IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 6, Ser. 1 (Nov. – Dec. 2025), PP 44-55
www.iosrjournals.org

A Machine Learning Framework for the Derivation of Strategic Business Models from Customer Interaction Data

[1]Prof. Anjali Landge
Research Scholar
Sinhgad Institute of business administrator and research
Savitribai Phule Pune University
ORCID ID: 0009-0000-3693-9253

[2]Dr. Sharada Patil
PhD Guide and Associate Professor
Sinhgad Institute of Business Administrator and Research
Savitribai Phule Pune University

Abstract
Within developing nations such as India, business-to-business (B2B) commerce has historically been backed by personal relationships, trust, and agreements based on credit. Businesses are increasingly relying on telecallers and digital platforms in order to keep up with the ever-increasing amount of online orders, tailor promotions, and maintain connections with their customers. In order to achieve sustainable growth and competitiveness, it is necessary to derive strategic business models from data information on interactions with consumers. This study presents a machine learning framework that combines an ensemble approach of XGBoost with a modified Poisson-Gamma Bayesian model. The purpose of this framework is to anticipate and assess trends in client order placement. Through the strategic integration of predictive analytics, empirical Bayesian learning, and deliberately designed features, this paradigm enhances the accuracy of behavioral forecasting and makes it possible to derive a business model plan that is driven by data. This approach has the ability to reshape business-to-business (B2B) e-commerce in developing nations in terms of value creation, consumer engagement, and operational efficiency. One evidence of this strategy's game-changing potential is the doubling of client order rates, which is a sign of the potential for this strategy to change the game.

Keywords: *Machine Learning Framework, Customer Interaction Data, Strategic Business Models, Empirical Bayesian Approach, XGBoost, Predictive Analytics, B2B E-commerce*

I. INTRODUCTION

As the digital economy has grown, the data that pertains to the interactions that customers have with a company has become an extremely useful resource for businesses who are looking to acquire a competitive advantage. From a customer's interaction data, which may include things like digital engagement, purchase history, credit records, and telecaller logs, you may be able to learn a great deal about the customer's routines and preferences [1]. Personal ties, mutual trust, and established credit conditions have been integral to business-to-business transactions in developing countries such as India for a considerable amount of time [2]. On the other hand, all of this communication is now taking place online, which results in the accumulation of mountains of data that machine learning may utilize to enhance business models. This is because e-commerce platforms are growing more popular.

It is possible for businesses to make advantage of the powerful predictive capabilities of machine learning (ML) in order to enhance their operations, develop new business models, and anticipate the behavior of prospective customers. Previous research has shown that machine learning has the potential to be beneficial in forecasting customer turnover, demand, and customer segmentation [3, 4]. Recent recommendations have been made for frameworks that would enable businesses to go from descriptive analytics to prescriptive insights by merging machine learning with strategic decision-making [5]. The business-to-business e-commerce market has experienced improvements in order placement estimates and simplification of customer engagement strategies as a result of the use of machine learning technologies [6].

Despite the fact that there exist predictive models for customer behavior, the process of deriving strategic business models from customer contact data is a neglected field of research. Prediction is the exclusive topic of the majority of investigations. A large amount of research has been conducted on the prediction of customer attrition [7] and customer lifetime value [8], but there is a scarcity of examples that demonstrate how predictive

data may be used to develop innovation frameworks for business models. It is of the utmost importance for Indian e-commerce enterprises to maintain the trust of their clients; nevertheless, there is a dearth of research that investigates how to transform the copious amounts of interaction data that are generated into data-driven, long-term business strategy [9]. An all-encompassing machine learning framework is being developed because there is a pressing need to close the gap that exists between data-driven projections and the process of deriving business models.

This work provides a contribution to the area in both a theoretical and practical sense by proposing a paradigm that is driven by data and that integrates the creation of strategic business models with predictive analytics. A methodical approach is provided by the framework for practitioners to use in order to transform data gathered from encounters with customers into long-term growth strategies. In the realm of academia, it contributes to the expanding body of work on data-driven business model innovation. This is particularly relevant in the context of developing nations, where traditional relationship-based trading is being replaced by online markets.

OBJECTIVES

1. To create a machine learning framework that can accurately forecast consumer behavior by analyzing data from customer interactions.
2. To use predictive analytics to determine key business model elements like price, customer interaction, and segmentation.

II. RESEARCH METHODOLOGY

The purpose of the machine learning approach that is provided in this research is to derive strategic business models from interaction data in order to anticipate the purchasing behavior associated with customers. There are three pillars that support the framework, and these are engineering features, predictive modeling, and the process of developing a strategic business model [10].

Feature Engineering

We break down customer contact data (including calls, purchase history, and product interaction) into 48-hour periods so that we may spot patterns that emerge over the course of time via this process. Indicators that may be put into action about the intention to make a purchase are offered by elements such as the frequency of interactions, product replenishment cycles, and order history [11].

Predictive Modeling

Two models are used that work together:

XGBoost: An algorithm for estimating the likelihood of a purchase based on customer characteristics that takes into account non-linear correlations [12] and is tree-based.

Considering a dataset that is shown as $D = \{(x_i, y_i); i \in \{1, 2, 3 \ldots n\}; x_i \in \mathbb{R}^w, y_i \in \mathbb{R}\}$. Every single one of the F samples has the properties of F, and the label is F. One possible way to represent the prediction on sample b is as follows:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i) \qquad (1)$$

Where a CART is denoted by the symbol Zk and K is the total number of trees that are used in the process of prediction. Due to the fact that the technique is iterative, the objective function is supplied by at a certain iteration, which we will refer to as M.

$$\text{minimize} \sum_{i=0}^{n} L\left(y_i, \widehat{y_i}^{(t-1)} + f_t(x_i)\right) + \omega(f_t) \qquad (2)$$

The canonical form of the tree is denoted by (Zb), whereas the predictions from previous trees are denoted by $\boldsymbol{\xi}bi$ (t−1), and $\boldsymbol{\nu}$ is understood to be an appropriate loss function. Data on processed features and labels that indicate whether or not a customer has placed a minimum order inside the desired time range are the inputs that this model takes into consideration.

