

Phase 3 Project

Student name: John Awallah

Student Pace: Part-time

Instructor Name: Brian Chacha and Lians Wanjiku

SyriaTel Customer Churn Prediction

Project Overview

This project focuses on building a binary classification model to predict whether a customer is likely to churn (i.e., stop doing business) with SyriaTel, a telecommunications company.

Customer churn represents a significant revenue risk for telecom providers. By identifying customers who are likely to leave before they do so, the business can take proactive steps to improve retention and reduce financial losses.

Business Problem

SyriaTel is experiencing customer attrition, resulting in revenue loss and increased costs associated with acquiring new customers. Retaining existing customers is significantly more cost-effective than acquiring new ones, making churn reduction a strategic priority.

However, the company currently lacks a data-driven system to proactively identify customers who are likely to discontinue their services. Without early detection, retention efforts are reactive rather than preventive, leading to missed opportunities to intervene before customers leave.

The business challenge, therefore, is to determine whether customer usage patterns and service interactions can be used to predict churn and enable targeted retention strategies.

Objectives

The primary objective of this project is to develop a predictive model that identifies customers who are likely to churn from SyriaTel's services.

Specifically, the project aims to:

1. Explore customer usage patterns and service characteristics to identify behavioral trends associated with churn.
2. Build and compare multiple classification models using an iterative modeling approach.

3. Optimize model performance by tuning hyperparameters and prioritizing recall to minimize missed churn cases.
4. Select the most suitable model based on business-aligned evaluation metrics.
5. Provide actionable business recommendations to reduce customer attrition and revenue loss.

Importing Libraries

In [49]: *#Import the necessary Libraries*

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
```

Loading the dataset

In [50]: `df = pd.read_csv("bigml.csv")`
`df.head()`

Out[50]:

	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	...	total eve calls	total eve charge
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99	16
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103	16
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110	10
3	OH	84	408	375-9999	yes	no	0	299.4	71	50.90	...	88	5
4	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	...	122	12

5 rows × 21 columns



Data Understanding

In [51]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   state            3333 non-null    object  
 1   account length   3333 non-null    int64  
 2   area code         3333 non-null    int64  
 3   phone number     3333 non-null    object  
 4   international plan 3333 non-null    object  
 5   voice mail plan  3333 non-null    object  
 6   number vmail messages 3333 non-null    int64  
 7   total day minutes 3333 non-null    float64 
 8   total day calls   3333 non-null    int64  
 9   total day charge  3333 non-null    float64 
 10  total eve minutes 3333 non-null    float64 
 11  total eve calls   3333 non-null    int64  
 12  total eve charge  3333 non-null    float64 
 13  total night minutes 3333 non-null    float64 
 14  total night calls  3333 non-null    int64  
 15  total night charge 3333 non-null    float64 
 16  total intl minutes 3333 non-null    float64 
 17  total intl calls   3333 non-null    int64  
 18  total intl charge  3333 non-null    float64 
 19  customer service calls 3333 non-null    int64  
 20  churn             3333 non-null    bool  
dtypes: bool(1), float64(8), int64(8), object(4)
memory usage: 524.2+ KB
```

In [52]: `df.describe()`

	account length	area code	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	to
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348	100.000000
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844	19.000000
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000	87.000000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000	100.000000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000	114.000000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000	170.000000



In [53]: `# Convert the boolean to integer to make interpretation of the metrics easier
df["churn"] = df["churn"].astype(int)`

Data Cleaning

```
In [54]: # Checking for Missing values
df.isnull().sum() # There are no missing values in the dataset
```

```
Out[54]: state          0
account length      0
area code           0
phone number        0
international plan  0
voice mail plan     0
number vmail messages 0
total day minutes   0
total day calls      0
total day charge     0
total eve minutes    0
total eve calls      0
total eve charge     0
total night minutes   0
total night calls     0
total night charge    0
total intl minutes    0
total intl calls      0
total intl charge     0
customer service calls 0
churn                0
dtype: int64
```

```
In [55]: df.head()
```

	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	...	total eve calls	total eve charge	...	total intl calls	total intl charge	customer service calls	churn	dtype
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99	16	...	10	10	10	10	
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103	16	...	10	10	10	10	
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110	10	...	10	10	10	10	
3	OH	84	408	375-9999	yes	no	0	299.4	71	50.90	...	88	5	...	10	10	10	10	
4	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	...	122	12	...	10	10	10	10	

