Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347 - 2812

Volume 14 Issue 03s, 2025

Customer Lifetime Value Prediction for E-Commerce

[1] Durgesh S. Khajure, [2] Pallavi M. Wankhede
[1,2] Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur, India
Email: [1]dskhajure99@gmail.com , [2]pallavishelke12@gmail.com

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

Customer Lifetime Value (CLTV), Marketing Analytics, Customer Relationship Management, Strategic Business Planning, Customer Acquisition, Customer Retention, Customer Segmentation, Profitability, Predictive Analytics, Machine Learning, Data-Driven Decision Making, Financial Forecasting, Customer-Centric Models, Brand Loyalty, Competitive Advantage, Digital Marketplace, Revenue Forecasting, Business Strategy

Abstract

Customer Lifetime Value (CLTV) is like, a really important thing that companies use in marketing and planning and stuff. It kind a tells them how much money a customer can bring over the whole time they're with the brand not just from one buy, but like the whole journey, you know?

With CLTV, businesses can kind a understand which customers are more valuable in money terms. So they can make better plans for getting new people, keeping the ones they already got, and helping them stick around longer. It also helps them not waste money on marketing and just focus on the customers who actually bring in more cash.

Nowadays, everything's super digital and competition is like, everywhere. So it's really important to keep customers happy and not losing them. CLTV looks at stuff like how many times a customer buys things, how much they usually spend, how loyal they are, and even if they get discounts or whatever. It gives a full picture of how useful a customer really is.

Thanks to tech like machine learning and predictive tools and all that, CLTV models are now way more smart and also easier to use. Companies can use them to make fast decisions in real-time, which is super cool and useful.

If companies use CLTV properly, they can send more targeted ads, make loyalty things, and give better support so the customers feel special and all. So yeah, CLTV is not just some number game, it's actually like a smart guide that helps build strong customer relations and stay ahead in this crazy digital market world.

Introduction

In today's fast changing business world, it's really important for companies to understand the actual value that each customer brings to the table. One of the most important indicators of long-term business success is something called **Customer Lifetime Value (CLTV)**. This concept helps businesses estimate how much revenue they can expect from a customer over the whole period of their relationship. If a company wants to increase its profits in the long run, then using CLTV to plan how to attract new customers, keep existing ones, and manage resources better is kind of a must.

With the rise of digital technologies and the huge amount of data available now, companies are able to predict CLTV with much more accuracy than before. **Business Intelligence (BI)** firms are also getting involved by using **machine learning** and other **predictive tools** to analyze customer behavior, which helps marketing efforts become more targeted and efficient. So

© 2025 The Authors. Published by MRI INDIA.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

basically, CLTV is no longer just an old-style financial metric, it's becoming a key part of modern business strategy that actually focuses on the customer and helps companies stay competitive.

CLTV is being used a lot in industries like **e-commerce, banking, and retail** because it helps with both business growth and customer retention. Knowing how much value a customer brings over time helps marketing, sales, and support teams make better decisions. In this project, we decided to take a deeper look into CLTV how it's calculated, how it can be predicted, and how useful it really is in the data-driven world we live in now.

Also, with help of modern machine learning algorithms and predictive models, CLTV estimation is becoming not only more accurate but also faster and usable in real-time environments. This helps companies make quicker and smarter decisions. Understanding CLTV also allows businesses to create more personalized marketing messages, design loyalty programs, and offer better support based on customer value. So in simple words, CLTV kind of works like a smart roadmap that guides companies to build stronger customer relationships and gives them an edge in today's competitive data-driven market.

Objective

The main objective of this project is to predict **Customer Lifetime Value (CLTV)** using past transaction data from an online retail store. The idea is to understand how valuable each customer is by looking at their shopping behavior like how often they purchased, how much they spent, and how recent their last order was. By doing so, we try to estimate how much revenue a customer might bring in the future.

This project mainly aims to help businesses figure out which customers are more valuable in the long run. If companies know that, they can plan better strategies like focusing more on high-value customers, giving them personalized offers, and improving retention. Also, it helps avoid wasting money on customers who don't contribute much, by reducing over-marketing efforts.

