FinTech Customer Life Time Value Analysis

Bernadetta Dewi| Rukshana Fathima |Sujeevana Lekkala

Team Members

Bernadetta Dewi
Data Scientist

Sujeevana Lekkala
Data Scientist

Rukshana Fathima
Syed Mohamed
Data Scientist

CRISP DM – Phases, Tasks and Milestones

Project Timeline

Business Understanding	**Data Understanding**	**Data Preparation**	**Modeling**	**Evaluation**	**Deployment/Final**

04/24 – 05/02/2025	**05/03- 05/09/2025**	**05/10 -05/20/2025**	**05/21 – 06/17/2025**	**06/18 – 06/30/2025**	**07/01- 07/17/2025**
▪ Determine Business Objectives ▪ Assess Situation ▪ Determine Data Mining Goals ▪ Produce Project Plan	▪ Collect Data ▪ Describe Data ▪ Explore the Data ▪ Verify Data Quality Data Description	▪ Load the Data ▪ Clean Data ▪ Integrate the Data ▪ Format the Data	▪ Choose the right modeling methods. ▪ Prepare Milestone Presentation ▪ Generate Test Design ▪ Create the predictive or analytical model.	▪ Evaluate the model's performance. ▪ Evaluate the results ▪ Review the Process ▪ Prepare Final Presentation	▪ Deployment ▪ Monitoring and Maintenance ▪ Final Documentation (Pdf) ▪ Review Project

06/05/2025	**06/26/2025**	**07/17/2025**
Milestone Presentation	Final Presentation	Final Documentation (Pdf) Submission

Business Understanding

Business Objective

A FinTech company provides digital financial services to a large customer base. The company's primary business objective is to maximize long-term profitability by focusing on customer retention, targeted marketing, and resource optimization.

To achieve this, the company aims to:

- **Identify high-value customers based on their Customer Lifetime Value (CLTV).**

  Approach: Build a regression model to predict each customer's future revenue.

  Example use case: Predict that Customer A will generate significantly higher revenue over time → prioritize for retention initiatives.

- **Improve personalization of financial products and services**

  Approach: We applied clustering on key customer features, including total spend and lifetime value (LTV), to uncover natural customer segments.

  Example use case: Clustering uncovered three distinct customer groups with varying behaviors, ranging from moderate spenders suitable for upselling, to premium but infrequent buyers ideal for loyalty strategies, and highly active, high-value users requiring focused retention efforts.

- **Optimize marketing strategies to increase ROI and reduce churn.**

  Approach: Leverage CLTV estimates and customer clusters to focus marketing efforts on high-value segments and personalize campaign content.

  Outcome: Marketing strategies can be optimized by targeting customers who offer the highest long-term value, adjusting messaging based on cluster behavior, and reducing churn through more relevant, data-driven engagement.

Situation Assessment

- **Inventory of Resources:**
  Access to historical customer data including transaction frequency, revenue, tenure, and engagement metrics, availability of internal data science team and computing infrastructure.

- **Requirements:**
  Business stakeholders require an interpretable and actionable CLTV model to segment customers and support strategic decisions in marketing and customer service.

- **Assumptions and Constraints:**
  Assumes data is accurate and reflects real customer behavior. Constraints include potential data privacy policies and limited customer demographic information.

- **Risks and Contingencies:**
  Risks include biased data, overfitting of the predictive model, and inaccurate CLTV estimates that may lead to poor marketing investments. To mitigate this, the team will validate models on unseen data and monitor performance over time.

Situation Assessment , Cont.

- **Terminology:**
  - CLTV: Predicted net profit attributed to the entire future relationship with a customer.
  - Segmentation: Grouping customers based on shared characteristics or predicted behavior.
  - Churn: When a customer stops using the service.
- **Costs and Benefits:**
  - Costs: Project time, data cleaning effort, and infrastructure usage.
  - Benefits: Improved customer targeting, increased revenue through better retention, and lower marketing spend.

Data Mining Goal

The goal of this data science project is to predict Customer Lifetime Value (CLTV) using regression techniques and then perform customer segmentation based on the predicted values.

- Primary Techniques:
  - Regression to estimate individual CLTV.
  - Segmentation (e.g., clustering) to identify customer groups with similar value profiles.
- Use Cases:
  - Enable targeted retention campaigns for high-value customers.
  - Allocate marketing budgets more effectively based on CLTV tiers.

Collect Initial Data

Data and description downloaded from Kaggle

https://www.kaggle.com/datasets/harunrai/fintech-customer-life-time-value-ltv-dataset

The dataset provides financial, demographic, and behavioral data for customers of a FinTech company. The goal is to predict the Customer Lifetime Value (CLTV) using available features. This dataset can be used for regression modeling, customer segmentation, and marketing strategy development.

File:

fintech_ltv_dataset.csv - historical customer data including the target variable LTV

Start Programming by Importing Python Libraries

import pandas as pd

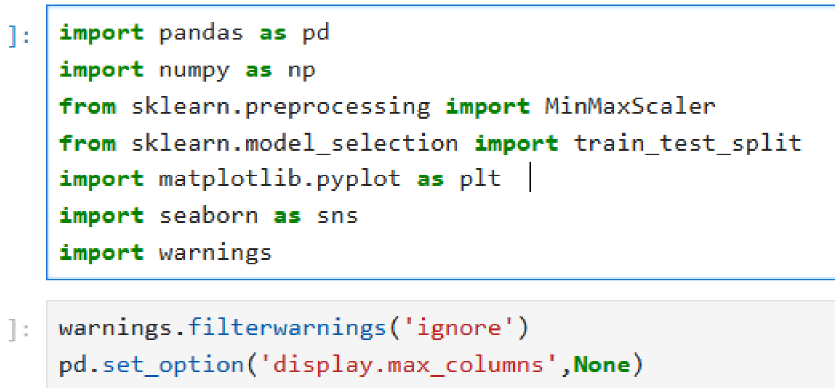
Import the pandas library for data manipulation and analysis

import warnings

Import the warnings module to manage warning messages

warnings.filterwarnings("ignore")

Suppress all warning messages to keep the output clean



```
]: import pandas as pd
   import numpy as np
   from sklearn.preprocessing import MinMaxScaler
   from sklearn.model_selection import train_test_split
   import matplotlib.pyplot as plt
   import seaborn as sns
   import warnings

]: warnings.filterwarnings('ignore')
   pd.set_option('display.max_columns',None)
```

Importing Data to The Notebook

1. Define the path to the dataset file

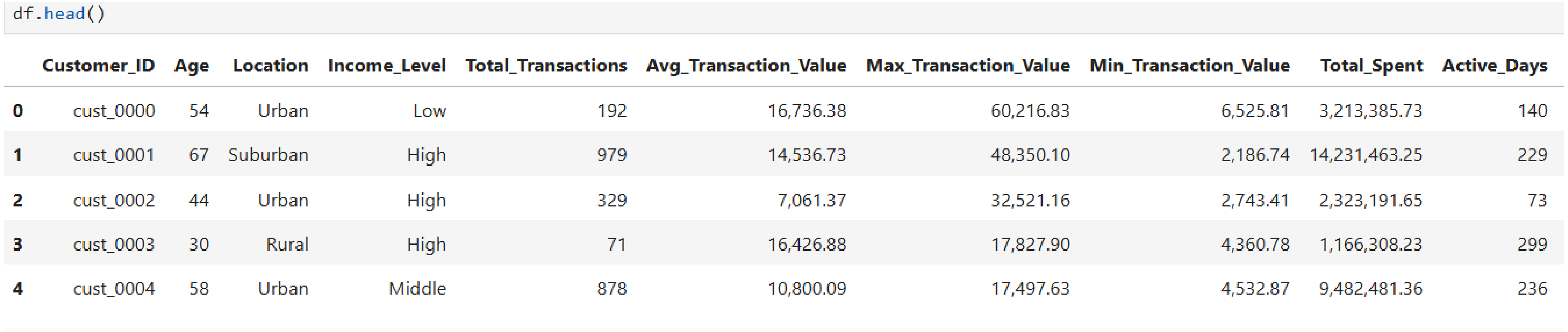
file_path = 'C:\data\digital_wallet_ltv_dataset.csv’

2. Load the dataset into a DataFrame

data = pd.read_csv(file_path, low_memory=False)

3. Display the first 5 rows of the dataset

data.head(5)



```
df.head()
```

	Customer_ID	Age	Location	Income_Level	Total_Transactions	Avg_Transaction_Value	Max_Transaction_Value	Min_Transaction_Value	Total_Spent	Active_Days
0	cust_0000	54	Urban	Low	192	16,736.38	60,216.83	6,525.81	3,213,385.73	140
1	cust_0001	67	Suburban	High	979	14,536.73	48,350.10	2,186.74	14,231,463.25	229
2	cust_0002	44	Urban	High	329	7,061.37	32,521.16	2,743.41	2,323,191.65	73
3	cust_0003	30	Rural	High	71	16,426.88	17,827.90	4,360.78	1,166,308.23	299
4	cust_0004	58	Urban	Middle	878	10,800.09	17,497.63	4,532.87	9,482,481.36	236

Describe Data – Fintech_ltv_dataset

No	Column Name	Data_type	Value Range	Null Values	Description
1	**Customer_ID**	object	['cust_0000' 'cust_0001' 'cust_0002' ... 'cust_6997' 'cust_6998' 'cust_6999']	No	Unique identifier for each customer
2	**Age**	int64	Min = 16, Max = 69	No	No missing values
3	**Location**	Object	['Urban' 'Suburban' 'Rural']	No	Geographical location of the customer
4	**Income_Level**	object	['Low' 'High' 'Middle']	No	Income classification of the customer
5	**Total_Transactions**	int64	Min = 1, Max = 1000	No	Total number of transactions by customers
6	**Avg_Transaction_Value**	float64	Min = 10.19, Max = 19996.45	No	Average value of each transaction in Rupees
7	**Max_Transaction_Value**	float64	Min = 31.86, Max = 98809.24	No	The highest single transaction value recorded in Rupees.

