ISTM 4214: Foundations of Artificial Intelligence

Project Report: Jonathan Beck

Table Of Contents

1. Data and Company Introduction

```
Rows: 7000 Columns: 20
```

Figure 1- Data Dimensions Upon Import

We utilized the [*Fintech Customer Lifetime Value Dataset*]() from Kaggle, which contained 7K customer instances, with no null values, and 20 descriptive attributes upon import, including various customer demographics, transaction histories, behavioral indicators, and service-interaction metrics. As a whole, the dataset provides a holistic view of customer engagement within a digital wallet environment that provides the necessary information to support our overarching project goal – to understand drivers of high-value customers and apply this knowledge to provide a competitive advantage to the owners of said data.

```
# convert rupees to USD
data <- data %>%
  mutate(
    avg_transaction_value  = avg_transaction_value  * 0.011,
    total_spent            = total_spent            * 0.011,
    max_transaction_value  = max_transaction_value  * 0.011,
    min_transaction_value  = min_transaction_value  * 0.011
  )
```

Figure 2 - Conversion of Indian Rupees to USD

Because the original monetary values were presented in Indian Rupees, all transaction-based features were converted into USD using a fixed 0.011 INR → USD exchange rate (Figure 2).

A data.frame: 15 × 9

variable	count	mean	std	min	25%.25%	50%.50%	75%.75%	max
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
age	7000	42.63371	15.516036	16.00000000	29.00000	43.00000	56.00000	69.00000
total_transactions	7000	501.22143	286.277311	1.00000000	252.00000	506.00000	744.00000	1000.00000
avg_transaction_value	7000	109.41515	63.589689	0.11204409	53.98213	108.25290	164.50554	219.96090
max_transaction_value	7000	331.75556	241.957197	0.35043233	134.46762	280.16331	489.66956	1086.90166
min_transaction_value	7000	32.79705	24.170764	0.05079763	13.56373	27.20499	48.16635	109.08732
total_spent	7000	55022.24073	48295.447818	16.47957988	15003.89836	41317.15423	84100.58490	214145.00447
active_days	7000	181.93486	105.102598	1.00000000	90.00000	182.00000	273.00000	365.00000
last_transaction_days_ago	7000	183.84771	105.063709	1.00000000	93.00000	184.00000	275.00000	365.00000
loyalty_points_earned	7000	2501.54543	1446.680026	0.00000000	1254.75000	2466.00000	3792.25000	5000.00000
referral_count	7000	24.83700	14.560352	0.00000000	12.00000	25.00000	37.00000	50.00000
cashback_received	7000	2496.52503	1440.651412	0.23434896	1269.42370	2478.94335	3749.37525	4999.69848
support_tickets_raised	7000	10.01757	6.037067	0.00000000	5.00000	10.00000	15.00000	20.00000
issue_resolution_time	7000	36.52801	20.389399	1.01985270	19.12730	36.25740	54.06859	71.97895
customer_satisfaction_score	7000	5.47900	2.860197	1.00000000	3.00000	5.00000	8.00000	10.00000
ltv	7000	511919.71684	439055.079701	3770.49542473	148205.83384	387817.99588	774857.82570	1956987.63826

Figure 3 - Descriptive Statistics: Continuous Features

The above table highlights meaningful variability across behavioral and monetary features.

A tibble: 4 × 6

statistic	customer_id	location	income_level	app_usage_frequency	preferred_payment_method
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
count	7000	7000	7000	7000	7000
unique	7000	3	3	3	4
top	cust_0000	Urban	Middle	Daily	UPI
freq	1	2368	2391	2346	1791

Figure 4 - Descriptive Statistics: Categorical Features

The above table highlights meaningful categorical distribution indicators. For example, urban customers form the largest customer segment, and UPI is the most commonly used payment preference.

In preparing the data for modeling, extensive data preparation was conducted – one-hot encoding of all categorical variables, keeping all dummies for clustering and using a baseline reference indicator for supervised models; scaling of all continuous variables, especially critical for K-Means clustering due to the algorithm's reliance on distance to determine similarity; and the instantaneous removal of clearly non-informative identifiers, such as *customer_id*. Our cleaned dataset provided the foundation for both unsupervised learning (cluster discovery) and supervised learning (classification of high- and low-valued customers and class drivers).