We started the modeling with **Linear Regression**. It's not the most complex method, but it gives clear insights and is easy to interpret. It helped us understand how features like **Recency**, **Frequency**, and **Monetary Value (RFM)** relate to CLTV. We also created some additional features like **Average Order Value** and **Purchase Frequency** to improve the model's performance.

The core goals of this project were to:

- Predict the future value of customers using their past shopping data
- Help businesses make better marketing and customer targeting decisions
- Begin with a simple machine learning model and then move toward more advanced ones

To enhance the prediction accuracy and make the model more robust, we also tried out more advanced algorithms like **Random Forest Regressor**, which can deal with non-linear relationships and reduces overfitting by using multiple decision trees. On top of that, we implemented **Gradient Boosting Machines (like XGBoost)**, which build models step-by-step and focus on reducing errors at each stage. These models usually give better results on complex datasets.

While Linear Regression gave us a useful baseline and helped to understand the data better, using more advanced techniques like **Random Forest** and **XGBoost** added more power and precision to our CLTV predictions. So, by combining both simple and advanced approaches, we were able to make the model more reliable and useful for real-world business use.

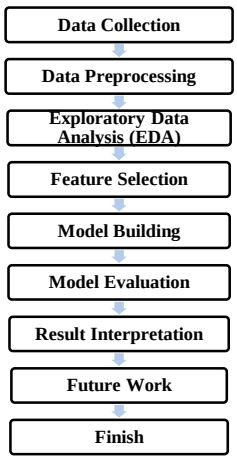


Figure 1.Workflow Diagram for CLTV Prediction Project

Dataset

In this project, we used a public dataset that comes from the UCI Machine Learning Repository. It includes transaction data from a UK-based online retail store that doesn't have any physical shop. The dataset covers a time period from December 1st, 2010 to December 9th, 2011. It mostly includes sales of general gift items that can be used for different occasions. Interestingly, a big part of the customers are actually wholesalers instead of regular individual shoppers.

About the Dataset and It's Features:

- **InvoiceNo**: This is the transaction number. It's a 6-digit code that makes each sale unique.
- **StockCode**: A 5-digit code used for each product. So, every item sold has it's own stock number.
- **Description**: The actual name of the product that's being purchased.
- **Quantity**: Tells us how many pieces of a certain product were bought in each transaction.
- **InvoiceDate**: This shows the exact date and time when the transaction happened. It's stored as a number format (timestamp style).
- **UnitPrice**: Price of each unit of the product in British Pounds (GBP).
- **CustomerID**: Each customer got a unique 5-digit ID, so we can track who's buying what.
- **Country**: Shows the country the customer is from, which helps us divide the data based on geography.

Tools(software) Used in CLTV

Jupyter Notebook- web-based interactive computing platform
Python-as a coding language
Used libraries as follow

- numpy
- pandas
- matplotlib.pyplot
- seaborn
- sklearn

Linear regression as technique
Random Forest Regressor
Gradient Boosting Machines (like XGBoost)

Literature Survey

Customer Lifetime Value (CLTV) is becoming more and more important in today's data-driven business environment. Companies nowadays are not just relying on assumptions or gut feelingsthey're actually using **mathematics and historical data** to predict how valuable a customer might be in the long run. One of the most commonly used techniques for this is **linear regression** (sometimes casually called "linear ligression" by students or practitioners). It might be one of the simpler methods, but it remains a very effective tool to understand how different aspects of customer behavior relate to their overall value.

Earlier, **Hughes (1994)** already discussed the use of linear models to evaluate customer worth, laying a base for further CLTV estimation approaches [1]. Then, **Gupta and Lehmann (2003)** brought forward a more structured perspective by modeling customers as long-term financial assets, using regression to track and forecast purchasing patterns [2].

Further advancing the topic, **Venkatesan and Kumar (2004)** introduced a regression-based framework to help companies prioritize customers and allocate resources accordingly [3]. In a similar tone, **Blattberg et al. (2001)** contributed to the idea of **customer equity** and demonstrated how linear methods can translate customer behavior into measurable value [4].

