figure 10: Comparative Single-Store Revenue Projections for Each Prescriptive Method

Walmart, or any retailer that applies this model to their outlets, stands to gain millions from the power of prescriptive machine learning. From our results, for instance, it can be seen that the average Walmart store generates a revenue of $837K per week using a regress-and-compare baseline compared to $915 in weekly revenue using ORTs for prescription. This means the deployment of ORTs would produce $78K more in revenue every week compared to the regress-and-compare baseline, which accumulates to a revenue increase of over $4.07 million per year. This is one store. In total, the impact of our model is even greater, since a retail conglomerate like Walmart would enjoy the boost in revenue from multiple outlets at once. This means that if Walmart, or any chain, implements prescription techniques based on optimal trees, they could be generating huge revenue increases in the order of millions.

Further, with ORTs, retailers stand not only to gain revenue from machine learning models but also to glean interpretable insights into what dictates how much of a product to stock at a given store in a given week. In Walmart's case, we learned that this important variable was the particular store, as well as some economic factors. This interpretability bonus is not a guarantee when the prevailing machine learning methods of the day include blackbox models like neural networks and random forests. In summary, we have found that predictive and interpretable machine learning methods for prescription contribute to business acumen just as much as business bottom lines, improving the efficiency of retailers while teaching them why.

5. Appendix and References

1. Walmart store sales dataset: https://www.kaggle.com/datasets/yasserh/walmart-dataset
2. https://corporate.walmart.com/askwalmart/how-much-do-walmart-associates-make "We are continuously investing in higher wages, and the average hourly wage for our U.S. frontline associates is more than $17."

6. Team Contributions

6.1 Zach Wayne

I was in charge of the trees. After implementing the regress-and-compare baseline and the store baseline, I trained both the OCT and CART models and then created the prescriptive framework in Gurobi so that we could measure our results. I then did the Approach and Results section as well as the Key Insights section of the report. I also ran an ORT model for each store, but the results were not better than the one ORT.

6.2 Daniel Chung

I performed EDA on the Walmart dataset and ideated the simplifying assumptions that allowed us to derive our cost function and scope out our prescription problem. Once we started modeling, I implemented the SAA baseline model so Zach could compare the results of the ML models against it, and once we finished the technical portion I handled writing for the problem, EDA, and impact portion of this report as well as the slidedeck for our presentation.