2. Initial Analysis & Insights

Our initial analysis had three major objectives: *1) identify natural customer segments through unsupervised learning, 2) characterize behavioral and demographic differences between high-value and low-value customers,* and *3) make attempts at developing supervised models that could be deployed to predict high LTV customers and/or their proxies.*

2.1 Unsupervised Learning: Customer Segmentation with K-Means

K-Means models for k = 2 through 9 clusters were evaluated:

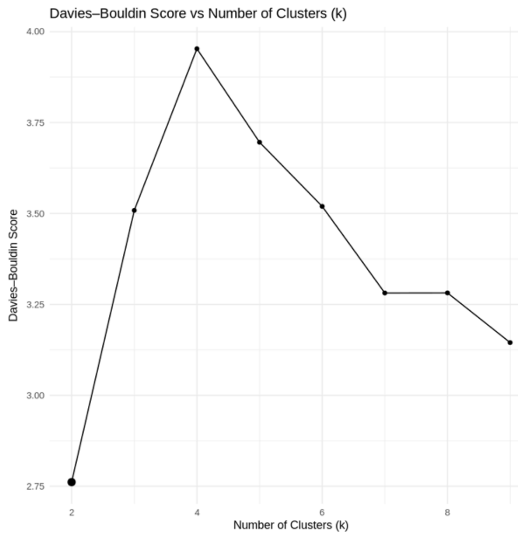


Figure 5 - Cluster Evaluation Metrics: Davis-Bouldin Index (cluster compactness - lower score better)

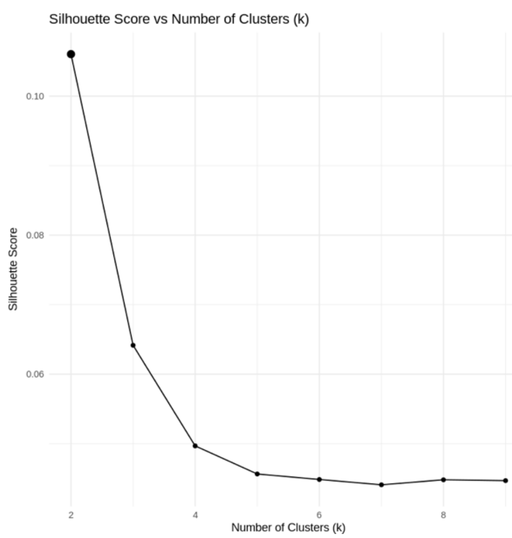


Figure 6 - Cluster Evaluation Metrics: Silhouette Score (cohesion/separation - higher score better)

Both metrics favored two clusters, indicating that two clusters formed the most natural customer segmentation (Figures 5 & 6).

	mean	median	count
High	777366.8	747014.7	3260
Low	280540.7	209660.7	3740

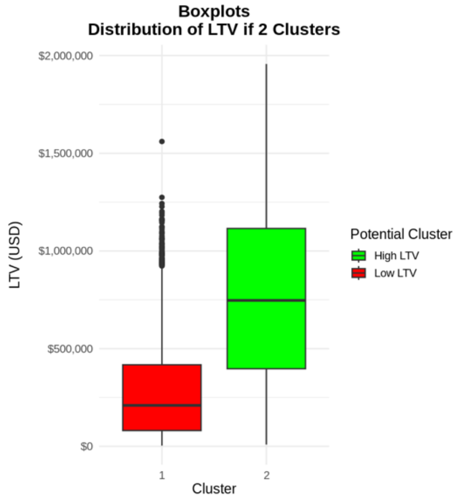


Figure 7 - Boxplot of 2-Cluster Solution

The above pivot table of aggregated metrics indicates that the clusters differ dramatically in CLV. Moreover, the “2-class” boxplots visually confirm separation, with minimal overlap between high and low LTV distributions between cluster segments.

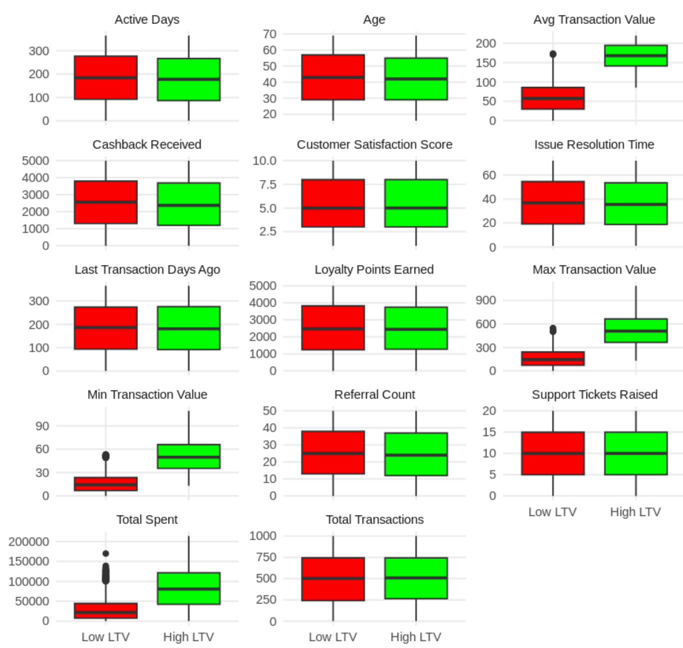


Figure 8 - Distributions of Continuous Features Grouped by Clusters

The above boxplots reveal systematic behavioral differences concerning monetary dimensions; specifically, high LTV customers transact with much higher average values, greater maxima, and substantially higher total spend. These patterns validate that monetary metrics drive LTV customer segmentation.

```
73.94% of the labels can truly be defined as 'accurate'.
26.06% of the labels can truly be defined as 'inaccurate'.
```