Modified Poisson-Gamma (MPG) Model: A method for modeling recurring purchases that makes use of Bayesian statistics and takes into account behaviors that are dependent on time as well as the frequency with which certain products are purchased during different time periods.

A Bayesian estimate of the customer's rate of repeat purchases may be obtained by integrating the prior distribution with the customer's own purchase history. The following methods can be used to get this estimate:

$$\lambda_{A,C}(t) = \frac{k + \alpha_A}{t + \beta_A}, t > 0 \quad (3)$$

the number of times that customer s has purchased product s; the amount of time that has elapsed since customer s made their first purchase of product s; and the shape and rate parameters of the gamma prior of product s, namely βA and βA, respectively. On the other hand, when we construct a buying activity, let's assume that a customer C bought a product A, K^{th} time after the passage of X time units after the first purchase of the product. A statement of the purchase rate is made both before and after a transaction.

$$\lambda_{A,C}(t) = \begin{cases} \frac{k-1+\alpha_A}{(t-\epsilon)+\beta_A}, & \text{Before purchase} \\ \frac{k+\alpha_A}{(t+\epsilon)+\beta_A}, & \text{After purchase} \end{cases} \quad (4)$$

Not only is the time interval ϵ relatively short, but it is also situated in close proximity to the time b at which the transaction was carried out. When we focus on the exact time of purchase ($\epsilon \to 0$), we see that the buy rate is greater immediately after the purchase than it was before the purchase. This is the correct interpretation of the data. This problem has been overcome by the implementation of a new version of the PG model.

The maximum likelihood estimates of the purchases made by repeat customers are fitted to the product-specific gamma distributions in order to identify the parameters of the product-specific gamma distributions by experimental means. As soon as the parameters have been evaluated, the computation of the parameter μ is carried out.

$$\lambda_{A,C}(t) = \begin{cases} \frac{k+\alpha_A}{t_{purch}+2*|t_{mean}-t|+\beta_A}, & 0 < t < 2 * t_{mean} \\ \frac{k+\alpha_A}{t+\beta_A}, & t \geq 2 * t_{mean} \end{cases} \quad (5)$$

$$P_{A,C}(m) = \frac{\lambda_{A,C}^m \exp(\lambda_{A,C})}{m!} \quad (6)$$

Hence, the following is the fundamental likelihood given in terms of the raw MPG output:

$$P_{A,C}(\text{m>=1}) = 1 - P_{A,C}(m = 0) \quad (7)$$

One way to describe a customer's ultimate purchase likelihood during the following 48 hours is:

$$P_C = \frac{1}{1 + \exp(-(W^T \mathbf{x} + b))} \quad (8)$$

When stacking is used, the accuracy of forecasts is significantly improved. For the purpose of providing calibrated purchase probabilities at the consumer level, we combine the outputs of XGBoost and MPG by using Logistic Regression as a meta-learner [13].

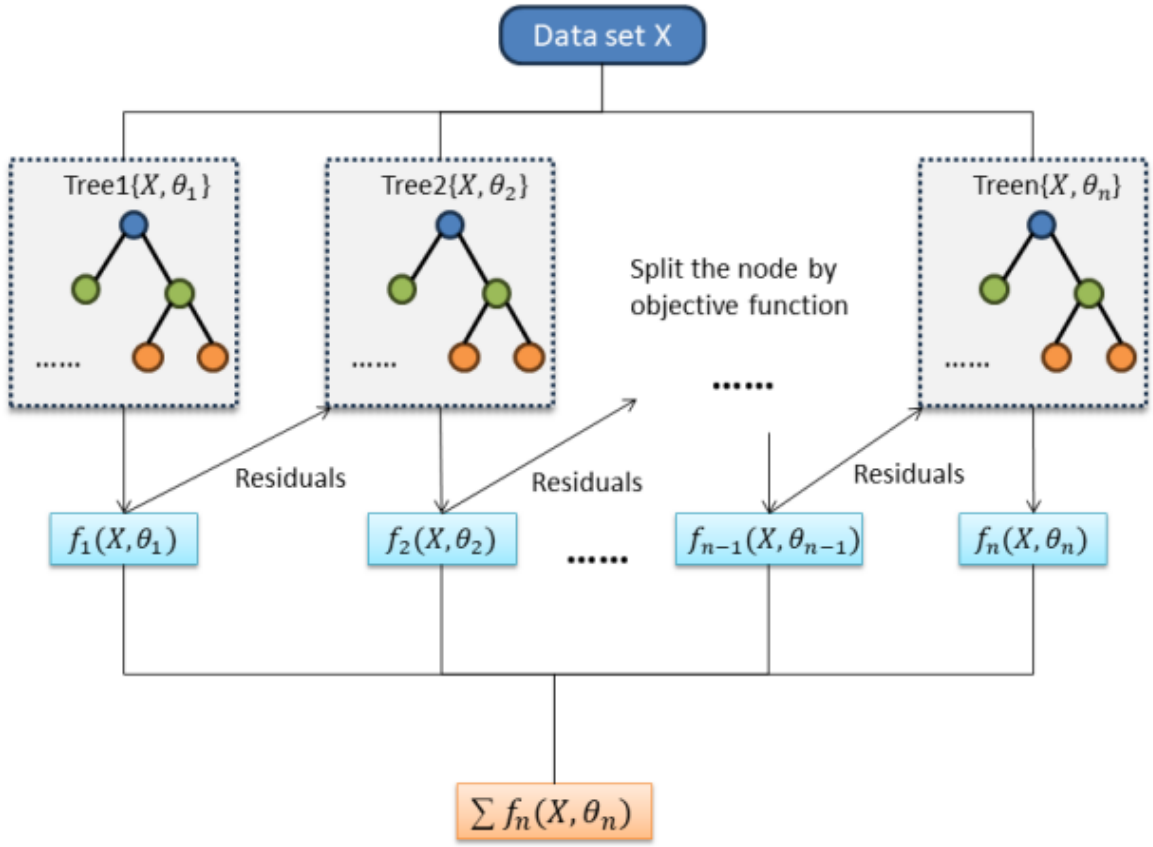


Figure 1. The XGBoost model's data flow using f trees.

