Machine Learning to Improve Marketing ROI

MIDS 207 Spring 2025 Section 6 Final Project

Saaketh Gunukula, Shreshta Keta, Vishal Saxena

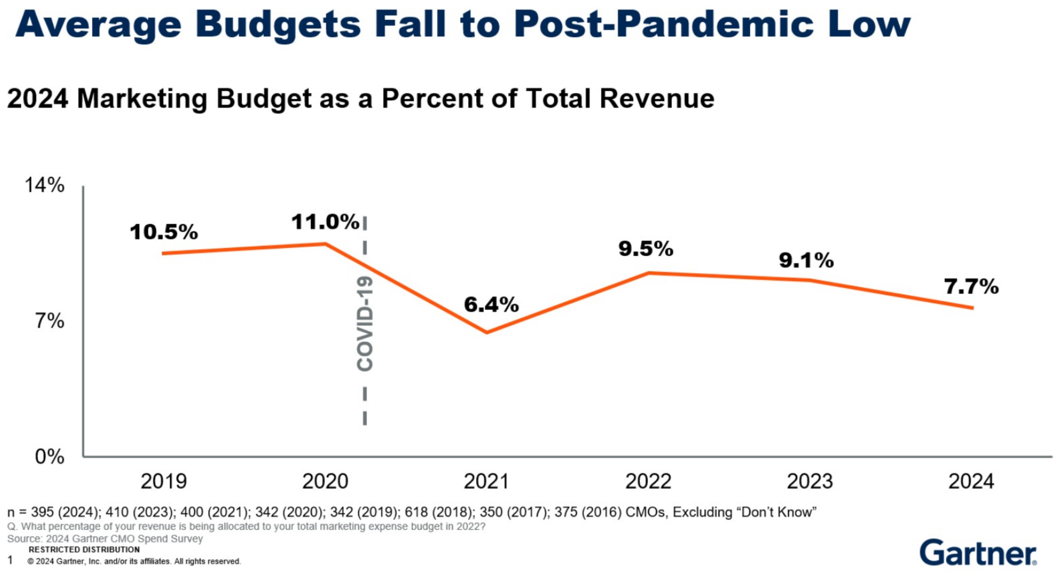
Contents

- Motivation - market need, vision, our approach
- Data, Modeling, and our experiments
- Conclusions
- Appendix

    References
    Git hub repository link
    Git hub file dictionary
    Contributions

Recurrent Marketing Imperative - *Do More with Less*

Average 2024 marketing spend was 7.7% of revenues, down from 9.1 % in 2023.



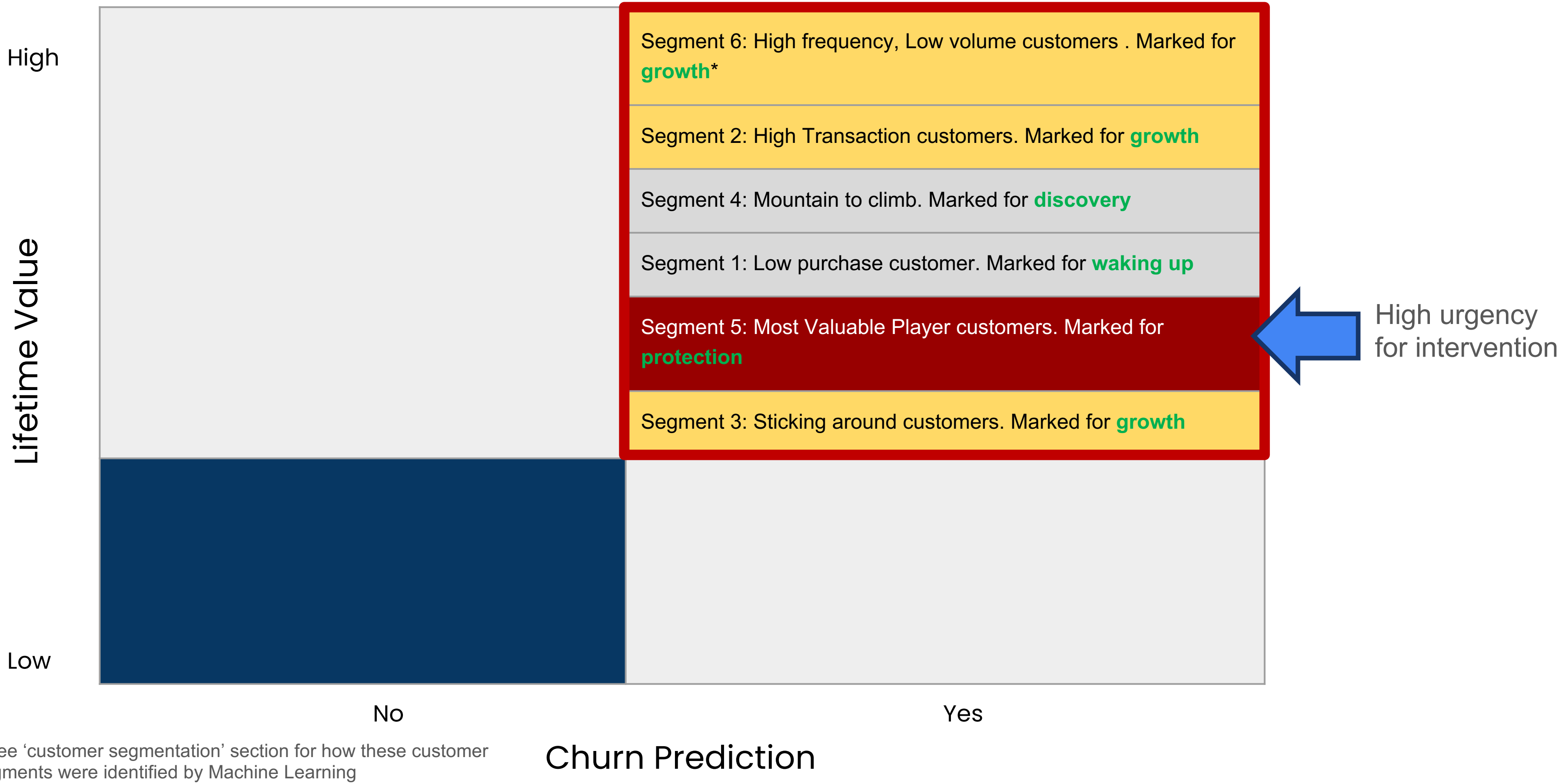
For enterprise companies, this spend *should* be 12-15 % of revenues[1].

Marketing has to do more with less. AI and Machine Learning can help.

Customer segmentation, churn prediction and retention, customer lifetime value (LFTV) collectively comprise 13-27% of a marketing budget.

See Appendix for all used references

Our vision combines all three – LFTV, Churn, and Segmentation – to identify urgency of marketing action



* See 'customer segmentation' section for how these customer segments were identified by Machine Learning

Introducing the work

Past Work: Use of ML for customer segmentation, churn and LFTV prediction has been around for some time. However, to our best knowledge, using them *collectively in a dashboard-style* use is missing or less prevalent.

This work: addresses above gap.

Data Challenges: Real-world datasets that cover all the three use cases are missing. Potential reasons:

- Different feature sets required by each use case. Customer segmentation (demographics+ interaction)
- Churn prediction (purchase behavior + customer service calls + returns + social media sentiments)
- LFTV (customer recency, frequency, monetary, purchase, lifespan related behavior)
- Data sets can be huge. Millions of transactions just from one store. Thousand to million product items.
- Little incentive for marketing teams to spend the effort to anonymize these huge data sets for ML learning.

Our Question

Our Goal: Improvement of Marketing Campaign ROI using ML

For this scope, in this work we address two questions:

- How can a Retail Chief Marketing Officer **improve their marketing campaign ROI** by launching effective intervention campaigns informed from Machine Learning models for customer lifetime value (LFTV) calculation and churn prediction?
- Use Machine Learning defin**ed** Customer Segmentation to increase specificity of intervention

Impact: An ML platform that can continuously incorporate new data to improve campaign performance using the above three models can learn from the market while creating a significant competitive advantage for any company.