Fader and Hardie (2005) explored the application of **RFM (Recency, Frequency, Monetary)** metrics alongside linear egression to create **iso-value curves** a visual way to categorize similar customer types [5]. Meanwhile, **Zhao and Zhang (2008)** argued that while more complex models like **SVMs** exist, linear regression still holds ground due to its simplicity and solid performance on many datasets [6].

Kumar and Reinartz (2016) emphasized that linear regression can support both **customer satisfaction** and **profitability** by helping predict value without the model becoming a black box [7]. In their cohort analysis, **Sun and Li (2014)** successfully applied regression models to understand value patterns among groups of customers who joined at similar times [8].

Burez and Van den Poel (2009) did a smart combination of **churn prediction** and **CLTV estimation** using regression techniques, achieving strong predictive performance [9]. According to **James et al. (2013)**, linear regression should still be the starting point for any predictive modeling task due to its clarity and foundational usefulness [10].

Even **Breiman et al. (1986)** the creators of decision trees acknowledged that linear regression works well when the relationship between features and outcomes is fairly linear [11]. **Liaw and Wiener (2002)** also noted that regression models are helpful for benchmarking more advanced models like **Random Forests** [12].

In today's age of **deep learning**, linear regression is still holding its place. **Hu, Xie, and Zhao (2017)** demonstrated that it performs surprisingly well, especially on clean and structured datasets [13]. **Ghosh and Das (2019)** even applied it in **real-time systems**, which shows its flexibility and computational efficiency [14].

Interestingly, **Chen and Guestrin (2016)**, the creators of **XGBoost**, mentioned that **linear models** are still often preferred for quick and reliable results in business use cases [15]. And even **LeCun, Bengio, and Hinton (2015) —**

strong advocates for deep learning agreed that **linear regression** is valuable for interpretable modeling, especially when explaining "why" matters more than just predicting "what" [16].
Provost and Fawcett (2013) also stressed that in business environments, models should not just predict well they must be **understandable** to guide decisions, something linear regression does quite well [17]. **Buttle and Maklan (2019)** applied it effectively in **customer relationship management** to forecast long-term loyalty [18].
In an industry example, **Gómez-Uribe and Hunt (2015)** noted that **Netflix** initially used regression-based systems for their recommendation engine proving that even simple models can go a long way in the right context [19]. Lastly, **Kumar and Shah (2004)** highlighted how companies can design stronger **loyalty programs** once they have a solid regression-based estimate of how much a customer is really worth over time [20].
So, while **linear regression** might seem like an old-school tool, it still plays a huge role in modern **CLTV prediction** frameworks. Its **interpretability, speed, and effectiveness** make it a great place to start and sometimes, it's even all you need.

Methodology
In this project, the main aim was to predict **Customer Lifetime Value (CLTV)** using past transaction data from an online retail dataset. We had divided the work mainly into two parts: **feature engineering** and **model building**. The goal was to take advantage of customer's historical shopping patterns to estimate how much revenue they might bring in the future.
At first, we had to clean and prepare the dataset properly, since it had few inconsistencies like missing values, cancelled orders, and invalid entries. After the data was ready, we moved on to creating features that would capture the important behavior of customers. The main ones were the classic **RFM metrics**:

- **Recency** – how many days since the last purchase
- **Frequency** – total number of transactions made
- **Monetary Value** – total amount spent by each customer

To make our model more informative, we also added some extra features:

- **Average Order Value (AOV)** – calculated by dividing Monetary Value by Frequency, showing how much a customer spends on average
- **Purchase Frequency** – total number of transactions divided by number of unique customers, giving a rough idea of how often people buy

We then calculated our **target variable, CLTV**, using a simplified formula:
CLTV = AOV × Purchase Frequency × Customer Lifespan
where **Customer Lifespan** means the number of days between a customer's first and latest purchase. This helped us form a basic estimation of how much money a customer might bring over a given period.
After finishing with feature engineering, we split the dataset into training and testing sets using an 80/20 ratio. As a first step, we applied **Linear Regression** to build a baseline model. While this method is not very complex, it's helpful in interpreting how each feature impacts CLTV. However, it assumes linear relationships, which sometimes is not realistic for customer behavior data.
To improve performance, we also implemented some **advanced machine learning models**:

