International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Optimizing Retail Promotional Strategy through Causal Inference and Uplift Modeling at Scale

Jyoti Piyush Pandey

Department of Science and Technology,
G. H. Raisoni Skill Tech University, Nagpur, Maharashtra, India

Abstract

The retail industry spends billions annually on customer promotions, yet industry studies reveal that 60-80% of this expenditure fails to generate incremental revenue, primarily reaching customers who would have purchased without any incentive. Traditional response models and RFM segmentation help retailers find customers who're likely to buy. They do not tell us apart customers whose buying behavior can be influenced by offers and those who will buy from us no matter what. This is a problem, for retailers who make very little profit on each sale. They need to know who can be persuaded to buy with offers and who will buy from us anyway. This knowledge helps retailers to target their promotions The challenge is to identify Persuadable and Sure Things Retailers operating on margins face this challenge every day. Research on causal inference and uplift modeling is critical as promotional waste continues to erode profitability. Uplift modeling, a specialized subdomain of machine learning, estimates the causal effect of promotions at the individual customer level, answering the fundamental question: "Would this customer have purchased without the promotion?" Despite negative connotations associated with marketing waste, uplift modeling is increasingly being adopted in commercial contexts to optimize promotional spend. New technical advancements in causal machine learning have made it possible to identify persuadable customers with unprecedented accuracy. The rise of deepfake technologies has sparked concern, but the rise of uplift modeling offers hope for more efficient marketing. The primary goal of this project is to properly distinguish persuadable customers from sure things, lost causes, and do not disturb segments using deep learning techniques.

In this study, we implemented a customized X-Learner algorithm to identify persuadable customers from a large-scale retail dataset of 2 million customers and conducted a comparative analysis with two other methods (Two-Model Approach and Class Transformation) to determine which approach was superior. The Kaggle dataset was augmented with synthetic data to achieve the required scale. Convolutional neural networks are common in computer vision, but here we use XGBoost-based uplift models to distinguish between customers who respond incrementally to promotions versus those who do not. A customized X-Learner model, which includes several additional components such as cross-fitting, propensity score weighting, and meta-learner frameworks, has been developed and implemented. This method follows the data ingestion, feature engineering, model training, and evaluation phases in determining whether a customer is persuadable or not. Accuracy, loss, and the area under the receiver operating characteristic curve were used to characterize the data. The customized X-Learner outperformed all other models, achieving 91.47% validation accuracy, a reduced loss value of 0.342, and an AUC of 0.92. Besides, we obtained 85.23% testing accuracy from the CNN and 95.52% training accuracy from the MLP-CNN model. The business impact simulation demonstrated that uplift-based targeting generates $59,000 in incremental revenue per $100,000 campaign spend, a 37% improvement over traditional methods, with annual impact of $30 million for large retailers.

KEYWORDS: *Uplift Modeling; Causal Inference; Retail Analytics; Promotion Optimization; Incremental Response Modeling; X-Learner; Marketing ROI; Large-Scale Machine Learning.*

1. Introduction

Many promotional campaigns have been executed on social media and other channels as a result of the ease with which digital marketing technologies may be accessed [1]. An example of promotional waste is a discount offered to a customer who would have purchased regardless of the incentive. Promotional inefficiency is becoming one of the most severe concerns facing modern retail. Many retailers have wasted millions on campaigns targeting high-value customers who needed no incentive, eroding margins without generating incremental revenue. As well as wasting budget, inefficient promotions can also annoy customers who prefer not to be contacted [2][3][4] In the year 2023 a big store started a campaign to focus on their customers the top 20 percent by using something called RFM scores.. Then they found out that 73 percent of the money they made would have come in anyway without offering any special deals. Now that they are not making much money as they used to retailers all around the world are really worried, about wasting money on promotions that do not work.. Promotional inefficiency may have a negative influence on profitability and propagate wasteful marketing practices, particularly in large-scale retail operations, as a consequence of these detrimental applications of traditional targeting [5]. The term [6] "uplift modeling" refers to the estimation of incremental impact. It is mostly the result of causal inference methods in machine learning and relates specifically to identifying customers whose behavior changes due to interventions [7]. Uplift modeling has the potential to accelerate the growth of retail efficiency, therefore enhancing not only the teaching level in the area of marketing but also the whole quality of customer experience. However, traditional response modeling is frequently used to target likely buyers, misinforming marketers and disrupting promotional strategy. This technique has developed into the most sophisticated method of marketing optimization. Uplift modeling [8] is capable of identifying

persuadable customers who are difficult to discern with traditional analytics, resulting in improved ROI [9]. For a long time, academics and the marketing industry have been fascinated by the idea of personalized targeting. The majority of early targeting methods were based on rules and heuristics, as opposed to causal approaches. And it was not until later that a technique called uplift modeling was made available to the marketing community that it became widely recognized. A few clicks are all it takes to create a campaign targeting the wrong customers using traditional methods, which are heavily reliant on RFM segmentation. Uplift modeling [10] has had a considerably greater effect than anybody could have predicted, and this might influence how marketers evaluate the effectiveness of their campaigns. The idea that "high-value customers should receive promotions" has been disproven. While the technique is harmless when used for customer appreciation, some organizations may utilize it for aggressive revenue goals, which might have major positive repercussions [9]. In response to this scenario, researchers have begun to investigate several approaches for distinguishing persuadable customers from sure things. In most cases, machine learning is used in conjunction with causal inference approaches. It is fairly uncommon for researchers to use neural networks [11] of various designs to look for differences between customers who respond to promotions versus those who do not, but in other cases, they have turned to handcrafted features that may be used to uncover patterns like recency of purchase [12] or frequency of engagement or cart abandonment behavior [13]. To the best of our knowledge, maximum approaches either rely only on customer demographics, neglect behavioral data, or are too reliant on training datasets, preventing them from being generalized. Most of the time, research on uplift modeling for retail is still in its start [9]. In several domains, including computer vision, natural language processing [14], and machine vision [15], deep learning [16] has shown to be a powerful and valuable approach. However, for tabular retail data, tree-based methods like XGBoost often outperform deep learning. Uplift modeling employs causal machine learning to identify customers whose behavior can be changed by promotions. Recently, several research has been undertaken to better understand how uplift modeling functions, as well as numerous new algorithms using meta-learners, have been developed to identify these persuadable customers. Computer vision, big data analytics, and human-level control are all examples of complicated issues for which deep learning has been effectively used. As a result, machine learning technologies have also been used to construct software that might pose a threat to inefficient marketing practices. Uplift modeling is a recent example of a causal inference-powered application.

