Identification of Poor Performing Products for Dilliard's

By Group 4:
Jason Huang, Rohit Sharma, Kexian Wu, and Kate Yee

Table of Contents

Executive Summary

Given the opportunity to analyze Dilliard's point of sale data, we chose to focus on predicting the amount spent in future transactions in order to pinpoint products with low profit percentages. By removing products that do not produce a profit, the company would be able to eliminate the cost of supplying those products and focus on advertising and supplying better performing products to its customers. We focused our analysis on transactions within Texas with a max threshold of $49.5, representing the 85th percentile for the original price in the training transaction data, and categorized products into two groups: 'Cheap' (original price <= $15) and 'Normal' ($15 < original price < $49.5) based on their original prices. During modeling, we used a grid search process to optimize hyperparameters for linear regression, Lasso regression, and random forest models, which we ran separately on each price category. We found that the Lasso model performed best on both price categories, as it was able to reduce overfitting and aid in feature selection. We performed a ROI analysis predicting that when we exclude the bottom 30% of products from each store, we make a significant increase in profit of 2.93 times as compared to our baseline results.

Introduction

Dilliard's is a prominent department store chain, which sells clothing, lifestyle, and home essentials products across the United States. Point of sale data was made available to our team consisting of five tables: strinfo, skstinfo, skuinfo, trnsact, and deptinfo. The schema of these tables is described in appendix (Figure 1 and Table 1). Given that data, we needed to perform exploratory data analysis to understand the shape of the data, choose a machine learning question related to profit at Dilliard's, and devise models to answer our chosen question. Once we had used feature engineering and modeling to predict answers to our machine learning question, we also needed to perform a Return on Investment (ROI) analysis on our predictions.

Data Processing

The Diliard's point of sale data was supplied to us as a collection of csv files, with one file containing the rows for each table. We employed a PostgreSQL database using pgAdmin for centralized data access and queries in our project, utilizing five tables (strinfo, skstinfo, skuinfo, trnsact, and deptinfo) for analysis. We encountered challenges with the eleven-gigabyte trnsact table. We resolved discrepancies in column names and order, allowing successful upload to the database. During skuinfo table upload, we resolved issues with vendor names containing commas, the same as the csv delimiter, by filtering out around 8000 rows to maintain data integrity. Our team members were then able to use pandas and psycopg2 with python Jupyter notebooks to access the datasets and begin conducting data analysis.

Exploratory Data Analysis and Data Cleaning

We began our data analysis by exploring the trnsact table, which contained transaction details including the notable values 'quantity,' 'saledate,' 'store,' 'orgprice' (original sku price), and 'amt'

(total amount spent). The trnsact table was difficult to query because it contained 120,916,896 rows, but we were able to begin working with the data by randomly sampling one million rows for exploratory analysis. We then merged our sampled trnsact dataframe with the strinfo table in order to gain insight into how stores are distributed across states, cities, and zip codes. We discovered that Texas was the state with the greatest sum of transaction 'amt's (Table 2). Based on that discovery, we decided to focus our analysis on Texas transactions in order to segment the transactions into a digestible amount, while still analyzing the largest state demographic. Analyzing sales within the same location would also aid our future predictions.

We then queried and merged the skuinfo, skstinfo, and deptinfo tables into the working Texas transaction dataframe to gain more specific info about the SKU's associated with each transaction. Further analyzing the created dataframe, we observed instances where 'orgprice' or 'amt' was recorded as zero. Furthermore, there were cases where 'amt' exceeded 'orgprice' or the 'cost' exceeded 'orgprice,' suggesting potential mislabeling of the original prices. To rectify this issue, we proceeded to remove these unreliable entries.

We continued our analysis on the cleaned transaction dataframe consisting of 26,482,728 rows. Looking for trends within the different columns, we saw a significant spike around December in the graph tracking the number of sales in a day overtime (Figure 2). Intrigued by both that spike and the more regular patterns, we grouped 'saledate' by weekday, month, and quarter (Figures 3, 4, 5). We kept those 'salesdate' classifications in mind for later feature engineering.

