MSiA 400: Everything Starts with Data

Project

Team 4: Yuexin Chen, Ziyan Liu, Weiyan Zhou, Ruben Nakano

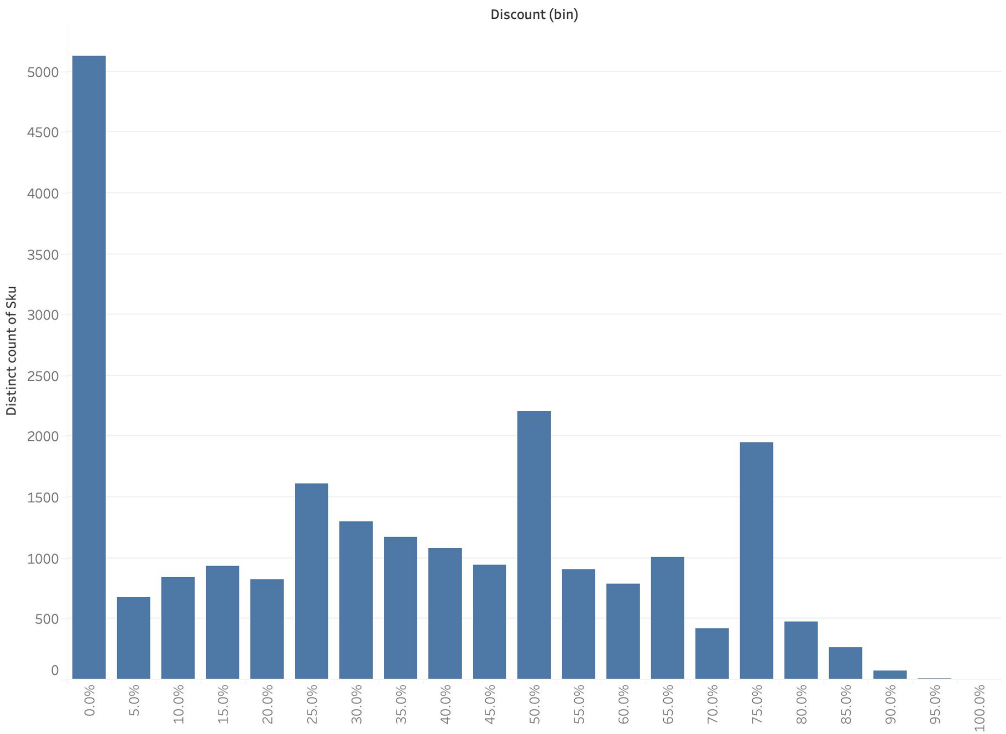
Introduction

- Dillard’s currently has 282 stores in 29 states
- Managing inventory is always a challenge for large retailers (inventory costs can make up for 20% of a products value)
- A model that can predict products that will need higher discounts based on their characteristics (size, color, brand, etc) would be helpful
  - Opportunity to optimize inventory by not keeping stock of underperforming products
  - Decrease rate of production of underperforming products ahead of time
  - Minimizes potential losses

- Baseline model has an R2 of 28%, we can do better!

EDA

- Large amount of items have no discount
- Discounts between 50-55% and 75-80% are the most common
    - Probably because it's uncommon to unrounded values