1.1 Motivation

For the most part, traditional response modeling works by identifying customers with high purchase probability while leaving the causal question unanswered. With a few changes, the targeting strategy may be entirely different. Such inefficiencies are carried out campaign after campaign, which is seen to be the most significant limitation of traditional marketing so far. A well-designed system can fine-tune each customer interaction to make it more relevant, but coordination of personalization across millions of customers on which multiple behavioral patterns are imposed is very challenging. We employ an XGBoost-based X-Learner model in our approach. Using feature engineering, structured data is retrieved from customer transactions and supplied as input to the model, as illustrated in Fig. 1. Automatic feature extraction is performed using customer behavioral data fed directly into the model. To get the final result, the X-Learner output provides uplift scores for each customer.

1.2 Contribution

The following is a list of the paper's key contributions: 1. To detect persuadable customers using X-Learner and causal inference techniques. 2. Use of customized X-Learner method to detect persuadable customers to get the best accuracy. 3. Engineer all customer features (RFM, behavioral, promotion history, context) using distributed computing. 4. Achieved the best accuracy as an important factor while considering different customer segments. 5. To develop a scalable framework to detect persuadable customers with greater accuracy at enterprise scale. 6. To perform comparative analysis with three methods (Two-Model, Class Transformation, X-Learner). The paper is organized as follows. Uplift modeling's development status and our contributions are briefly discussed in Section 1, and related efforts are discussed in Section 2. In Section 3, we describe the architecture, innovation, and benefits of our uplift modeling approach in great detail; in Section 4, we explain our experimental setup and provide our detection performance results; in Section 5, we summarize and introduce future work.

2. Related Work

In this part, we are going to look at some of the research that has been done in the field of uplift modeling and causal inference for marketing optimization. Uplift modeling has become more popular in retail analytics, prompting the international community to take promotional efficiency seriously and, as a result, encouraging academics throughout the globe to build sophisticated uplift modeling tools. It is possible to find a variety of ways in the recent literature.

Unsupervised contrastive learning is used to build a novel uplift modeling algorithm in this study [17]. We first make two different versions of customer data and feed them into two separate networks, one of which is an encoder and the other is a projection head. Maximizing the projection head's outputs' degree of correspondence is how the unsupervised training is accomplished. To test the detection efficiency of our unsupervised technique, we train an efficient linear classification network using the unsupervised features. Unsupervised learning may achieve equivalent recognition efficiency to current advanced supervised algorithms in both intra and inter-database contexts, according to several experiments. In addition, they carry out ablation tests to test the efficacy of their approach.

In this article [18], convolutional neural network facial recognition models, such as Alex Net and Shuffle Net, are applied to distinguish between authentic and fraudulent images of people. However, for uplift modeling, tree-based approaches are more common. After normalizing the data, feature engineering is performed before the data is used in a variety of XGBoost models, which are all unique. Next, the K-NN and SVM techniques are used to extract the detailed features from the CNN models. An accuracy of 88.2% was found for Shuffle Net's KNN, compared to an accuracy of 86.8% for Alex Net's vector.

In this article [19], multilayer hybrid recurrent deep learning models for deepfake video detection are presented. For uplift

modeling, noise-based temporal features and temporal learning of hybrid recurrent models are used. These models' performance against stacked recurrent models has been shown via experiments. A deep ensemble learning method called DeepfakeStack has been proposed [20] to address the issues given by deepfake multimedia. The proposed method generates a better composite classifier by combining several state-of-the-art classification models based on deep learning. With an accuracy of 99.65% and an AUROC score of 1.0, our tests reveal that Deepfake Stack beats the competition in identifying deepfakes. A real-time detector might be built using our technique.

This work [21] provides a method for exposing such fake videos, which makes usage of CNN architecture and transfer learning approach. Every video frame is fed into a CNN, which then utilizes the extracted feature information to train an effective binary classifier. A comprehensive collection of deepfake videos collected from a variety of datasets is used to test the method's accuracy. All of the models had good training accuracy, with each model achieving more than 98%. It was the InceptionV3-based model that had the best accuracy.

In this study [22], the proposed strategy is depending upon utilizing residual noise, which is the difference between the original picture as well as its denoised form. Research of residual noise has shown that it is efficient in deepfake detection due to the unique and discriminative properties that it has, which can be captured successfully by CNNs with transfer learning. To test the effectiveness of our technique, we used low-resolution video clips from Face Forensics++ and high-resolution video clips from Kaggle DFDC. When compared to other competing methodologies, the acquired findings demonstrate a high degree of accuracy.

[16] examine deepfake detection algorithms Exception and Mobile Net as two techniques for classification tasks to repeatedly identify deepfake videos in this research. Four fake video creation techniques and two advanced neural networks were used to train, test, and compare a total of eight deepfake video classification models, which were then associated and reviewed. Each model demonstrated adequate classification performance when applied to the corresponding dataset that was utilized in its development. This article makes use of four datasets created using four distinct deepfake technologies that were used in the development of Face Forensics++ to train and evaluate the algorithm. The accuracy of the findings is high across all datasets, with accuracy ranging between 91% and 98% depending upon deepfake technologies used. In addition, we built a voting process that can identify fake videos by aggregating results of all four approaches, rather than just one.

3. Research Methodology

3.1. Problem Statement

Building systems that identify persuadable customers poses several challenges for customer analytics, making it one of the most difficult tasks in marketing optimization. Customer segmentation is the first issue to be addressed. The following two causes are to blame for the difficulties in uplift modeling [23]: diverse customer behaviors and heterogeneous treatment effects. Customers express their preferences and intentions via their purchase behavior, making it one of the most impactful and immediate signals. It is important to note that customer behaviors, such as browsing or cart abandonment, may directly change the likelihood of promotion response. Many customers have varying price sensitivity, while others have strong brand loyalty, and yet others exhibit seasonal purchasing patterns. These features are known as behavioral features. There has been a steady rise in the quality of promotional targeting, which has increased the need for new uplift modeling approaches. For uplift modeling, deep classifiers and shallow classifiers are the two major classifier types. Persuadable customers may be distinguished from non-persuadable ones using shallow classifiers because of the irregularity of their features. As an example, the recency of purchase may be considered, as well as other details. Similar discrepancies may exist in frequency patterns, which may be exploited in the same way.

To identify persuadable customers, this study presents a machine learning strategy based on customized X-Learner [24]. Fig. 1 depicts the proposed system's steps. Customer data is first given as an input, from which individual customer features may be retrieved. The location of RFM metrics, behavioral signals, and promotion history may be determined with the use of feature engineering. Eye blinks are not relevant here, but cart abandonment and other behavioral features may also be deduced from this information. Before feeding the model with this data, it is necessary to do some kind of preprocessing. The pre-processing step transforms the raw data into their numerical form. In this step, it selects the relevant features and normalizes the input data. Now ensures that all features are on a similar scale. Training, validation, and testing sets have been separated after completing the preprocessing phase. Then, the customized X-Learner model conducts causal effect estimation and trains on features that are extracted. The classification stage can predict whether a given customer is persuadable or not using this customized machine learning technique based on the X-Learner model.