In order to better elucidate the discount being applied to each transaction, we also created a discount percentage column and a profit column following these equations:

$$'discount\ percentage' = \frac{'amt'}{'orgprice'} \times 'quantity' \times 100$$

$$'profit' = ('amt' - 'cost') \times 'quantity'$$

During those column creations, we removed rows that contains missing values that we eventually needed for those calculations above. We also decided to exclude transactions with excessively high discounts, such as selling items at prices less than 3% of the original price. This was necessary because we could not ascertain whether these transactions were a result of inventory clearance or pricing errors. We created a histogram graphing the distribution of discount percentages (Figure 6) and observed many instances of transactions with exceptionally high discount rates. Consequently, we considered the possibility of predicting the profit margin before stocking these items.

In order to further clean our data, we chose to remove all rows of 'stype' R in order to only focus on purchase transactions. We also removed outliers by filtering transaction 'quantity,' 'orgprice,' 'packsize,' and 'cost' on a .99 quantile and excluding rows where a single transaction is associated

with a specific ('sku', 'store') combination. We pickled the resulting dataframe to be used for the feature engineering and modeling steps of our process.

Modeling

Our next step was to focus on building multiple machine-learning models to predict sales outcomes. As observed in the EDA section, the dataset used in this project is extensive. Therefore, we decided to concentrate solely on purchase-type transaction data recorded in Texas. Our main objective was to employ SKU features and transaction history as key variables to predict the target variable, 'amt' (the actual amount paid by the customer during the transactions).

The data was split into a training set (first twelve months) and a testing set (last month), with random selection of 1,200,000 rows for training and 300,000 rows for testing. In addition to accurately predicting the 'amt' for each transaction entry, we aimed to compute the profit percentage for each of these entries. A critical aspect of this project involved identifying and removing the lowest 30% of the profit percentage distribution. Our hypothesis was that by identifying and eliminating less profitable products ahead of time, we could help the company reduce the cost of purchasing those products and the loss of selling them, as they will not contribute significantly/positively to the company's profitability. The effects of cost reduction will be further explored in our ROI section.

Feature Engineering and Pipeline

We trimmed the dataset by selecting a threshold of $49.5 (85th percentile) for the org-price feature. As we observed significant inconsistencies in store units with high prices, our analysis indicated that Dilliard's has a strong presence in mid to low-end product categories, characterized by a much lower volume of transactions in the higher-priced segments (Figure 7). Based on these findings, it was advisable to prioritize our model predictions for low-end product transaction records, as this strategy aligned more closely with our overarching goal of cost reduction.

We dropped numerous redundant features that serve no purpose and may contribute to model overfitting or complicated feature space. While categorical variables like size, color, style, and brand could potentially be useful, inconsistencies in their data quality led us to consider dropping them or conducting feature engineering. We chose to use the department description due to its consistent and concise definition for each SKU. Saledate itself was not used in our models. However, saledate was broken down into multiple features that could have predictive power, such as year, month, weekday, quarter, or even holidays if relevant. In addition to our primary features, we incorporated lagged features into our model for 'amt' and orgprice.' These lagged features provided valuable information about transactions for specific store units by capturing data from the last transaction in the training set. The rationale behind this addition was rooted in our belief that a temporal association exists within pricing patterns. By considering historical

transaction data leading up to the present, we aimed to uncover insights into how pricing evolved over time. This temporal perspective allowed us to better understand the dynamics of pricing within our dataset, ultimately enhancing the predictive power of our models.

We constructed a pipeline to streamline the feature engineering and model-building steps. For our column preprocessor, we standardized numerical variables such as quantity, orgprice, pack size, and cost. From our EDA, we observed right-skewness in these numerical variables, which can undermine the linearity assumption of linear regression models. For the remaining categorical variables, we conducted one-hot encoding to transform the categorical data into a binary format. This representation assigned each category as a unique binary variable, where each variable indicates the presence or absence of a specific category. We then employed different models to predict the 'amt.'