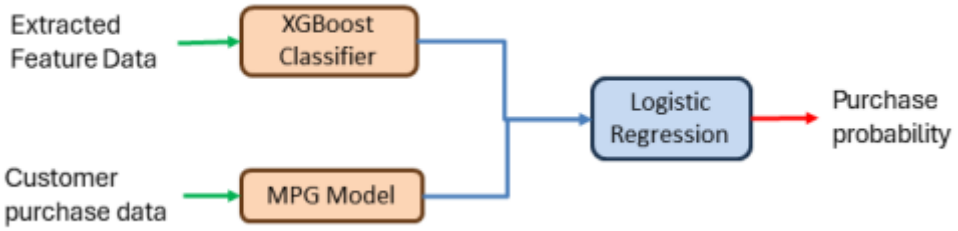


Figure 2: Using logistic regression as a meta-learner in a stack of XGBoost and MPG models

Strategic Derivation of Business Models

We combine the expected probability across all of our client categories in order to find:

- High-value buyers (frequent, predictable orders),
- Opportunistic buyers (irregular but high-margin orders), and
- At-risk buyers (low purchase probability) [14].

Through the process of mapping these insights into strategic business model components such as improved inventory, targeted promotions, and tailored telecaller allocation, this merges predictive analytics with management strategy.

III. RESULT

All of the models that were used in this investigation were trained in a Python environment by making advantage of the computational resources that were made available by the Microsoft Azure framework. For the purpose of accelerating this computationally heavy activity, we make use of a powerful computing cluster consisting of Azure D8asv4 computers. Each of our machines is equipped with 32 gigabytes of random access memory (RAM), top-of-the-line solid-state drives (SSD), and third-generation AMD EPYCTM 7763v processors that can reach speeds of up to 3.5 gigahertz.

Feature Dataset

For the purpose of this research, data that is wholly owned by Udaan.com and that has been meticulously retrieved from their large data warehouse is used. This dataset contains feature data that is two days behind the binary labels in terms of the temporal lag that exists between the two. To be more specific, a label that represents a '1' indicates that a customer has placed an order for at least one item over the last two days, while a label that represents a '0' shows that they have not done so. In addition, descriptive data for each of the characteristics that were included in our sample are shown in Table 1.

Table 1. Numerical Features Used by the XGBoost Algorithm and Their Descriptive Statistics

Feature	Mean	Std. Dev.	Min	Median	Max	Description
l1wo	1.04	2.39	0	0	103	Orders in last 1 week
l2wo	1.88	3.70	0	0	151	Orders in last 2 weeks
l3wo	2.65	4.93	0	1	174	Orders in last 3 weeks
l4wo	3.38	6.14	0	1	206	Orders in last 4 weeks
l5wo	4.11	7.35	0	1	224	Orders in last 5 weeks
l6wo	4.82	8.54	0	2	244	Orders in last 6 weeks
l7wo	5.53	9.71	0	2	268	Orders in last 7 weeks
l8wo	6.24	10.89	0	2	327	Orders in last 8 weeks
DSLO	69.66	78.71	0	20	182	Days since last order
l1w_fam	0.44	1.15	0	0	84	FAM visits in last 1 week
l2w_fam	0.77	1.80	0	0	130	FAM visits in last 2 weeks
l3w_fam	1.10	2.41	0	0	132	FAM visits in last 3 weeks
l4w_fam	1.41	3.01	0	0	133	FAM visits in last 4 weeks
call_engage_frac	0.54	0.45	0	0.87	1	Fraction of time with human agent
bot_engage_frac	0.00	0.01	0	0	1	Fraction of time with chatbot
credit_ratio	0.05	0.32	-30.04	0	2.63	Ratio of credit available
promise_diff	4.69	10.03	-3	1	15	Delivery deviation (days)
max_promise_diff	6.52	9.41	-3	4	16	Maximum delivery deviation
min_promise_diff	3.07	11.89	-3	0	15	Minimum delivery deviation
a2c_amount	141.00	32020.15	0	0	5,142,694	Total add-to-cart value
l7d_a2c	106.30	31961.60	0	0	5,142,694	Most frequent item in last 7d
l7d_a2c_2	63.08	31279.51	0	0	5,142,694	2nd most frequent item in last 7d
num_app_opens_l7d	11.45	15.41	0	7	2,223	App opens in last 7 days
num_listing_views_l7d	4.83	7.75	0	3	1,096	Product listing views (7d)
num_a2c_l7d	1.42	2.42	0	1	91	Items added to cart (7d)

The fact that the initial real-time dataset that was extracted had a considerable class imbalance is something that definitely needs to be stressed. A response that we utilized was an undersampling method, which includes eliminating data points from the majority class in a random fashion. This was done in order to correct the imbalance that we had. Examine Table 2, which provides information on the change, to get an idea of the characteristics of the labels both before and after the undersampling. Tomeklinks [15] is an alternate strategy that primarily seeks to improve class segregation by deleting samples from the majority class that are situated near the minority class. This is done in order to achieve the aforementioned results. In contrast to the circumstances we find ourselves in, this method gives rise to a relatively little numerical downsampling of the dominating class. Because the raw data contains a huge minority population (about 80:1), it is required to do undersampling on a much larger scale. The use of neighbor comparison was an innovative strategy that was proposed by [16]. With the help of our projection, we exclude adjacent neighbors who do not correspond to the others. Despite the fact that this preserves the integrity of vital data, it is very computer-intensive. We came to the conclusion that the random undersampling method would be the most suitable option to pursue in this particular scenario. This conclusion was reached after considering the parameters that were provided as well as the size of our majority sample. When the dataset is undersampled, the huge number of samples that are produced assures that the dataset will stay intact while simultaneously minimizing the class imbalance. This is something that should be mentioned.