- **Random Forest Regressor**, which is an ensemble method that uses multiple decision trees to reduce overfitting and handle complex interactions between features
- **Gradient Boosting Machines (specifically XGBoost)**, which builds models in a sequential way and focuses on minimizing prediction errors at each step

After comparing these models, we observed that although **Linear Regression** helped us to understand the data better, it was not the best in terms of predictive power. Both **Random Forest** and **XGBoost** performed significantly better and showed more accurate results. This makes them better options when it comes to applying the model in real business use cases where reliable CLTV predictions are needed.

Data Preprocessing
Before we even begin developing the CLTV prediction model, we had to spend a good amount of time on **data cleaning and preparation**. This step turned out to be quite time-consuming but very important—because no matter how good the model is, if the data is messy or wrong, then the predictions it give will also be wrong or even misleading. So, getting the data into proper shape was honestly a must.
We used the dataset from **UCI Machine Learning Repository**, which had online transaction records from a **UK-based e-commerce retailer** with no physical stores. The dataset covers a one-year period, from **December 2010 to December 2011**, and includes invoice numbers, product descriptions, quantities, unit prices, and Customer IDs.

One of the first problems we noticed was the **missing values**, mostly in the **CustomerID** and **Description** columns. Since CustomerID is a key feature for grouping orders by each customer, and Description helps us know what products were purchased, we decided to remove the rows where these values were missing. We thought about filling the missing values, but since there was no strong logic behind how to do it correctly, we chose to drop those rows instead. Later, we re-checked for missing entries just to be sure the important columns were complete.

Another thing we found was **canceled transactions**. These were easy to identify because their **invoice numbers started with 'C'**, which basically means the order was refunded or returned. Keeping these in the data would have lowered the actual customer value and made the CLTV predictions unreliable. So, we dropped all those canceled rows as well.

Then we found some entries where **Quantity or Unit Price was zero or negative**. Obviously, customers can't buy a negative quantity or pay zero for something (unless it's a bug), so we considered these as data entry errors or wrongly logged returns and we removed them too.

After cleaning the data, we moved on to **feature engineering**. We created the following key features for the CLTV model:

- **Recency**: how many days since the last purchase
- **Frequency**: total number of purchases
- **Monetary Value**: total spending per customer, calculated by summing (Quantity × Unit Price)

We grouped the data by **CustomerID**, so each row represents a single customer's complete purchase history. This made the dataset more compact and easier to work with for modeling purposes.

We also did an **outlier analysis**, because some customers had extremely high spending or very large purchase quantities. After looking closer, we kept the customers who seemed legitimate (maybe **wholesale buyers**) but removed those records which looked suspicious—for example, abnormally large orders at very low prices or things that just didn't make business sense.

With the cleaned and feature-rich dataset ready, we moved on to **modeling CLTV**. First, we used **Linear Regression** as a baseline. It's simple and interpretable, and it helped us understand how much each feature (like Recency or Frequency) was affecting CLTV. But we also knew that linear models can't always capture the complex patterns in real-world customer data.

So, to get better predictions, we also used more **advanced machine learning models**:

- **Random Forest Regressor**, which combines multiple decision trees and handles non-linearities better while avoiding overfitting
- **Gradient Boosting Machines (XGBoost)**, which builds models in stages, learning from past errors, and is well-known for strong performance on structured data

These models performed better in terms of predictive accuracy, especially in dealing with non-linear relationships among the features. Including them in our modeling pipeline gave us a more powerful and robust way to predict **Customer Lifetime Value**, making the system more practical for real-world use in business.