Model Evaluations

Within our model pipeline, we incorporated a grid search process to optimize the models. This optimization focused on tuning various hyperparameters to achieve the best model fit. We evaluated the models based on Mean Squared Error (MSE), R-squared, and Mean Absolute Error (MAE). These metrics provided insights into the accuracy and reliability of the models. In addition to linear regression, we implemented Lasso regression to address potential overfitting issues and aid in feature selection. This involved adjusting parameters like alpha and fit_intercept. Furthermore, we also implemented the Random Forest algorithm due to its ability to handle complex datasets with higher accuracy. The optimization of Random Forest models included a grid search to determine the optimal number of trees (n_estimators) and the maximum depth of the trees. Initial observations indicated that applying a single model to the entire trimmed dataset is not effective, as evidenced by a low R-squared value. This led us to develop individual models for each price category based on the 'orgprice' features, categorized as 'Cheap' (orgprice <= $15) and 'Normal' ($15 < orgprice < $49.5), to enhance accuracy and predictive capabilities.

Results and Comparative Analysis

The linear regression models exhibited variable performance across different price categories. For instance, the models for the 'Cheap' category all showed high R-squared values, indicating a good model fit. The Lasso regression models proved to be helpful in reducing overfitting and identifying significant features and exhibited better performance in both categories. To our surprise, the Random Forest models did not demonstrate improved performance over the linear models, even though the model is more robust in handling complex, non-linear relationships and interactions between variables in the dataset. Among all the models, the individual Lasso model applied to the 'cheap' and 'normal' types demonstrated the best performance (Table 2). Consequently, we utilized predictions from these two models to estimate the profit margin.

ROI Analysis

Our ROI analysis perfectly examined the financial outcomes of inventory management decisions and uses the results from our lasso regression model. Our analysis revealed a clear pivot point at the 0.3 quantile threshold, where profit shifts from negative to positive. This suggests that products falling below this threshold are a drain on resources, as their inclusion leads to losses. The strategic exclusion of these products means they are not purchased in the first place, eliminating the incurred costs altogether.

Our baseline model estimated performance of previously non-profitable products and involves eliminating the bottom % of underperforming items in each store by targeting (sku, store) pairs during training, and applying the same criteria in test sets. Baseline model data indicated a total cost of $293,519.23 and a negative total profit at the 0.35 quantile, shifting to a positive total profit at the 0.4 quantile with a total cost of $378,213.07 and a total profit of $25,826.15. We saw that it continues to experience negative profits up to the 0.35 quantile, implying that holding onto the bottom 35% of products led to an opportunity cost, which is the lost potential revenue from more profitable investments. It required adjusting the quantile cutoff to 0.4 to start realizing profits, which means holding more inventory for longer, with associated costs and risks. As compared to the lasso regression model, we achieved profitability at 0.3 indicating substantial savings.

After excluding the bottom 30% of products for each store and interpreting the total costs and profits from our ROI table, we inferred a significant revenue increase of 3.64x folds and subsequently a rise in profit of 2.92x folds, as indicated in our ROI table, with better returns at an earlier threshold compared to the baseline model values. Moreover, focusing on products priced under $50 aligns with our strategic approach to discount management and profit margin consistency. Concentrating on this price segment helped to ensure stable profitability across the product range. Moreover, as per our ROI table, we have identified the investments in hiring qualified data scientists and managers and allocated costs efficiently so that our proposed model yields a greater profit.

In conclusion, our model identified the 0.25 to 0.3 quantile range as a profitable margin, advocating for a product selection that resides in the top 70-75% in terms of profit margins. Our targeted approach promises a better ROI by achieving profitability at a lower quantile threshold and minimizing opportunity cost. This approach not only prevents losses but also allows for a more focused investment in inventory that will yield higher returns.