Table 2. Quantity of Pre- and Post-Sampling Samples in Each Class

Class	Before Sampling	After Sampling
0's	2,767,563	69,144
1's	34,572	34,572
Total	**2,802,135**	**103,716**

Activation_type, which indicates the type of credit, day_of_week, which indicates the specific day when customers placed the most orders, and brand_1, brand_2, and brand_3—the names of the most frequently purchased brands—are some of the variables that are included in our dataset. In addition to the numerous

categorical features, our dataset also contains other variables. Without the presence of these category criteria, we would not have been able to conduct our analysis. The top fifteen qualities are shown in Figure 3, which is arranged in descending order of relevance. These characteristics were chosen based on their relative value in lowering impurity when split. It is essential that we bring to your attention the fact that we have observed a number of components that are really fascinating.

The bot_engage_frac feature, which displays the percentage of a client's total customer support time that was spent engaging with bots, is one of the top 15 features. Despite the fact that this quality may have a negative link with our objective, it does illustrate the huge influence that the quality of customer service can have on businesses [17]. After that, we separated the dataset into three sections, with eighty percent going to training, five percent going to validation, and fifteen percent going to testing. This was done to ensure that resources were distributed appropriately. Both the validation subset and the training subset were utilized solely for all training operations, including hyperparameter fine-tuning. The training subset was used exclusively for all training activities. The best-fitting model that was produced via this approach was used to make predictions on the test set. This ensured that an objective evaluation of the model's performance was carried out.

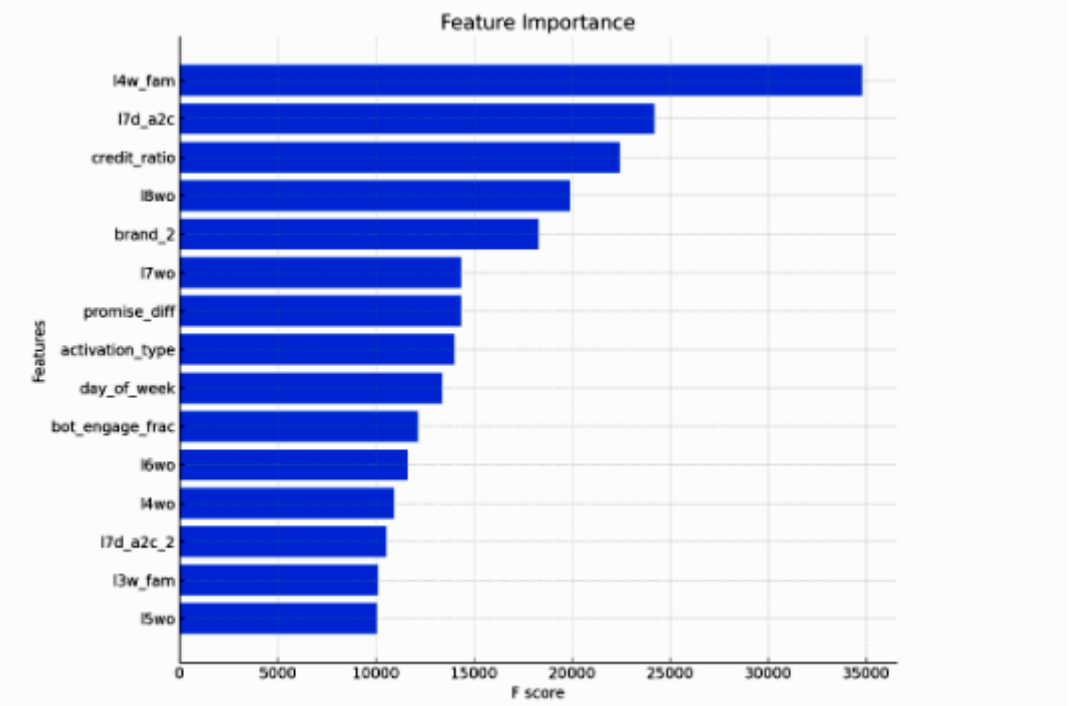


Figure 3: The fifteen most important input characteristics derived from the solo XGBoost prediction, ordered from most important to least important.

Performance of XGBoost alone

The XGBoost model is driven for the most part by two key components: the feature data that has been suitably tagged and the feature data itself. Identifying the hyperparameters that increase model performance and are most compatible with our dataset is accomplished via the use of a thorough 5-fold grid search cross-validation approach. For this all-encompassing strategy, it is necessary to investigate and alter hyperparameters in a systematic manner. These hyperparameters cover subjects such as the learning rate, maximum tree depth, maximum number of leaves per tree, and total number of trees. Additionally, in order to better investigate the use of features across a variety of levels, we have included a complete strategy [18]. This includes sampling columns at various tree levels, while the tree is being constructed, and even as nodes are being torn apart. For the purpose of ensuring that our model is capable of reaching our business objectives, we have modified the eval metric parameter such that it focuses on the region under the precision-recall curve, which is pertinent to the categorizing task that we are doing. It is important to keep in mind that our training program is adaptable enough to accommodate the ever-evolving data requirements of our customers. Training our model takes place once every three weeks, in contrast to the fact that customer information is updated every single day. The predictive model, on the other hand, is put to good use once every two days in order to deliver insights in real time.