Describe Data – Fintech_ltv_dataset

No	Column Name	Data_Type	Value Range	Null Values	Description
8	**Min_Transaction_Value**	float64	Min = 4.62, Max = 9917.03	No	The lowest single transaction value recorded in Rupees.
9	**Total_Spent**	float64	Min = 1498.14, Max = 19467727.68	No	The total amount spent by the customer in Rupees
10	**Active_Days**	int64	Min = 1, Max = 365	No	Number of days the customer has been active on the platform.
11	**Last_Transaction_Days_Ago**	int64	Min = 1, Max = 365	No	Days since the customer's last transaction.
12	**Loyalty_Points_Earned**	int64	Min = 0, Max = 5000	No	Total loyalty points earned by the customer.
13	**Referral_Count**	int64	Min = 0, Max = 50	No	Number of new customers referred by the user.
14	**Cashback_Received**	float64	Min = 0.23, Max = 4999.7	No	Total cashback received by the customer.

Describe Data – Fintech_ltv_dataset

No	Column Name	Data_Type	Value Range	Null Values	Description
15	**App_Usage_Frequency**	Object	['Monthly' 'Weekly' 'Daily']	No	Frequency of app usage
16	**Preferred_Payment_Method**	object	['Debit Card' 'UPI' 'Wallet Balance' 'Credit Card']	No	The most frequently used payment method by the customer.
17	**Support_Tickets_Raised**	int64	Min = 0, Max = 20	No	Number of support tickets raised by the customer.
18	**Issue_Resolution_Time**	float64	Min = 1.02, Max = 71.98	No	Average time taken to resolve customer issues
19	**Customer_Satisfaction_Score**	int64	Min = 1, Max = 10	No	A score (1-10) reflecting customer satisfaction.
20	**LTV**	float64	3770.5, Max = 1956987.64	No	The target variable representing the estimated Lifetime Value of the customer.

Data Visualization

- ❑ Our interactive dashboard and all subsequent visualizations were developed using Plotly, complemented by Matplotlib and Seaborn for static plots.

1. **Correlation Heatmap** :To quickly identify which pairs of variables are strongly related (positively or negatively), and which are not. Useful for feature selection, understanding multicollinearity, and initial business insights.

2. **Histogram :** To understand the shape, spread, and central tendency of one variable (e.g., "how many customers spend between $0 and $100?").

3. **scatter plot :** To identify patterns, trends, or correlations between two variables (e.g., "does higher customer satisfaction lead to higher LTV?").

4. **Box Plot :** A box plot is a standardized way of displaying the distribution of numerical data.

Correlation Heatmap

- A correlation heatmap is a colored grid that shows the relationship (correlation) between numerical features in a dataset.

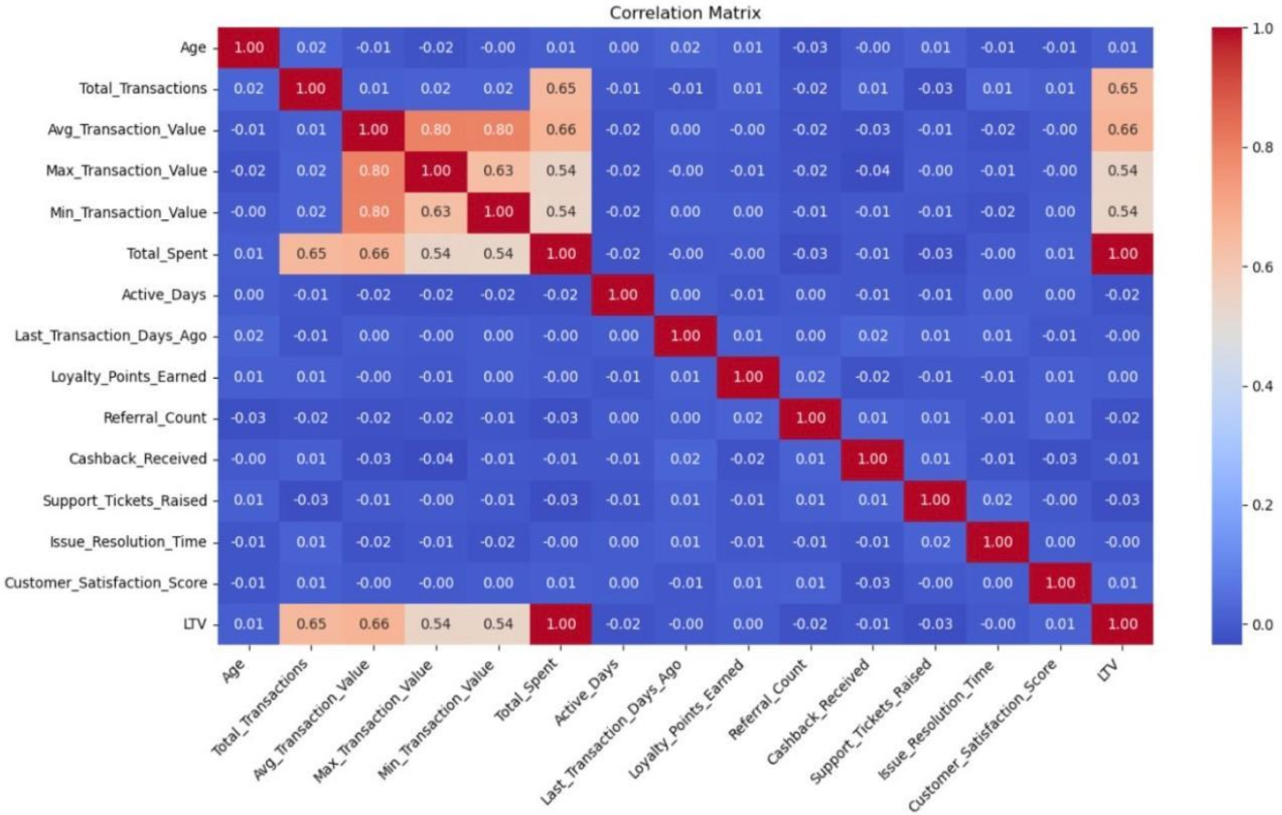
- In the FinTech LTV project, we used a correlation heatmap to **identify which features are most strongly related to Customer Lifetime Value (LTV)**.

- This helps to:

  - Pick **important features** for my prediction model.

  - Avoid **redundant or highly similar features.**

Better understand the **underlying business patterns.**



	Age	Total_Transactions	Avg_Transaction_Value	Max_Transaction_Value	Min_Transaction_Value	Total_Spent	Active_Days	Last_Transaction_Days_Ago	Loyalty_Points_Earned	Referral_Count	Cashback_Received	Support_Tickets_Raised	Issue_Resolution_Time	Customer_Satisfaction_Score	LTV
Age	1.00	0.02	-0.01	-0.02	-0.00	0.01	0.00	0.02	0.01	-0.03	-0.00	0.01	-0.01	-0.01	0.01
Total_Transactions	0.02	1.00	0.01	0.02	0.02	0.65	-0.01	-0.01	0.01	-0.02	0.01	-0.03	0.01	0.01	0.65
Avg_Transaction_Value	-0.01	0.01	1.00	0.80	0.80	0.66	-0.02	0.00	-0.00	-0.02	-0.03	-0.01	-0.02	-0.00	0.66
Max_Transaction_Value	-0.02	0.02	0.80	1.00	0.63	0.54	-0.02	-0.00	-0.01	-0.02	-0.04	-0.00	-0.01	-0.00	0.54
Min_Transaction_Value	-0.00	0.02	0.80	0.63	1.00	0.54	-0.02	0.00	0.00	-0.01	-0.01	-0.01	-0.02	0.00	0.54
Total_Spent	0.01	0.65	0.66	0.54	0.54	1.00	-0.02	-0.00	-0.00	-0.03	-0.01	-0.03	-0.00	0.01	1.00
Active_Days	0.00	-0.01	-0.02	-0.02	-0.02	-0.02	1.00	0.00	-0.01	0.00	-0.01	-0.01	0.00	0.00	-0.02
Last_Transaction_Days_Ago	0.02	-0.01	0.00	-0.00	0.00	-0.00	0.00	1.00	0.01	0.00	0.02	0.01	0.01	-0.01	-0.00
Loyalty_Points_Earned	0.01	0.01	-0.00	-0.01	0.00	-0.00	-0.01	0.01	1.00	0.02	-0.02	-0.01	-0.01	0.01	0.00
Referral_Count	-0.03	-0.02	-0.02	-0.02	-0.01	-0.03	0.00	0.00	0.02	1.00	0.01	0.01	-0.01	0.01	-0.02
Cashback_Received	-0.00	0.01	-0.03	-0.04	-0.01	-0.01	-0.01	0.02	-0.02	0.01	1.00	0.01	-0.01	-0.03	-0.01
Support_Tickets_Raised	0.01	-0.03	-0.01	-0.00	-0.01	-0.03	-0.01	0.01	-0.01	0.01	0.01	1.00	0.02	-0.00	-0.03
Issue_Resolution_Time	-0.01	0.01	-0.02	-0.01	-0.02	-0.00	0.00	0.01	-0.01	-0.01	-0.01	0.02	1.00	0.00	-0.00
Customer_Satisfaction_Score	-0.01	0.01	-0.00	-0.00	0.00	0.01	0.00	-0.01	0.01	0.01	-0.03	-0.00	0.00	1.00	0.01
LTV	0.01	0.65	0.66	0.54	0.54	1.00	-0.02	-0.00	0.00	-0.02	-0.01	-0.03	-0.00	0.01	1.00