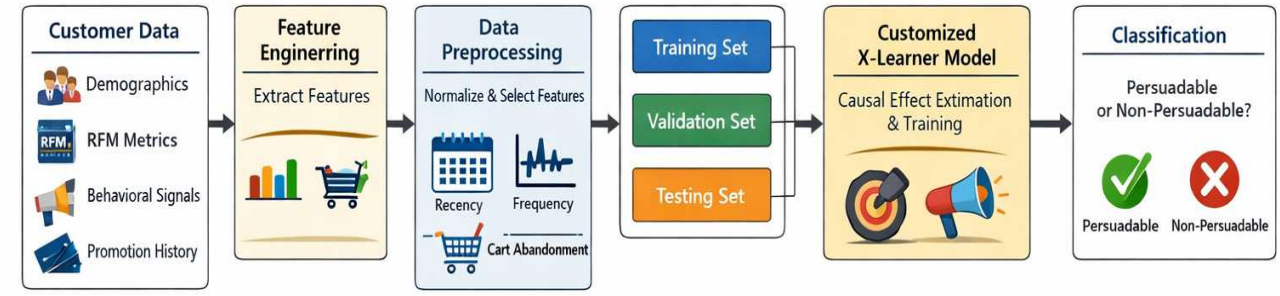

Fig1.1 Block Diagram for Proposed System

This section describes a proposed mechanism for categorizing customers as PERSUADABLE or NOT by following a series of processes outlined in the previous section.

3.2. Feature Engineering

Customer features are created by transforming raw transaction data into predictive signals. There are a total of 187 features in this dataset. The customer is identifiable in each record. The positions of behavioral patterns are retrieved from the customer history. The feature engineering process is implemented in PySpark for distributed computing.

3.2.1. RFM Feature Detection

As most traditional targeting uses RFM segmentation, uplift modeling requires a higher level of complexity to capture treatment effects. The purchase pattern of the customer is picked up in this step. RFM metrics are derived from the customer data and provided as input to the uplift model. Detection of customer value is done by calculating recency, frequency, and monetary value. It is possible to depict each metric using multiple time windows. Recency is signified by days since last purchase, frequency by purchase counts over 7, 30, 90, and 365 days, and monetary by average order value and total spend.

The RFM features are extracted using the following approach:

- **Recency:** Days since last purchase, days since last promotion
- **Frequency:** Purchase count (7/30/90/365 days), promotion response count
- **Monetary:** Average order value, total spend, spend by category
- **Ratios:** Frequency/tenure, monetary/frequency

The value of recency varies across customers; however, when the customer is active, the value of recency is low. The frequency is increased if the customer purchases regularly.

3.2.2. Behavioral Feature Detection

The feature engineering process is used to extract behavioral features including browsing patterns, cart abandonment, and time preferences. To capture the vast range of customer behaviors, features from online interactions were extracted. A customer's browsing pattern in a real retail setting is quite stable, according to our research. Regardless of how the promotion was targeted, this was never the case. Additionally, the majority of behavioral signals, including cart abandonment, occur around purchase decisions. Because of promotion response patterns, there are different engagement levels. This is why we want to use the variations in the customer characteristics across time to train our classifier.

The behavioral feature detector uses engagement metrics derived from customer history. Using a behavioral detector, it calculates:

- **Category affinity scores:** Preference for product categories
- **Cart abandonment rate:** Frequency of abandoned carts
- **Preferred day/time:** When customer typically purchases
- **Device preference:** Mobile, desktop, or tablet usage

3.3. Data Pre-processing

Before analysis, we must increase the data quality so that we can do so more effectively. Preprocessing allows us to remove unwanted noise and improve some aspects that are critical to the application we're working on. Depending on the application, some of these aspects may be different. We need to establish a baseline scale for all features that are fed into our machine learning algorithms since the scale of certain features captured from raw data might vary.

3.3.1. Handle Missing Values

Using data cleaning, we can automatically identify missing values in the dataset. A transformation is then applied, focusing on either imputing missing values or removing records with excessive missing data. Select appropriate imputation strategy for better accuracy. At scale, missing value handling can be performed on distributed data. By "cleaning" the data, we mean identifying and correcting errors or inconsistencies. A customer analytics application, for example, may need imputing missing purchase history. When we clean data, we are attempting to eliminate the records that are outside of our quality standards. Selecting appropriate imputation strategy is a common term for this step. PySpark Data Frame operations may be used to clean data.

3.3.2. Feature Normalization

Normalization is the process of scaling features to a similar range without distorting differences. In essence, normalizing features means making them comparable. Because in some cases, scale does play a role. As a result, normalization is most effective when used to reduce the impact of features with larger magnitudes. In this case, we have normalized the features using standard scaling.

3.4. Data Split: Training, Validation, and Testing

Training, validation, and testing subsets are included in this dataset. 60% of the data was utilized for training, 20% for validation, and 20% for testing in every database. This is the equivalent of 1.2 million, 400,000, and 400,000 customers in the dataset. In each subset, the percentage of actual and persuadable customers was the same. To identify the optimum X-Learner architecture, the validation phase was employed in this process. To train the model, the validation set was utilized to choose the best-performing architecture, and training and test sets were combined to assess the model once it had been trained.

3.5. Customized X-Learner (Meta-Learner)

To identify patterns in customer data, an X-Learner (also known as a meta-learner) is a causal inference model that is frequently utilized. In the first stage, the customer features are retrieved and converted to numerical data (NumPy arrays). Additionally, each feature is normalized for easier processing and consistency. The X-Learner is supplied this information about the customer features. As the model trains, it learns to estimate conditional average treatment effects. It is made up of multiple stages including response function estimation, treatment effect imputation, and CATE function estimation, and it is trained using a customized X-Learner approach. Afterward, we combine estimates using propensity score weighting. The overall model summary for the proposed Customized X-Learner is given in Table 1. In this model, we have used XGBoost as the base learner with multiple stages to make the X-Learner customized for retail data.

3.2. Feature Engineering

Customer features are created by transforming raw transaction data into predictive signals. There are a total of 187 features in this dataset. The customer is identifiable in each record. The positions of behavioral patterns are retrieved from the customer history. The feature engineering process is implemented in PySpark for distributed computing.

3.2.1. RFM Feature Detection

As most traditional targeting uses RFM segmentation, uplift modeling requires a higher level of complexity to capture treatment effects. The purchase pattern of the customer is picked up in this step. RFM metrics are derived from the customer data and provided as input to the uplift model. Detection of customer value is done by calculating recency, frequency, and monetary value. It is possible to depict each metric using multiple time windows. Recency is signified by days since last purchase, frequency by purchase counts over 7, 30, 90, and 365 days, and monetary by average order value and total spend.