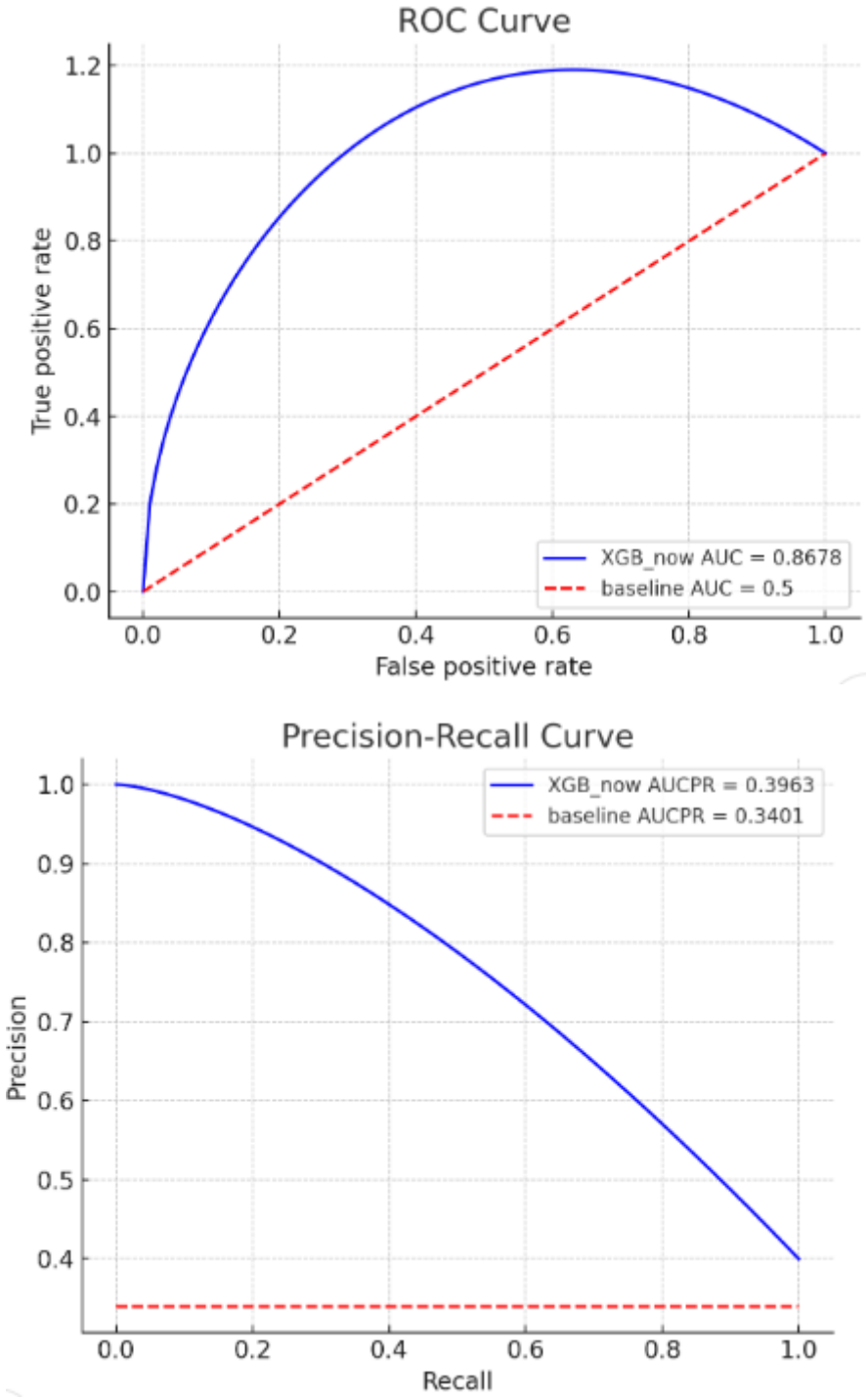


Figure 4: The area under the curve for the XGBoost model's ROC and AUC-PR

Through the use of the processing capabilities of the cluster that was previously described, the whole training operation is successfully completed in a total of three hours. As a result of this reduced process, our prediction model is able to continue to be accurate and effective. Additionally, this methodology enables us to be flexible and responsive to changing consumer behaviors. With regard to the performance of XGBoost, the theoretical conclusions are shown in Figure 4. The AUC-ROC for our study is 0.8678, while the AUC-PR for our study is 0.3963.

Performance of the stack model

In order to fine-tune the hyperparameters of the stack model during its separate training cycle, we use a process known as 5-fold grid search crossvalidation. At this point in time, our primary focus is on fine-tuning the hyperparameters of the XGBoost and Logistic Regression models; the MPG model is being left alone. It is

important to note that the MPG model has already been trained independently prior to this phase, and it will continue to perform in the same manner while the hyperparameters of the stack model are being modified.