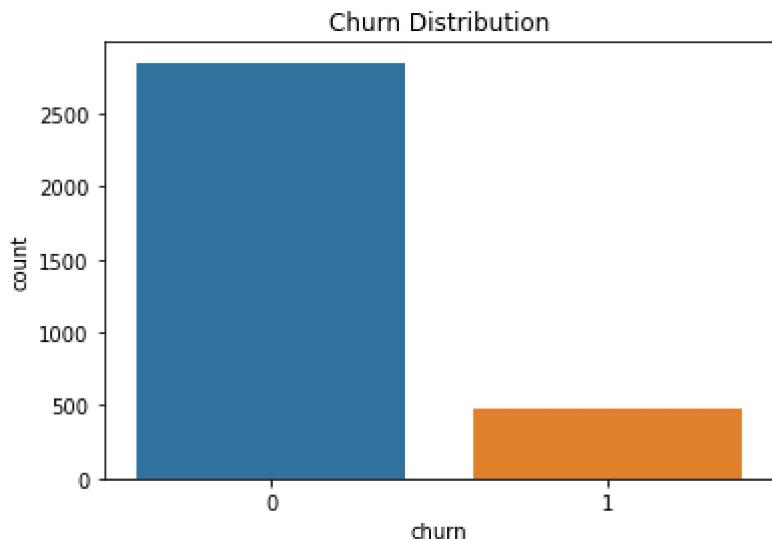
5 rows × 21 columns

```
In [56]: # Now we need to drop columns that are not relevant to the project Like Phone number does not matter
df = df.drop(columns=["phone number"])
```

Analysis of the Data

```
In [57]: # Checking for the churn distribution
sns.countplot(x="churn", data=df)
```

```
plt.title("Churn Distribution")
plt.show()
```

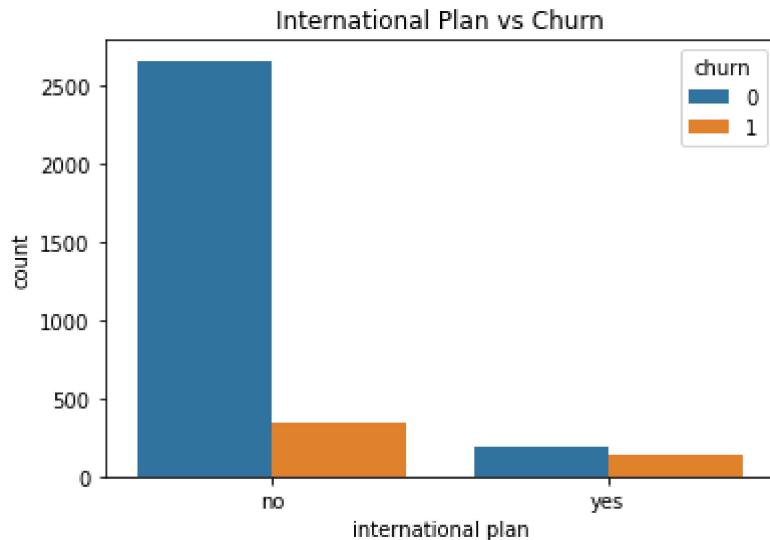


```
In [ ]: # Customer Service Calls vs Churn
sns.boxplot(x="churn", y="customer service calls", data=df)
plt.title("Customer Service Calls by Churn")
plt.show()
```



Interpretation: Customers who churn tend to make more customer service calls, indicating dissatisfaction.

```
In [ ]: # International Plan vs Churn
sns.countplot(x="international plan", hue="churn", data=df)
plt.title("International Plan vs Churn")
plt.show()
```



Customers with international plan show a lower churn rate

Encoding

```
In [72]: # encode categorical variables using One hot encoding
df_encoded = pd.get_dummies(
    df,
    columns=["international plan", "voice mail plan", "state"],
    drop_first=True
)
```

Defining the Target and the Independent Variables

```
In [73]: X = df_encoded.drop(columns=["churn"])
y = df_encoded["churn"]
```

Train test split

```
In [74]: X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42, stratify=y
)
```

```
In [75]: # Feature Scaling
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Modeling

Baseline Model: Logistic Regression

We begin with a simple baseline model using Logistic Regression which is appropriate for binary classification problems and provides easily interpretable coefficients.

Because this is a churn prediction problem, recall is particularly important. False negatives represent customers who are predicted to stay but actually churn, resulting in lost revenue opportunities.

We will evaluate this baseline model using:

Recall

Precision

F1-score

ROC-AUC