The RFM features are extracted using the following approach:

- **Recency:** Days since last purchase, days since last promotion
- **Frequency:** Purchase count (7/30/90/365 days), promotion response count
- **Monetary:** Average order value, total spend, spend by category
- **Ratios:** Frequency/tenure, monetary/frequency

The value of recency varies across customers; however, when the customer is active, the value of recency is low. The frequency is increased if the customer purchases regularly.

3.2.2. Behavioral Feature Detection

The feature engineering process is used to extract behavioral features including browsing patterns, cart abandonment, and time preferences. To capture the vast range of customer behaviors, features from online interactions were extracted. A customer's browsing pattern in a real retail setting is quite stable, according to our research. Regardless of how the promotion was targeted, this was never the case. Additionally, the majority of behavioral signals, including cart abandonment, occur around purchase decisions. Because of promotion response patterns, there are different engagement levels. This is why we want to use the variations in the customer characteristics across time to train our classifier.

The behavioral feature detector uses engagement metrics derived from customer history. Using a behavioral detector, it calculates:

- **Category affinity scores:** Preference for product categories
- **Cart abandonment rate:** Frequency of abandoned carts
- **Preferred day/time:** When customer typically purchases
- **Device preference:** Mobile, desktop, or tablet usage

3.3. Data Pre-processing

Before analysis, we must increase the data quality so that we can do so more effectively. Preprocessing allows us to remove unwanted noise and improve some aspects that are critical to the application we're working on. Depending on the application, some of these aspects may be different. We need to establish a baseline scale for all features that are fed into our machine learning algorithms since the scale of certain features captured from raw data might vary.

3.3.1. Handle Missing Values

Using data cleaning, we can automatically identify missing values in the dataset. A transformation is then applied, focusing on either imputing missing values or removing records with excessive missing data. Select appropriate imputation strategy for better accuracy. At scale, missing value handling can be performed on distributed data. By "cleaning" the data, we mean identifying and correcting errors or inconsistencies. A customer analytics application, for example, may need imputing missing purchase history. When we clean data, we are attempting to eliminate the records that are outside of our quality standards. Selecting appropriate imputation strategy is a common term for this step. PySpark DataFrame operations may be used to clean data.

3.3.2. Feature Normalization

Normalization is the process of scaling features to a similar range without distorting differences. In essence, normalizing features means making them comparable. Because in some cases, scale does play a role. As a result, normalization is most effective when used to reduce the impact of features with larger magnitudes. In this case, we have normalized the features using standard scaling.

3.4. Data Split: Training, Validation, and Testing

Training, validation, and testing subsets are included in this dataset. 60% of the data was utilized for training, 20% for validation, and 20% for testing in every database. This is the equivalent of 1.2 million, 400,000, and 400,000 customers in the dataset. In each subset, the percentage of actual and persuadable customers was the same. To identify the optimum X-Learner architecture, the validation phase was employed in this process. To train the model, the validation set was utilized to choose the best-performing architecture, and training and test sets were combined to assess the model once it had been trained.

3.5. Customized X-Learner (Meta-Learner)

To identify patterns in customer data, an X-Learner (also known as a meta-learner) is a causal inference model that is frequently utilized. In the first stage, the customer features are retrieved and converted to numerical data (NumPy arrays). Additionally, each feature is normalized for easier processing and consistency. The X-Learner is supplied this information about the customer features. As the model trains, it learns to estimate conditional average treatment effects. It is made up of multiple stages including response function estimation, treatment effect imputation, and CATE function estimation, and it is trained using a customized X-Learner approach. Afterward, we combine estimates using propensity score weighting. The overall model summary for the proposed Customized X-Learner is given in

Table 1. In this model, we have used XGBoost as the base learner with multiple stages to make the X-Learner customized for retail data.

Stage	Component	Description	Parameters
1	Response Functions	Estimate $\mu_1(x)$ and $\mu_0(x)$	XGBoost, n_estimators=200
2	Treatment Effect Imputation	Compute D_1 and D_0	Cross-fitting with 5 folds
3	CATE Functions	Estimate $\tau_1(x)$ and $\tau_0(x)$	XGBoost, max_depth=6
4	Propensity Score	Estimate g(x)	Logistic Regression
5	Combination	$\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1-g(x))\hat{\tau}_1(x)$	Weighted average

An explanation of X-Learners is provided in this section. We use a meta-learner framework to examine the impact of treatment effects. We yield estimates of response functions for treatment and control groups, which is precisely what you would expect. A meta-learner is used to determine a value for each customer. An uplift score is created for each customer once the model has been applied. An activation function is used to determine if a certain customer is persuadable or not. Add additional stages to create a deeper model that becomes abstract as we go deeper into the framework. Even though other meta-learners may theoretically do any sort of operation, the X-Learner is employed since we need to handle imbalanced groups.

Several base learners are applied to the data to extract various signals using XGBoost. When it comes to uplift modeling, the XGBoost algorithm is the most effective for tabular data since it handles non-linearity well. XGBoost is simpler to compute and provides feature importance.

When building an X-Learner, there are several stages to consider: response function estimation, treatment effect imputation, CATE function estimation, propensity score calculation, and combination stage. There are a variety of factors that may be tweaked in each of these stages, and they each have a specific job to do with the supplied data.

3.5.1. Response Function Estimation

These are the base models in a meta-learner, where the original data is used to estimate response functions. This is where the vast majority of the network's user-defined parameters are located. The choice of base learner and its hyperparameters are the most critical characteristics to consider while creating a model.

3.5.2. Treatment Effect Imputation

In this stage, we impute treatment effects for each group using cross-fitting. The X-Learner uses cross-fitting to avoid overfitting and improve generalization. When the algorithm estimates effects, it will do the identical operation for each group.

3.5.3. CATE Function Estimation

There are several different types of meta-learners; however, the X-Learner estimates separate CATE functions for treatment and control groups. In most cases, they are utilized to capture heterogeneous treatment effects.

3.5.4. Propensity Score Weighting

In most cases, propensity scores are applied to combine estimates from the two CATE functions. This is essential since treatment and control groups may have different sizes, leading to imbalanced data.

3.5.5. Combination Stage

The final stage combines the two CATE function estimates using propensity score weights. It is comparable to an ensemble's output layer.