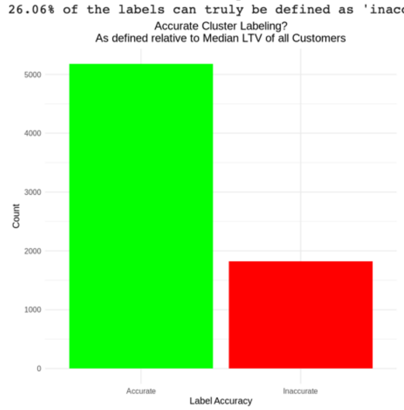


Figure 9 - Validation of Cluster Labels

To assess segmentation reliability, the cluster labels were validated against a true median LTV threshold: thus, we discovered that 73.94% of clustering labels were aligned with true high/low LTV definitions and 26.06% were misaligned (Figure 9). Given that clustering is unsupervised, this level of agreement suggests that the clusters were successful in identifying meaningful structure without relying on outcome labels.

2.2 Transition to Supervised Learning – Decision Tree Classifier to Determine Feature Importance

```r
# Creating binary target ltv_01
# (1 = at/above median LTV, 0 = below)
data_for_supervised <- data_for_supervised %>%
  mutate(
    ltv_01 = if_else(
      ltv >= median(ltv, na.rm = TRUE),
      1L, 0L
    )
  )
```

Figure 10 - Creating our Binary Classification Label

Because the firm ultimately requires a predictive model to classify customers as high/low LTV value as accurately as possible, LTV was binarized at a threshold of the median LTV to ensure class validity (Figure 10). As such, this derived binary feature was used as our response variable; moreover, the split produced a perfectly balanced dataset (50% of data in each class), facilitating fair model training. Lastly, for each supervised model, 90% of the data was used for training and 10% was set aside as a validation/test dataset.

	feature	importance
	<chr>	<dbl>
total_spent	total_spent	3112.1526
avg_transaction_value	avg_transaction_value	1698.3489
total_transactions	total_transactions	1583.3411
max_transaction_value	max_transaction_value	1464.3674
min_transaction_value	min_transaction_value	1457.4273
support_tickets_raised	support_tickets_raised	100.1362
age	age	0.0000

Figure 11 - Single Decision Tree: Initial Feature Importance (Top 5 features shown)

Above (Figure 11), we see that using *total spent* as a predictor for high/low CLTV makes the use of any other predictors either minimal or unnecessary. This is because *total spent* is an almost perfect proxy for CLTV - knowing the total amount that a customer has spent with the company provides all the information necessary to determine a customer's LTV. As such, we dropped this variable and *total transactions*, for similar reasons, as candidate regressors to obtain a more nuanced understanding of the relationship between the independent and dependent variables.

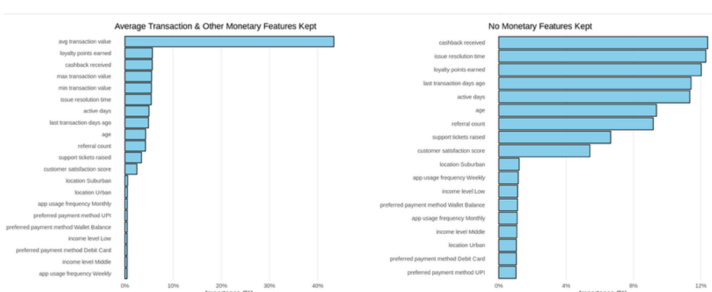


Figure 12 - Ensemble Feature Importance via Bagging

To improve stability and reduce variance in the *feature importance* metrics, an ensemble of 100 bagged decision trees was applied. This strengthened our confidence in the stability, by reducing the variance, of the importance rankings for each feature. In the resulting plots (Figure 12), we see that, when monetary features are kept as regressors, *Avg Transaction Value* drowns out the influence of all other predictors. When monetary features were excluded, importance was shifted to a variety of descriptive customer engagement features. This two-model perspective was critical: leveraging monetary predictors would produce a model with the best predictive accuracy, whilst removing them would provide enhanced customer characteristic and behavioral insights.