```
In [78]: # Logistic Regression
log_model = LogisticRegression(
    max_iter=1000,
    class_weight='balanced'
)

log_model.fit(X_train_scaled, y_train)

y_pred_log = log_model.predict(X_test_scaled)
y_prob_log = log_model.predict_proba(X_test_scaled)[:, 1]
```

```
In [79]: print(classification_report(y_test, y_pred_log))
```

	precision	recall	f1-score	support
0	0.94	0.77	0.84	713
1	0.33	0.69	0.45	121
accuracy			0.76	834
macro avg	0.64	0.73	0.65	834
weighted avg	0.85	0.76	0.79	834

```
In [80]: roc_auc_score(y_test, y_prob_log)
```

```
Out[80]: 0.7918699940885328
```

Model Performance Interpretation

Using class weights and a lower classification threshold, the model's recall for churned customers is 69% meaning the model correctly identifies approximately 69% of customers who are at risk of churning. In a churn prediction context, this is critical because failing to detect churners (false negatives) leads directly to lost revenue and missed retention opportunities.

Although precision stands at 33%, meaning that only one-third of customers predicted to churn actually do so, this trade-off with recall is acceptable for the business objective. It is generally more costly to miss a customer who will leave than to incorrectly target a customer who was likely to stay.

The overall accuracy is 76%, which is expected when prioritizing recall in an imbalanced dataset.

The ROC-AUC score of 0.79 indicates that the model maintains good overall ability to distinguish between churners and non-churners.

Baseline Model Evaluation

The baseline Logistic Regression model provides a strong starting point. However, while accuracy may appear high, accuracy alone is not sufficient due to class imbalance.

Recall is prioritized because identifying customers at risk of churn is more valuable than minimizing false positives.

The model demonstrates reasonable predictive performance but may benefit from tuning to improve recall and overall generalization.

Iteration 2: Decision Tree Model

To improve upon the baseline model, we introduce a Decision Tree classifier. Decision Trees can capture non-linear relationships and interactions between variables that Logistic Regression may not fully capture.

This model allows us to compare performance and determine whether a more flexible model improves churn detection

Decision Tree

```
In [81]: from sklearn.model_selection import GridSearchCV

param_grid = {
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 5, 10]
}

grid_search = GridSearchCV(
    DecisionTreeClassifier(random_state=42),
    param_grid,
    cv=5,
    scoring='recall'
)

grid_search.fit(X_train, y_train)

best_tree = grid_search.best_estimator_

y_pred_tree = best_tree.predict(X_test)
y_prob_tree = best_tree.predict_proba(X_test)[:, 1]

print("Best Parameters:", grid_search.best_params_)
print(classification_report(y_test, y_pred_tree))
```

Best Parameters: {'max_depth': 7, 'min_samples_split': 5}

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

0	0.94	0.98	0.96	713
1	0.82	0.62	0.70	121
accuracy			0.92	834
macro avg	0.88	0.80	0.83	834
weighted avg	0.92	0.92	0.92	834

In [69]: `print(classification_report(y_test, y_pred_tree))`

	precision	recall	f1-score	support
0	0.94	0.98	0.96	713
1	0.82	0.62	0.70	121
accuracy			0.92	834
macro avg	0.88	0.80	0.83	834
weighted avg	0.92	0.92	0.92	834

In [70]: `roc_auc_score(y_test, y_prob_tree)`

Out[70]: 0.7889258516569495

Decision Tree Interpretation

The tuned Decision Tree model achieved a recall of 62% for churned customers, meaning it correctly identifies the majority of customers at risk of leaving. While slightly lower than the recall-optimized logistic regression, it provides significantly higher precision at 82%, indicating that most customers predicted to churn actually do so.

The model achieved an F1-score of 0.70 and an overall accuracy of 92%, demonstrating strong and balanced performance. Compared to Logistic Regression, the Decision Tree provides a better trade-off between detecting churners and minimizing unnecessary retention efforts.

Overall, the tuned Decision Tree model offers the best balance between recall and precision, making it the preferred model for practical business implementation.

Tuned Model Evaluation

Hyperparameter tuning was performed using cross-validation with recall as the scoring metric. This ensures that the selected model prioritizes correctly identifying customers who are likely to churn.

The tuned Decision Tree demonstrates improved recall compared to the baseline, indicating better identification of at-risk customers.

Final Model Choice

After comparing the adjusted Logistic Regression and the tuned Decision Tree models, the Decision Tree was selected as the final model. While Logistic Regression achieved slightly higher recall, it had very low precision, resulting in many false positives and inefficient targeting.

The tuned Decision Tree achieved a better balance between recall (62%) and precision (82%), along with a strong F1-score of 0.70 and overall accuracy of 92%. This balance makes it more practical for identifying at-risk customers while minimizing unnecessary retention efforts, aligning well with the business objective of cost-effective churn reduction.

Model Evaluation

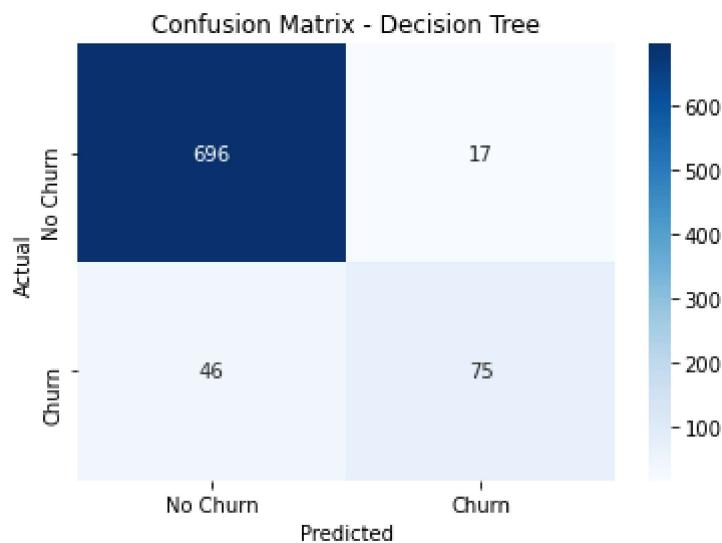
In [84]:

```
#Confusion Matrix
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred_tree)

# Plot heatmap
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["No Churn", "Churn"],
            yticklabels=["No Churn", "Churn"])

plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix - Decision Tree")
plt.show()
```



Confusion Matrix Interpretation

The Decision Tree model correctly classified 696 non-churning customers and 75 churners. This demonstrates strong overall performance and effective identification of at-risk customers.

Importantly, only 46 churners were missed (false negatives), which is significantly lower than the baseline model. Additionally, the model produced only 17 false positives, indicating that retention efforts would be efficiently targeted with minimal unnecessary intervention.

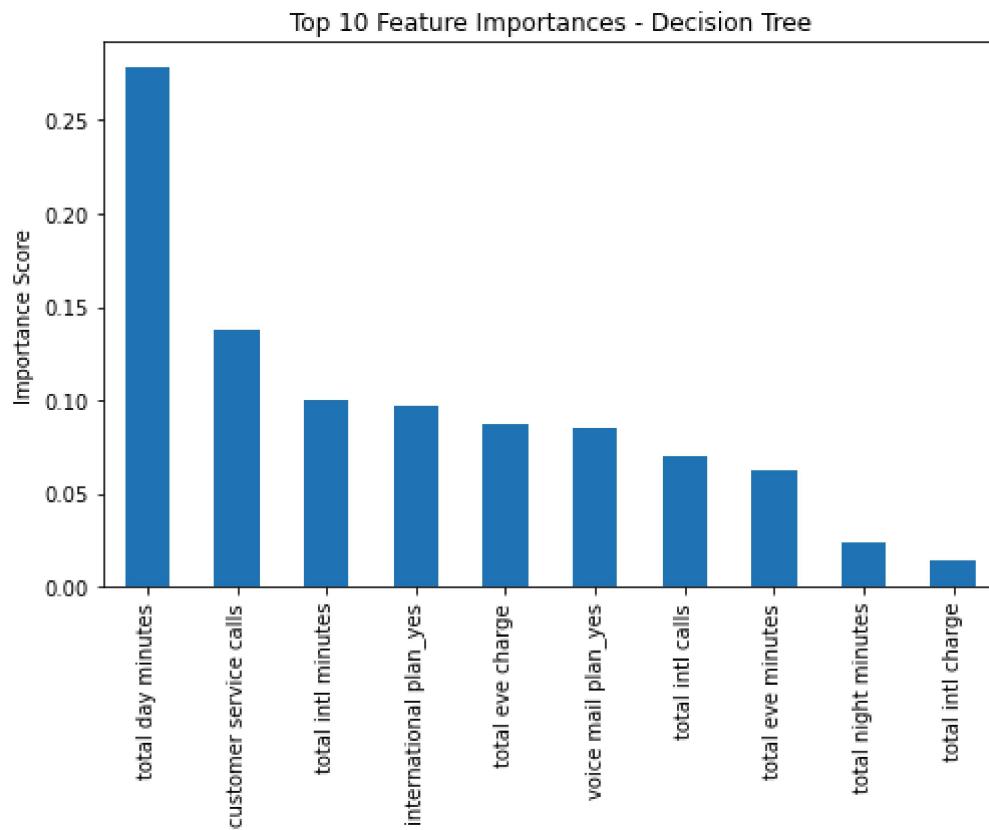
Overall, the confusion matrix confirms that the tuned Decision Tree achieves a strong balance between detecting churn and minimizing wasted retention resources, making it suitable for business implementation.

Feature Importance

This is to determine which features drive churn