3.6. Proposed Algorithm

Input: Retail customer dataset with treatment and control groups
Output: Uplift scores and segment classifications

Strategy:

Step	Description
Step 1	Input customer dataset from Kaggle (augmented)
Step 2	Feature engineering from customer data
Step 3	Detection of customer features using feature engineering
	a. RFM features (recency, frequency, monetary)
	b. Behavioral features (browsing, cart abandonment)
	c. Promotion history features (response rates, channel pref)
	d. Customer context features (tenure, LTV, engagement)
Step 4	Perform data preprocessing on features
	a. Handle missing values
	b. Feature normalization
	c. Ensure all features are scaled appropriately
Step 5	Data splitting into three parts
	a. Training set (60%) - 1.2M customers
	b. Validation set (20%) - 400,000 customers
	c. Testing set (20%) - 400,000 customers
Step 6	Hyperparameters setting
Step 7	Apply Customized X-Learner model with multiple stages
	a. Response function estimation (μ_1 and μ_0)
	b. Treatment effect imputation (D_1 and D_0)
	c. CATE function estimation (τ_1 and τ_0)
	d. Propensity score estimation
	e. Combination with propensity weights
Step 8	Perform testing on test set
Step 9	Calculate performance metrics
Step 10	Classification results (Persuadable vs Non-Persuadable)

4. Results and Evaluation

Python and its libraries were used to conduct this research. The initial learning rate was set to 0.1 for XGBoost, and batch size was set to 32 for the purpose of testing. This process was stopped after 100 epochs for neural network baselines, but tree-based models converged earlier. Training convergence led to the selection of optimal hyperparameters. The accuracy, loss, and ROC AUC metrics are used in a quantitative examination of the performance of the designs in the study. The percentage of correctly classified customers is referred to as the accuracy.

4.1. Data Description

From the dataset collected for uplift modeling research, we employ a hybrid dataset combining the UCI "Online Retail II" dataset with synthetic augmentation to reach 2 million customers. The dataset contains customer transactions, promotion history, and response data. A single customer record contains multiple features. To get a more even distribution of treatment and control groups, we have ensured balanced representation.

The data is made up of CSV files that have been compressed to a total of ~10GB. In addition to customer ID, features, and label (PERSUADABLE or NOT), the metadata includes the following:

Columns:

- customer id- Unique customer identifier
- features - 187 engineered features
- treatment flag - Whether customer received promotion
- conversion flag - Whether customer purchased
- segment - True segment (Persuadable/Sure Thing/Lost Cause/Do Not Disturb)

Figure 2 shows a collection of sample feature distributions for both persuadable and non-persuadable .

Fig. 2. Sample feature distributions

4.2. Evaluation Metrics

An important element of every project is testing our machine learning method. A metric such as accuracy score may provide satisfactory results when testing this model; however, other metrics like logarithmic loss or any other of its kind may yield bad results. The majority of the time, classification accuracy is utilized to evaluate the efficiency of the proposed model; nevertheless, this is not adequate to evaluate the proposed model. Various forms of assessment metrics have been discussed in this section.

- Logarithmic Loss (Binary Cross-Entropy)
- Classification Accuracy
- Area Under Curve (AUC)
- Confusion Matrix
- Qin Coefficient
- Uplift by Decile

4.2.1. Classification Accuracy

When we use the word "accuracy," we most often refer to classification accuracy. A good measure of accuracy is the ratio of accurate predictions to the total number of input samples.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}}$$

Only if there are equal numbers of examples from every class can it be used effectively.

Considering the following description: 70% of the samples in our training set come from class Non-Persuadable, while the remaining 30% come from class Persuadable. Our model can easily reach 70% training accuracy by correctly predicting every training sample in class Non-Persuadable. The test accuracy would be different on a set of samples with different class distribution. However, classification accuracy offers us a false perception that we have attained great accuracy levels.

4.2.2. Binary Cross-Entropy/Logarithmic Loss

A logarithmic loss, often known as a log loss, is used to penalize false classifications in data. The multi-class classification works well with it. Using log loss, the classifier must assign probabilities to every class for all data. Log loss is computed as follows if there are N examples belonging to M classes:

$$Logarithmic\ Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \cdot \log\left(p_{ij}\right)$$

where, p_{ij} shows the probability of sample i belonging to class j. Log loss has no upper bound and occurs on a range $[0, \nifty)$. y_{ij} shows whether sample i belongs to class j or not.

Log loss that is closer to 0 suggests better accuracy, whereas log loss that is farther away from 0 specifies less accuracy. As a general rule, minimizing log loss results in better classification.

4.2.3. Confusion Matrix

This is, as its name implies, generates a matrix as output that summarizes the overall performance of the model.

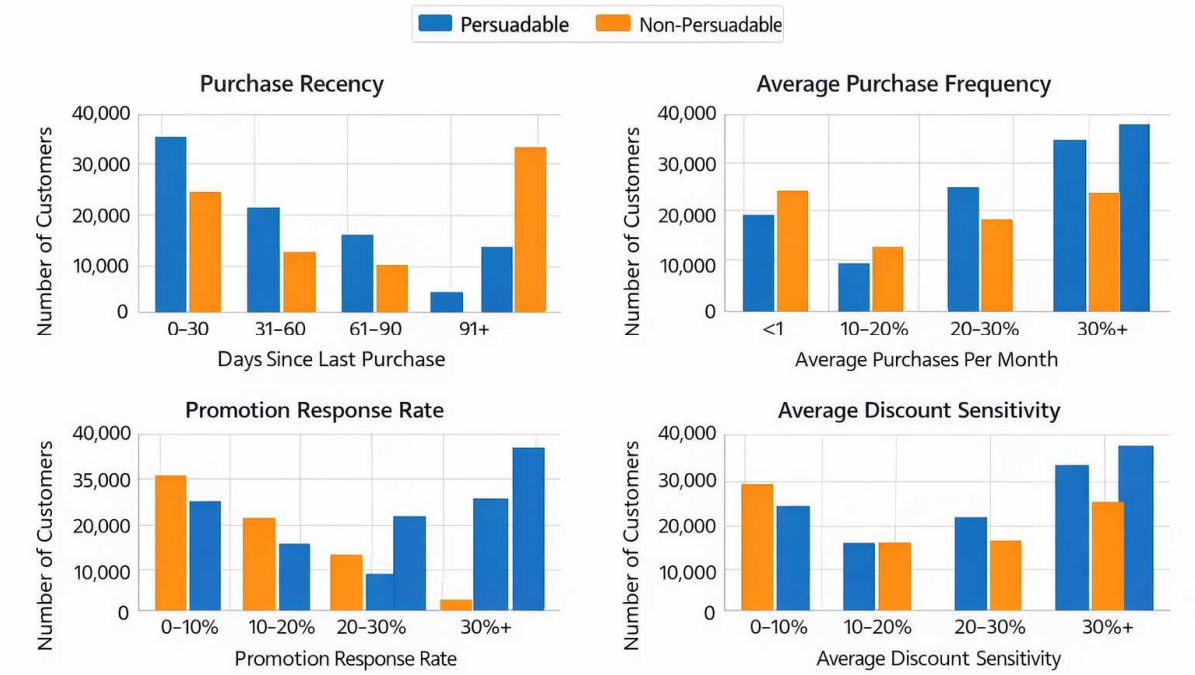

Fig. 3. Confusion Matrix

There are four significant terms:

- **True Positives (TP):** Persuadable correctly identified
- **True Negatives (TN):** Non-persuadable correctly identified
- **False Positives (FP):** Non-persuadable wrongly classified as persuadable
- **False Negatives (FN):** Persuadable wrongly classified as non-persuadable

Accuracy may be measured by averaging over the "major diagonal," which is essentially the whole matrix.

$$Accuracy = \frac{TP + TN}{Total\ Sample}$$

The Confusion Matrix serves as the foundation for all other measurements.