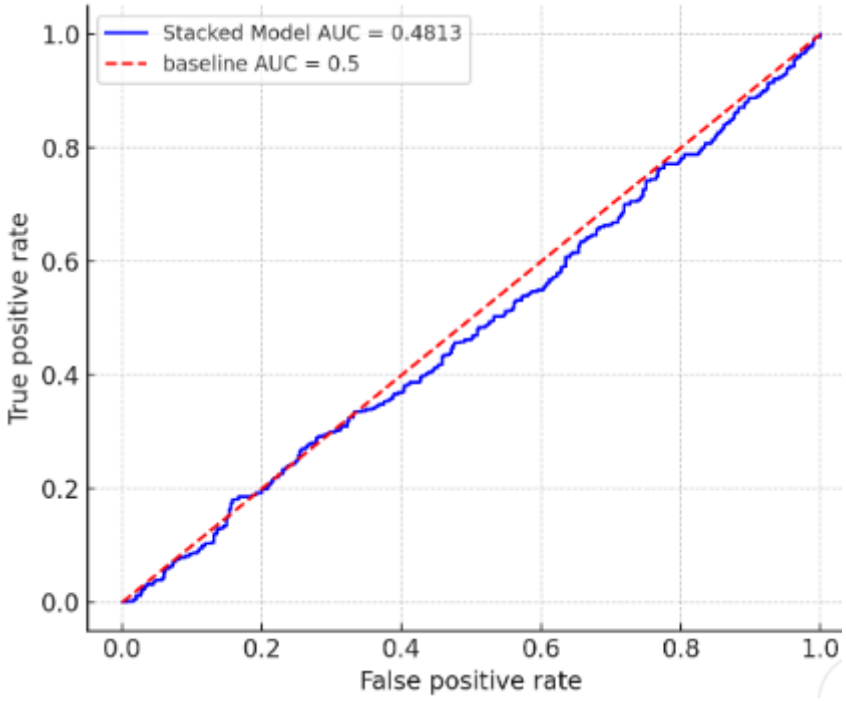
Table 3. Features Derived from MPG and Their Descriptive Statistics

Feature	Mean	Std	Min	Max
#p_greater than 0.5	0.01	0.18	0	55
#p_greater than 0.75	0	0.06	0	12
#p_greater than 0.9	0	0.02	0	7
p_mean	0.01	0.03	0	1
p_std	0.04	0.06	0	1.53
p_sum	0.01	0.06	0	5.26

Preventing over-fitting is the objective of our experiments with the regularized version of Logistic Regression, which we are doing. In order to implement an elastic penalty, we make use of the strength as a hyperparameter. Because the number of samples we had was far more than the number of features we had, we decided to employ the Newton-Cholesky solver for quadratic and finer convergence. It took the computer cluster around four hours to complete the task in its entirety. Figure 5 depicts the curves that are connected to the stacked model at various points. The AUC-ROC score that we get is 0.8753, and the AUC-PR score that we obtain is 0.4262 [19]. The contributions of XGBoost and MPG features are evaluated with the use of the meta learner's softmax normalized weights in order to establish the relative significance of each of these contributions. A breakdown of the percentages contributed by each is shown in Figure 6. Feature significance is defined as the ratio of the total weights of normalized MPG features to the total weights in the logistic regression model. This ratio is used in the context of MPG statistics.

Table 4. Comparative Analysis of the Stacked Approach and XGBoost

Models	Accuracy	F1 Score	Precision	Recall	AUC	AUC-PR
XGBoost	0.79	0.76	0.77	0.75	0.8678	0.3963
Stacked	0.80	0.77	0.78	0.76	0.8753	0.4262