Our Plan: Develop optimized models for each of the three use cases.

Our Models and their results

Churn Prediction

Goal: Predicting whether a customer will stop using the company's products

Highest Correlation with Target: Tenure, Complaint, Marital Status

Final model: Random Forest Classifier with Grid Search

Validation Accuracy: 95%
AUC: 98%
Improvement over baseline: 32.51%

Inference Accuracy: 97%

Customer Segmentation

Goal: Understand customer demographics and consumption patterns via clustering algorithms.

Final model: K-Means with 6 clusters

Output: 6 Customer segments with meaningful action steps

Inertia: 2.94 on elbow-selected 6 clusters

LFTV Prediction

Goal: Determine customer lifetime value based on purchasing habits (continuous variable)

Highest Correlation with Target: Purchase frequency, membership years, purchase value

Final model: Feed Forward Neural Network with Feature Selection

Validation RMSE: 23.07%
Improvement over baseline: 74.9%

Test RMSE: 23.32%

Due to data challenges mentioned earlier, our modeling for Churn prediction uses a Kaggle data set that is different than the Kaggle data set used for modeling customer segmentation and lifetime value predictions.

Churn Prediction

Data Statistics and Preprocessing

Source: Kaggle[2]

Key Statistics:

Original Shape: (5630,20)
After cleaning, and feature engineering: (5630, 9)

Churn Model Data Cleaning

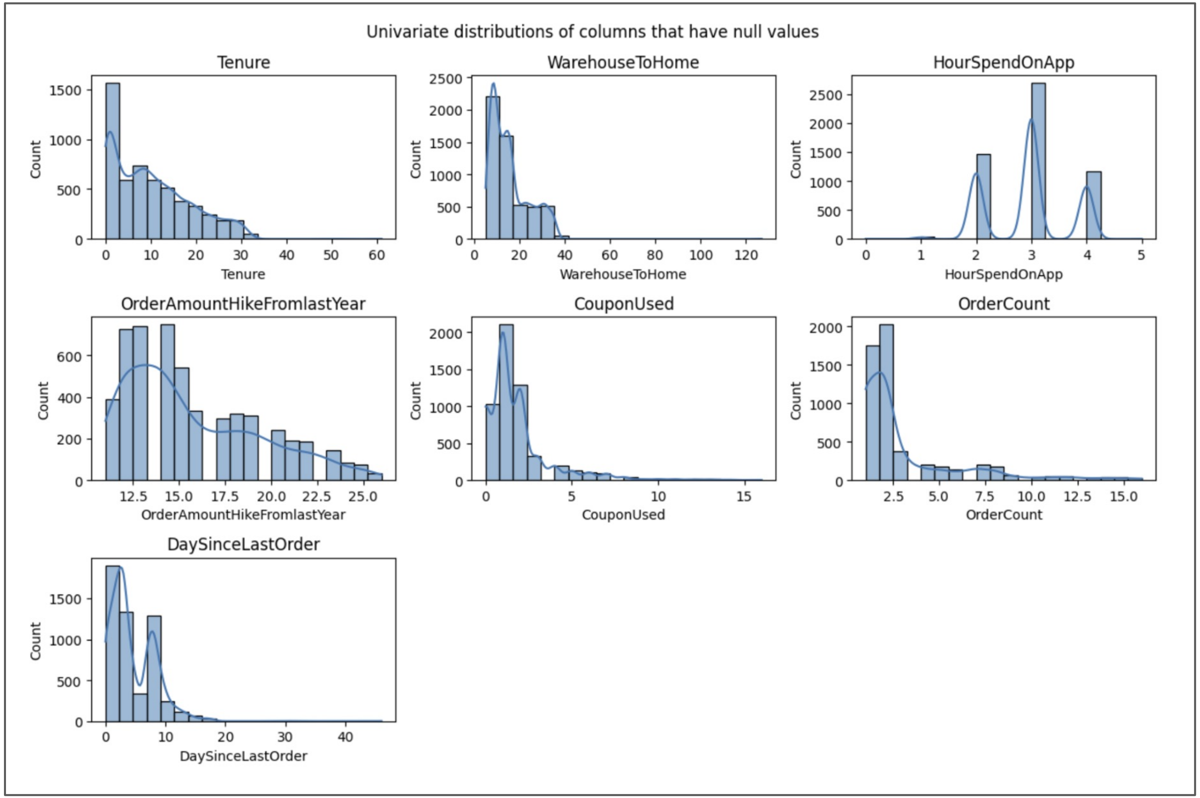
33% of the data had at least one null value in the row

The data to the right are the distributions of variables with null values

Since none had a normal distribution, we **imputed** null values using the **median**, which is better for skewed distributions

Additionally, we **One-Hot Encoded** the following (mostly nominal) categorical variables:

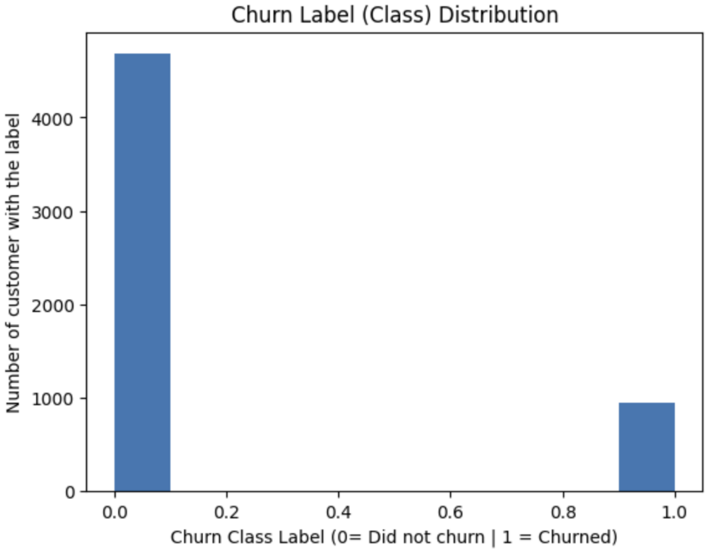
- Preferred Login Device
- Preferred Payment Mode
- Gender
- Preferred Order Category
- Marital Status'