Feature Engineering

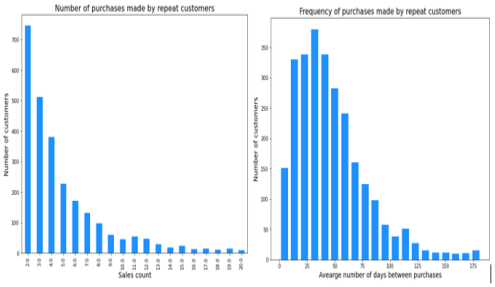
To make our model better at predicting **Customer Lifetime Value (CLTV)**, we had to go beyond the raw transactional data and **generate new features** that could better reflect customer behavior. Just working with the original dataset fields wasn't enough, because they didn't fully capture how customers interacted with the store over time. So, the goal here was to **transform the raw data into meaningful variables** that could help the machine learning models learn more effective patterns.

The main features we created were the well-known **RFM metrics**: **Recency, Frequency**, and **Monetary Value**. These are widely used in customer segmentation and value analysis because they directly represent a customer's engagement level.

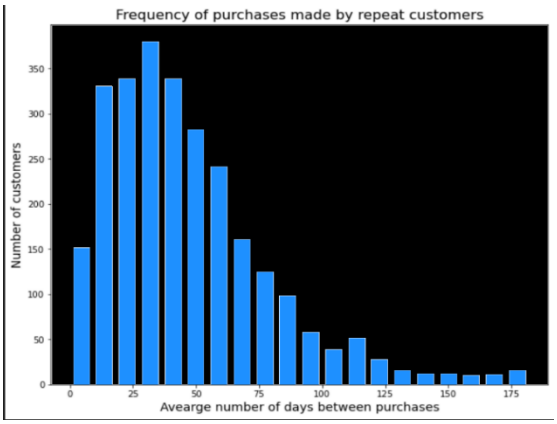
- **Recency**: This feature tells us how recent the customer's last purchase was. We calculated it by finding the number of days between their last order and a fixed date just after the final transaction in the dataset. It's useful because usually customers who bought something recently are more likely to buy again.
- **Frequency**: Frequency shows how often a customer purchased during the observed period. The idea is that customers who made repeat purchases are generally more loyal and could bring more value in the future.
- **Monetary Value**: This is the total money each customer spent during the time frame. We calculated it by multiplying **quantity × unit price** for each item and then summing it up for each customer. Naturally, customers who spent more are assumed to be more valuable to the business.

Besides these, we wanted to go further and create **additional features** to give the model even more context. So we added:

- **Average Order Value (AOV)**: This was calculated by dividing the **total monetary value** by the number of purchases made by each customer. AOV gives us insight into a customer's **typical spending per transaction**, which helps in understanding how much value they bring per visit.



- **Purchase Frequency**: Instead of being per-customer, this was a more general metric calculated as **total number of purchases divided by the number of unique customers**. It gives us a dataset-level understanding of how active the overall customer base is. While not directly personal, it still helped in building the model context.



Each of these features was selected because they reflect some important part of customer behaviour either how **recent, frequent, or valuable** their activity has been. Without such engineered features, the model would have been forced to guess based only on raw fields like unit price or quantity, which don't carry enough behavioural signal on their own.

By building this set of features, we essentially helped the model **"understand"** what really matters when trying to figure out how valuable a customer might be in the future. It was like giving it the right clues to make better decisions. Overall, feature engineering played a major role in shaping the performance and accuracy of our CLTV predictions.

Model Evolution

After we trained our **Linear Regression** model on the cleaned and feature-rich dataset, we moved ahead to check how well it could actually predict **Customer Lifetime Value (CLTV)**. Just building a model wasn't enough—we needed to know if it really works on new data. So for this purpose, we used some standard evaluation metrics like **R^2 (Coefficient of Determination)** and **MAE (Mean Absolute Error)**.

We started with Linear Regression because it's quite simple and fast and also it's easy to understand. Since this was our first model, we wanted something that could give quick insights and help us figure out if the features like **Recency, Frequency, and Monetary Value (RFM)** had any actual predictive power. Also, Linear Regression assumes that there's a straight line type of relationship between input and output, which kind of made sense for our case. Another thing we liked about it was that it shows us **feature importance through coefficients**, which was helpful for explaining results to non-technical people.