2.3 Decision Tree Modeling with Cross-Validation (No Monetary Regressors Used)

```
# ---------------------------------------------------
# 2. Define hyperparameter grid
# ---------------------------------------------------
param_grid <- expand.grid(
  maxdepth = c(2, 4, 6, 8, 10, 12),
  minsplit = c(2, 5, 10),
  minbucket = c(1, 2, 5)
)
```

```
# 5-fold CV, stratified
folds <- createFolds(y_train, k = 5, list = TRUE, returnTrain = FALSE)

cv_results <- param_grid
cv_results$mean_accuracy <- NA_real_
```

Figure 13 - Decision Tree Design

A Decision Tree, using a Grid Search for automated hyperparameter tuning to ensure a model was induced with the ability to generalize to unseen data, and 5-fold CV for performance validation, was implemented to create an interpretable customer segmentation framework and initial prototype predictive model (Figure 13).

```
          Reference
Prediction   0   1
         0  69  86
         1 281 264

               Accuracy : 0.4757
                 95% CI : (0.4382, 0.5135)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.9071

                  Kappa : -0.0486
```

Figure 14 - Model Evaluation: Confusion Matrix (Holdout Sample)

The Decision Tree was evaluated on the 10% validation set using a confusion matrix (Figure 14). The results show an overall accuracy of ~47.57%, precision of ~48.44%, and recall of ~75.43%. Cumulatively, these performance metrics indicate that, at least this classifier, fails exponentially when all monetary predictors are removed as regressors. From an early-stage analytics modeling design perspective, accurate classification may only be feasible if monetary indicatorsr are incorporated.

2.4 Evaluating a Regularized Logistic Regression Model, including Monetary Regressors

When monetary features from the raw dataset (not including *total spent* and *total transactions)* were used as predictors, the grid search using Stratified K-Fold CV selected the following parameters to induce the regularized logistic model:

A data.frame: 1 × 4

	log__C	log__l1_ratio	log__penalty	log__solver
	<dbl>	<dbl>	<chr>	<chr>
1	10	0.7	elasticnet	glmnet

Figure 15 - Regularized Lasso Logistic Regression Model

The weights of the scaled coefficients of the top five most influential regressors, including the intercept, can be shown below:

A data.frame: 18 × 2

Feature	Coefficient
<chr>	<dbl>
avg_transaction_value	-2.6099187
intercept	1.4768764
min_transaction_value	-0.3064622
max_transaction_value	-0.2854515
age	0.0000000

Figure 16 – Feature Weights

We see that only the three monetary regressors had non-zero weights, effectively removing the influence of all other regressors used in the model to distinguish high vs low CLTV. In other words, we have valid support to infer that even a finely-tuned classification model’s ability to distinguish high from low CLTV is driven almost entirely by direct transactional proxies of the target rather than by the support of demographic or behavioral attributes.

The best regularized logistic regression model induced by our pipeline had its performance assessed and validated against a confusion matrix, and its results on our holdout sample can be seen below (Figure 17).

```
          Reference
Prediction HighLTV LowLTV
   HighLTV     260     90
   LowLTV       90    260

Accuracy: 0.7429
Sensitivity (Recall): 0.7429
Specificity: 0.7429
Precision: 0.7429
F1: 0.7429
```

Figure 17 - Confusion Matrix Results

At first glance, these metrics appear strong; however, the model's apparent success is not practical from a business standpoint – the model is essentially re-coding CLTV proxy regressors from the customers' historical transaction patterns to distinguish customers as high vs low LTV.

2.5 Decision Tree Analysis (to Visualize Monetary Influence - Total Spent & Total Transactions Dropped)

```r
dt_grid <- expand.grid(
  maxdepth  = seq(2, 12, by = 2),
  minsplit  = c(2, 5, 10),
  minbucket = c(1, 2, 5),
  cp        = 0
)
```

```r
ctrl_dt <- trainControl(
  method = "cv",
  number = 10,
  classProbs = FALSE,
  savePredictions = "final"
)
```

```r
set.seed(0)

dt_model <- caret::train(
  x = X_train_dt,
  y = factor(ifelse(y_train == 1, "HighLTV", "LowLTV"),
             levels = c("HighLTV", "LowLTV")),
  method = "rpart2",
  trControl = ctrl_dt,
  tuneGrid = expand.grid(
    maxdepth = seq(2, 12, by = 2)
  ),
  metric = "Accuracy"
)
```

Figure 18 - Parameters provided to induce Decision Tree

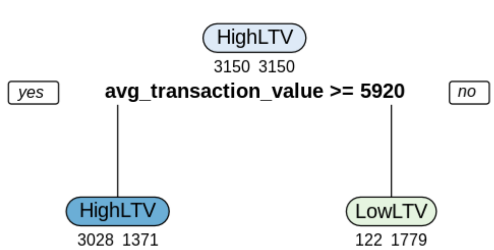


Figure 19 - Induced Decision Tree

As seen in Figure 18, even when given the opportunity to grow to a maximum depth of 12, the optimized decision tree (using all possible regressors other than *total spent* and *total transactions*) need not grow past a depth of 2 to optimize classification accuracy – it strictly relies whether each customer has an *average transaction value* greater than or equal to $5920 to distinguish high vs low CLTV (Figure 19).