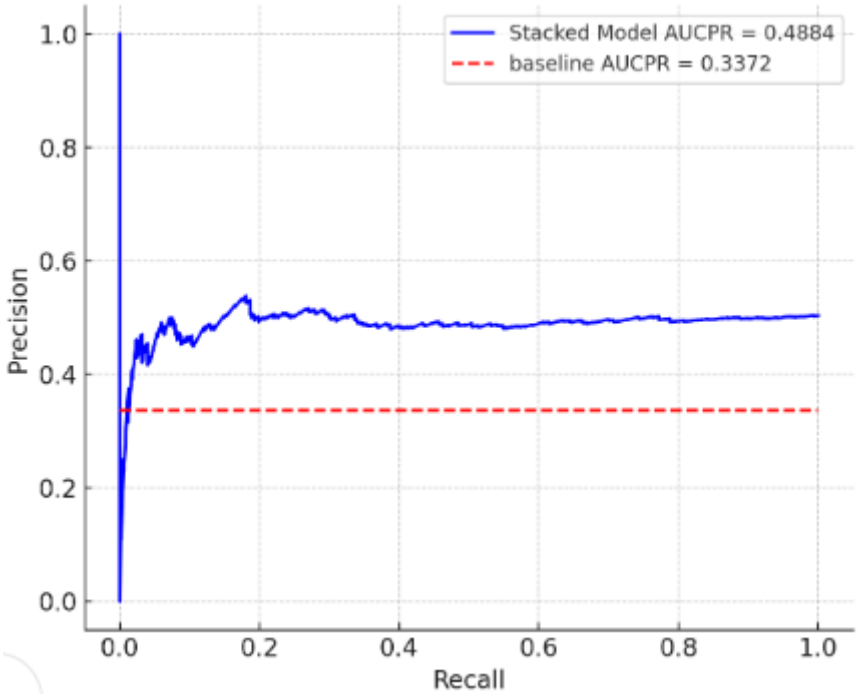


Figure 5: The Stacked model's AUC-ROC and AUC-PR calculations

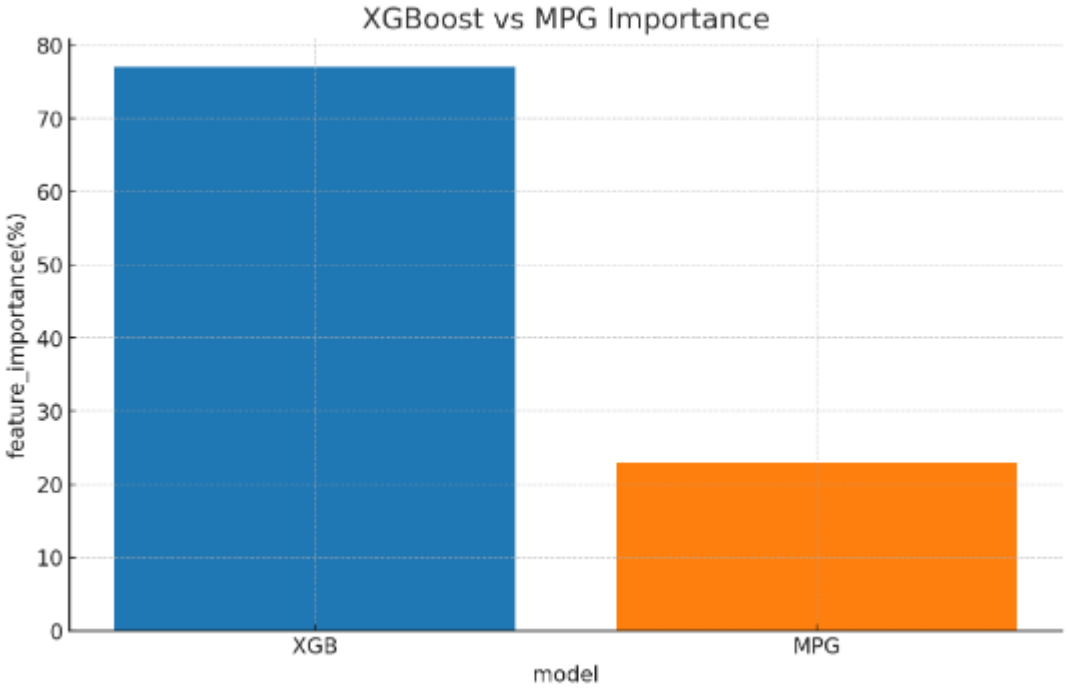


Figure 6: Analysis of the final stacked model's feature significance using the XGBoost and MPG models

The results of our comprehensive comparison of the two approaches are shown in Table 4. According to our results, the performance of the layered model is superior to that of the single XGBoost model. A relative improvement of 7.5% in the AUC-PR score and a relative improvement of 1.3% in the F-1 score are both shown by our study at this time. In the future, we will see that the higher performance of the stacked model becomes more visible and relevant in our real-world commercial application, despite the fact that these numerical advantages may first appear to be insignificant. For the purpose of providing a robust idea of classification, we employ a threshold of 0.5 on the chance of making a purchase. The probability of a sample is assigned a value of one if it is more than half, and a value of zero if it is less than half. True Positives, True Negatives, False Positives, and False Negatives are each represented by four different variables: True Positives, True Negatives, False Positives, and False Negatives, respectively. In classification jobs, the most important metrics that are used are

the F-1 Score, Accuracy, True Positive Rate (Recall), and Precision levels. These may be determined by using the formula that is shown below:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \quad (12)$$

The ability to compare is made possible by implementation in real-time business. Up to this point, we have provided a comprehensive summary of the theoretical results that our models have generated. However, from a commercial aspect, our primary aim is to examine how these techniques truly function, particularly when it comes to customer retention, which is a big element in growing sales revenue. The purpose of this study is to determine whether or not these models have the potential to boost sales revenue via the customisation of unique offers for each individual customer [20]. In order to do this, we take these models from the world of theory and put them into production so that telecallers may design call prompts for a specific group of customers. Following this, we compare the results of these projects to those of existing approaches that have been created in the past.

During the course of this experiment, which will last for a period of three weeks, a broad range of customer types will participate. Before the experiment begins, both a solo XGBoost model and an MPG model are trained according to their respective specifications. After that, we fine-tune the hyper-parameters of the new XGBoost and Logistic Regression models, but we do not make any changes to the MPG model. Following that, we train the stacked model on its own. The fact that Equation 7 is dependent on time is something that should be kept in mind. For this reason, we retrain the MPG model every two days in order to ensure that it remains in sync with the ever-changing data from consumers. The solo XGBoost and stacked models, on the other hand, make use of training less often than the other models. Given that our client database is constantly evolving, this arrangement enables us to get projections that are based on the most recent data every two days, which is a fantastic feature. The predictive models provide probabilities about the likelihood of a customer making a purchase. These probabilities indicate the likelihood of a consumer making a purchase. The next step that we will do is to categorize our clients into several probability categories. Consumers who have a probability that falls between 0.9 and 1.0, for instance, may be seen as having a significant amount of potential, and so on. According to the findings of the comprehensive analysis of the performance of these models in the real world, Table 5 displays the results.

Table 5. Number of Calls Made by Each Model

Model	Probability Range (%)	# Customers Called	# Orders in 48 Hours	% Orders
Non-Model	90–100	9	1	11.11
	80–90	165	4	2.42
	70–80	256	13	5.08
Total	–	430	18	4.19
XGBoost	90–100	11	3	27.27
	80–90	157	10	6.37
	70–80	291	25	8.59
Total	–	459	38	8.28
Stacked	90–100	12	5	41.67
	80–90	149	18	12.08
	70–80	287	37	12.89
Total	–	448	60	13.39

As a result of the restricted bandwidth and personnel resources, we have made the decision to focus on business customers whose possibility of making a transaction is more than seventy percent when it comes to telecalling. Because of this astute move, we will be able to decrease the complexity of our telemarketing operations and concentrate on customers with high potential. Particularly noteworthy is the fact that the manual procedure, which was used prior to the implementation of this methodology, had an average conversion rate of 4.19 percent. An approach that is fundamental and fundamentally simple is the manual method that the organization utilizes. It does not possess any of the powers of artificial intelligence and instead relies on primitive heuristics, such as

arranging customers in descending order based on certain characteristic data. On the other hand, the conversion rate increases to 8.28% when XGBoost is used by itself. The exceptional 13.4% conversion rate is the point at which the stacked model achieves its highest level of performance. Through the use of information at both the feature level and the score level, we are able to improve our consumer targeting strategies. The advantages of using a layered design are brought to light by this major increase.