Appendix

PK – Primary key

FK – Foreign key

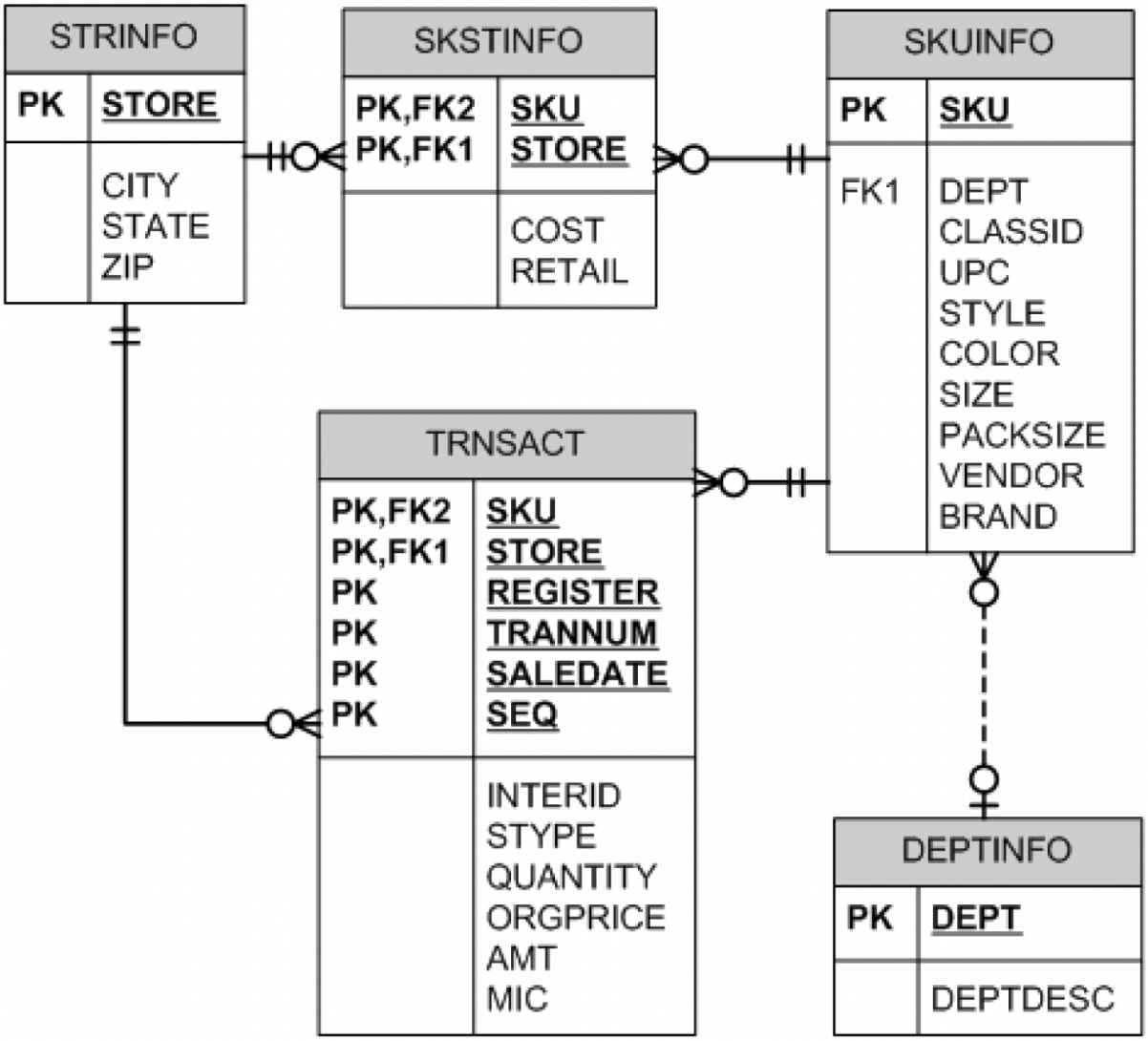

Figure 1: Database Diagram

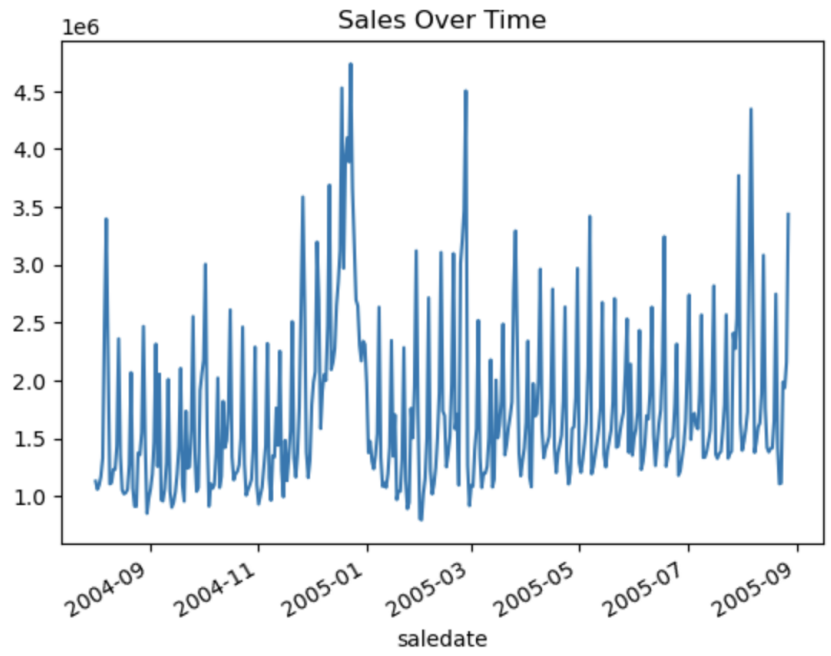

Figure 2: Sales per Day Over Time

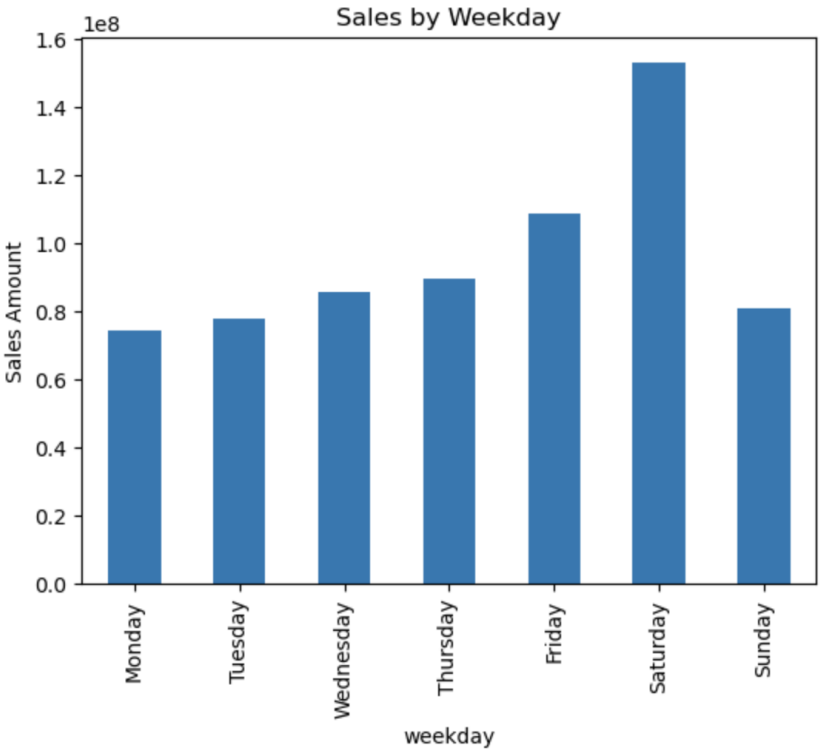

Figure 3: Number of Sales by Weekday

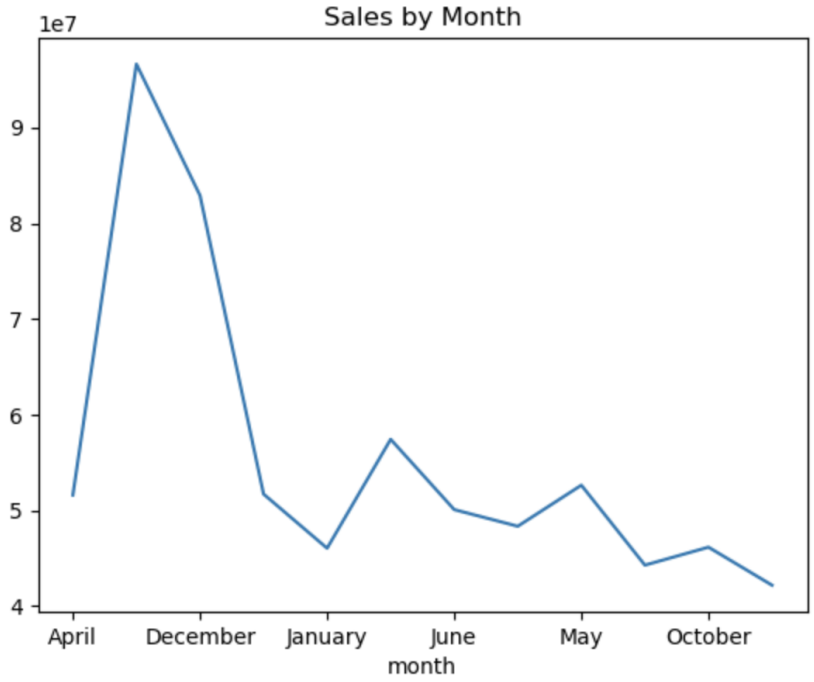

Figure 4: Number of Sales by Month

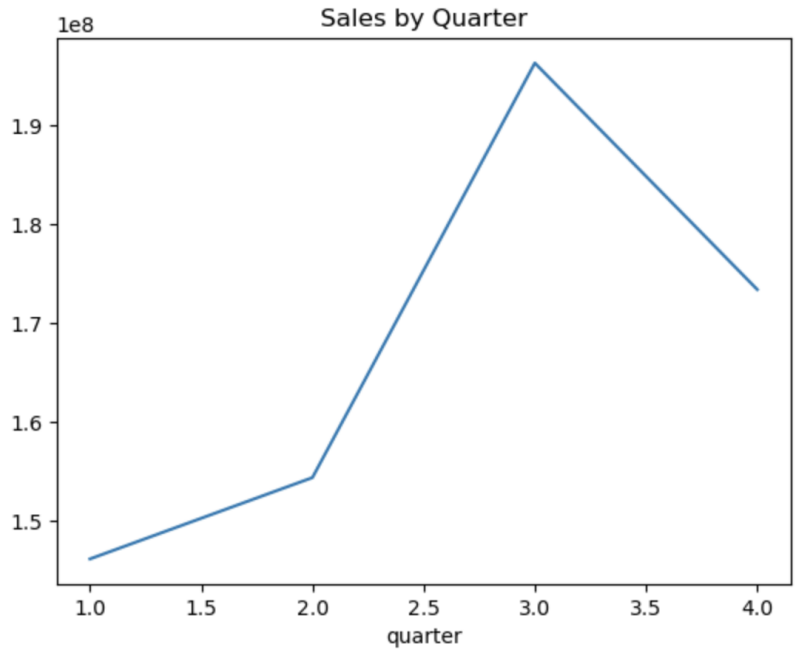

Figure 5: Number of Sales by Quarter

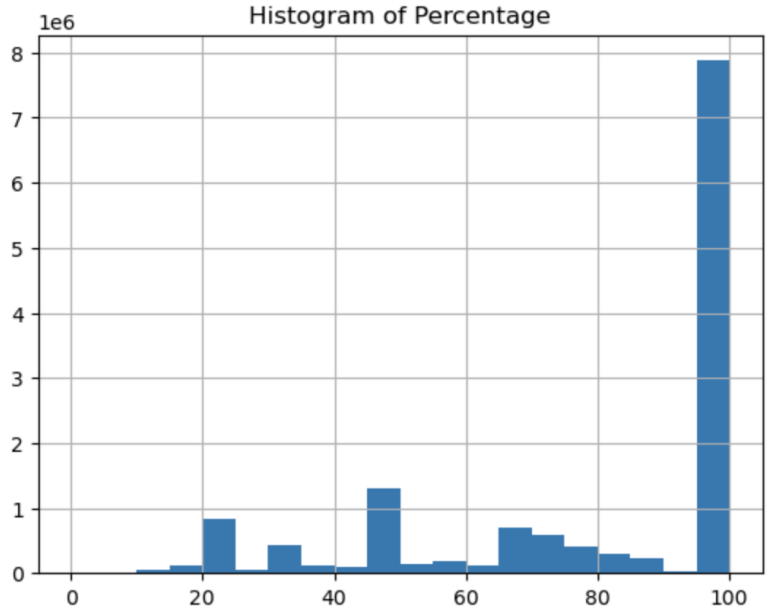

Figure 6: Histogram of Discount Percentages

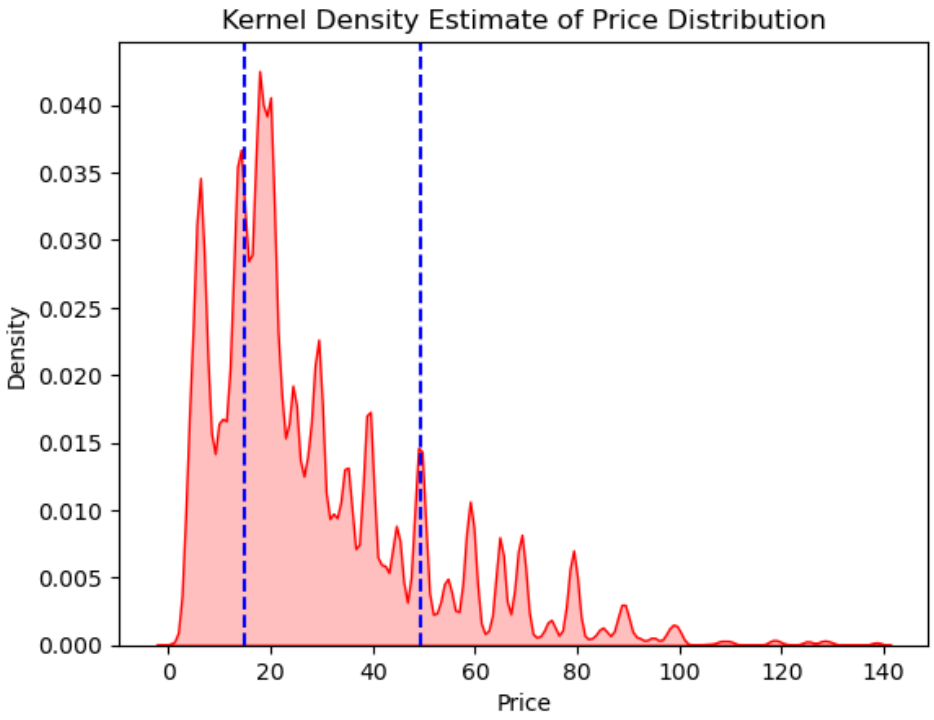

Figure 7: Density Estimate of Price Distribution

Table 1: Column/Attribute Description

Attribute	Description	Value Types
AMT	Total amount of the transaction charge to the customer	26.25, 44.00, ...
BRAND	The brand name of the stock item	TOMMY HI, MARK ECK, ...
CITY	City where the store is located	ST. LOUIS, TAMPA, ...