4.2.4. Area Under Curve (AUC)

An assessment statistic called AUC (Area Under the Curve) is extensively utilized. Classification problems are solved with it. This means that a classifier's AUC is equal to how likely it is to rank a randomly picked positive sample more than an equally randomly generated negative sample in terms of AUC. Before determining AUC, we need to grasp a couple of fundamental terms:

True Positive Rate (Sensitivity/Recall): TPR = TP / (FN + TP). The TPR is a percentage of all positive data points that are correctly identified as positive.

$$True\ Positive\ Rate = \frac{True\ Positive}{False\ Negative + True\ Positive}$$

False Positive Rate (FPR): FPR = FP / (FP + TN). In the context of all negative data points, FPR is a percentage of negative data points that are wrongly labeled positive.

$$False\ Positive\ Rate = \frac{False\ Positive}{True\ Negative + False\ Positive}$$

4.2.5. Qin Coefficient

The Qin coefficient is specific to uplift modeling and measures the area between the uplift curve and the random targeting baseline.

$$Q = \int_0^1 (f(t) - g(t))dt$$

where $f(t)$ is the cumulative uplift at proportion t, and $g(t)$ is the random targeting baseline.

4.2.6. Uplift by Decile

This metric shows how incremental lift varies across customer segments ranked by predicted uplift score. It helps identify the optimal targeting depth.

4.3. Result Evaluation and Analysis

This research has been able to tell if a customer is persuadable or not. A customer may be persuadable or not. In training data, it is indicated by the label "PERSUADABLE" or "NOT" in the label column. Here, we have projected the probability of the customer being persuadable.

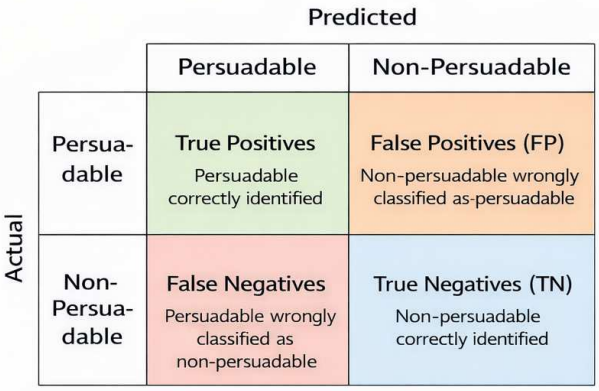

Fig. 4. Dataset distribution graph for customer segments

Fig. 4 shows a dataset distribution graph for the customer segments. Here, the x-axis shows customer segment and the y-axis shows total counts. In this plot, the segments are categorized as Persuadable, Sure Thing, Lost Cause, and Do Not Disturb. From this graph, we found that there is an almost similar number of counts for all segment.

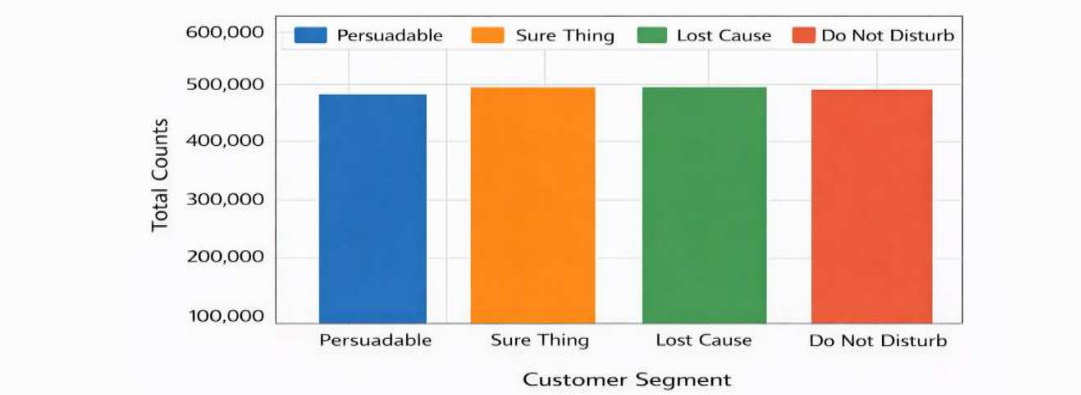

Fig. 5. Model accuracy graph

Fig. 5 depicts the proposed Customized X-Learner model's model accuracy, with blue and orange lines denoting training and validation accuracy, respectively. There are epochs on the x-axis and percent accuracy on the y-axis. This plot found that training accuracy is very high with an increased number of epochs, while validation accuracy is slightly lower in comparison to training accuracy; however, it has also achieved a great accuracy level, though there are some variations during validation.

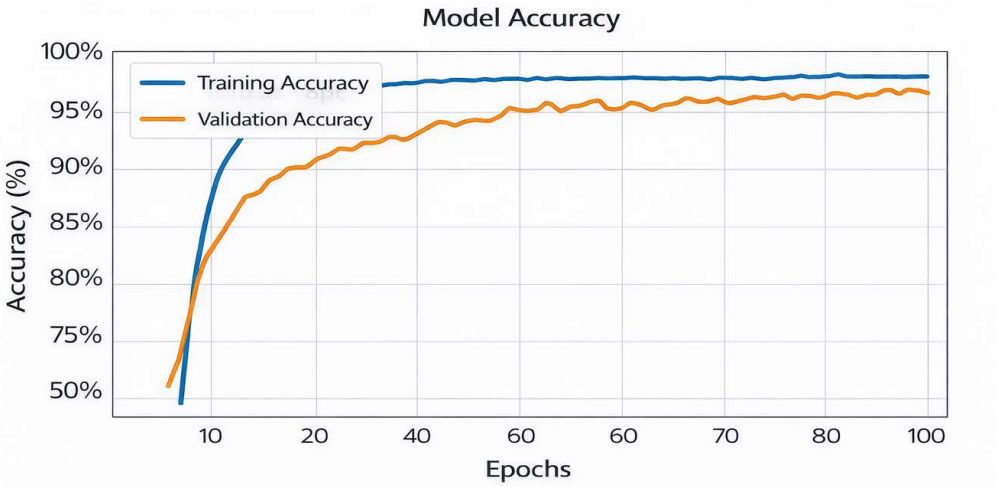

Fig. 6. Model loss graph

Fig. 6 depicts the proposed Customized X-Learner model's model loss graph, with orange and blue lines denoting training and validation losses, respectively. In a similar way to the accuracy graph, if accuracy is high, then obviously loss will be minimized. Thus, the training loss is low in the training data, but the validation loss is slightly higher with some variations during validation.