Churn Model EDA - SMOTE reduced class imbalance

Distribution of Target Variable

Original: 83% not churned, 17% churned
After SMOTE: 71% not churned, 29% churned

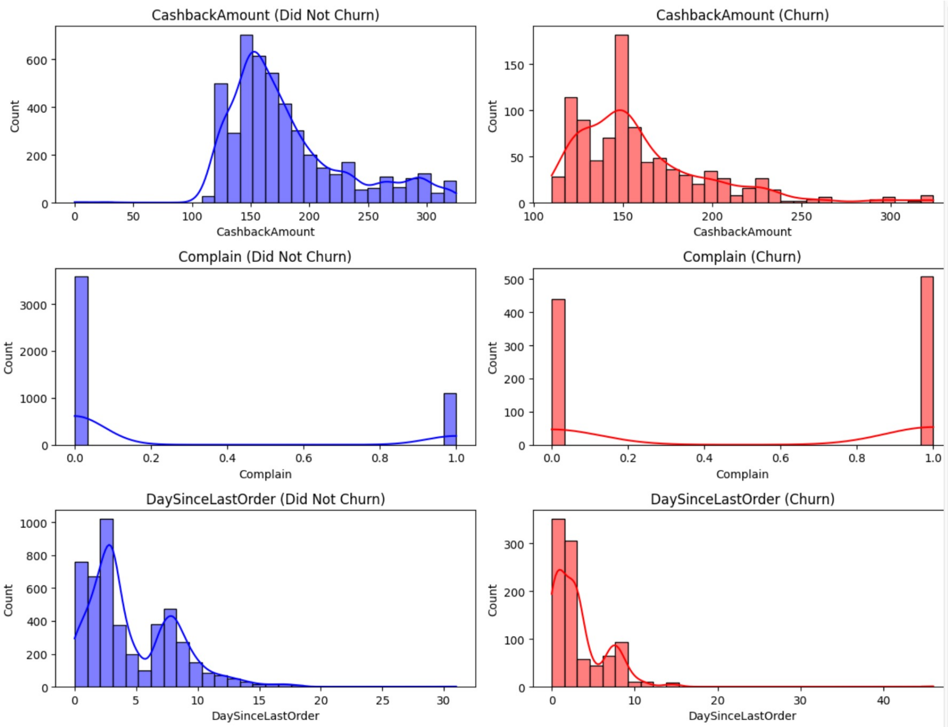


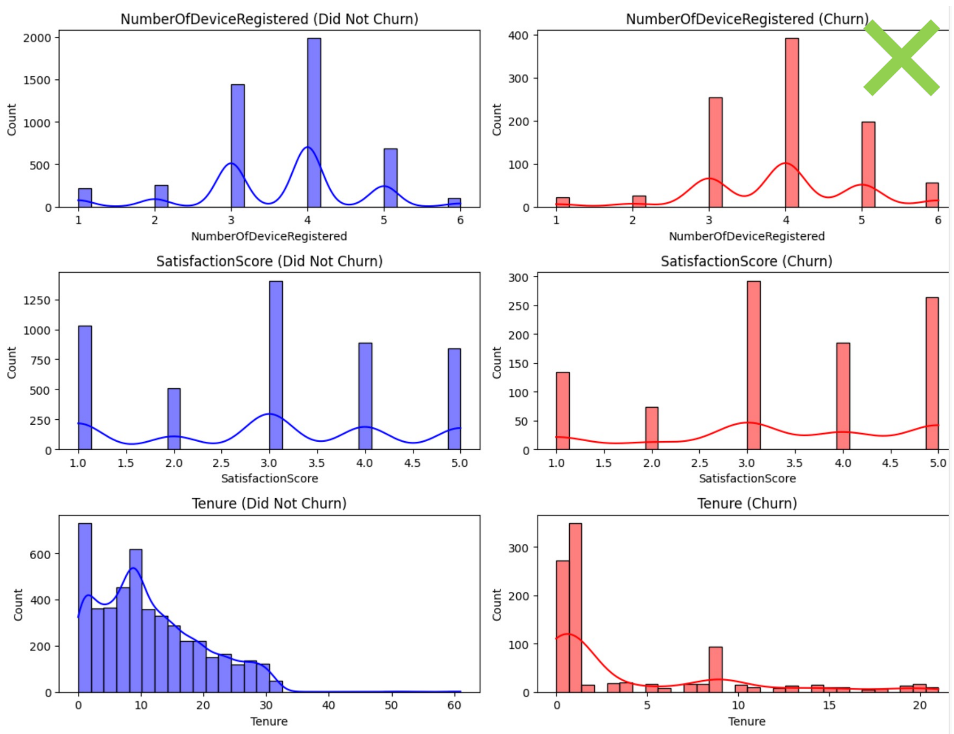
Variables with the Highest Correlation to Churn

Dropped columns with corr < 0.1 to reduce noise

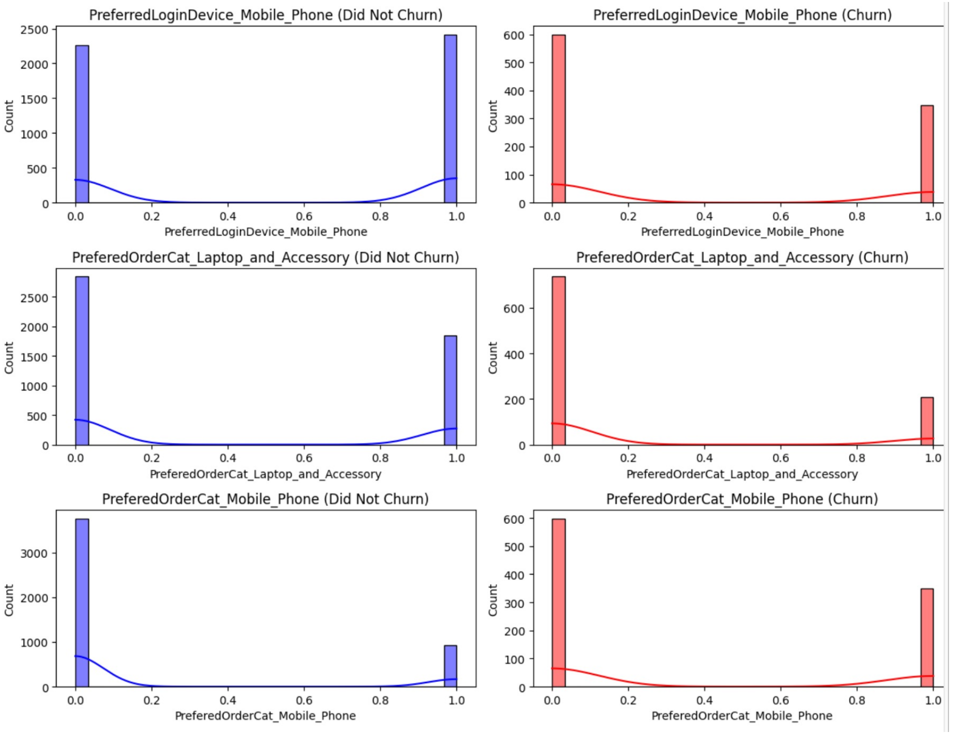
SatisfactionScore	0.105481
NumberOfDeviceRegistered	0.107939
PreferredLoginDevice_Mobile_Phone	0.111639
PreferedOrderCat_Mobile	0.113364
PreferedOrderCat_Laptop_and_Accessory	0.133353
CashbackAmount	0.154118
PreferedOrderCat_Mobile_Phone	0.154387
DaySinceLastOrder	0.155871
MaritalStatus_Single	0.180847
Complain	0.250188
Tenure	0.337831
Churn	1.000000

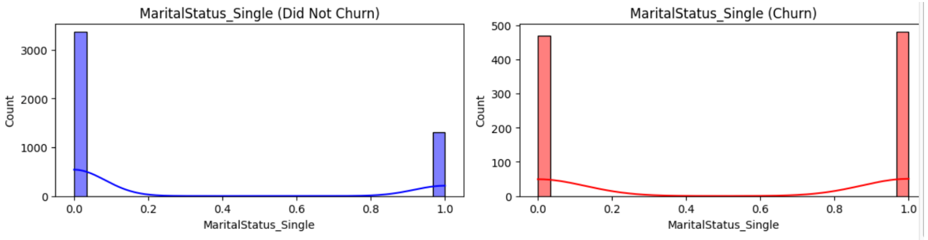
Feature Selection based on distribution by Class (1/2)





Feature Selection based on distribution by Class (2/2)





Outcome:

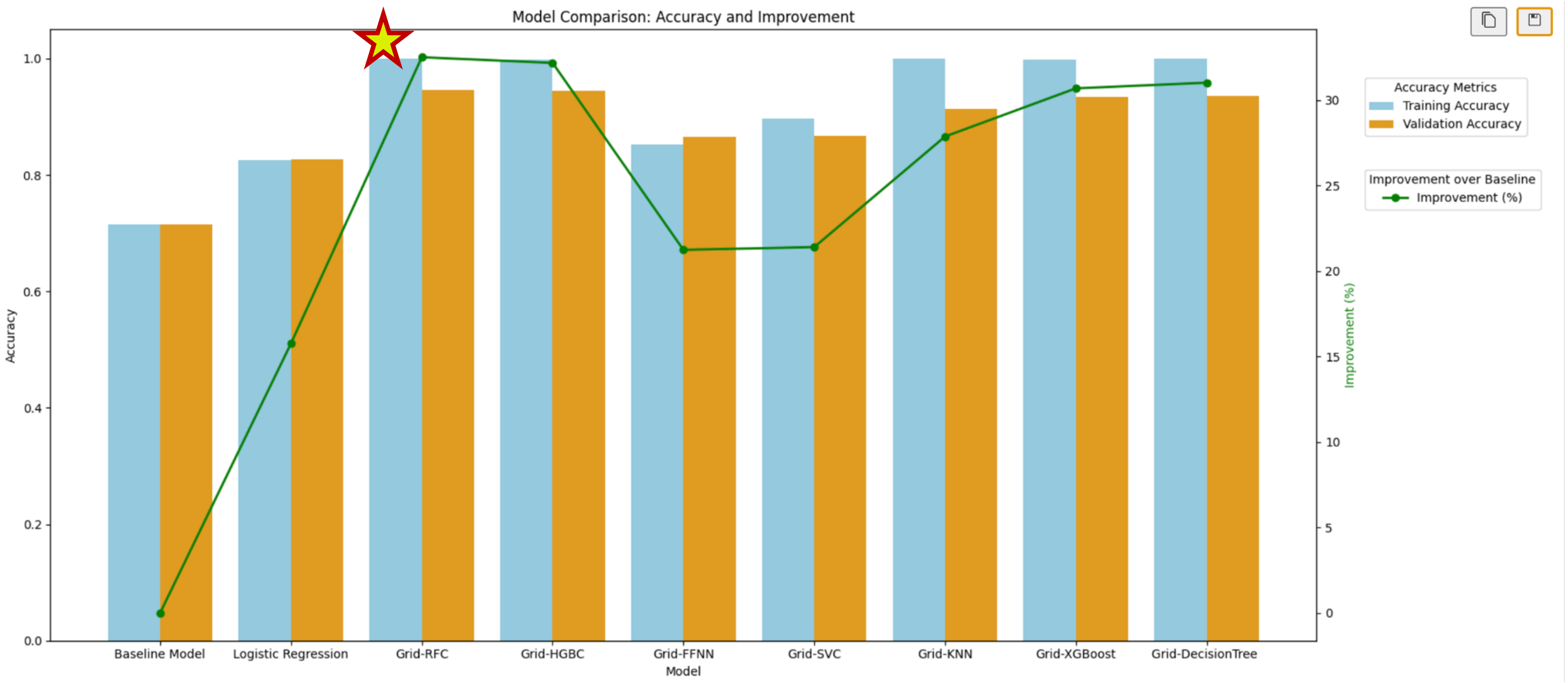
Distribution of all features are dissimilar across the two class labels (churn, did not churn) except for one feature - Number Of Devices Registered - indicating that all features except for number of registered devices can contribute information to the model.

=> We will drop feature for number of registered devices

Baseline and Experimentation

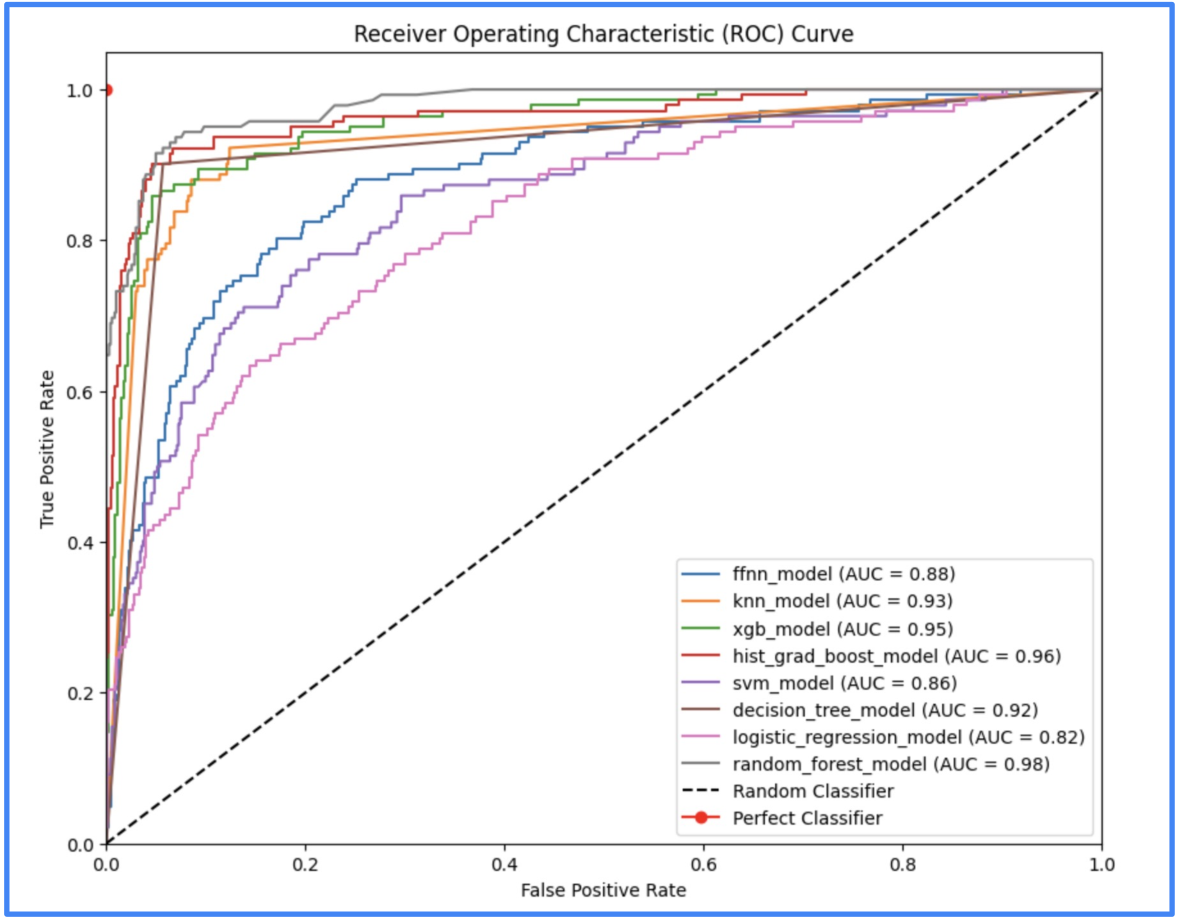
Model	Hyperparameters	Training Accuracy	Validation Accuracy	Improvement over Baseline %
Baseline: Majority Class Classifier	N/A	0.71	0.71	0.00
Logistic Regression	Sigmoid activation, Cross Entropy Loss, 50 Epochs, Early Stopping	0.83	0.83	15.76
Grid Search Random Forest Classifier	**200 estimators, max depth = 0, max features = sqrt, min split = 2**	**1.00**	**0.95**	**32.51**
HistGradientBoostingClassifier comprehensive grid search	Learning rate = 0.2, max iterations = 300, min samples leaf = 20, l2 reg.	1.00	0.94	32.18
FFNN with Grid Search	2 hidden layers of 32 units, lr = 0.01	0.85	0.87	21.24
Support Vector Machine, grid searched, 5 fold cross validation	C = 10, degree = 3, gamma = auto, kernel = rbf	0.90	0.87	21.40
K-Nearest Neighbors, grid searched, 5 fold cross validation	Manhattan distance, 3 neighbors, p = 1, weights = distance	1.00	0.91	27.87
XGBoost Classifier, grid searched, 5 fold cross validation	Learning rate = 0.2, Max depth = 9, log loss, n_estimators = 150, col sample by tree = 0.7	1.00	0.93	30.69
Decision Tree, grid searched, 5-fold cross-validation	Entropy criteria, no max depth, no max features, min sample split = 2	1.00	0.94	31.02