Taken with the results of the regularized logistic regression model, we have reached an affirmative conclusion: whenever monetary or transaction-derived regressors are included, supervised models are inclined to "cheat" by relying on variables that are, in practice, perfect proxies of the target. While this naturally will yield respectable accuracy on historical data, these models provide very little strategic insight, nor do they support the firm's goal of understanding drivers of high LTV customers. This motivated our new strategy: leverage the non-monetary features to understand demographic, behavioral, and engagement drivers of high vs low CLTV.

2.6 Immediate Recommendations to Employ

It's very clear that including monetary features as regressors in this predictive modeling task is both uninteresting and uninformative. Proxies of the response variable (CLTV) should *not* be used as predictors in a supervised model. Could likely just keep a random selection of any three of these variables and produce a model that is "perfectly" able to distinguish a customer as high or low LTV. Greater investment needs to be immediately put towards obtaining higher-quality data attributes to define customers, regardless of the time they have been with the firm. As is, using any of these monetary features as regressors induces a model that is not optimized to generalize to truly new onboarding customers. Need to leverage higher-quality customer data that provides insightful and descriptive attributes for all customers. Monetary features need to be removed as regressors for any supervised modeling purposes across all business departments immediately.

2.7 Redefining "High CLTV" Definition

To increase the precision of our definition of a "High LTV" customer, we redefined "High LTV" customers as the top 25th percentile of LTV.

```
# 75th percentile (top 25%) of LTV
top_25 <- quantile(data$ltv, probs = 0.75, na.rm = TRUE)

# Binary label: 1 = in top 25% of LTV, 0 = otherwise
ltv_high <- ifelse(data$ltv >= top_25, 1, 0)
```

This redistributed the class proportions to 76% of customers belonging to the Low CLTV class, and 24% of them belonging to the High CLTV class.

2.8 Supervised Learning Using No Monetary Features

With a new and more precise definition of high CLTV in hand, we created a new dataset that removed all monetary and transaction-derived variables as candidate regressors from both the training and holdout samples of our data, giving one final shot at inducing practical and deployable supervised models using strictly non-monetary features as classifiers for high and low CLTV and stratified cross-validation to ensure class proportions were maintained across training and validation splits.

2.8.1 Parameters & Performance of CV Regularized Logistic Regression ROC-AUC-Optimized Model

REGULARIZED LOGISTIC REGRESSION (no monetary features)

A data.frame: 1 × 4

	log__C	log__l1_ratio	log__penalty	log__solver
	<dbl>	<lgl>	<chr>	<chr>
1	0.001	NA	l2	glmnet

Figure 20 - Parameters Selected of Best Model After Performing Grid Search Stratified CV (scoring criterion set to ROC-AUC)

Feature	Coefficient
<chr>	<chr>
intercept	1.09438395
support_tickets_raised	0.00000000
referral_count	0.00000000
last_transaction_days_ago	0.00000000
customer_satisfaction_score	-0.00000000
app_usage_frequency_Monthly	0.00000000
income_level_Middle	-0.00000000

Figure 21 - Regressors in Model with largest coefficients (top 5 most influential)

As shown in Figure 21, without the signal provided by the monetary features, every feature's coefficient, aside from the intercept, has been effectively driven to zero. This is a direct consequence of the induced regularization logistic regression procedure: when the available predictors contain only weak or inconsistent signals for distinguishing the top quartile of LTV customers from the remaining population, the penalty term overwhelms the small influence of the regressors, inducing the model to suppress the coefficients almost entirely. The resulting effect is an intercept-only classifier, where predictions for all customers are determined almost entirely by the baseline log-odds of the intercept, rather than by any measurable behavioral or demographic influence from the candidate regressors. Thus, again, this model highlights the important empirical reality – within this data, non-monetary information alone provides an insubstantial signal for identifying the highest-value customers.

```
          Reference
Prediction HighLTV LowLTV
   HighLTV       0      0
   LowLTV      170    530

Accuracy: 0.7571
Sensitivity (Recall): 0
Specificity: 1
Precision: NA
F1: NA
```

Figure 22 - Resulting Confusion Matrix on Holdout Data

Without signal from the regressors, the logistic regression defaults to predicting the majority class (Low LTV) for all customers (Figure 22). This results in a misleading accuracy of ~75%, which reflects the underlying class distribution rather than predictive capabilities, a recall of 0%, meaning the model failed to identify any high LTV clientele; specificity of 100% because all low LTV customers in the holdout sample are correctly identified; and a precision of NA, since the model never predicted high LTV. Thus, the model is impractical and cannot support any realistic downstream data-driven decision-making processes.