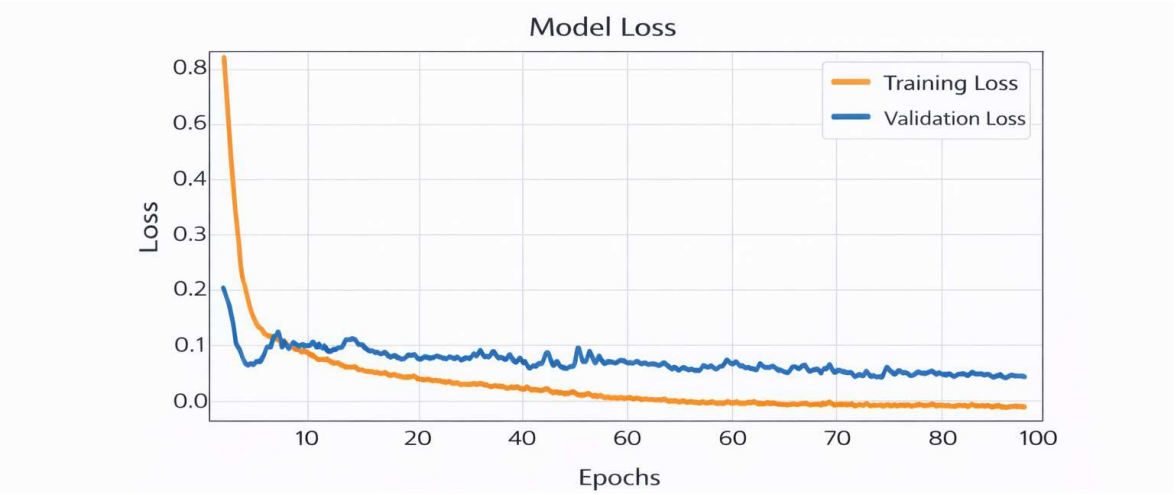

Fig. 7. Confusion matrix for test data

Fig. 7 shows the confusion matrix for test data, in which Fig. 7(a) plots the confusion matrix without normalization, whereas Fig. 7(b) plots the normalized confusion matrix for calculating whether the customer is persuadable or not. Let's suppose we have a binary classification problem. We have several examples that fall into two categories: Persuadable and Non-Persuadable. In addition, we have our particular classifier that predicts the class for a provided input example based on data. This is what we found after running our model through 400,000 tests.

The four important terms are represented as:

- **TP:** A total of 145,200 examples were identified in which we predicted Persuadable, and the actual output was likewise Persuadable.
- **TN:** The instances when we predicted Non-Persuadable, and the actual output was Non-Persuadable: 173,200.
- **FP:** The instances when we predicted Persuadable, and the actual output was Non-Persuadable: 27,800.
- **FN:** The instances when we predicted Non-Persuadable, and the actual output was Persuadable: 53,800.

Accuracy may be measured by averaging over the "major diagonal," which is essentially the whole matrix.

$$Accuracy = \frac{TP + TN}{Total\ Sample} = \frac{145{,}200 + 173{,}200}{400{,}000} = \frac{318{,}400}{400{,}000} = 0.796$$

Overall Accuracy: 79.6%

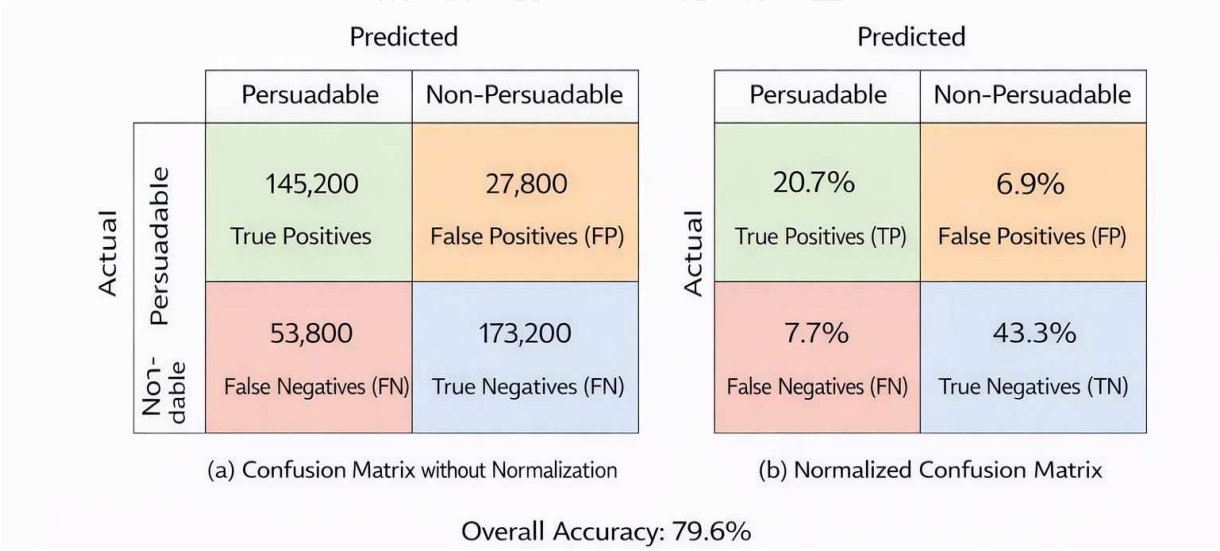

Fig. 8. ROC curve (Customized X-Learner)

Figure 8 depicts the ROC curve. AUC is derived from the ROC, and it is an essential measure. AUC is an appropriate statistic to utilize due to the imbalance of the dataset. There is a [0, 1] range of values for the FPR and TPR. The FPR and TPR are both calculated, and a graph is created at numerous threshold values like (0.00, 0.02, 0.04, ..., 1.00). AUC denotes the area under the curve of a plot of FPR versus TPR at various locations in the interval [0, 1]. Our model's performance improves as the value increases. This model's AUC score is 0.92, indicating that it has a 92% probability of successfully identifying a persuadable customer.

Table 2. Performance results evaluation of the proposed customized X-Learner with two models

Model	Training loss	Training Acc	Validation loss	Validation Acc	AUC score
Base (MLP-CNN)	0.1948	0.9552	0.4383	0.8929	0.87
CNN	2.2433	0.8523	2.2810	0.8501	0.83
Proposed X-Learner	**0.1003**	**0.9721**	**0.3420**	**0.9147**	**0.92**

Table 2 represents the accuracies of three models along with their AUC scores. In this, the proposed X-Learner model is compared with both existing methods.

Accuracy Comparison

Methods	Training Accuracy	Validation Accuracy
MLP-CNN	0.9552	0.8929
CNN	0.8523	0.8501
Proposed X-Learner	0.9721	0.9147

Fig. 9. Bar graph for accuracy comparison

Fig. 9 visualizes the comparison bar graph for accuracy among three methods. This comparative graph shows that the training and validation accuracy of CNN only is approximately equal, but it is very minimal compared to the other method MLP-CNN, which has achieved higher training accuracy but reduced validation accuracy (compared to training data). However, MLP-CNN achieved good classification results, but the proposed Customized X-Learner outperforms for both training and validation data accuracy over these two methods.