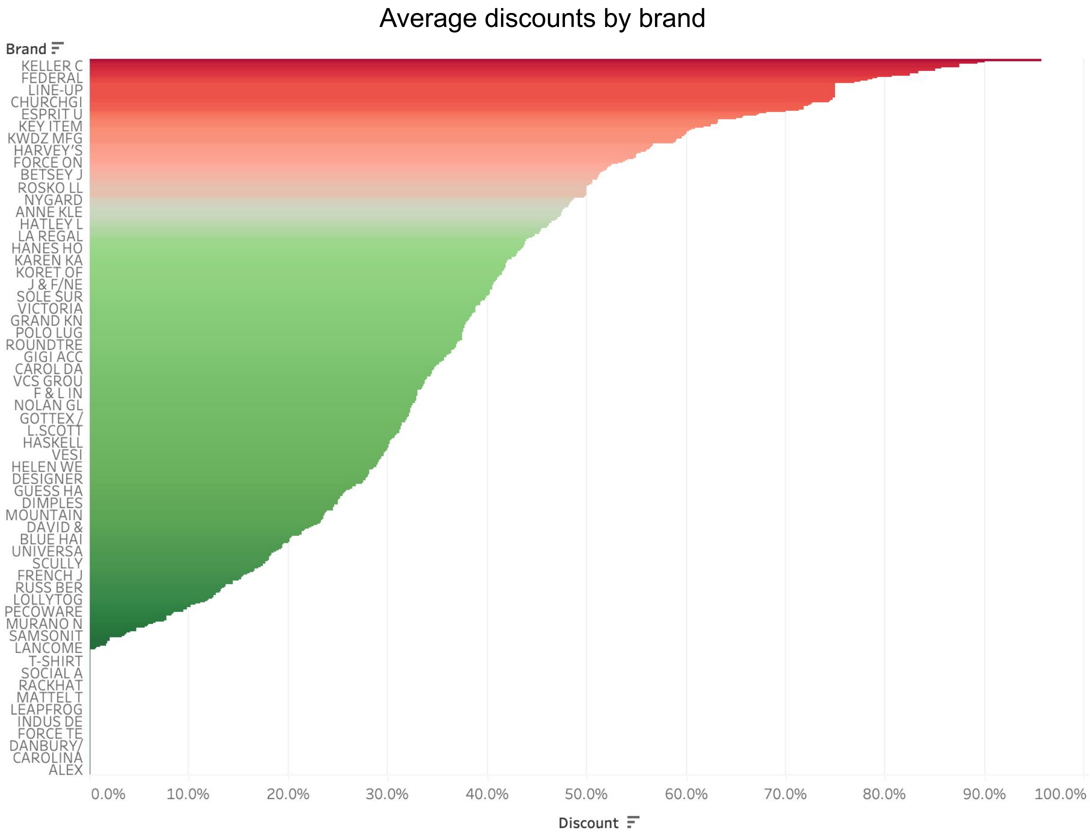
EDA

- No visible trend in average discounts by color and size
  - Some combinations of color and size are uncommon

	New Color											
Cleaned Size	BLACK	BLUE	BROWN	GOLD	GREEN	MULTI	other	PINK	RED	SILVER	WHITE	YELLOW
ALL	31.0%	28.9%	31.1%	22.8%	31.9%	37.1%	33.3%	33.6%	33.1%	33.6%	29.3%	27.3%
L	40.5%	37.7%	32.7%	39.9%	36.6%	41.0%	37.2%	37.8%	37.2%	30.8%	30.0%	31.4%
M	38.9%	34.0%	27.5%	38.4%	30.4%	45.3%	36.6%	38.1%	37.1%	42.6%	32.0%	39.8%
ONE	33.2%	18.7%	10.6%		22.9%	13.7%	23.5%	29.6%	29.0%	0.0%	28.4%	0.0%
S	41.3%	37.9%	21.6%	29.3%	31.4%	42.6%	38.1%	39.0%	43.1%	0.0%	30.5%	42.9%
XL	37.4%	32.1%	31.0%	28.6%	32.4%	35.7%	35.5%	42.1%	36.6%	41.9%	29.4%	36.2%
XS	52.1%	30.3%			50.0%	36.8%	39.1%	28.6%	13.8%		0.0%	27.1%

EDA

- There are brands that have more discounts than others
- Some brand sell few SKUs, so percentage might be misleading

