Dillard's

Business Project

Hanxi (Elsie) Lin, Judy Zhu, Yuanzhu (Zoe) Li

& Xenia Vrettakou

Agenda

1. Business Question
2. Exploratory Data Analysis (EDA)
3. Feature Engineering
4. Data Modeling
5. ROI
6. Conclusion

Northwestern

Business Question

Project Scope:

Return rates serve as a critical indicator of the effectiveness of product selection strategies and the alignment of products with consumer demands. Our business focus employs sophisticated data analysis techniques to predict the probability of a return with the focus to reduce Dillard's total return costs.

Descriptive Statistics

- Date range: 01/08/2004 - 01/08/2005
- Number of samples: 66 million
- Number of states represented: 29
- Number of purchase for each transaction: 1

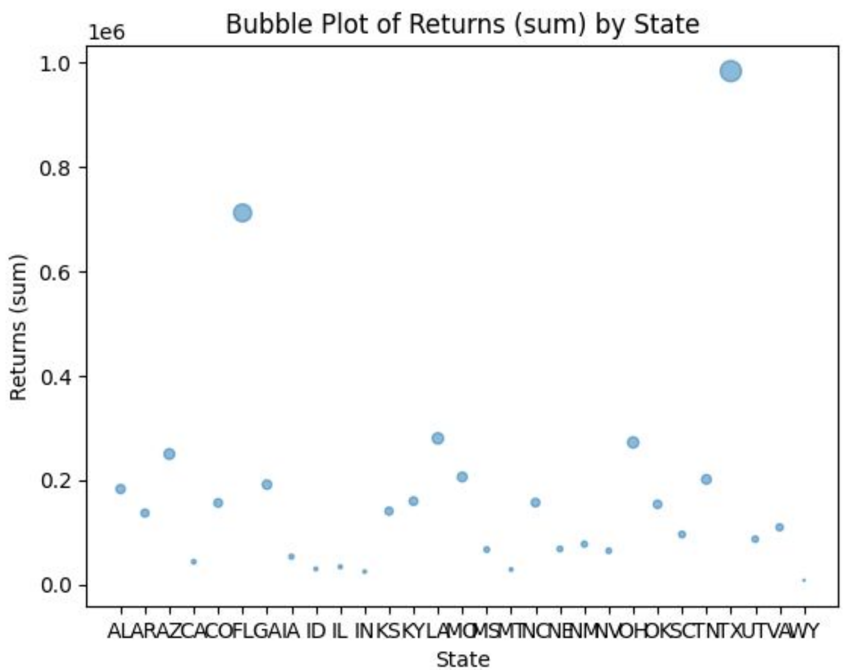
EDA

Return rates across Months

- Date range: 01/08/2004 - 01/08/2005
- Peak: July,higher return in first half of the year

Return rate across States

- Total 29 States
  - Highest: TX
  - Second highest: FL



Feature Engineering

Binning and Categorization
Feature: BRAND_dummy
Process: Bin 'ORGPRICE' into 'Cheap', 'Affordable', 'Luxury' using quartiles.
Purpose: Categorize price ranges for analysis or modeling.

Filtering Data
Process: Remove rows where 'ORGPRICE' > 1000.
Purpose: Focus on specific price range and remove outliers for improved analysis.

Date Manipulation
Feature: Salemonth
Process: Convert 'SALEDATE' to datetime and extract month.
Purpose: Enable seasonality analysis and recognize monthly patterns.

Calculating Differences
Features: amt_diff, retail_diff
Process: Difference between 'AMT', 'RETAIL', and 'ORGPRICE'.
Purpose: Understand pricing strategies and customer behavior.

Calculating Return Rate
Feature: Return rate
Process: Calculate return rate by state, using 'Stype' for purchase/return.
Purpose: Analyze return variations geographically.

Data Preparation

1. Subset data:
   - Random sample selection (100k).
     - Reducing future effect of computationally intensive operations.
2. Applied Holdout CV
   - Train set (80%)
   - Test set (20%)
3. Train set
   - Further subset the data by:
     - Randomly sample 15k where *Returns* = 0 (Purchase)
4. Resolved class imbalance issues
   - Applied SMOTe methodology to balance our target feature.
     - Oversampled minority class: *Returns* = 1 (Return) to reach 15k.
   - SMOTe was applied on the training set only.
     - Aimed to train our model to best classify the 2 classes.
     - Then applied the trained model to our test set to evaluate the model's performance.