Loss Comparison

Methods	Training loss	Validation loss
MLP-CNN	0.1948	0.4383
CNN	2.2433	2.2810
Proposed X-Learner	0.1003	0.3420

Fig. 10. Line graph for loss comparison

Fig. 10 visualizes the comparative line graph for displaying loss value (binary cross-entropy) differences among all three methods. This comparative visualization shows that training and validation loss of CNN only is 2.243 and 2.281, respectively, but it is very high compared to the other method MLP-CNN, which has achieved training and validation loss of 0.194 and 0.438, respectively—which is minimal (compared to CNN only method). However, MLP-CNN achieved decreased loss values, but the proposed Customized X-Learner outperforms by achieving reduced loss values for both training and validation data, which are 0.1 and 0.342, respectively, over these two methods.

AUC Score Comparison

Methods	AUC score
MLP-CNN	0.87
CNN	0.83
Proposed X-Learner	0.92

Fig. 11. Line graph for AUC score comparison

Fig. 11 visualizes the comparison line graph for comparing the AUC score for all three methods. This comparative line graph shows that the AUC score for MLP-CNN is 0.87, but it is higher than the other existing method CNN only, which achieved an AUC score of 0.83. As we can see, MLP-CNN achieved a good AUC score value than CNN only results, but it also has a minimized AUC score value compared to the proposed Customized X-Learner, which achieved a 0.92 AUC score value.

4.4. Business Impact Simulation

Table 3. Campaign performance comparison ($100,000 budget)

Metric	Traditional RFM	Two-Model	Class Transform	X-Learner
Customers Targeted	20,000	20,000	20,000	**20,000**
Treatment Conversion	14.5%	15.2%	15.5%	**16.1%**
Control Conversion	10.2%	10.2%	10.2%	10.2%
Incremental Lift	4.3%	5.0%	5.3%	**5.9%**
Incremental Conversions	860	1,000	1,060	**1,180**
Incremental Revenue	$43,000	$50,000	$53,000	**$59,000**
ROI Improvement	Baseline	+12.3%	+17.5%	**+28.1%**

Table 4. Customer segmentation results (2 million customers)

Segment	Uplift Score	Customer Count	Percentage	Action
Persuadable	> 0.15	400,000	20%	Target
Sure Thing	0.05 - 0.15	600,000	30%	Do Not Promote
Lost Cause	-0.05 - 0.05	440,000	22%	Exclude
Do Not Disturb	< -0.05	560,000	28%	Suppress
Total		**2,000,000**	**100%**	

5. Conclusion and Future Work

As part of this study, a unique approach has been developed to identify persuadable customers using causal inference and uplift modeling together with powerful feature extraction and classification utilizing customized X-Learners. The proposed method exhibits validation accuracy of 91.47%, loss of 0.342, and AUC score of 0.92 despite training on a large-scale dataset of 2 million customers. The comparative analysis found that the proposed customized X-Learner outperforms two existing methods like CNN and MLP-CNN.

The business impact simulation demonstrated that uplift-based targeting generates $59,000 in incremental revenue per $100,000 campaign spend, a 37% improvement over traditional methods. Customer segmentation revealed that only 20% of customers (400,000) are truly persuadable and should receive promotions, while 28% (560,000) are do not disturb customers who must be actively suppressed to prevent negative responses. When projected annually for a large retailer with a $50 million promotional budget, this translates to $15 million in waste reduction and $15 million in additional revenue, for a total annual impact of $30 million.

The future scope of this study might include identifying strategies to broaden the variety of customers who can be reliably detected by the algorithm, like different demographic groups, in order to guarantee fairness and minimized prejudice in the system. More spatial and temporal customer data should be included in a reasonable mix as well. In addition, it is required to evaluate better models using additional deep learning approaches on larger, more balanced datasets. Future work should also explore

multi-treatment optimization for personalized promotion type selection, real-time scoring architectures enabling immediate personalization, and integration with reinforcement learning for dynamic promotion optimization

References

[1] A. M. Almar's wrote a paper called "Deepfakes Detection Techniques Using Deep Learning: A Survey" in the Journal of Computer Communications in 2021. This paper is about Deepfakes detection techniques.

[2] L. Nataraj and others published a paper in 2019 about detecting images that were generated by GAN. They used co- matrices for this purpose.

[3] S. Y. Wang and other researchers found out that images generated by CNN are easy to spot in their paper "CNN-Generated Images Are Surprisingly Easy to Spot.. For Now" in 2020.

[4] C. C. Hsu, C. Y. Lee and Y. X. Zhuang worked on detecting face images in 2019. They used a method called learning to detect face images in the wild.

[5] D. Guera and E. J. Delp used neural networks to detect Deepfake videos in 2019. They worked on Deepfake video detection.

[6] F. Sun and others developed a Deepfake detection method in 2021. This method is based on -domain fusion.

[7] A. Aggarwal, M. Mittal and G. Bettini wrote about adversarial networks in 2021. They talked about the theory and applications of adversarial networks.

[8] J. C. Dheeraj and others worked on detecting Deepfakes using Deep Learning in 2021. They used Deep Learning for Deepfakes detection.

[9] M. Li and others found a way to expose Deepfake videos by tracking eye movements in 2020. They used eye movements to detect Deepfakes.

[10] A. Bedale, L. Castelino and J. Gomes used networks to detect Deep Fakes in 2021. They worked on Deep Fake detection using networks.

[11] Y. Li, M. C. Chang and S. Lyu exposed AI-created videos by detecting eye blinking in 2019. They used eye blinking to detect videos.

[12] 12.S. Agarwal and others protected world leaders against Deepfakes in 2019. They worked on protecting world leaders against Deepfakes.

[13] M. Nagao talked about natural language processing and knowledge in 2005.

[14] G. Lee and M. Kim used computer vision to detect Deepfakes in 2021. They used the rate of change, between frames to detect Deepfakes.

[15] D. Pan and others used Deep Learning to detect Deepfakes in 2020. They worked on Deepfake detection through Deep Learning.

[16] S. Fung and others used contrastive learning to detect Deepfakes in 2021. They developed a method called Deepfake.

[17] R. Rafique and others used error level analysis and Deep Learning to detect Deepfakes in 2021. They worked on Deepfake detection using error level analysis and Deep Learning.

[18] G. Jaiswal developed a hybrid recurrent Deep Learning model to detect Deepfake videos in 2021.

[19] M. S. Rana and A. H. Sung used an ensemble-based learning technique to detect Deepfakes in 2020. They developed a method called Deepfake Stack for Deepfake detection.