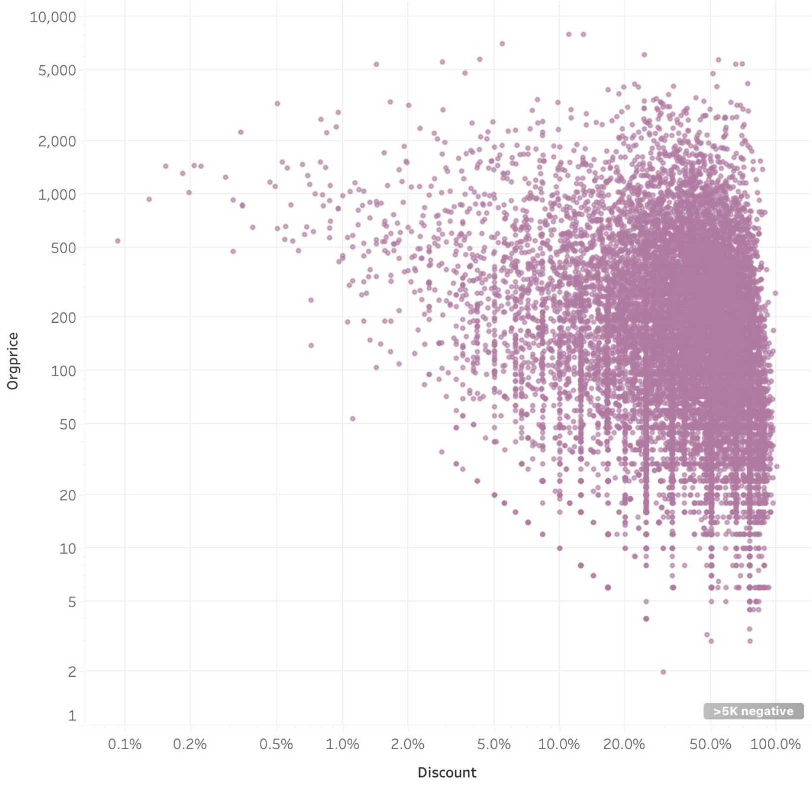


EDA

- Are more expensive products discounted less? More?

- Data shows that the discount percentage tends to increase as price of the product decreases

Average discount x Original price (log scale)



Feature Selection

We grouped the dataset by the remaining variables:

- ’brand’,
- ‘vendor’,
- ‘dept’,
- ‘classid’,
- ‘color’
- ‘size’

Resulted in 186410 groups out of the 266225 rows.

Size Cleaning

- Select out non-numerical size first
- Numerical value of size is inconsistent for different types of items.
- List comprehension to extract only
  - 'ALL', 'L','M','ONE', 'S', 'XL', and 'XS' for size

Color cleaning

- Define new color system: includes majority of popular colors
  - 13 colors: BLACK, BLUE, WHITE, PINK, RED, MULTI, SILVER, GREEN, NOCOLOR, GOLD, BROWN, YELLOW and OTHER
- Regular expressions to map the current colors to new system. Examples:
  - BLACK, BLAK, 100BLACK → BLACK
  - NAVY, DARKBLUE → BLUE
  - WHITE/PINK → PINK
  - CHOCO → OTHER
- Colors that can’t be classified → OTHER

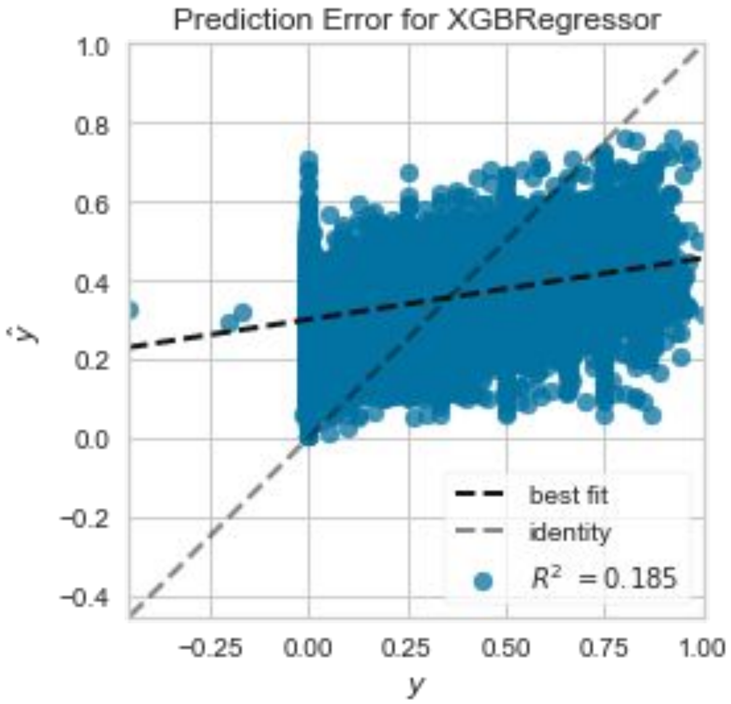
Modeling

- Decision tree is considered the baseline model (R2 = 28%)
- Our proposed final model has R2 = 48%
- Less overfitting issue