Data Modeling

Logistic Regression

Model 1

Model with features

1. Salemonth
2. Original Price
3. Amount
4. Amount Difference
5. Retail Difference
6. Brand Dummy Affordable
7. Brand Dummy Luxury
8. Return Rate (per State)

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                Returns   No. Observations:                30000
Model:                          Logit   Df Residuals:                    29992
Method:                           MLE   Df Model:                            7
Date:                Thu, 07 Dec 2023   Pseudo R-squ.:                 0.02019
Time:                        18:49:23   Log-Likelihood:                -20375.
converged:                      False   LL-Null:                       -20794.
Covariance Type:            nonrobust   LLR p-value:                 5.035e-177
=========================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Salemonth              -0.0022      0.004     -0.605      0.545      -0.009       0.005
ORGPRICE               -0.0019   1.17e+04  -1.63e-07      1.000    -2.3e+04     2.3e+04
AMT                     0.0043   1.17e+04   3.66e-07      1.000    -2.3e+04     2.3e+04
amt_diff                0.0064   1.17e+04   5.46e-07      1.000    -2.3e+04     2.3e+04
retail_diff            -0.0048      0.001     -6.489      0.000      -0.006      -0.003
BRAND_dummy_Affordable  0.4248      0.032     13.479      0.000       0.363       0.487
BRAND_dummy_Luxury      0.5401      0.047     11.412      0.000       0.447       0.633
ReturnRate             -6.6285      0.455    -14.583      0.000      -7.519      -5.738
=========================================================================================
```

Observations

- First 4 variables are insignificant.
- Multicollinearity between ORGPRICE, AMT, & amt_diff.
- 4/8 features are significant.
- Salemonth is still not significant.

Data Modeling

Logistic Regression

Model 2

Model with 5 features

1. Salemonth
2. Original Price
3. Brand Dummy Affordable
4. Brand Dummy Luxury
5. Return Rate (per State)

Observations

- Remove highly correlated features with ORGPRICE.
- 4/5 features are significant.
- Salemonth is still not significant.

Logit Regression Results

Dep. Variable:	Returns	No. Observations:	30000
Model:	Logit	Df Residuals:	29995
Method:	MLE	Df Model:	4
Date:	Thu, 07 Dec 2023	Pseudo R-squ.:	0.01589
Time:	18:58:01	Log-Likelihood:	-20464.
converged:	True	LL-Null:	-20794.
Covariance Type:	nonrobust	LLR p-value:	9.816e-142

	coef	std err	z	P>|z|	[0.025	0.975]
Salemonth	-0.0059	0.004	-1.664	0.096	-0.013	0.001
ORGPRICE	0.0029	0.000	6.631	0.000	0.002	0.004
BRAND_dummy_Affordable	0.4518	0.031	14.440	0.000	0.390	0.513
BRAND_dummy_Luxury	0.6136	0.046	13.345	0.000	0.523	0.704
ReturnRate	-6.4078	0.453	-14.161	0.000	-7.295	-5.521

Logistic Regression

Key Takeaways:

- Using logistic regression as our baseline model does not perform well.
  - Pseudo R^2 extremely low for both models.
  - Consider other metrics for evaluating the two models.
    - Confusion matrix.

Solutions:

- Remove highly correlated variables from model 1.
  - Feature significance improved in Model 2.
  - Evaluation metrics worsen.
- Use Random Selection instead of SMOTe for class balance issues.
  - From 100k dataset randomly select 15k of minority class instead of imputing them.
    - We observed worse evaluation metrics.
- Implement different model: Random Forest.

Evaluation Metrics

Observations:

- All of the metrics are better in model 1 than model 2.
  - Despite the highly correlated features in model 1.
- Accuracy is relative good for model 1.
  - Accuracy is not a good metric for imbalanced data.
- Low Precision & F1 for minority class.
  - Logistic models not good predictors for probability of return.
- High Recall
  - Models are good at identifying positive instances.

Overall:

- Models 1 & high likelihood of false positives.

Model 1