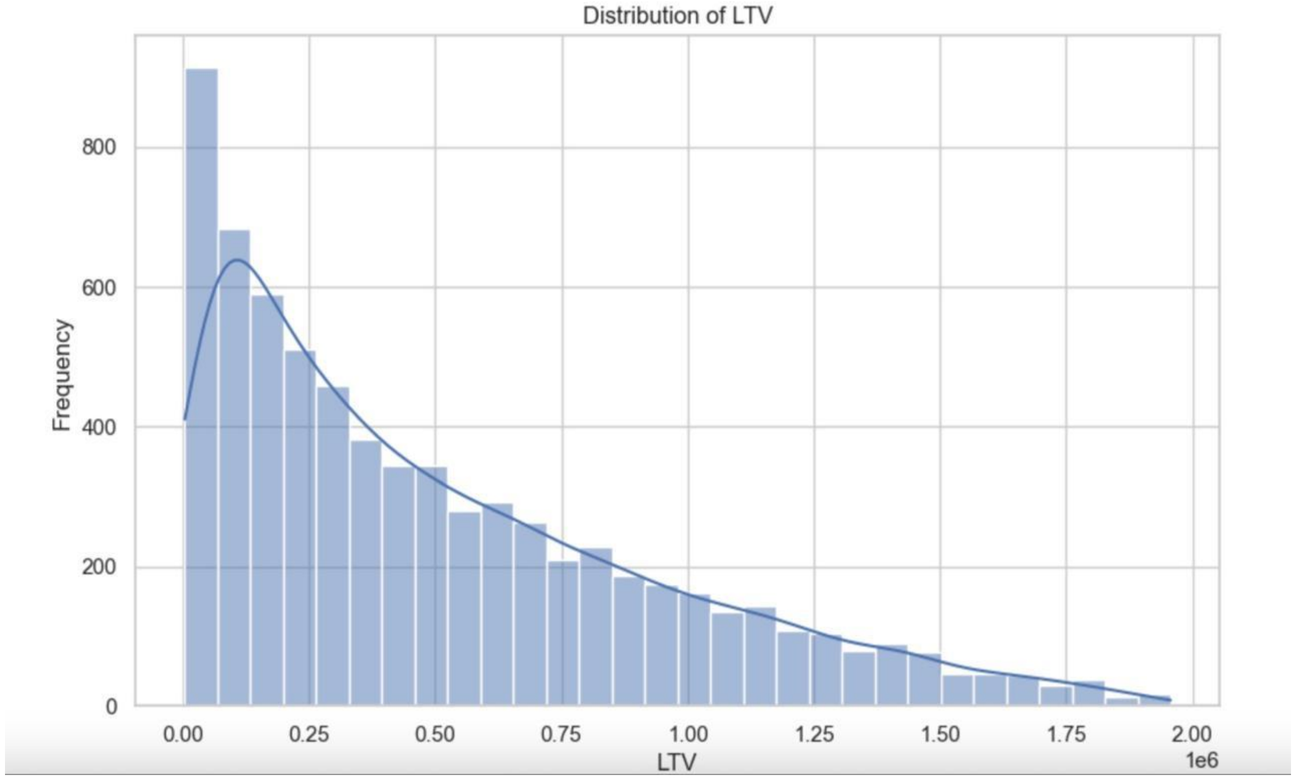
Distribution of Customer Lifetime Value (LTV)

- This chart shows how LTV is distributed across all 7,000 customers in the dataset.

- Most customers have a **low to moderate LTV**, and only a small number of customers have **very high LTV**. This is a classic case of a **right-skewed distribution**, which is common in real business data:

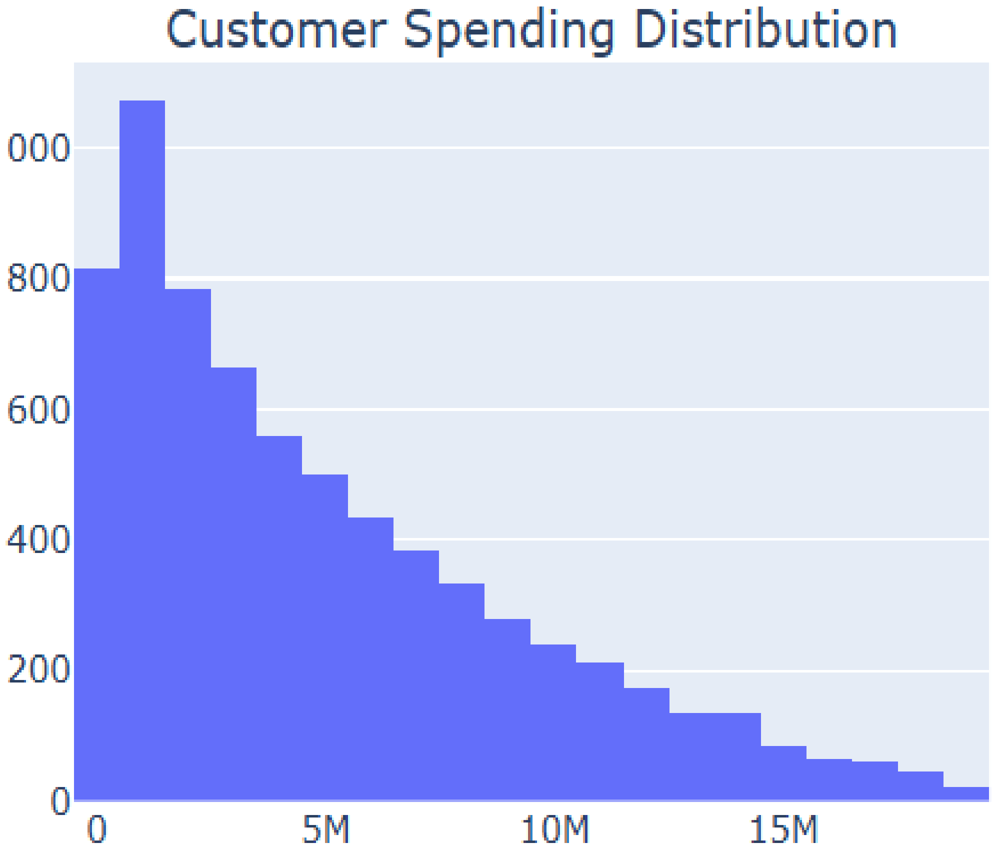
  - A few customers bring **very high value.**

  - Most customers bring **low or average value.**



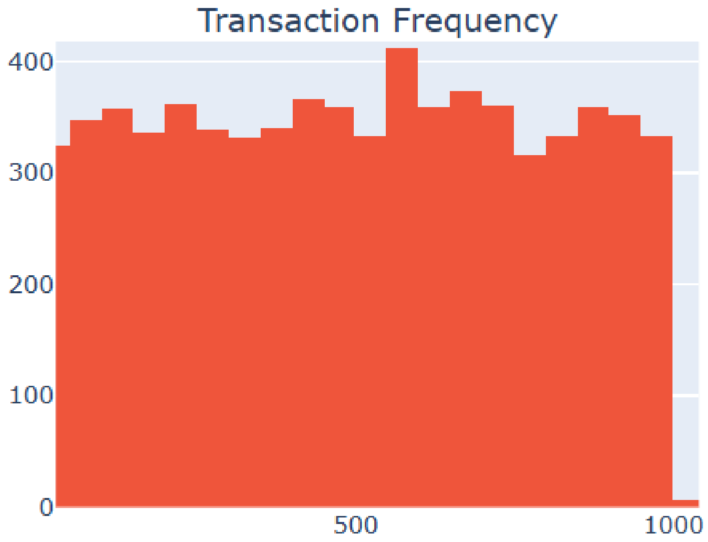
1. **Customer Spending Distribution**

- This is a histogram showing the distribution of customer spending.
- The x-axis represents spending in millions (e.g., 0, 5M, 10M, 15M).
- The y-axis represents the count of customers.
- The chart indicates that a large number of customers have **relatively low spending**, with the count decreasing significantly as spending increases.
- **This suggests a typical distribution where a few customers spend a lot, and many spend less.**