Churn Prediction - Model Performance



Model Selection Criteria: (1) Highest Validation Accuracy, (2) Least gap between validation and training accuracy

Churn Prediction - Our Model’s Performance



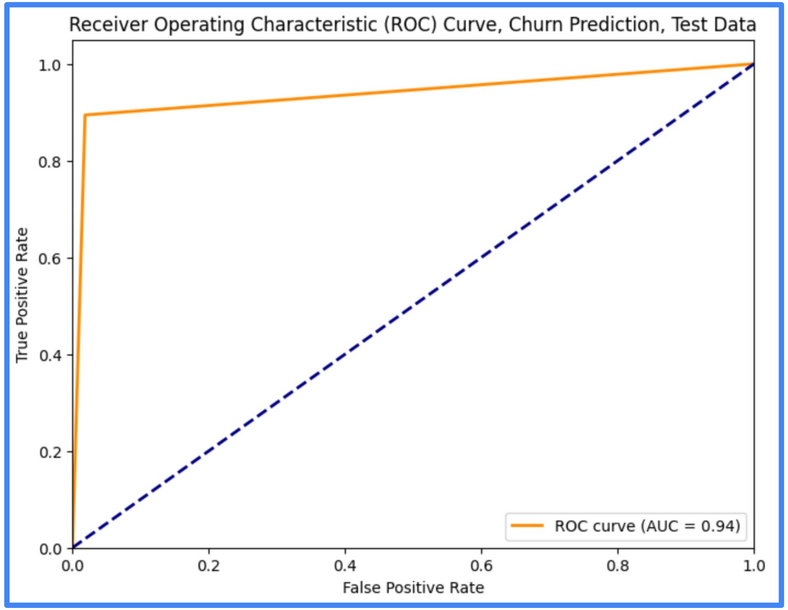
Our final selected model –Random Forest Classifier – has an AUC of 0.98 on validation data.

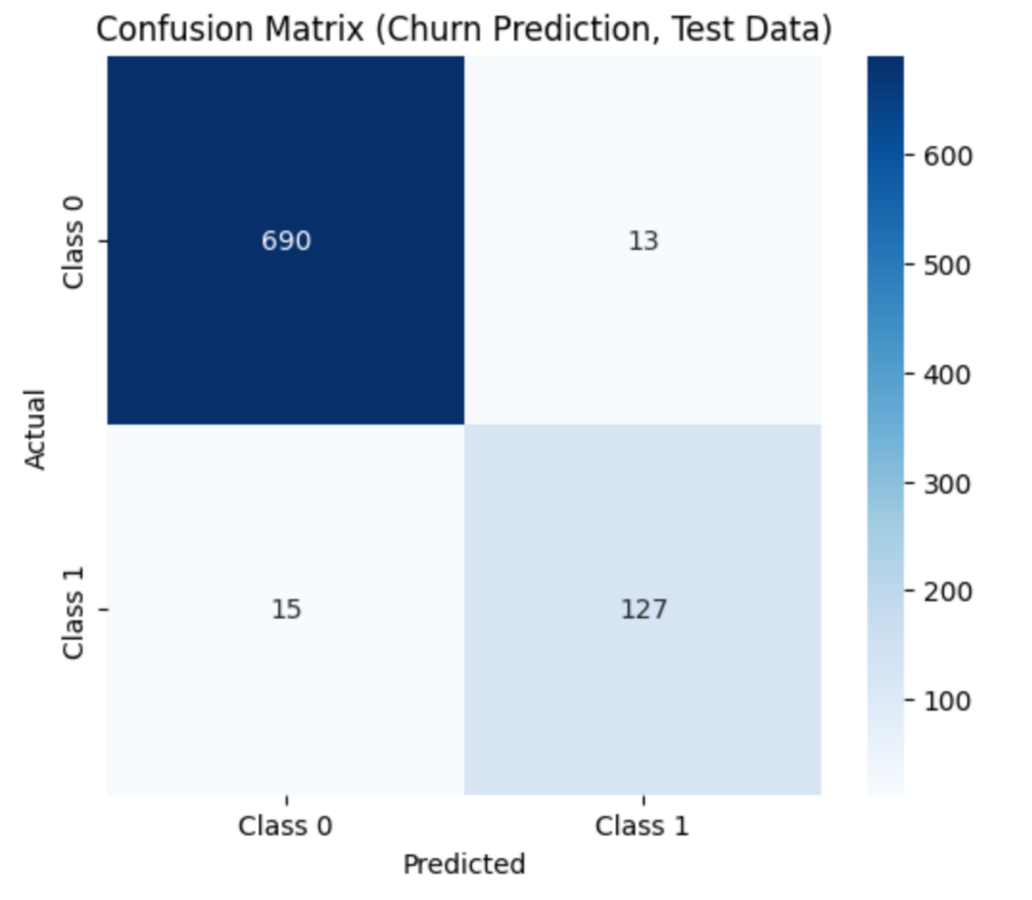
Churn Prediction - Inference Accuracy **97%**, AUC 94%

```
Classification Report (Inference)
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       703
           1       0.91      0.89      0.90       142

    accuracy                           0.97       845
   macro avg       0.94      0.94      0.94       845
weighted avg       0.97      0.97      0.97       845
```





Customer Segmentation

Customer Segmentation Data Preprocessing

Same dataset as Customer Lifetime Value, but different feature engineering and selection process, as described below:

- Created new (derived) features: Recency, Frequency, Monetary
- Nominal (one hot) encoded loyalty program feature (Yes/No)
- Ordinal Encoded the following features:
  - app_usage, social_media_engagement, income_bracket, promotion_effectiveness
- Standard Scaled all numerical features
- Original and post processing data shapes: `(500000, 80)`, `(500000, 15)`

Column Name	Data Type
recency	int64
frequency	int64
monetary	float64
customer_lifetime_value	float64
customer_lifespan	float64
total_transactions	int64
loyalty_program	object
avg_discount_used	float64
app_usage	object
social_media_engagement	object
income_bracket	object
age	int64
promotion_effectiveness	object
online_purchases	int64
in_store_purchases	int64

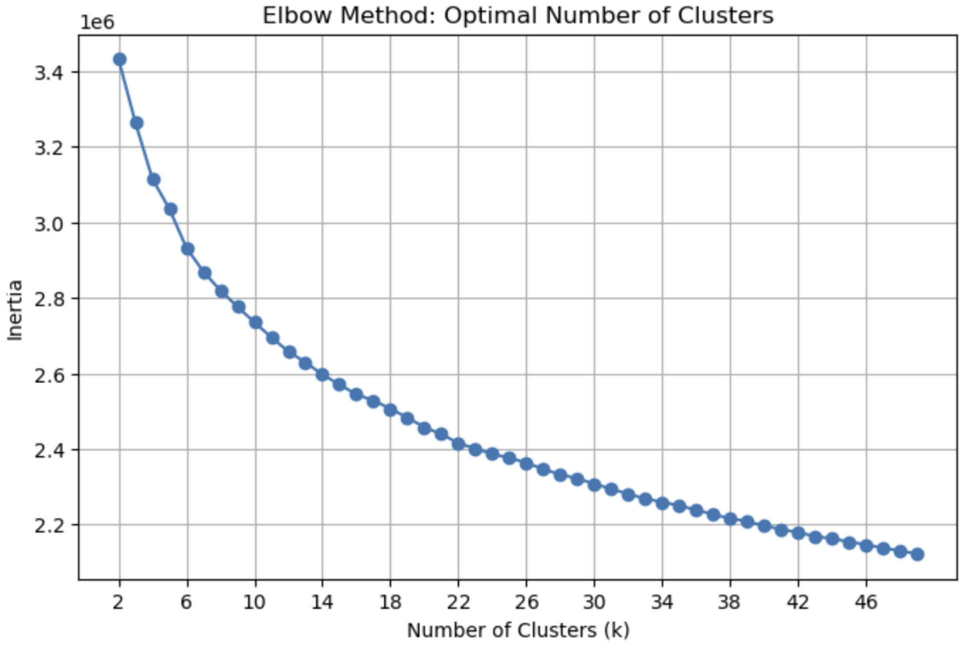
Domain Knowledge based feature Selection for Customer Segmentation