CLASSID	Stock Item Classification	5305, 4505, 8306, ...
COLOR	The color of the stock item	BLACK, KHAKI, ...
COST	The cost of the stock item	9.00, 15.00, ...
DEPT	Department where the stock item belong	800, 801, 1100, ...
DEPTDESC	Description of the department	CLINIQUE, LESLIE, ...
INTERID	Internal ID	265005802, 671901998, ...
MIC	Master Item Code	862, 689, ...
ORGPRICE	Original price of the item stock	75.00, 44.00, ...
PACKSIZE	The quantity of item per pack	1, 3, ...
QUANTITY	Item quantity of the transaction	1, 2, 3, ...
REGISTER	Register Number of the current transaction	580, 30, 460, ...
RETAIL	The retail price of the stock item	19.75, 34.00, ...
SALEDATE	Sale date of the item stock	2005-01-20, 2005-06-02, ...
SEQ	Sequence number	298100028, 213500030, ...
SIZE	The size of the stock item	L, 070N, 22, ...
SKU	Stock Keeping Unit number of the stock item	4757355, 2128748, ...
SPRICE	Sale price of the item stock	26.25, 65.00, ...
STATE	State where the store is located	FL, MO, AR, ...
STORE	Store Number	2, 3, 4, 100, ...
STYLE	The specific style of the stock item	51 MERU08, 9 126NAO, ...
STYPE	Type of the transaction (Return or Purchase)	P, R
TRANNUM	Transaction Code	09700, 01800, ...
UPC	Universal Product Code for the stock item	000400004087945, ...
VENDOR	The vendor number of the stock item	5511283, 2726341, ...
ZIP	ZIP Code	33710, 63126, ...

Table 2: Sampled Total Transaction Amount by State

state	
TX	5564009.12
FL	3435391.28
LA	1420495.11
OH	1324056.37
AZ	1255785.47
TN	952154.98
OK	881879.30
MO	877461.90
AR	840495.58
AL	779322.11

Table 3: Model Evaluations

	Model	Mean Squared Error	R-squared	Mean Absolute Error
0	Linear Regression (all)	30.558	0.669	3.283
1	Linear Regression (cheap)	2.260	0.862	0.928