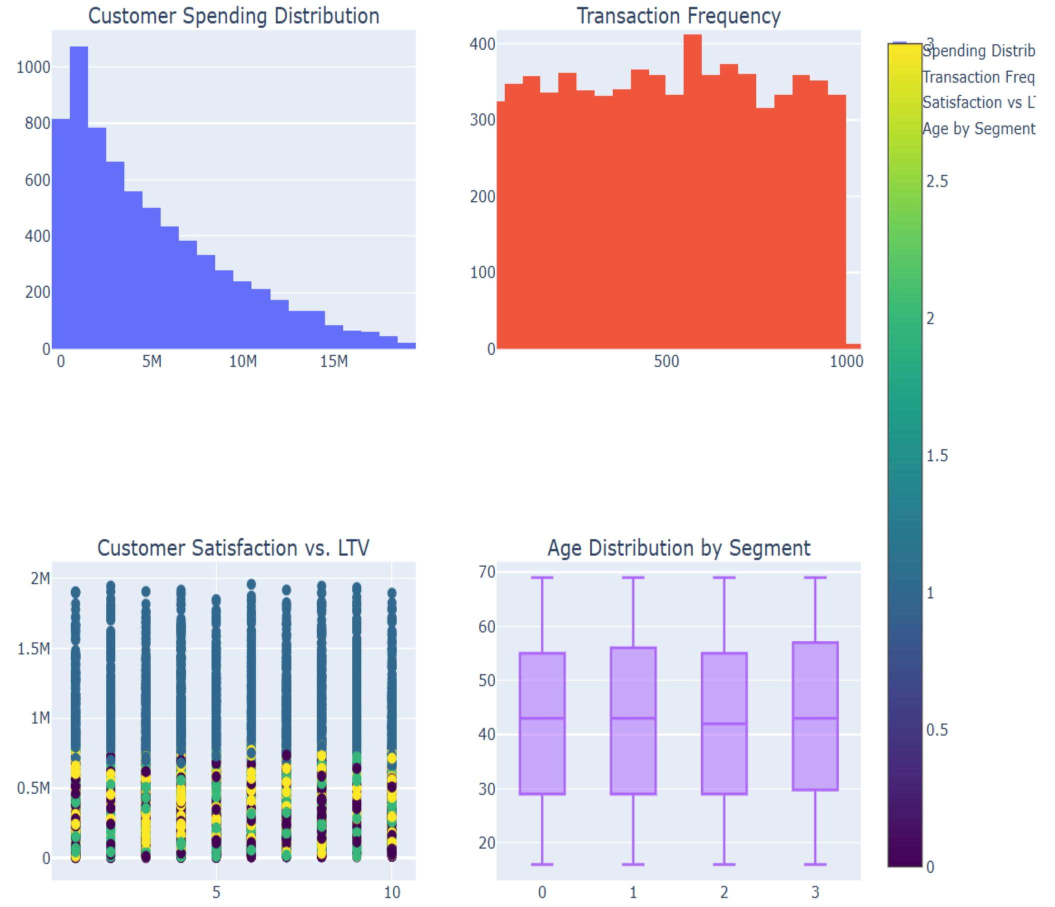
2. Transaction Frequency

- This is another histogram, likely showing the frequency of transactions.
- The x-axis ranges from 0 to 1000
- The y-axis represents the count or frequency.
- The bars are fairly consistent in height, suggesting a somewhat uniform distribution of transaction frequencies.



3. Customer Satisfaction vs. LTV

- The x-axis represent different levels or categories of customer satisfaction
- The y-axis represents LTV, ranging from 0 to 2 million.
- The different colored dots (green, yellow, various shades of blue/purple) likely represent different satisfaction levels, as indicated by the color bar legend.



Customer Analytics: Key Insights

- The "**Customer Spending Distribution**" highlights a strong concentration of customers in lower spending tiers, indicating a significant "long tail" where a few high-value customers likely drive substantial revenue.
- While "**Transaction Frequency**" appears somewhat consistent across various rates,
- the "**Customer Satisfaction vs. LTV**" plot suggests that while satisfaction is important, it doesn't solely determine Lifetime Value, implying other critical factors are at play.
- Finally, the "**Age Distribution by Segment**" clearly shows distinct age profiles across our customer segments, necessitating tailored approaches.
- These insights underscore the need for **segment-specific strategies in marketing**, **product development, and customer engagement** to maximize value across the diverse customer base.

Data Preparation: Features for Modeling

- **Objective:** To transform the raw data into a clean, numerical format suitable for our machine learning models.
- **Uniqueness:** No duplicate data
- **Completeness:** No missing data
- **Accuracy:** Potential Accurate data
- **Conformity:** Data format is correct
- While the data contains some values that appear as outliers, further investigation is needed to determine if these points represent potential customers. Removing them prematurely could lead to the loss of valuable information.
- The identified outliers in the **'Total_Spent', 'Max_Transaction_Value', 'Min_Transaction_Value', and 'LTV'** columns represent valuable data points for potential customers and should not be removed.
- **Categorical Feature Encoding:** Transformed nominal and categorical features into numerical format using **One-Hot Encoding.**
- **Feature Engineering:** Introduced new derived features, such as RF_score (Recency-Frequency Score), to capture valuable customer behaviour patterns.

Data Post-Cleaning: Ready for ML

- **Following our comprehensive Data Quality Assessment and Data Preparation steps, the dataset has been cleaned and transformed, making it fully ready for machine learning model training.**

No	Column Name	Data_type
1	**Age**	Int64
2	**Max_Transaction_Value**	Float64
3	**Min_Transaction_Value**	Float64
4	**Active_Days**	Int64
5	**Last_Transaction_Days_Ago**	int64
6	**Loyality_Points_Earned**	Int64
7	**Referral_Count**	Int64
8	**Cashback_Received**	Float64
9	**Support_Tickets_Raised**	Int64
10	**Issue_Resolution_Time**	Float64
11	**Customer_Satisfaction_Score**	int64
12	**Cluster**	Int 32
13	**RF_Score**	Int 32
14	**Location_Suburban**	Bool
15	**Location_Urban**	Bool
16	**Income_Level_Low**	Bool
17	**Income_Level_Middle**	Bool
18	**App_Usage_Frequency_Monthly**	Bool
19	**App_Usage_Frequency_Weekly**	Bool
20	**Preferred_Payment_Method_Debit Card**	Bool
21	**Preferred_Payment_Method_UPI**	Bool
22	**Preferred_Payment_Method_Wallet Balance**	Bool
23	**LTV (Target Variable)**	float64

Applying Clustering: Customer Segmentation

- Our **primary goal** in this phase is to **segment customers into meaningful groups** using the **K-Means Clustering algorithm**, enabling personalized marketing, strategic retention, and efficient resource allocation based on behavioral patterns.
- We selected **7 behavioral and spending-related features** that are highly relevant for business impact:
  'Total_Transactions', 'Avg_Transaction_Value', 'Total_Spent', 'Max_Transaction_Value', 'Min_Transaction_Value', 'Loyalty_Points_Earned', and 'LTV'.

Segmentation Focus: Customers are clustered based on **frequency** and **spending**, while **LTV is used as a reference metric** to assess business value of each segment

K-Means Clustering Algorithm

Type	**Unsupervised** – no target label, groups customers by similarity.
How it works	Iteratively assigns points to the nearest centroid, then moves centroids until positions stabilize.
Pros	Fast on large data; easy to implement; good at finding spherical clusters; results scale well.
Cons	Must pre-choose **k**; sensitive to scaling & outliers; assumes equal-size clusters.
Data Challenges	• Strong numeric skew → required StandardScaler. • LTV dominates magnitude → included but scaled. • Mixed behaviour/value features (transactions vs. spend) – need to balance feature set.

Model Evaluation Strategy

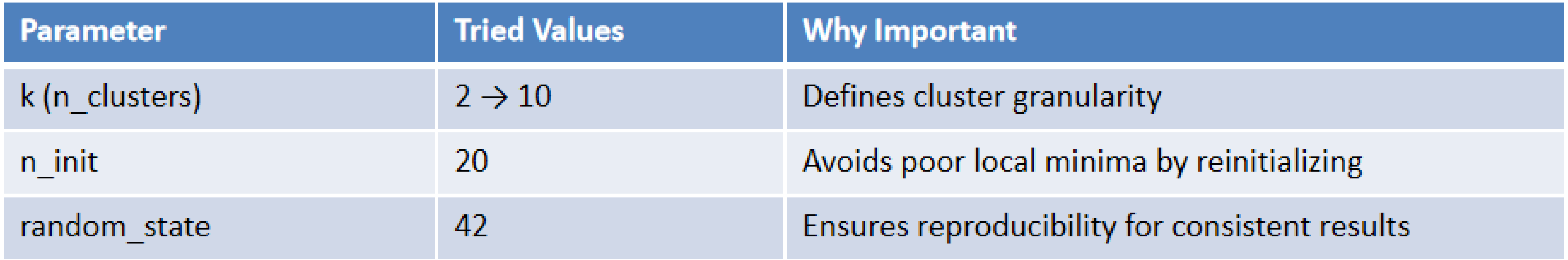
To ensure a fair and comprehensive comparison of clustering performance, we designed our model evaluation around **two key scenarios**:

- **Scenario 1 – Clustering on Full Feature Space (Non-PCA)**:
  - Clustering was performed directly on the scaled dataset using all selected features.
  - This approach maintains interpretability and preserves the original feature structure, allowing for easier business insight.
- **Scenario 2 – Clustering on Reduced Dimensional Space (PCA)**:
  - Principal Component Analysis (PCA) was applied before clustering to reduce dimensionality and noise.
  - This method aims to improve clustering performance by projecting data onto the most informative components.

Each scenario was evaluated using standard clustering metrics, including **Silhouette Score**, and **Davies-Bouldin Index**, to assess compactness, separation, and overall quality of the clusters. The model with higher Silhouette and lower DBI was selected for final clustering

Key Hyper-parameters Tuned

- The **same hyperparameter** tuning strategy was applied to both **PCA-based** and **Non-PCA** clustering pipelines.
- We perfomed a manual grid search over k (number of clusters), with fixed values for n_init = 20 and random_state = 42 to ensure fair comparison.
- This consistency allows us to isolate the effect of dimensionality reduction (via PCA) without introducing tuning bias.



Parameter	Tried Values	Why Important
k (n_clusters)	2 → 10	Defines cluster granularity
n_init	20	Avoids poor local minima by reinitializing
random_state	42	Ensures reproducibility for consistent results

Same hyperparameters were used for both PCA and Non-PCA to ensure fair comparison.

PCA-Based Clustering Results

We assessed clustering performance for *k = 2–10* using three internal metrics. **k= 3** is selected as a practical balance between performance and business interpretability.