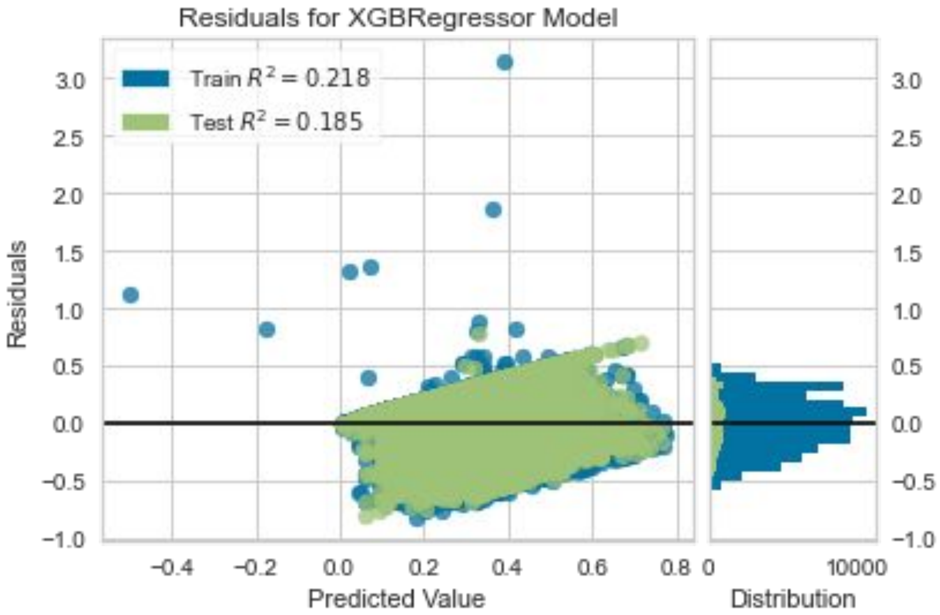
Table 2 Modeling results and hyperparameters

Model / Metrics	Decision Tree	Random Forest	XGBoost (base model) XGBRegressor with objective='reg:squarederror'	**XGBoost (final model)**** XGBRegressor with objective='reg:squarederror'
RMSE (train)	0.197684	0.205070	0.238745	**0.219994****
RMSE (test)	0.258360	0.248859	0.243273	**0.235858****
R^2 (train)	0.681214	0.650584	0.467211	0.579880
R^2 (test)	0.284971	0.384112	0.430557	0.484026
Hyperparameter choice	No tuning with default hyperparameter	n_estimators = 5, random_state = 42	random_state = 2, tree_method = 'hist'	n_estimators = 2500, random_state = 22, learning_rate = 0.18, colsample_bylevel = 0.8, max_depth = 5, tree_method = 'hist'

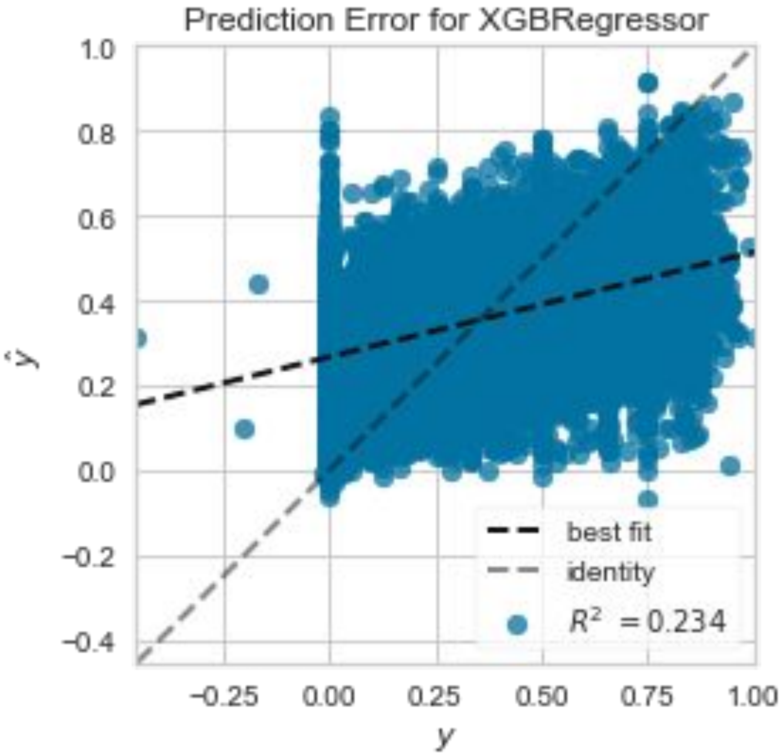
Prediction Error Plot for XGBoost (base model)