2.8.2 Bagging Ensemble (No Monetary Features)

```r
# 10-fold stratified CV with ROC as scoring
ctrl_bag <- trainControl(
  method          = "cv",
  number          = 10,
  classProbs      = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)


# caret's treebag tuning param (number of base estimators)
bag_grid <- expand.grid(
  nbagg = c(10, 50, 100)
)
```

```r
set.seed(0)

bag_model <- caret::train(
  x         = X_train_without,   # no monetary features
  y         = y_train_bag,
  method    = "treebag",         # bagged decision trees
  trControl = ctrl_bag,
  metric    = "ROC"              # optimize ROC AUC
)
```

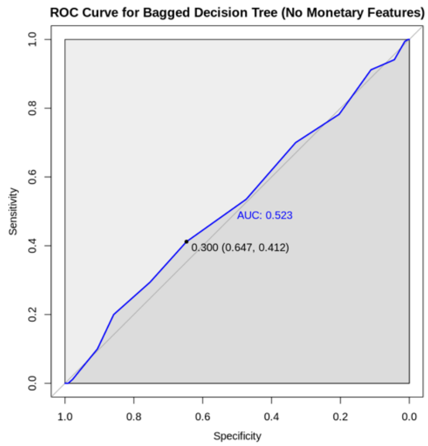
Figure 23 - Model Inducement

```
          Reference
Prediction HighLTV LowLTV
   HighLTV       2     12
   LowLTV      168    518

Accuracy: 0.7429
Sensitivity (Recall): 0.0118
Specificity: 0.9774
Precision: 0.1429
F1: 0.0217
```

Figure 24 - Performance on Holdout Sample (Positive Class Set to High LTV)

A bagged ensemble of decision trees was trained to assess whether variance reduction could perhaps improve classification accuracy. While the ensemble achieved a deceptively high overall accuracy of 74.29%, aggregated across all base estimators, closer inspection of the confusion matrix shows that it predicted the class of interest only fourteen times. This “increased predictive accuracy” therefore reflects exploitation of the imbalanced class proportions rather than true predictive learning.



The ROC curve for the bagging classifier verifies our concerns – the model, at best, is equivalent to random guessing. The AUC value of 0.523 indicates that the model cannot distinguish High/Low LTV customers any better than a coin-flipping machine. As initially declared, this stands in sharp contrast to the model's deceptively strong accuracy, which was driven almost entirely by its tendency to predict the majority class (Low CLTV) for nearly all customer instances. Moreover, because the ROC curve has a roughly diagonal shape, we can conclude that the bagging ensemble provides no actionable discriminatory power across any and all thresholds and therefore can conclude affirmatively that, without the signal provided by transactional proxies of the dependent variable, these models are useless for deriving actionable insight into customer LTV.

3 Expanded Analysis: Modeling Customer Behaviors and Experience to Derive Value

While both the regularized logistic regression and decision tree algorithms ultimately collapsed into majority-class predictions, due to minimal signal derived from the features when used to predict High vs Low CLTV, it is premature to declare that these variables lack business relevance. Instead, we deferred to examine how these same features may influence operational outcomes that shape CLTV over time. Thus, this section shifts the analytical focus: rather than attempting to predict High LTV directly, we've chosen to examine how these non-monetary factors can be leveraged to predict three foundational drivers of CLTV – support ticket volume, customer satisfaction, and issue resolution time.

3.1 Predicting Support Tickets Raised (Using Decision Tree: Non-Monetary Features)

This modeling endeavor was motivated by a desire to understand which features correlate with more frequent support tickets raised by customers – a key indicator of customer frustration that can be used to flag potential low CLTV risk.