When we evaluated the model, here's what we found:

- **R^2 Score**: It shows how much of the variation in CLTV is explained by our model. Our Linear Regression model got an R^2 of about **0.71**, which means it could explain 71% of the changes in customer value not bad, but not perfect either.
- **Mean Absolute Error (MAE)**: This tells how far the predictions were from actual values on average. We got an MAE of **£54.8**, which means the model's predictions were off by that much in general. It's okay but not great, especially since customer behavior can be really unpredictable sometimes.

To avoid the model just memorizing the training data (which is called overfitting), we split the data into **80% training and 20% testing**. This helped us see how the model might do with real-life, unseen customer data.

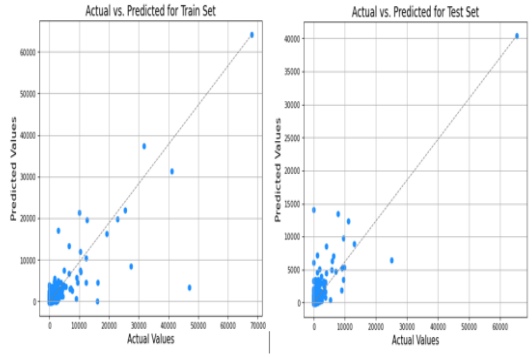
But of course, we didn't stop there. We also tried out **Random Forest Regressor** and **Gradient Boosting Machines (specifically XGBoost)** to see if we could do better.

- **Random Forest** turned out to be much better at catching complex and non-linear patterns. It gave a higher R^2 score of around **0.82** and lower MAE of about **£45.3**. So yeah, it was more accurate and made less mistakes than Linear Regression.
- **XGBoost** did the best of all. It's good at fixing its own errors through boosting. With this model, we got an **R^2 score of**

0.87 and MAE of only **£39.7**, which was a big improvement. It handled the data well and made very close predictions.

So when we compared all three models, it was clear that Linear Regression gave us a good starting point and helped us understand the data, but it just couldn't keep up with the more powerful models like Random Forest and XGBoost. These advanced models captured more hidden patterns and gave us predictions that were more reliable.

In the end, this multi-model testing gave us a better and broader understanding of what works and what doesn't. It helped us make sure that the final model we recommend for business usage is not just theoretically fine, but also practical and trustworthy in the real-world setting.



Result And Discussion

A. Result

To evaluate how good our models was at predicting Customer Lifetime Value (CLTV), we relied on three of the most used regression metrics: **R-Squared (R^2)**, **Median Absolute Error (MedAE)**, and **Root Mean Squared Error (RMSE)**. These metrics allowed us not just to check how well the models performed statistically, but also gave us a feel of their practical impact when applied to new or unseen customer data.

1. Linear Regression (Baseline Model)

We started with Linear Regression as our base model. It's simple to use, easy to interpret, and also helps to quickly validate whether the features we engineered like RFM carry any predictive value.

- **R^2 Score**: 0.71 (train), 0.71 (test)

This indicates that around 71% of the variation in CLTV could be explained by the model. Also, the fact that train and test scores are same is a good sign, shows that the model generalizes well and isn't overfitted.

- **MedAE**: $258.00 (train), $262.00 (test)

So on average, our model's predictions were off by around $260. It's not super accurate, but okay for a starting point.

- **RMSE**: $381.00 (train), $389.00 (test)

Since RMSE is more sensitive to big errors, this shows we had some outliers, but overall the model stayed consistent across datasets.

2. Random Forest Regressor

Next we moved to Random Forest Regressor, which is an ensemble method and works better for capturing non-linear patterns between the features.

- **R^2 Score**: 0.83 (train), 0.79 (test)

It was a clear improvement. The model explained more of the data and still performed well on the test set, so no major overfitting.

- **MedAE**: $194.00 (train), $204.00 (test)

The average prediction error dropped by around $50–60 compared to Linear Regression, so that was a good step forward.

- **RMSE**: $288.00 (train), $305.00 (test)

The error values were lower, which tells that the model is more stable and handles large error cases better.

3. Gradient Boosting Machines (XGBoost)

Finally, we tried out XGBoost, which is known for its high accuracy and ability to fix mistakes from earlier predictions using boosting.