Core Customer Metrics	Recency – Days since the last purchase Frequency – Total number of purchases Monetary – Average or total amount spent Customer Lifetime Value – Estimated value of the customer over time Customer Lifespan – Duration between first and most recent purchase Total Transactions – Total number of transactions (measures engagement level)
Loyalty & Commitment	Loyalty Program – Indicator of brand commitment
Price Sensitivity	Avg Discount Used – Customer's price sensitivity
Digital Engagement	App Usage – Level of digital engagement
Brand Interaction	Social Media Engagement – Measures brand interaction level
Economic Status	Income Bracket – Proxy for customer spending power
Life Stage Behavior	Age – Indicates life stage & related buying patterns

Segmentation Experimentation

Initially ran K Means with 5 clusters as a baseline

Used **inertia score** to select the number of clusters via **Elbow Plot**

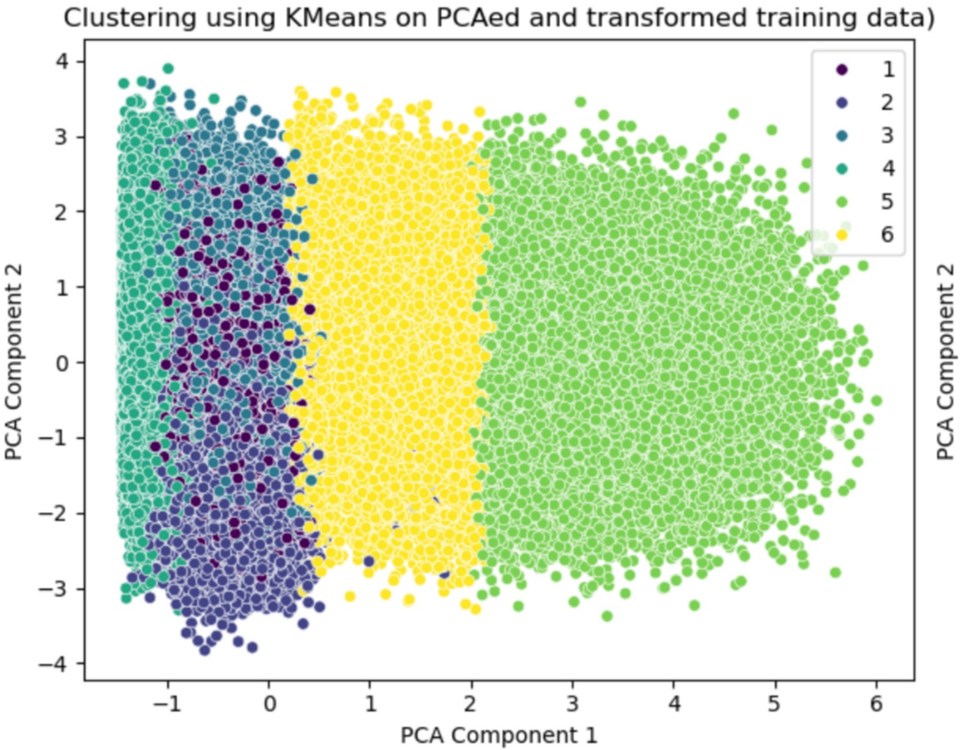


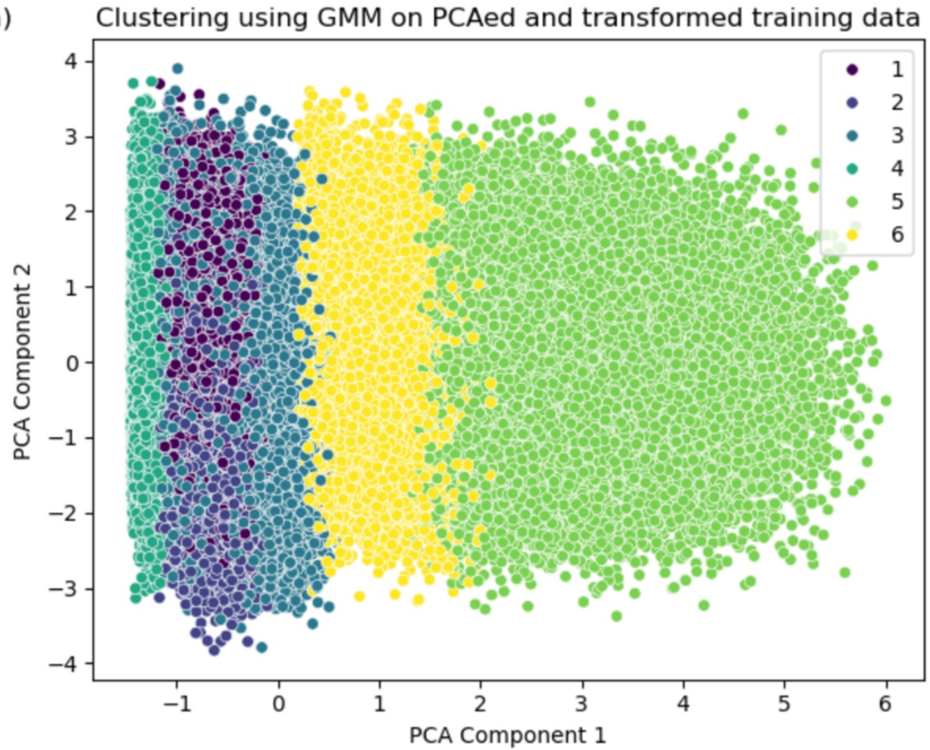
Based on these results, we ran K Means on 6 clusters, which has inertia 2.94. This is 38% higher than lowest inertia (highest accuracy).

To keep number of customer segments meaningful, we accept this accuracy, and chose 6 as an optimal number of clusters.