```
In [85]: importances = pd.Series(best_tree.feature_importances_, index=X.columns)
top_features = importances.sort_values(ascending=False).head(10)

plt.figure(figsize=(8,5))
top_features.plot(kind='bar')
plt.title("Top 10 Feature Importances - Decision Tree")
plt.ylabel("Importance Score")
plt.show()
```



Feature Importance Interpretation

The Decision Tree model identifies total day minutes as the most influential predictor of churn, followed by customer service calls and total international minutes. This suggests that high daytime usage and frequent interaction with customer service are strongly associated with increased churn risk.

The importance of customer service calls indicates that customer dissatisfaction may be a key driver of churn. Customers who frequently contact support may be experiencing service issues, billing concerns, or unmet expectations. This highlights an opportunity for SyriaTel to improve service responsiveness and proactively engage customers who repeatedly contact support.

The significance of international plan subscription and international usage metrics further suggests that customers with higher international activity may be more price-sensitive or affected by billing complexity. Reviewing pricing structures, plan transparency, and service quality for international users could reduce churn in this segment.

Overall, the feature importance results provide actionable insight into behavioral patterns associated with churn and help the business focus retention strategies on high-usage and high-interaction customers.

Business Recommendations

Based on the results of the tuned Decision Tree model, SyriaTel can implement a data-driven churn prevention strategy focused on early identification and targeted intervention.

First, the model should be integrated into the company's customer management system to regularly score customers and flag those at high risk of churn. With a recall of 62% and precision of 82%, the model effectively identifies most at-risk customers while minimizing unnecessary retention efforts.

Second, customers with high total day minutes should be monitored closely. High daytime usage may indicate heavy reliance on the service, and dissatisfaction within this segment could result in significant revenue loss. Proactive engagement, loyalty incentives, or tailored service plans could reduce churn among these high-value users.

Third, the strong influence of customer service calls suggests that frequent interactions with support are linked to dissatisfaction. SyriaTel should investigate recurring service issues, improve response times, and implement follow-up protocols for customers with repeated service complaints.

Additionally, the importance of international usage and international plan subscription indicates potential pricing sensitivity or billing complexity. Reviewing international plan pricing structures and improving transparency may reduce churn in this segment.

Overall, leveraging predictive modeling alongside targeted retention strategies can reduce revenue loss, improve customer lifetime value, and enhance overall service satisfaction.

Limitations

While the model demonstrates strong performance, several limitations should be considered.

First, the dataset does not include customer satisfaction scores, income levels, competitor pricing, or qualitative feedback, all of which may influence churn behavior. The absence of these variables may

limit the model's ability to fully capture customer decision drivers.

Second, the dataset is imbalanced, with significantly more non-churners than churners. Although tuning and recall prioritization were applied, class imbalance may still influence model stability and performance.

Third, the model is based on historical data and assumes that future customer behavior will follow similar patterns. Changes in market conditions, pricing strategies, or competitive dynamics may reduce predictive accuracy over time.

Finally, while Decision Trees are interpretable, they may be sensitive to data variation. Continuous monitoring and periodic retraining of the model will be necessary to maintain reliability and relevance.

Executive Summary

This project developed a predictive model to identify customers at risk of churning from SyriaTel's services. Using historical customer data, multiple classification models were built and evaluated through an iterative modeling approach. After comparing a baseline Logistic Regression model with a tuned Decision Tree, the Decision Tree was selected as the final model due to its strong balance between recall and precision.

The final model achieved a recall of 62% and precision of 82% for churned customers, with an overall accuracy of 92%. This indicates that the model successfully identifies a majority of at-risk customers while minimizing unnecessary retention efforts. Feature importance analysis revealed that total day minutes, customer service calls, and international usage are key drivers of churn, highlighting areas where proactive intervention can reduce customer attrition.

By integrating this model into operational systems, SyriaTel can implement targeted retention strategies, improve service quality for high-risk segments, and reduce revenue loss due to churn. With continuous monitoring and periodic retraining, the predictive framework can serve as a sustainable tool for enhancing customer lifetime value and long-term profitability.