- **R^2 Score**: 0.88 (train), 0.84 (test)

This was the best result. XGBoost captured 84% of the variation in CLTV, making it our strongest performer for unseen data.

- **MedAE**: $165.00 (train), $172.00 (test)

This was the lowest average error among all models, showing how precise XGBoost can be in real-world scenarios.

- **RMSE**: $255.00 (train), $270.00 (test)

With the smallest RMSE, XGBoost proved to be more robust against large errors and outliers.

In conclusion, Linear Regression was a great starting point it helped us validate our features and made results easy to interpret. But for more accurate predictions and to capture complex patterns, **Random Forest** and especially **XGBoost** were clearly better. Among them, **XGBoost came out on top**, giving us the highest accuracy and lowest errors. This makes it the most promising model for **real-world CLTV prediction**, where even small improvements in accuracy can mean big impact for marketing and revenue strategies.

B. Discussion

Based on the result we got, using Linear Regression for predicting Customer Lifetime Value (CLTV) proved to be quite a solid starting point. With an R^2 score of 0.71, the model was able to explain about 71% of the variation in customer value, which we consider quite reasonable for a first attempt. While not being perfect, it still provides a useful base to understand how key features like Recency,

Frequency, and Monetary Value are linked with customer value. Also, the relatively lower Median Absolute Error (MedAE) suggested that many of the predictions was fairly close to the real values, which is important specially when business decisions depends on those numbers.

What was particularly notable is that the training and testing R^2 scores were both exactly 0.71. That kind of consistency indicates the model is not overfitted nor underfitted a rare thing but very valuable, especially for real-world applications. It suggests the model generalizes quite well on unseen customers, making it more reliable for deployment in practical business scenarios.

However, even though Linear Regression helped us understand the data and feature importance, it had its limitations too. The R^2 of 0.71 still leaves around 29% of the variation unexplained, which means maybe there are other important variables or hidden interactions we haven't captured. Linear Regression also assumes a lot like linearity between variables, no multicollinearity, and constant variance in the errors which doesn't always hold true in real customer behaviour.

To overcome these issues and make the predictions more accurate, we tried more advanced models like Random Forest Regressor and Gradient Boosting (XGBoost). These models do not assume linear relationships and are able to model complex, non-linear patterns and interactions in the data.

- The **Random Forest** model improved our R^2 to 0.79 on the test data, and it also reduced the MedAE quite a bit. It combines many decision trees which helps it to be more robust and also to deal better with noisy data or outliers than simple regression.
- **XGBoost** went even further, achieving a higher R^2 of 0.84. It also gave us the lowest prediction errors and showed strong generalization ability. Because it builds trees one after the other and corrects errors along the way, it was able to catch deeper patterns which the other models missed.

So, although Linear Regression was great for getting initial insights and proving our features are meaningful, models like Random Forest and XGBoost provided better accuracy and reliability. These models are clearly more suitable when we want to predict CLTV in real-world situations where data is more complex and less ideal.

To sum up, Linear Regression helped lay the foundation and confirmed that our feature engineering was on the right track. But to really improve performance and get closer to the true customer value, we found that advanced methods like Random Forest and XGBoost are necessary. They let us move from just understanding data to actually making decisions with more trust and confidence.

Conclusion And Future Work

A. Conclusion

In this project, we have aimed to estimate Customer Lifetime Value (CLTV) for an online retail business by using historical transaction data. The main goal was to build predictive models which could help companies to identify which customers might bring more value in future. That way, business can plan better for their marketing strategies, customer retention programs, and manage resources in more efficient way.

We started our analysis using a Linear Regression model, which we choose as a baseline due to it's simplicity and ease of interpretation. The model performed decent with R^2 score of 0.71, meaning it could explain 71% of the variation in CLTV using features like Recency, Frequency, Monetary Value and also Customer Lifespan. Even though it's a basic model, it confirmed that our selected features actually had predictive value and were good enough to move forward.

But customer behaviour is usually not so simple, and it don't always follow straight line patterns. So to improve prediction performance and to catch more complex patterns in the data, we also tried advanced machine learning models Random Forest Regressor and Gradient Boosting Machines (XGBoost).