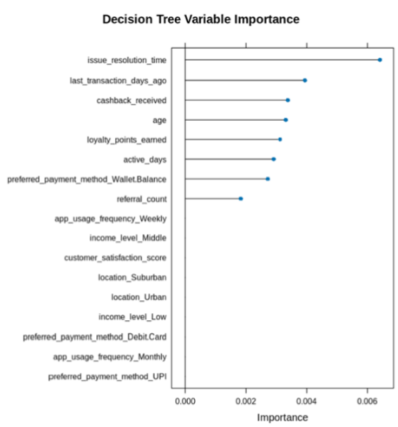


Figure 25 - Key Influential Variables

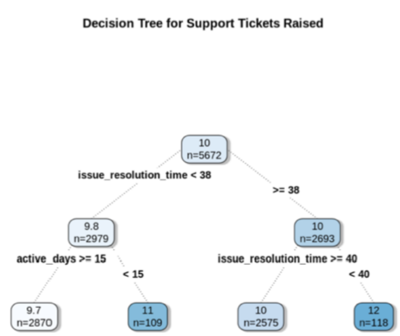


Figure 26 - Best Induced Tree Structure

Figures 25 and 26 cumulatively demonstrate the influence of a key driver: when Issue Resolution Time is < 38 days, Support Tickets Raised decrease marginally on average. Longer resolution times seem to push customers into nodes with higher ticket volumes, indicating escalating customer dissatisfaction and perhaps repeated unresolved issues.

3.2 Predicting Customer Satisfaction (Using Decision Tree: Non-Monetary Features)

This second modeling endeavor was motivated by a desire to understand which features correlate with a higher (or lower) Customer Satisfaction Score – a key indicator of customer engagement, churn risk, and LTV.

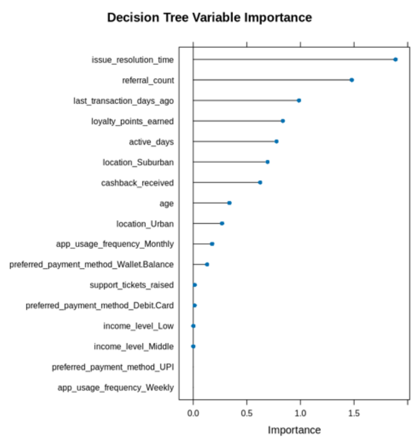


Figure 26 – Key Influential Variables

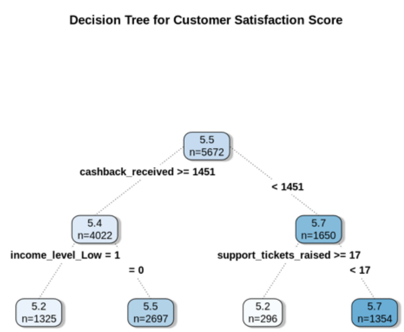


Figure 27 - Best Induced Tree Structure (capped at depth of 2 for interpretability)

As Figures 26 and 27 demonstrate, the decision tree classifier provides rich insight and interpretability: again, we see Issue Resolution to be the key influencer in determining Customer Satisfaction; moreover, the tree structure reveals Support Tickets Raised also influence Customer Satisfaction, particularly if more than 16 tickets are raised, there is a noticeable drop in mean Customer Satisfaction Score and, even with less cashback received than the split at the root node, having less than 17 tickets raised induces the highest mean average Customer Satisfaction Score.

3.3 Predicting Issue Resolution Time

Because issue resolution time emerged as a first-order driver of both support tickets and customer satisfaction, we modeled it as a dependent variable to assess what influences it in both a regularized linear regression model and a decision tree classifier.

	Coefficient
support_tickets_raised	0.084960066
preferred_payment_method_UPI	-0.068711295
cashback_received	-0.061643871
app_usage_frequency_Monthly	0.056947008
referral_count	-0.051915493
app_usage_frequency_Weekly	0.047281467
loyalty_points_earned	-0.036870483
income_level_Low	-0.035621101
last_transaction_days_ago	0.024767267
location_Suburban	-0.021939838
customer_satisfaction_score	0.019000320
age	-0.018699055
location_Urban	0.015107390
preferred_payment_method_Wallet.Balance	-0.014379929
income_level_Middle	-0.008908482
active_days	0.007846215
preferred_payment_method_Debit.Card	0.005104719