```
Confusion Matrix on Test Data:
 [[10151  8286]
 [  571   992]]
Classification Report on Test Data:
              precision    recall  f1-score   support

           0       0.95      0.55      0.70     18437
           1       0.11      0.63      0.18      1563

    accuracy                           0.56     20000
   macro avg       0.53      0.59      0.44     20000
weighted avg       0.88      0.56      0.66     20000
```

Model 2

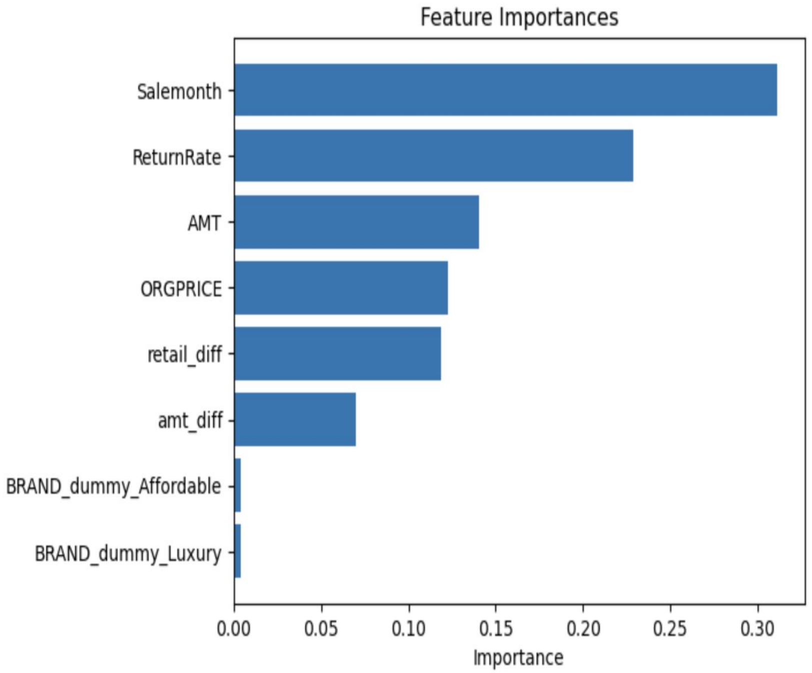
```
Confusion Matrix:
[[ 7803 10634]
 [  415  1148]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.42      0.59     18437
           1       0.10      0.73      0.17      1563

    accuracy                           0.45     20000
   macro avg       0.52      0.58      0.38     20000
weighted avg       0.88      0.45      0.55     20000
```

Data Modeling

Random Forest

Model with features & Importance:

- **Salemonth**: most important, specific months or seasonal trends
- **Return Rate (per State)**: important, possibly derived from historical return data or customer behavior
- **Amount** & **Original Price**: transactional attributes maintain notable importance
- **Retail Difference** & **Amount Difference**: moderate importance, potentially related to discounts or fluctuations
- **Brand Dummy Affordable** & **Brand Dummy Luxury**: minimal importance

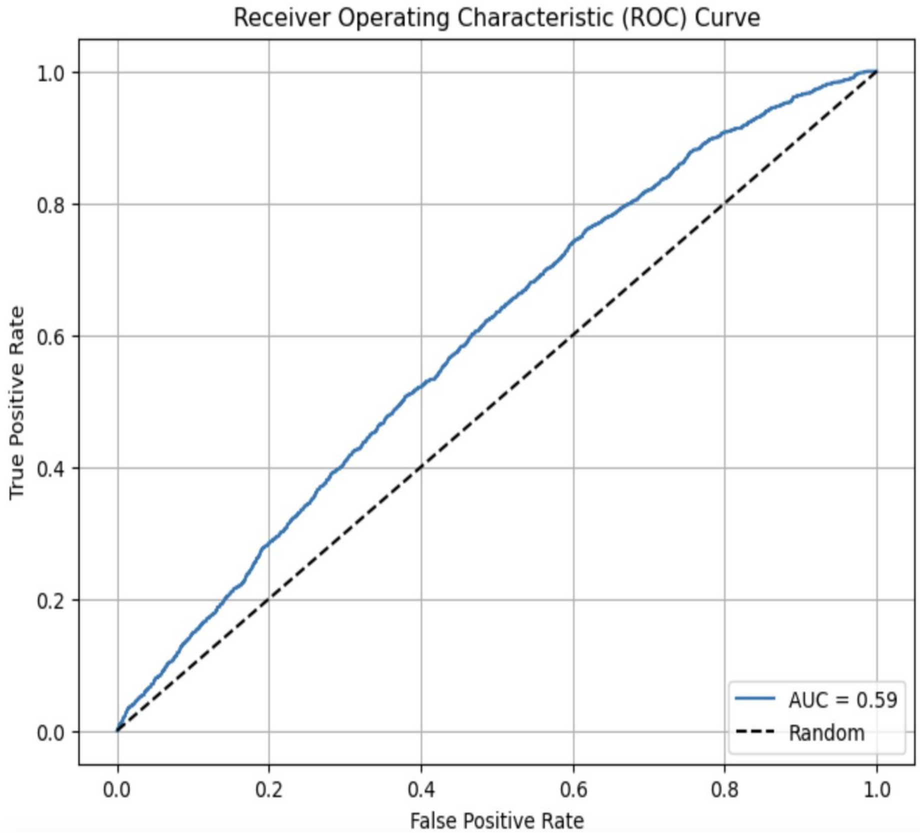


Model:

```
▾ RandomForestClassifier
RandomForestClassifier(bootstrap=False, class_weight='balanced',
                       min_samples_leaf=2, min_samples_split=10,
                       n_estimators=200, random_state=42)
```

Random Forest
Performance

- AUC: 0.5926
- Accuracy: 0.77025
- Precision: 0.1071
- Recall: 0.2642
- F1: 0.1524

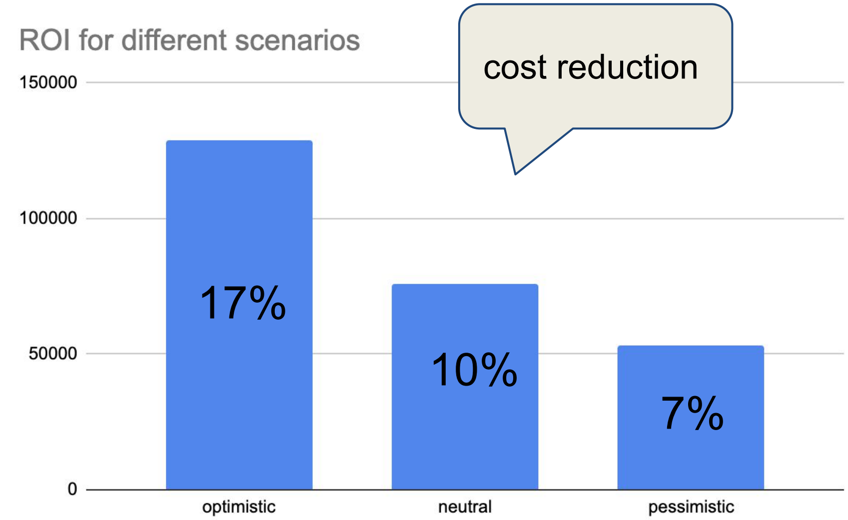


Summary

1. Strength in overall predictions
2. Potential for false positives
3. May capture more actual return cases

Conclusions

- The correlation between seasonal trends, specific months, and return rates underscores the pivotal role of timing in precise return predictions.
- Aligned with insights from Exploratory Data Analysis (EDA), evident seasonal return trends in June, July, and December were identified.
- This temporal pattern coincides with promotional activities, notably during the Christmas season.
- Leveraging the Random Forest model and key features, a concise ROI analysis was conducted.



Strategic Recommendation

Seasonal Inventory Management
Strategy: Adjust inventory levels based on projected return rates during distinct seasons or specific months.

Dynamic Pricing Strategies:
Strategy: Implement dynamic pricing based and adjust pricing during high-return periods to account for potential returns, ensuring profitability even if returns occur.

Thank you!