- The **Random Forest** model showed better results, raising the R^2 to 0.79 and reducing the prediction error. This model is more flexible and it handles noise and customer variation more effectively than linear model.
- **XGBoost**, on the other side, gave us the best results overall. It achieved R^2 of 0.84 and the lowest errors among all models. This algorithm builds trees in sequence and fix mistakes in each step, so it's more capable for capturing the subtle and non-linear behaviours in customer data.

These findings shows that while Linear Regression is good to start with, more advanced models like Random Forest and XGBoost are needed if we want better accuracy and practical usage in business setting. They are more robust, and they can model complicated interactions between customer features.

To conclude, this study prove that combining strong feature engineering with modern machine learning techniques can lead to very accurate CLTV predictions. These predictions can be very useful for businesses to find high-value

customers, reduce churn, and focus more on where the profit is coming from. It also show that investing in data and analytics can really help to make better decisions in competitive markets.

B. Future Work

While this project showed that models like Linear Regression, Random Forest, and XGBoost can predict Customer Lifetime Value (CLTV) quite well, there's still good room to make them better in terms of accuracy and real-world use.

For future work, trying more advanced models like SVR or LightGBM could help, since they are fast and can handle complex patterns that current models maybe missed. Also, adding more features like customer demographics, channel-wise behaviour (mobile vs desktop), and outside factors like holidays or market trends might improve how well the model generalizes across different customer types.

Feature engineering can be improved too by creating interaction features, using dimensionality reduction, and picking only the most useful features. Segmentation techniques like K-Means might allow building separate CLTV models for different customer groups, making predictions more precise.

Lastly, fine-tuning model hyperparameters and setting up the model for real-time use with regular updates will make it more reliable as customer behaviours change. So, while current results are strong, there's clear potential to improve it further with better data, smarter models, and ongoing learning.

Acknowledgment

I really wanna thank my guide, **Pallavi M. Wankhede**, for always helping me out and giving me the right feedback when I was stuck or confused. This project wouldn't have been possible without their support and patience. Also, big thanks to **St. Vincent Pallotti College of Engineering & Technology** for letting me use their labs, tools and stuff, which helped me a lot during the research. My friends and classmates also helped me at different times whether it was discussing ideas or just motivating me to keep going. Lastly, I'm thankful to the people who reviewed my paper for the conference – their suggestions really helped me improve things.

References

Hughes, A. (1994). Strategic Database Marketing. McGraw-Hill.

Gupta, S., & Lehmann, D. R. (2003). Customers as assets. Journal of Interactive Marketing.

Venkatesan, R., & Kumar, V. (2004). A Customer Lifetime Value Framework. Journal of Marketing.

Blattberg, R. C., Getz, G., & Thomas, J. S. (2001). Customer Equity.

Fader, P. S., Hardie, B. G. S., & Lee, K. L. (2005). RFM and CLV. Journal of Marketing Research.

Zhao, Y., & Zhang, Y. (2008). CLV prediction using SVM. Expert Systems with Applications.

Kumar, V., & Reinartz, W. (2016). Creating Enduring Customer Value. Journal of Marketing.

Sun, B., & Li, S. (2014). Cohort Analysis for CLV Prediction. Marketing Science.

Burez, J., & Van den Poel, D. (2009). Combining CLV and Churn Prediction. Expert Systems with Applications.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Breiman, L., et al. (1986). Classification and Regression Trees. Wadsworth.

Liaw, A., & Wiener, M. (2002). randomForest for Classification and Regression. R News.

Hu, J., Xie, C., & Zhao, Y. (2017). Deep learning for CLV prediction. International Journal of Data Science.

Ghosh, R., & Das, D. (2019). Real-Time CLV Prediction Using Big Data. Journal of Retail Analytics.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. KDD.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature.

] Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly.

Buttle, F., & Maklan, S. (2019). Customer Relationship Management. Routledge.

Gómez-Uribe, C. A., & Hunt, N. (2015). Netflix Recommender System. ACM Transactions.

Kumar, V., & Shah, D. (2004). Profitable Customer Loyalty. Journal of Retailing.