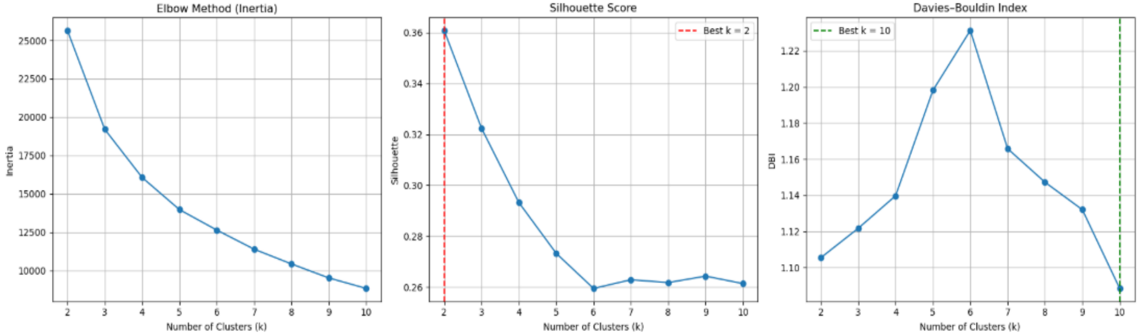
Scenario	Silhouette Score	Davies–Bouldin Index	Chosen k
Non-PCA	0.2875	1.2747	3
PCA	**0.3224**	**1.1217**	**3**

Evaluation Metrics Summary:

- Silhouette Score (Higher is better) → PCA produces **more cohesive and well-separatedclusters.**
- **DBI** (Lower is better) → PCA results in **more compact and distinct clusters**.

Based on these metrics, **PCA is selected for final clustering**

PCA Clustering Evaluation Charts:



Metric Insights:

- **Elbow:** Inertia drops sharply until **k = 3–4**, then flattens - suggesting diminishing returns.
- **Silhouette:** Best at **k = 2**, but **k = 3** still shows a strong score (~0.32), allowing finer segmentation.
- **DBI:** Lowest at **k = 10**, but **k = 3** remains compact (~1.12) and avoids over segmentation.

Customer Segments & Strategic Insights

Customer segment insights based on Clustering Results:

Cluster 0 – High-Value Power Users:

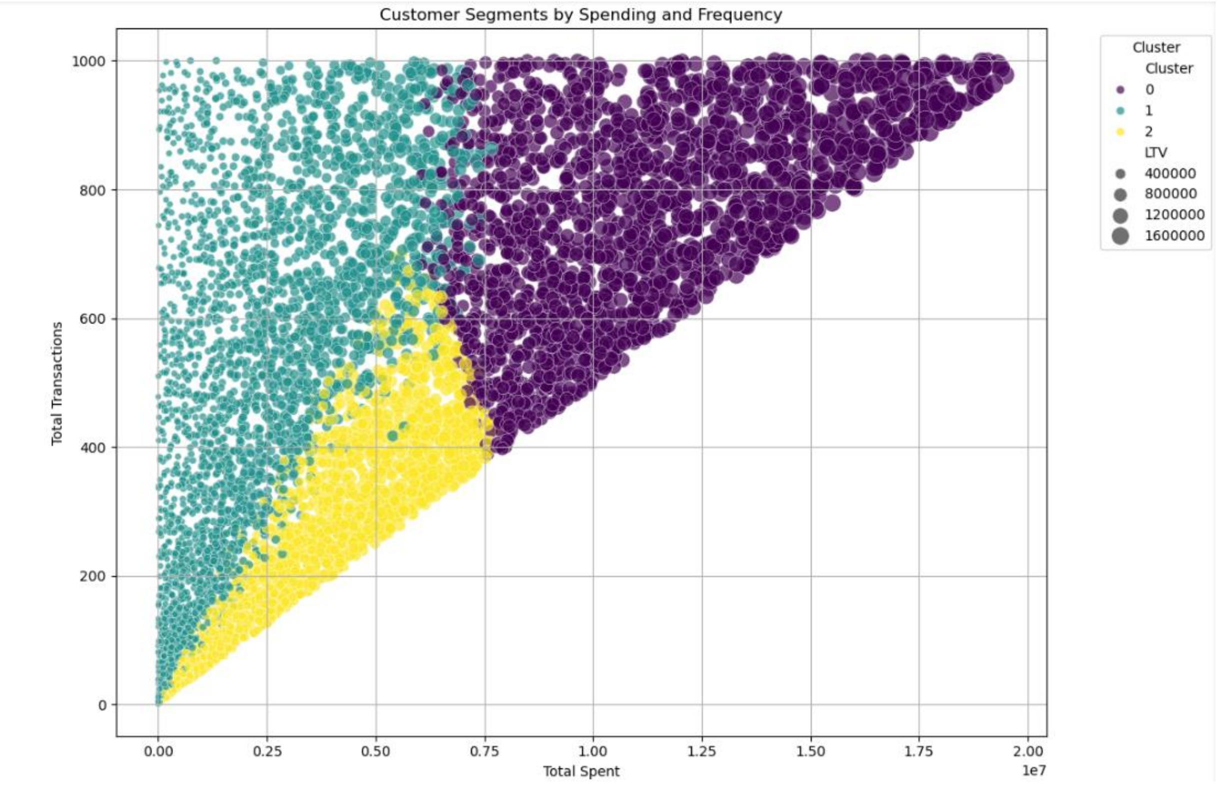
- These are frequent and high-spending customers with the highest LTV. Likely B2B or loyal top-tier users.
- **Suggested Strategy:** Retain this valuable group with loyalty programs, personalized rewards, and exclusive perks.

Cluster 1 – Moderate Spenders:

- Customers with low spending per transaction, moderate activity. LTV level is low.
- **Suggested Strategy:** Focus on reactivation campaigns using discounts, educational product content, or free trials.

Cluster 2 - **Premium Infrequent Buyers:**

- Customer who buy less often but spend more per transaction. Mid-to-high LTV.
- **Suggested Strategy:** Engage through VIP tiering, exclusive offers, or concierge-level service.



Applying the Regression Models: Predicting CLTV

•In the first part of our project, we explored the data and uncovered key insights into our customer base. Now, we're moving into the **predictive modeling phase.**

•Our primary goal in this phase is to predict Customer Lifetime Value (CLTV) using various regression techniques on our prepared dataset.

- **Linear Regressor :** Utilized as a simple baseline model for initial interpretation.

Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

- **Random Forest Regressor:**Selected for its high-performance capabilities, especially with structured data, aiming for robust predictive accuracy.

Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

- **XGBoost Regressor:**Chosen for its ability to capture complex non-linear patterns and handle feature interactions effectively.

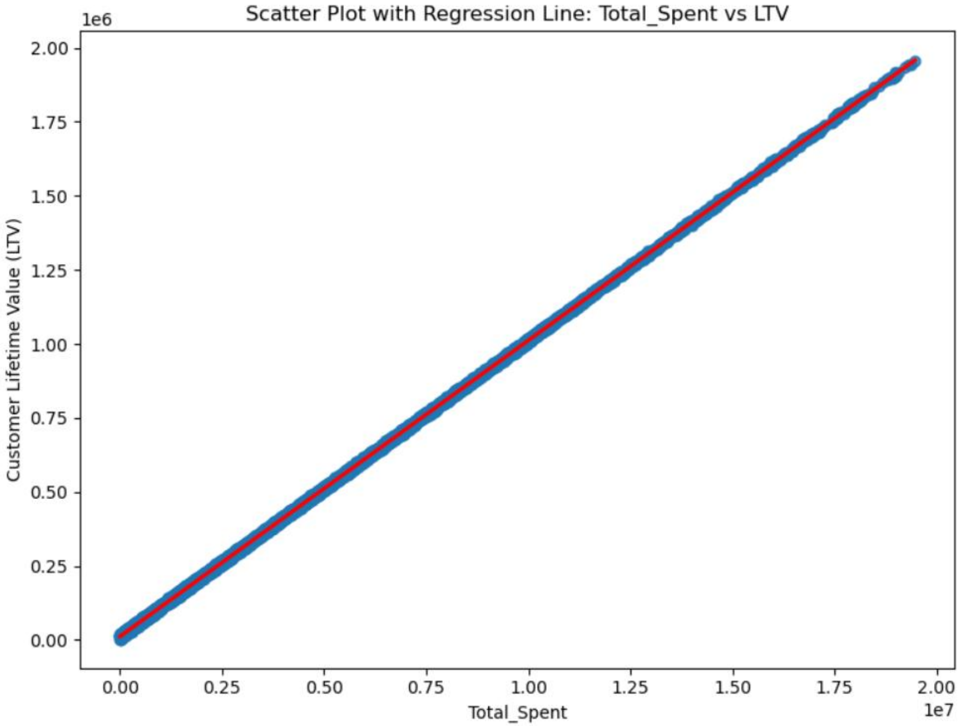
Documentation: https://xgboost.readthedocs.io/en/stable/parameter.html

Visualizing LTV's Dependence on Spending

- This scatter plot illustrates a strong positive linear relationship between **'Total Spent' and 'Customer Lifetime Value (LTV)',**

- indicating that higher spending customers generally exhibit a higher lifetime value.

- Note: **Total_Spent** is typically removed from the feature set when predicting Customer Lifetime Value (LTV) due to **data leakage** or **target leakage**.



Key Findings

- If Total_Spent (which represents past total revenue/spending) is included as a feature, the model might simply learn to predict LTV directly from Total_Spent rather than learning the underlying factors that *drive* LTV.
- It is a direct component of or highly correlated with Customer Lifetime Value (LTV), which would lead to an **artificially inflated model performance** rather than truly predictive insights.

Regularized Linear Model (Ridge Regressor)

- **Type**: Supervised Learning (trained on labeled data to predict continuous outcomes)

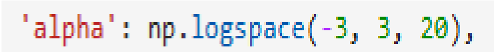
- **How it Works**:
  A linear regression model enhanced with **L2 regularization** to penalize large coefficients, reducing overfitting and improving generalization. If regularization parameter (alpha) = 0, it behaves the same as standard linear regression

- **Pros:**
  - Simple, fast, and interpretable (clear coefficient-based feature influence)
  - Handles multicollinearity well.