Next, we ran Gaussian Mixture Model (GMM). GMM actually outperformed K-Means

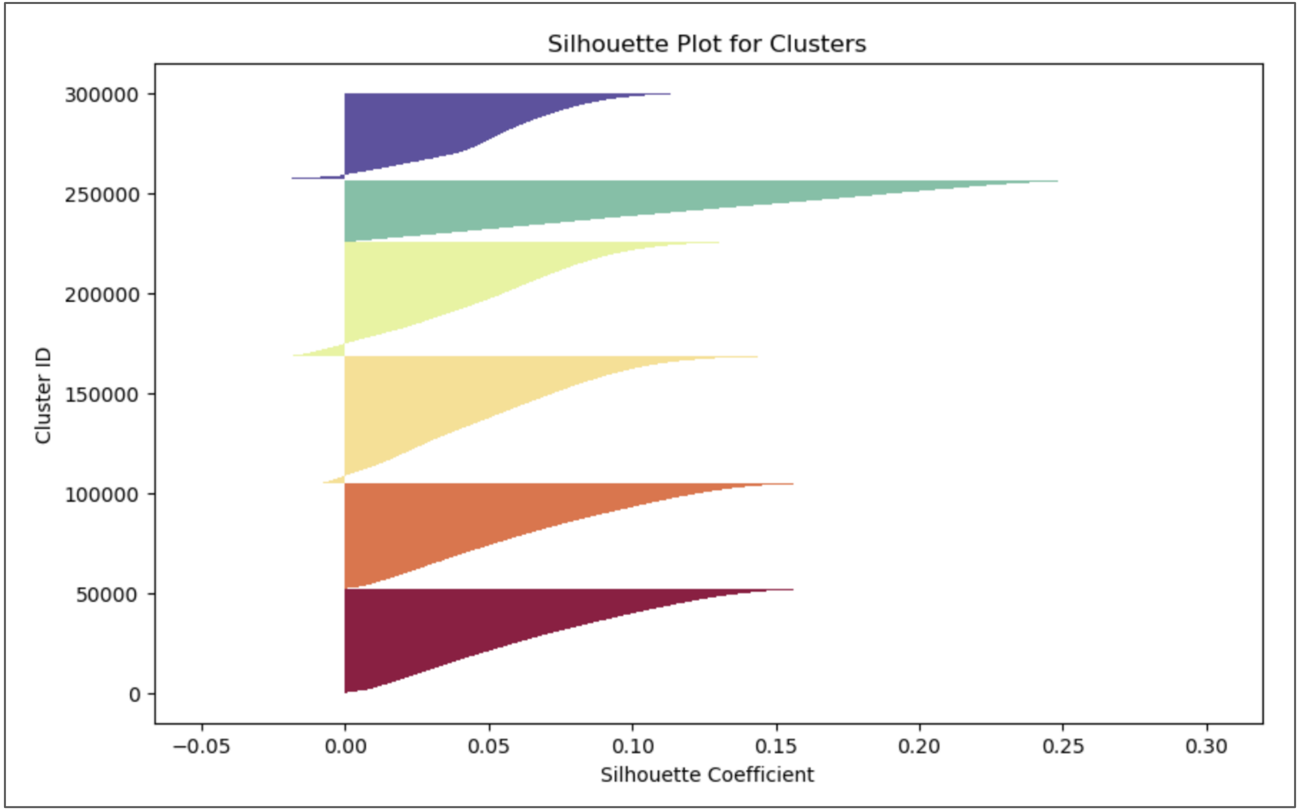
Clusters 1, 2, and 3 that have poor separation in K-Means have better quality cluster separation using GMM





Better machine would run GMM on our full data set, w/o PCA. Absent that, we stick with K-Means

Silhouette Plot (K-Means)



Our clusters' silhouette interpretations:

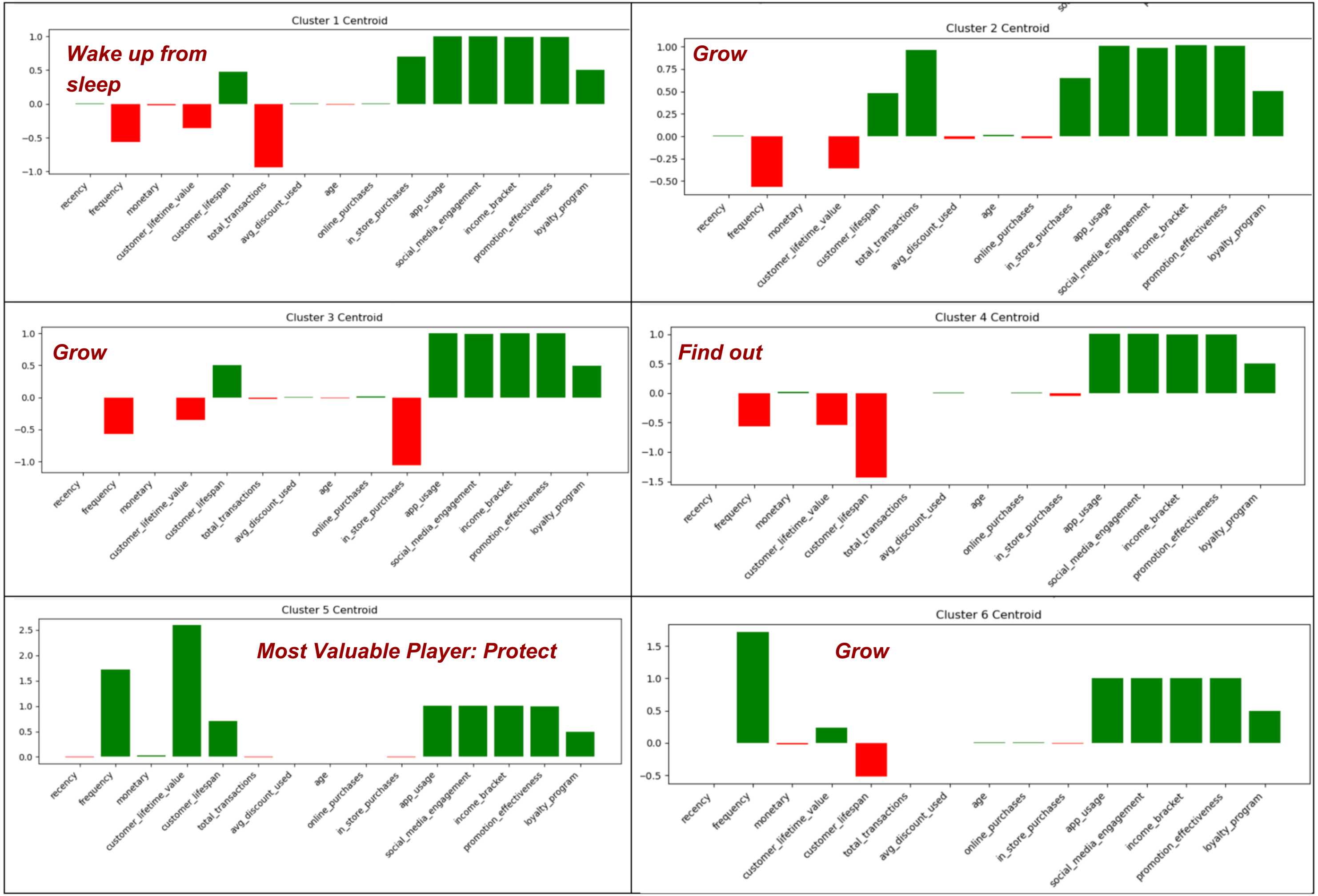
Very few points have negative scores, and are far away from -1, indicating that very few points are misclassified (are in the wrong cluster)

But most silhouette scores are between 0 and 0.15 indicating that most of the points are closer to the decision boundary.

Similar to PCA analysis findings, as we include features in the data set (and in our clustering) that have good univariate variation (e.g. are not close to uniformly distributed), quality of clustering should improve.

Also, selecting grid searched K-Means, and perhaps even a higher number of clusters (from 6) should improve silhouette-coefficient scores.

Customer Segments from K Means



Cluster 1, Segment: **Moderation Molly, Digitally active, Doesn't buy**

Action: Use social media, app to increase online purchases and monetary, frequency values.

Cluster 2, Segment: **High transaction, low frequency buyer.**

Action: Launch promotions to increase frequency (that should drive up monetary scores as customer tends to buy a lot when she makes a purchase).

Cluster 3, Segment: **In-store hating, low lifetime value customer who somehow still sticks around**

Action: Launch promotions (online and digital channels) to increase frequency and monetary scores.

Cluster 4, Segment: **'Mountain to climb' customer segment.**

Most difficult segment: lowest lifespans, low customer lifetime value, low frequency.

Action: Conduct surveys to discover reasons for low traction.

Cluster 5. Segment: **Most Valuable Players Highest Lifetime Value, high frequency customers**

Action: Drive interventions if needed to minimize churn. Launch new campaigns to increase loyalty scores

Cluster 6, Segment: **High frequency, low volume buyers.** Purchases frequently

Action: Use online and digital channels for campaigns to increase retention rates, monetary scores.

Customer LFTV

Data Statistics and Preprocessing

Data Source: Retail Sales and Customer Behavior Analysis
https://www.kaggle.com/datasets/utkalk/large-retail-data-set-for-eda

Initial Shape: (1,000,000, 78)
After cleaning and EDA: (500,000, 24)

Feature Engineering to Develop our Ground Truth Customer LFTV:

- LFTV = Average Purchase Value x Purchase Frequency x Customer Lifespan (CL)
- Customer Lifespan was calculated using conditional logic based on loyalty membership years
- Final LFTV = mean of calculations from APV x PF x CL, and Total sales x CL, plus some noise.