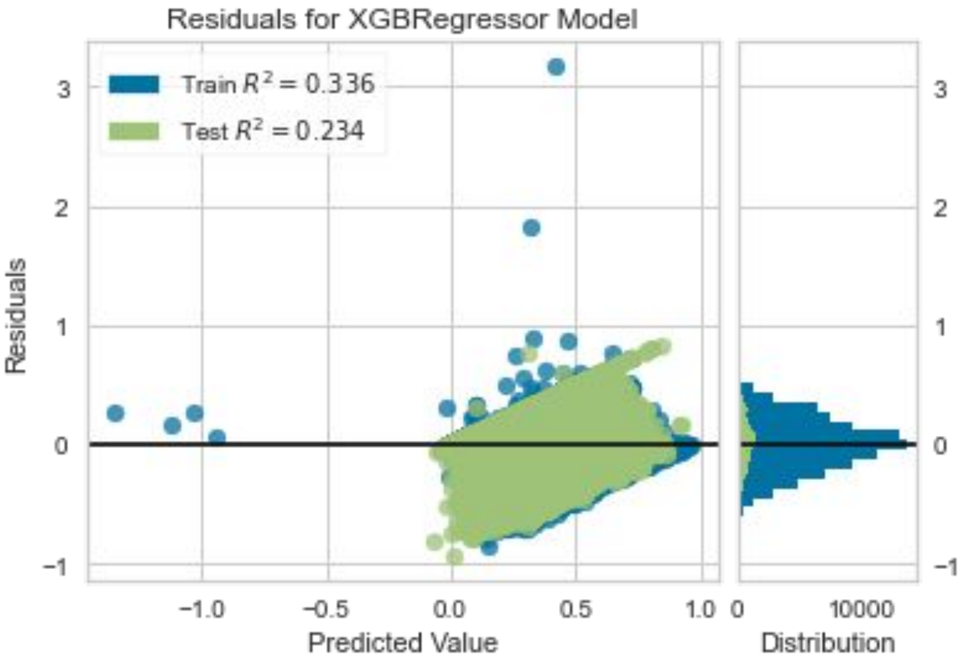
Residuals Plot for XGBoost (base model)



Prediction Error Plot for XGBoost (final model)



Residuals Plot for XGBoost (final model)



ROI Analysis

- Assumption
  - Dillard's has limited storage space for inventory → need to choose which products to keep
  - Uses model to choose which products they will not stock and decrease their production rate
  - Proposed model outperforms baseline model → identifies underperforming products better → reduces loss (or reduces profits "left on the table")
- Details in the spreadsheet

FINAL ROI	$	6,862,134		1538%	
RETURNS			$	7,281,159	
		BASELINE MODEL		**PROPOSED MODEL**	
Test set comparison					
Cost of underperforming items	$	423,596	$	422,401	
Revenue of underperforming items	$	703,957	$	695,938	
Profit of underperforming items	$	280,361	$	273,537	
Difference (proposed - baseline)			$	6,824	
Profit if production reduced by 50%			$	3,412	*
Total profit (test set)			$	1,286,466	
Percent increase in profit				0.27%	
Entire list of company products					
Annual total profit (2021)			$	2,745,300,000	
Total profit with proposed model			$	2,752,581,159	

Results and Risks

- Proposed model can identify the products that need discounts better than the baseline model (R^2 = 28% vs 48%)
- This results in more profit for the company: ROI = US$ 6.8 MM
- Risks
  - Train set performance is still relatively better than test set for the final model (58% vs 48%), so there are still overfitting issues (although this was largely improved compared to the baseline model)
  - Calculation of ROI is based on the extrapolation of the test set performance to company-level profits. If we had more time/resources, comprehensive tests should be made to ensure we achieve similar results