- **Cons:**
  - Only captures straight-line (linear) patterns.
  - Needs feature scaling and tuning (alpha)

- **Challenge**:
  - **RF_Score** and **Last_Transaction_Days_Ago** dominate the model. This imbalance may overshadow smaller but still valuable predictors.
  - Ridge regression is **sensitive to scale** – StandardScaler was necessary to ensure fair penalty application across coefficients

Hyperparameter Tuning

- **alpha** (Regularization Strength):
  - 

    ```
    'alpha': np.logspace(-3, 3, 20),
    ```

    , this generates 20 numbers from 10^{-3} to 10^{3} on a logarithmic scale. Controls how much the model penalizes large coefficients.

    Smaller values → more flexible model (risk of overfitting)
    Larger values → more regularization (risk of underfitting

- **fit_intercept**:

  Tested both *'True'* and '*False'* to check if the model should learn an intercept term.

- **solver**: Used *'auto'* and *'saga'*, both efficient solvers for linear models.
  'saga' works well with large datasets and is robust.

Model Evaluation Strategy - 1

Train-Test Split
Dataset was split into 75% training and **25% testing** using train_test_split.
This ensures fair evaluation on unseen data.

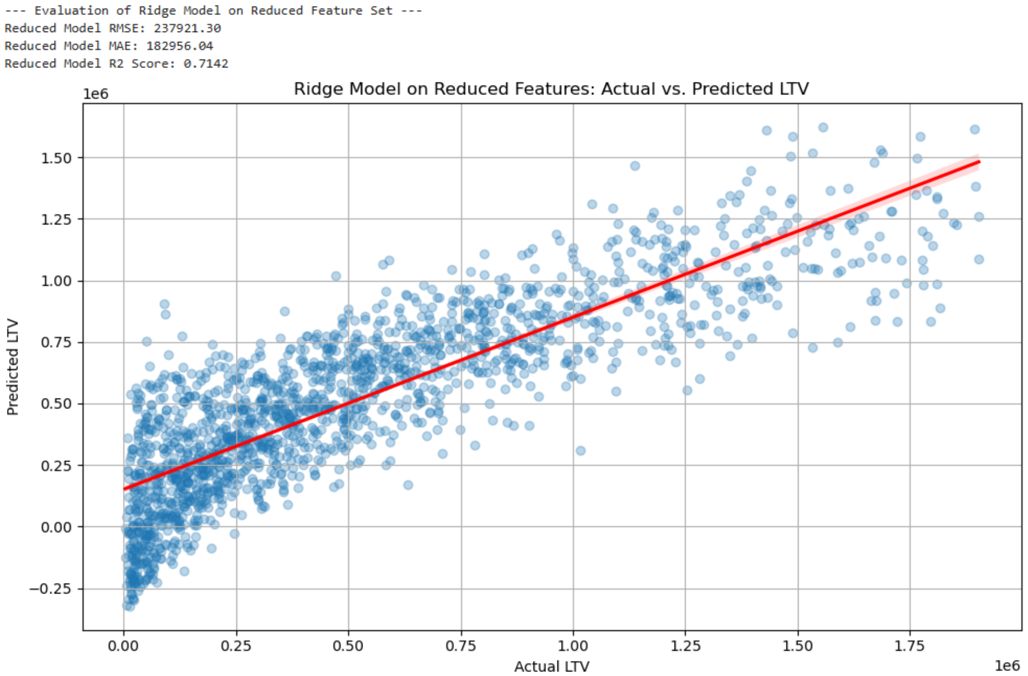
To assess model performance comprehensively, we implemented a **three-step scenario evaluation approach. Performance evaluated** with**:** **R²** (model fit), **RMSE** (penalizes large errors), **MAE** (average error).

Scenario steps:

1. **Base Model: Ridge Regression (Regularized Linear Regression)**
   Used as a simple and interpretable baseline model.

2. **Model Tuning: Ridge Regression with optimized hyperparameters**
   Applied Ridge Regression to reduce overfitting.
   Used GridSearchCV with 5-fold cross-validation to tune **alpha**. Optimal alpha improves generalization.

Model Evaluation Strategy - 2



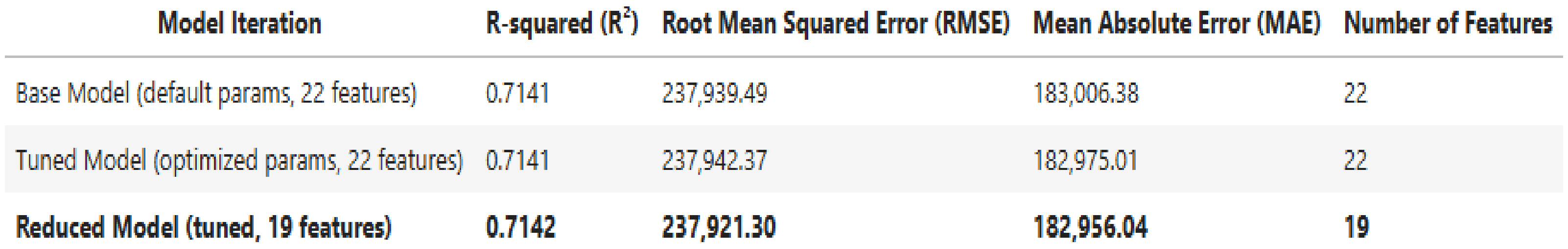
```
--- Evaluation of Ridge Model on Reduced Feature Set ---
Reduced Model RMSE: 237921.30
Reduced Model MAE: 182956.04
Reduced Model R2 Score: 0.7142
```

3. Feature Selection

Selected top features based on Ridge coefficients.

- Reduced from **22 to 19 features**
- Resulted in slightly **better RMSE and MAE**

R^2 (R-squared) = 0.714 indicates that approximately **71.4% of the variation in Customer Lifetime Value (LTV)** in the original data can be explained by the predictive model.



Model Iteration	R-squared (R^2)	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	Number of Features
Base Model (default params, 22 features)	0.7141	237,939.49	183,006.38	22
Tuned Model (optimized params, 22 features)	0.7141	237,942.37	182,975.01	22
Reduced Model (tuned, 19 features)	**0.7142**	**237,921.30**	**182,956.04**	**19**

Key Finding

- **Tuning Ridge Regression** didn’t improve performance — R^2 stayed at **0.7141**, and RMSE/MAE only changed slightly. *>> Default parameters were already close to optimal, likely due to the dataset's strong linear patterns.*

- **Feature reduction** gave a **small boost** in RMSE/MAE without hurting R^2. *>> Removing low-impact features helped reduce noise and simplified the model.*

- **All models performed similarly**, showing **stable and reliable results**. *>> The data was well-prepared and highly suited for linear modeling.*

- Top 4 features contribute the most to predicting Customer Lifetime Value (CLTV): **RF_Score, Last_Transaction_Days_Ago, Min_Transaction_Value, Max_Transaction_Value.** They highlight the importance of recent, frequent, and high-spending behavior, helping businesses focus retention and upsell strategies on the right segments.

Random Forest Regressor

- Type: Supervised Learning (learns from labeled historical data).