Figure 28 - Coefficients of Regularized Linear Regression Model

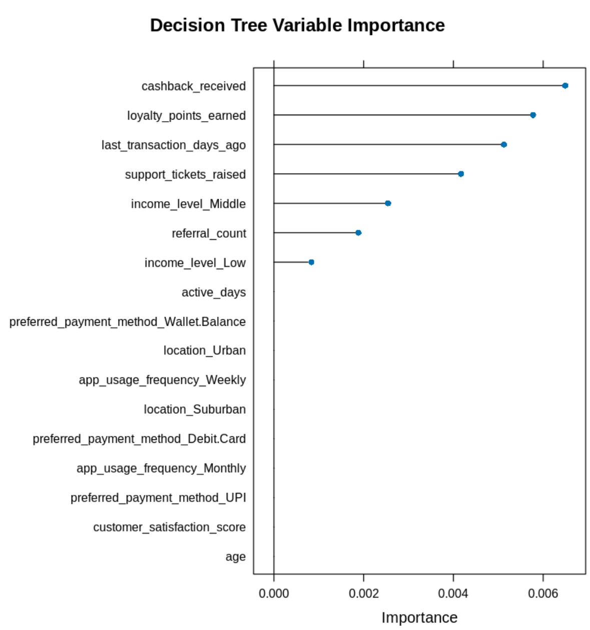


Figure 29 – Key Influential Variables Extracted from Decision Tree Classifier

Analysis of the coefficients and their importance supports our ongoing declarations – there is a major correlation between Support Tickets Raised and Issue Resolution Time. Though the effect remains small due to regularization in the linear model, the coefficients indicate intuitive directional effects: customers raising more tickets tend to experience longer resolution times (Figure 28). Moreover, Figure 29 shows that Support Tickets Raised again appears as a key determinant of undesired customer resolution times.

Our hypotheses: higher ticket volumes → longer resolution times → lower customer satisfaction → more tickets → lower CLTV

4 Cross-Model Chain of Events and Node Analysis

Collectively, the three modeling tasks have produced insight into dissecting a robust chain of events: *1) Issue Resolution Time →Support Tickets Raised*: Faster resolutions reduce the frequency of issues. *2) Issue Resolution Time →Customer Satisfaction*: Decision Tree Feature Importance analysis supports that longer resolution times negatively impact customer satisfaction. *3) Support Tickets Raised →Customer Satisfaction*: Frequency of tickets indicates diminished customer satisfaction scores.

Node	Avg_Total_Spent	Avg_Avg_Transaction_Value	Avg_LTV	Count
<int>	<dbl>	<dbl>	<dbl>	<int>
3	4990092.	9827.	510630.	2870
4	5442271.	10284.	556812.	109
6	4977441.	9951.	509538.	2575
7	5328203.	10033.	544445.	118

Figure 30 - Using Indexes of Customers in Each Node from Induced Decision Tree (Reference --> Figure 26)

A deeper analysis of customer profiles aggregated in the leaf nodes of the Decision Tree induced in Section 5.1 (Figure 26) reveals a major insight: customers with lower resolution times and fewer days active, on average, have the highest LTV, total spend, and average transaction value. Notably, customers with lower resolution times and *more* active days reveal the lowest average monetary metrics. This pattern suggests that even when issues are resolved relatively quickly, extended active engagement with the platform correlates with diminishing economic value, perhaps due to ongoing unresolved problems.

5 Business Recommendations and Strategic Insight

Synthesizing insights across our entire modeling pipeline reveals a unified conclusion: across all models, Issue Resolution Time, is the central operational variable driving ticket load, customer satisfaction and engagement, transaction behaviors, and CLTV. Reducing resolution times below key thresholds (~33-40 days) should be an enterprise-level objective.

5.1 Support Ticket Reduction Strategies

Because high ticket volume reinforces slower resolution and lower customer satisfaction, we recommend the firm devote resources to improving first-contact resolution, identifying high-issue categories, and targeting improvements in those categories to induce the greatest marginal benefits. With per-customer LTV uplift between $33K and $47K (discussed in the next section), management can justify technology upgrades, staffing increases, and enhanced customer experience training initiatives to induce these operational upgrades – the financial upside is large enough that even moderate improvements will yield substantial returns.

6 Quantifying Financial Impact of Operational Improvements

By assessing the node-level LTV differences (Figure 30), we can effectively quantify the financial upside from improving operational metrics that naturally move customers from lower-LTV nodes (3 and 6) into higher-LTV nodes (4 and 7). For example:

- **Migration of customers from Node 3 to Node 4:**
    - ***How?*** Correct ongoing issues so that more app engagement and activity == higher economic value of customer. This is a platform-dependent issue that an

investment in enhanced UX and full-stack training can resolve with the appropriate capital resources.

  - ***Return:*** Average LTV increase per customer: \$556,812 - \$510,630 = **\$46,182.**

- **Migration of customers from Node 6 to Node 4:**
  - ***How?*** Implement strategies outlined in section 7.1 – reducing ticket load by increasing staffing levels and training is a leading driver in bringing down issue resolution times.
  - ***Return:*** Average LTV increase per customer: \$556,812 - \$509,538 = **\$47,274**

- **Migration of customers from Node 3 to Node 7:**
  - ***How?*** Implement strategic recommendations outlined in section 7.1 and migration plans for the previously mentioned nodes.
  - ***Return:*** Average LTV increase per customer: \$544,445 - \$510,630 = **\$33,815**

If operational strategies moved even 1,000 customers currently in Node 3 to Node 4, the potential LTV uplift = ~\$46,182 x 1,000 customers = **~\$46.18M in revenue**

7 Conclusion

The business implication is clear → improving support operational processes is the soundest path to improving customer value, given the current structure of the data on hand.