IV. CONCLUSION

The capacity to accurately estimate the purchasing patterns of customers is an essential component in the process of developing strategic business models for online retailers. Through the rigorous examination of data derived from customer interactions, businesses have the opportunity to discover valued consumers, enhance engagement strategies, and develop policies that are driven by data. It was proposed that the comparative framework for this research include a hybrid stacking model that incorporates machine learning and an empirical Bayesian approach. Additionally, a strategy that is solely based on machine learning, known as XGBoost, was also included. Based on the findings of our study, the layered architecture is more applicable to real-world situations and has superior predictive potential compared to the single XGBoost algorithm. To be more exact, it beat traditional manual telecalling approaches by a factor of three, and it exceeded XGBoost by a factor of eight, in terms of the proportional increase in aggregate conversion rate (from eight percent to thirteen percent). These results provide evidence that predictive modeling has the potential to serve as a tool for strategic decision-making. For instance, it may make it easier to simplify advertising campaigns, conduct targeted telemarketing, and efficiently allocate credit. The significance of this research lies in the fact that it provides the framework for future strategic business model deductions based on data on customer interactions. It is possible that further research might widen the framework to include product-level predictions by integrating recommendation systems and advanced parameter-estimated models. The use of real-time interaction data that is enabled by the Internet of Things is one method that may be utilized to enhance the strategic agility and serviceability of e-commerce companies functioning in developing nations such as India. With the help of this data, it may be possible to develop adaptive decision-making systems that are based on deep learning.

REFERENCES

[1]. Manyika, J., et al. (2021). The future of data-driven business: Strategic applications. McKinsey Global Institute Report.
[2]. Kshetri, N. (2018). 1 Big data's role in expanding access to financial services in India. IT Professional, 20(2), 54–59.
[3]. Verbeke, W., Martens, D., & Baesens, B. (2012). Social network analysis for customer churn prediction. Applied Soft Computing, 14(3), 431–446.
[4]. Kumar, A., et al. (2019). Predicting consumer behavior with machine learning: Applications in e-commerce. Information Systems Frontiers, 21(5), 1125–1140.
[5]. Wamba, S. F., et al. (2020). Big data analytics and firm performance: Effects of dynamic capabilities. Journal of Business Research, 131, 355–365
[6]. De, A., Singh, A., & Patel, P. (2024). A Machine Learning and Empirical Bayesian Approach for Predictive Buying in B2B E-commerce. arXiv preprint arXiv:2403.07843.
[7]. Ansari, A., Jedidi, K., & Dube, J. P. (2018). Customer-base analysis in a discrete-time noncontractual setting. Marketing Science, 37(4), 601–620.
[8]. Kumar, V., & Reinartz, W. (2016). Creating Enduring Customer Value. Journal of Marketing, 80(6), 36–68.
[9]. Jain, R., & Kumar, N. (2022). Digital transformation in Indian B2B commerce: Challenges and opportunities. Journal of Business Research, 148, 320–330.
[10]. Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. 2018. Buy It Again: Modeling Repeat Purchase Recommendations. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (London, United Kingdom) (KDD '18). Association for Computing Machinery, New York, NY, USA, 62–70. https://doi.org/10.1145/3219819.3219891
[11]. Tianqi Chen and Carlos Guestrin. 2016. XGBoost. ACM. https://doi.org/10.1145/ 2939672.2939785
[12]. Donald G. Morrison and David C. Schmittlein. 1988. Generalizing the NBD Model for Customer Purchases: What Are the Implications and Is It Worth the Effort? Journal of Business & Economic Statistics 6, 2 (1988), 145–159. https: //doi.org/10.1080/07350015.1988.10509648
[13]. R. Dunn, S. Reader, and N. Wrigley. 1983. An Investigation of the Assumptions of the NBD Model as Applied to Purchasing at Individual Stores. Journal of the Royal Statistical Society. Series C (Applied Statistics) 32, 3 (1983), 249–259. http://www.jstor.org/stable/2347947
[14]. Gary L. Grahn. 1969. NBD Model of Repeat-Purchase Loyalty: An Empirical Investigation. Journal of Marketing Research 6 (1969), 72 – 78. https://api. semanticscholar.org/CorpusID:167983897
[15]. Ivan Tomek. 1976. Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics SMC-6, 11 (1976), 769–772. https://doi.org/10.1109/TSMC. 1976.4309452
[16]. Dennis L. Wilson. 1972. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics SMC-2, 3 (1972), 408–421. https://doi.org/10.1109/TSMC.1972.4309137
[17]. Jing Liu, Yingnan Zhang, Jin Hu, Xiang Xie, and Shilei Huang. 2017. A FastTraining Approach Using ELM for Satisfaction Analysis of Call Centers. In Proceedings of the 2017 International Conference on Machine Learning and Soft Computing (Ho Chi Minh City, Vietnam) (ICMLSC '17). Association for Computing Machinery, New York, NY, USA, 143–147. https://doi.org/10.1145/3036290.3036313
[18]. Chi On Chan, H. C.W. Lau, and Youqing Fan. 2020. Implementing IoT-Adaptive Fuzzy Neural Network Model Enabling Service for Supporting Fashion Retail. In Proceedings of the 4th International Conference on Machine Learning and Soft Computing (Haiphong

City, Viet Nam) (ICMLSC '20). Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3380688.3380692

[19]. Suvodip Dey, Pabitra Mitra, and Kratika Gupta. 2016. Recommending Repeat Purchases Using Product Segment Statistics. In Proceedings of the 10th ACM Conference on Recommender Systems. Association for Computing Machinery, 357–360. https://doi.org/10.1145/2959100.2959145

[20]. Suvodip Dey, Pabitra Mitra, and Kratika Gupta. 2016. Recommending Repeat Purchases Using Product Segment Statistics. In Proceedings of the 10th ACM Conference on Recommender Systems. Association for Computing Machinery, 357–360. https://doi.org/10.1145/2959100.2959145