- How it Works: It's an "ensemble" method that builds hundreds of decision trees and averages their predictions for a more stable and accurate result.

Pros:

- High accuracy and robustness to outliers.
- Excellent at capturing complex, non-linear patterns.
- Provides clear feature importance rankings.

Cons:

- Less interpretable than simpler models (a "black box").

Hyperparameter tuning

- We optimized our model by tuning key hyperparameters using RandomizedSearchCV to find the best settings.
- Key Parameters Tuned:
- n_estimators (Number of Trees): Set to 200 for a balance of performance and efficiency.
- max_depth (Tree Depth): Limited to 20 to capture detail without overfitting.
- min_samples_split & min_samples_leaf: Kept at robust defaults to ensure generalized learning.
- max_features: 'sqrt': This was crucial. By using only a subset of features for each tree, we de-correlated the trees and built a more robust overall model.

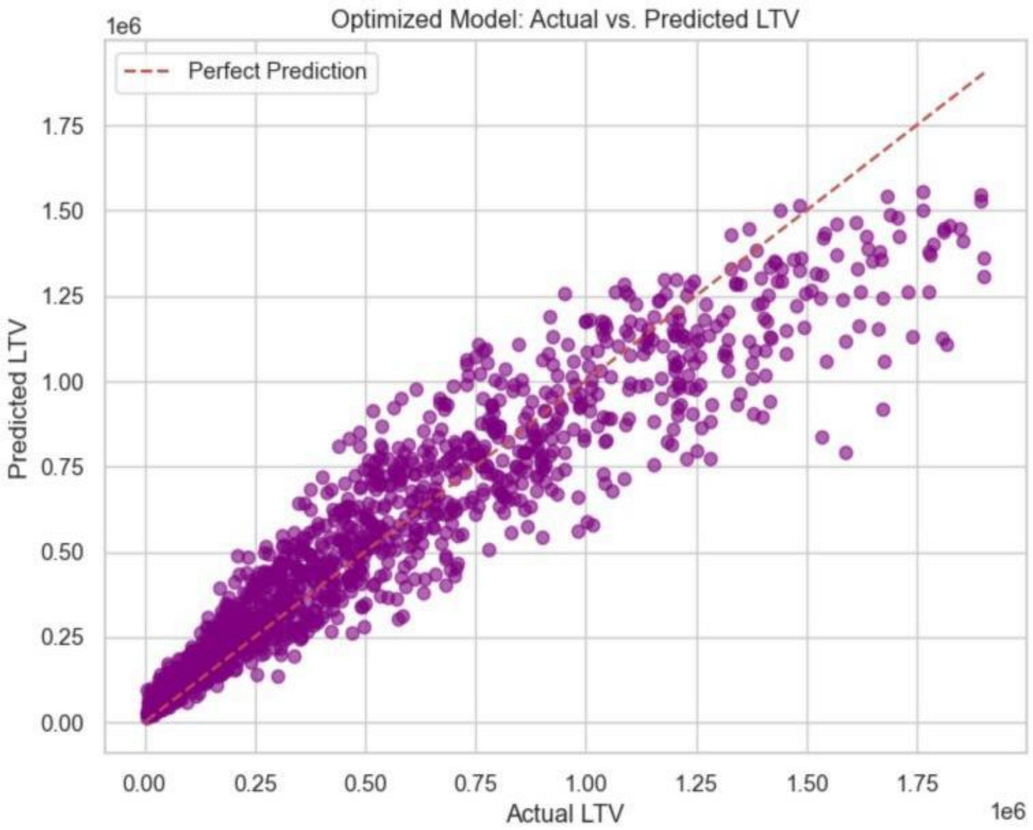
Model Evaluation Strategy

- Testing Strategy:
- 80/20 Train-Test Split: The model was trained on 80% of the data and evaluated on a 20% "hold-out" set it had never seen, simulating real-world performance.
- K-Fold Cross-Validation: Used during tuning to ensure the model's stability and prevent dependency on a single data split.
- Key Performance Metrics:
- R-squared (R^2): 0.8862 — Our model explains 89% of the variance in LTV. This is a strong, realistic score.
- Mean Absolute Error (MAE): $104,257 — On average, our predictions are off by this amount, which is acceptable given the LTV range.

End Result Visualisation

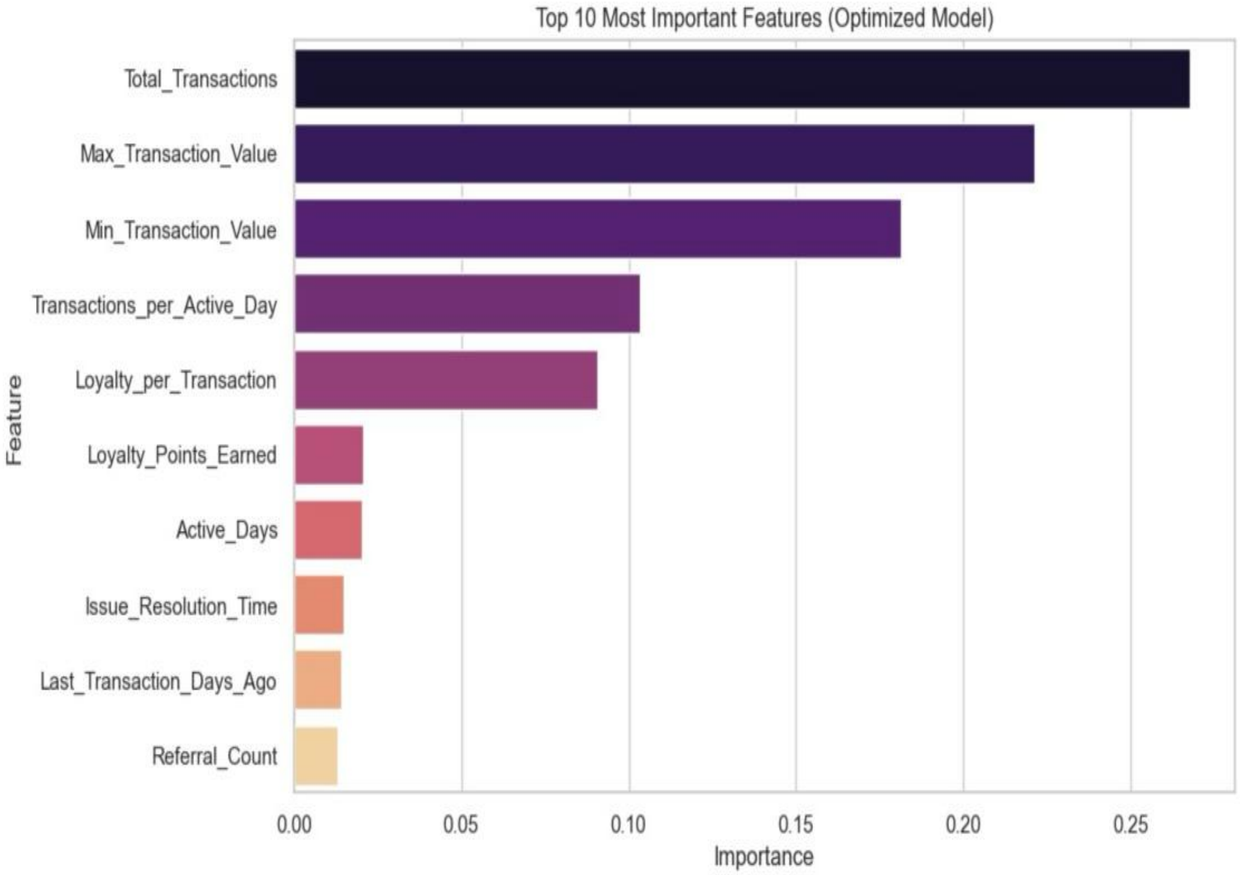
- This plot is our final validation. It shows a strong positive correlation between the actual LTV (x-axis) and our model's predicted LTV (y-axis).

- The data points cluster tightly around the red "perfect prediction" line, visually confirming the model's accuracy across different customer value levels.



Key Findings

- This feature importance plot is the most valuable output of our balanced model. It shows what truly influences LTV.
- Total_Transactions and Max_Transaction_Value are the most powerful predictors. This confirms that high-frequency, high-value customers are our most valuable.
- The initial model based on Total_Spent was completely misleading. It had perfect accuracy but provided zero strategic insight. Our balanced approach successfully overcame this.



CLTV Prediction: XGBoost Model Training

- **Type: Supervised Learning (Regression)**
- XGBoost is a supervised learning algorithm. It learns patterns from our labeled historical data.

How it works?

- An advanced **ensemble method** based on **Gradient Boosting**, it builds a series of weak predictive models (decision trees) sequentially.
- The final prediction is an aggregation of all individual tree predictions, yielding highly stable and accurate results.

❖ **Pros :**

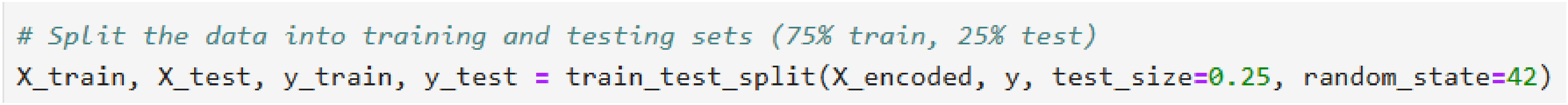
✓ XGBoost stands out for its **exceptional predictive accuracy** on structured data, effectively capturing complex patterns while being **robust to outliers and missing values**. Its **built-in feature importance** provides valuable insights, and it is highly **optimized for speed and scalability**.

❖ **Cons:**

○ Despite its power, XGBoost is generally considered a **"black box" model**, making its internal decision process less interpretable than simpler algorithms. Achieving optimal performance often **requires extensive hyperparameter tuning**, which can be a time and resource-intensive process.

Model Evaluation Strategy

- Our dataset was strategically partitioned, allocating **75% of the data for model training** to learn underlying patterns, and reserving the remaining **25% for rigorous testing** to ensure the model's ability to generalize to unseen customer data.