2	Linear Regression (normal)	42.932	0.490	4.229
3	Lasso Regression (all)	30.523	0.670	3.275
4	Lasso Regression (cheap)	2.251	0.862	0.919
5	Lasso Regression (normal)	42.842	0.491	4.219
6	Random Forest (all)	40.048	0.567	3.244
7	Random Forest (cheap)	2.539	0.845	0.792
8	Random Forest (normal)	54.201	0.356	4.145

Table 4: ROI Analysis

Baseline Model	**Profitability cutoff value is 0.4**				
Lasso Regression Model	**Profitability cutoff value is 0.3**				
Threshold	**0.3**	To evaluate returns based on our proposed model			
RETURNS					
	BASELINE MODEL		**PROPOSED MODEL**		
Test set comparison					
Cost of underperforming items	$ 229,662	$ 981,396			
Revenue of underperforming items	$ 217,017	$ 1,005,915			
Percent increase in revenue between baseline and proposed models		363.52%			
Profit of underperforming items	$ (12,645)	$ 24,520			
Difference (proposed - baseline)		$ 37,165			
Percent increase in profit between baseline and proposed models		293.91%			
Previous month cost	$ 3,052,840	$ 2,071,445			
Previous month profit	$ 1,632,529	$ 1,608,009			
Based on this analysis, we observe a significant percent profit increase of 2.93 folds between the baseline and our lasso model, which suggest that excluding the bottom 30% of products for each store results in higher revenue and minimized opportunity costs					
INVESTMENTS				1/12 Years	$ 14,515
LABOR COSTS					
	Annual wage		**FTE**	**Project duration (years)**	**Total**
Data Scientist	$ 120,000.00		1	0.25	$ 30,000.00
Manager	$ 175,000.00		1	0.25	$ 43,750.00
Model maintenance					
Data Engineer	$ 100,000.00		1	1.00	$ 100,000.00
INFRASTRUCTURE COST					
	Cloud hourly rate		**Computing hours/day**	**Days computing***	**Total**
AWS	$ 0.10		24	180	$ 432.00