Our LFTV Modeling Journey

- 2 Complete feature overhauls
- Derived features, noise-adjusted label, and noise adjusted features

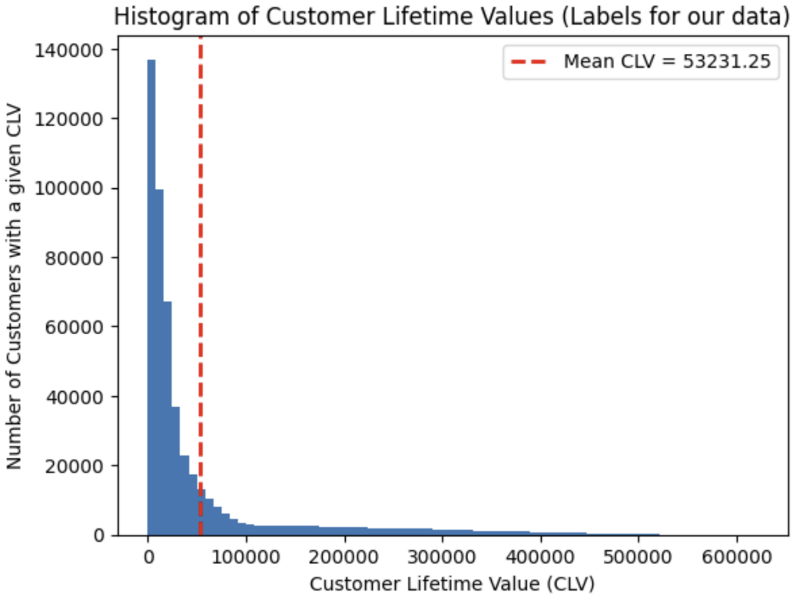
- Brought down first baseline validation R-MSE of 178.29%, to 23.07%

LFTV Data Cleaning

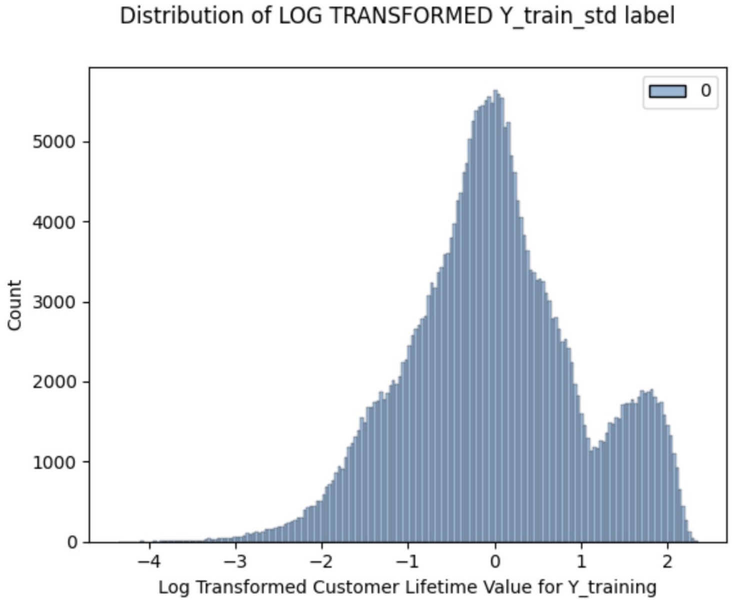
- Added noise-adjusted columns. Dropped their 'pure' counterparts:
  - Used normal distribution to add noise
  - Drop avg_purchase_value, purchase_frequency, membership_years, total_sales
- One Hot Encoded (ordinal) categorical features
  - income_bracket, app_usage, social_media_engagement
- Addressed skewness, outliers in Y through transformation
  - Log transformation of Y (customer lifetime value)
- Standard scale numeric features

Customer LFTV EDA

Distribution of LFTV Model Target Variable before transformation



Distribution of LFTV Model Target Variable after log transformation



Customer LFTV Experimentation

Model	**Hyperparameters**	**Training RMSE**	**Validation RMSE**
Baseline-I: Simple Linear Regression with No Feature Selection	N/A	179.01%	178.29%
Baseline-II: Simple Linear Regression with Feature Selection	N/A	93.02%	92.52%
Feed Forward Neural Network with Feature Selection	2 hidden layers, unit sizes 32 and 16, relu activation, adam optimizer, MSE loss function, Early Stopping, 3585 parameters, 50 epochs, batch size of 32	178.73%	178.00%
Feed Forward Neural Network with Feature Selection	**2 hidden layers, unit sizes 32 and 16, relu activation, adam optimizer, MSE loss function, Early Stopping, 3329 parameters, 100 epochs, batch size of 32**	**23.00%**	**23.07%**

Final selected model (FFNN, 100 epochs) had an improvement over baseline-II of 74.9%, and test RMSE of 23.23%

Conclusion

Key Results

Churn prediction model of high accuracy. LFTV model of reasonable accuracy (23% RMSE on LFTV, a directional indicator, is generally considered acceptable). Business relevant customer segments. See future work for further model optimizations.

Takeaways

Feature engineering has the most impact. Incremental feature addition is a good practice. Selected data set should have features that steer away from uniform distribution. SMOTE to address class imbalance is helpful, as also rejecting features that have similar distribution per class.

Future Work

LFTV model can benefit from higher epoch runs. Segmentation model can benefit from rationalizing feature set, and running GMM over the full data set on a more powerful machine. Our churn model is production ready.

Use a real singular data set to run all three models. Develop an app dashboard that identifies if a customer or a set of customers are candidates for urgent marketing action.

Appendix

1. References
2. Git hub repository link
3. Git hub file dictionary
4. Contributions

1. References

1. https://boomcycle.com/blog/right-percentage-of-gross-revenue-to-invest-in-marketing?utm_source=chatgpt.com
2. **https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction/data**

2. Git Hub Repository

https://github.com/Vsaxena702/mids_207_final_project.git

3. Git Hub File Dictionary

	Git Hub File Name	Comments
Churn Prediction	207FPCHURN_CLASSIFIER_2_VS3_ModelRunA.CL.ipynb	Definitive file for churn prediction
Customer Segmentation	207_FP_CUSTSEG_VSAPR7.ipynb	First file with full EDA, feature processing and K Means, and created customer segments
	207_FP_CUSTSEG_GMM_VSAPR16.ipynb	Second file that builds upon first, uses GMM, and compares K Means and GMM
Lifetime Value	207_FP_LFTV_Run_2_VSAPR6_CL_Final.ipynb	Final inference file for lifetime value using 100 epochs on GPU of Colab
	207_FP_LFTV_Run_2_VSAPR5.ipynb	Pre-final version, using Apple Mac M3. Inferior model performance.
	207_FP_LFTV_LR_VS2.ipynb	Earlier runs, changed feature selection
	207_FP_LFTV_LR_VS1.ipynb	First run, original feature selection

4. Contributions

Team Member	Contributions
Shreshta Keta	• Created the slide template and first comprehensive and good quality draft of the presentation • Sifted through each notebook to create a detailed Experiments log for churn, customer segmentation, and lifetime prediction • Captured EDA summary
Saaketh Gunukula	• Provided inputs to slides, and modeling • Participation in all team meetings and dry run.
Vishal Saxena	• Ownership: idea, data selection, business scenario • Soup-to-nuts model development for Churn prediction, Customer Segmentation, and Lifetime Value. Coverage: original data load, EDA, feature engineering, baseline and selected model runs, hyper tuning, final model selection, and inference. Sole authorship of all notebook files on git hub. • Authorship: Conclusions. • Final editor of presentation. Final inputs for business case, and vision.

Thank you!