```
# Split the data into training and testing sets (75% train, 25% test)
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.25, random_state=42)
```

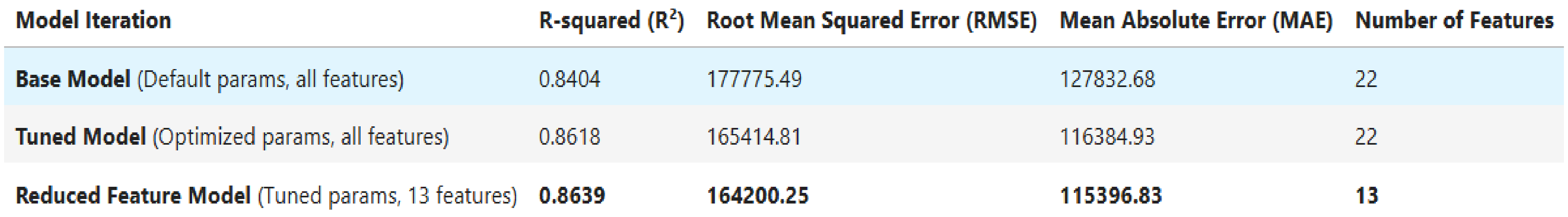
Key Performance Metrics:

- **R-squared (R2):** Our model's R-squared value indicates the percentage of variance in Customer Lifetime Value (CLTV) that our features successfully explain, reflecting its overall explanatory power.
- **Root Mean Squared Error (RMSE):** The RMSE quantifies the typical magnitude of errors in our CLTV predictions, expressed in the same currency units as LTV, with larger errors being penalized more.
- **Mean Absolute Error (MAE):** MAE represents, on average, the absolute difference between our model's predicted CLTV and the customer's actual CLTV, providing a straightforward measure of prediction accuracy.

CLTV Prediction: XGBoost Model Performance

- **Objective:** To quantify how well our XGBoost model predicts Customer Lifetime Value (CLTV) on unseen data and to evaluate the impact of model tuning and feature selection.

- **Model Iteration & Evaluation Metrics:**

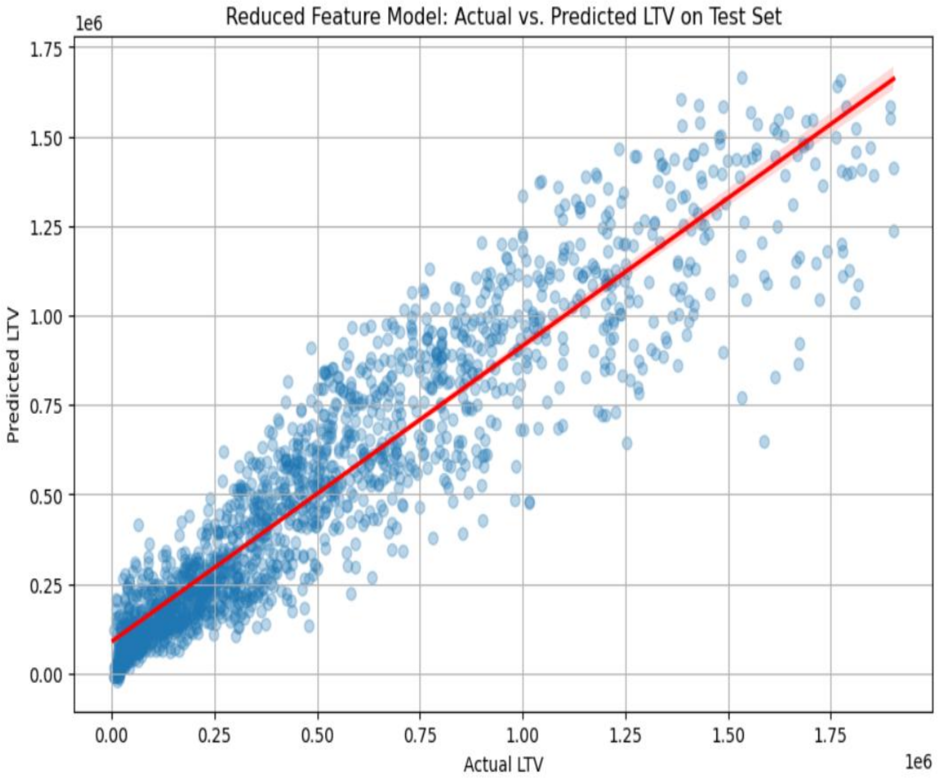


Model Iteration	R-squared (R^2)	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	Number of Features
Base Model (Default params, all features)	0.8404	177775.49	127832.68	22
Tuned Model (Optimized params, all features)	0.8618	165414.81	116384.93	22
Reduced Feature Model (Tuned params, 13 features)	**0.8639**	**164200.25**	**115396.83**	**13**

Hyperparameter Tuning Impact:

- Our rigorous hyperparameter tuning process, leveraging techniques like Grid Search Cross-Validation, involved **fitting 3 folds for each of 72 candidate hyperparameter combinations, totaling 216 model fits**. This systematic optimization of key XGBoost configurations (e.g., n_estimators, learning_rate, max_depth) was crucial in preventing overfitting and directly achieving the superior performance of our 'Tuned Model'.

XGBoost Model: Visualizing Prediction Accuracy



- With an impressive **R-squared of 0.8639**, our XGBoost model effectively predicts approximately **86.4% of the variance in Customer Lifetime Value (CLTV)**, showcasing its strong predictive power.

- This high accuracy enables precise **data-driven decision-making**, transforming CLTV into a proactive strategic tool.

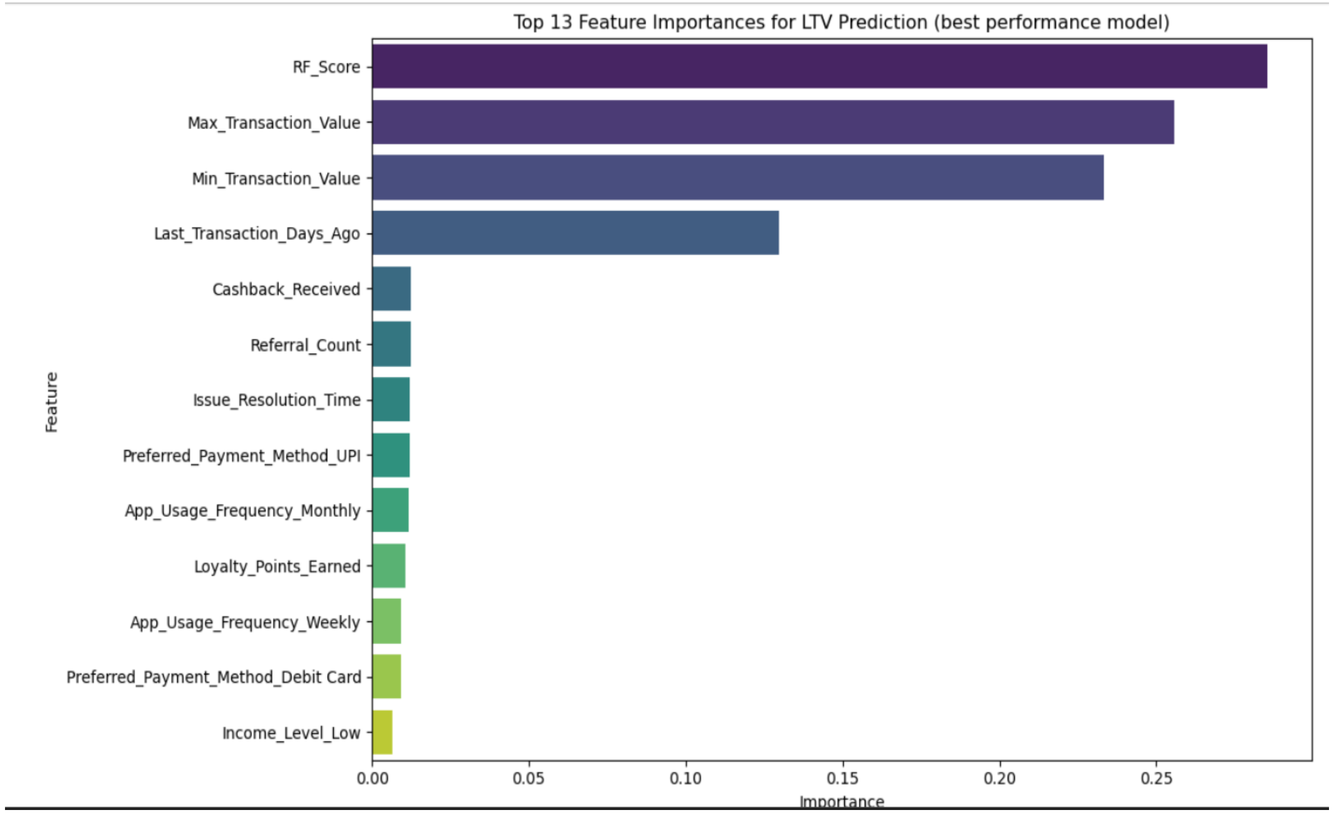
- Businesses can now optimize customer acquisition, personalize marketing efforts, and allocate resources more efficiently to maximize profitability and sustainable growth.

CLTV Prediction: Key Feature Importance

- **Objective:** To identify which features are most influential in predicting Customer Lifetime Value (CLTV) according to our optimized XGBoost model. Understanding these drivers allows for more targeted strategies.

- **Methodology:** XGBoost inherently calculates feature importance based on how frequently a feature is used to split nodes in the ensemble of trees and how much it contributes to reducing prediction error.

- The XGBoost model highlights that **Recency-Frequency Score (RF_Score)**, **transactional value metrics (Max/Min Transaction Value)**, and **customer engagement (Last_Transaction_Days_Ago)** are the paramount drivers of Customer Lifetime Value, enabling businesses to focus strategies on fostering higher value transactions, frequent interactions, and customer loyalty.



Comprehensive Model Comparison: Linear Regression, Random Forest, & XGBoost

- **Objective:** To compare the performance of distinct regression model types to identify the most robust and accurate approach for predicting Customer Lifetime Value (CLTV).

Model	R^2	RMSE	MAE	Note
Random Forest	0.8862	–	104,257	Highest accuracy, lowest error
XGBoost	0.8639	164,200	115,396	Strong performance, tunable & scalable
Linear Regression (Ridge)	0.7142	237,921	182,956	Simple baseline, least accurate

Summary

- We compared three models—Random Forest, XGBoost, and Ridge Regression—for predicting Customer Lifetime Value. What worked best was **Random Forest**. It had the **highest R² score of 0.8862** and the **lowest MAE of 104,257**. This means it gave the **most accurate** predictions with minimal error.

- **XGBoost** also performed well, with an **R² of 0.8639** and a decent MAE. It's a **tunable and scalable** option, making it a strong alternative when more control is needed.

- What didn't work well at all was **Ridge Regression**. It had the **lowest R² of 0.71** and the **highest MAE**, showing poor predictive ability. It couldn't capture the complex relationships in the data.

- In conclusion, **tree-based models, especially Random Forest worked very well**, while **linear models were too simplistic** for this task. Therefore, **Random Forest is recommended for production deployment**

Conclusion: Actionable Insights & Next Steps

Project Recap:

❑ We successfully developed and optimized a Customer Lifetime Value (CLTV) prediction model, with the Random Forest model emerging as the most accurate. It explains an impressive 89% of CLTV variance, providing highly reliable forecasts.

Business Impact:

❑ This model empowers data-driven decisions, enabling precise customer targeting, optimized resource allocation, and proactive retention strategies. By understanding customer value, the business can significantly enhance profitability and growth.

Future Work & Enhancements:

❑ We recommend continuous model monitoring and retraining with fresh data to ensure sustained accuracy and relevance over time. Future efforts could explore integrating the model into live systems for real-time actionable insights.