ASEAN
Journal of Scientific and Technological Reports
Online ISSN:2773-8752

Research article

Predicting E-Commerce Purchase Intention Using Machine Learning

Om Ratna Sheshagiri Gupta Alamuri[1*], and Chaitanya Krishna Bondalapu[1]

[1] Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
* Correspondence: alamurigupta129@gmail.com

Citation:
Alamuri, O.; Bondalapu, C. Predicting e-commerce purchase intention using machine learning. *ASEAN J. Sci. Tech. Report.* **2026**, *29*(1), e260159. https://doi.org/10.55164/ajstr.v29i1.260159.

Article history:
Received: July 7, 2025
Revised: August 30, 2025
Accepted: September 6, 2025
Available online: December 14, 2025

Abstract: The fast-evolving digital commerce environment demands precise predictions of consumer buying intentions to develop personalized experiences and boost conversion rates and user satisfaction on e-commerce platforms. The field has extensively utilized traditional statistical models in conjunction with behavioral theories; however, these methods fail to adapt to high-dimensional, imbalanced session-level data. This learning proposes a machine learning-based approach to predict online purchase intention using the widely recognized Online Shoppers Intention dataset. The methodology involves a reproducible pipeline that integrates data preprocessing, the synthetic minority over-sampling technique (SMOTE) to address class imbalance, chi–square–based feature selection, and a comparative evaluation of multiple classification models. The pipeline was tested on a 70:30 train-test split using stratified sampling to maintain class distribution, and further validated through 10-fold cross-validation to enhance robustness. The Support Vector Classifier (SVC) was found to be the best model in terms of both ROC-AUC and F1 score, achieving an ROC-AUC of 0.886 and an F1 Score of 0.633, thereby efficiently discriminating between purchase and non-purchase sessions. We also explore Random Forest, Decision Tree, and Ridge Classifier models to support a more holistic understanding of performance across a variety of complexity and interpretability levels. Importantly, the research also uncovers important behavioral predictors, including product-page engagement and whether the visitor is a returning one, providing interpretable insights that are consistent with real-world e-commerce practices. These results suggest the potential for implementing machine learning models for real-time behavior forecasting in online retail contexts and show that a data-driven pipeline can add value to traditional behavioral modeling counterparts.

Keywords: Customer purchase intention; e-commerce analysis; ML pipeline; SMOTE; chi-square test.

1. Introduction

The growing dominance of e-commerce has transformed consumer interactions with products and services, as digital platforms now serve as the primary channel for retail transactions. Online shopping environments have become increasingly competitive, so businesses now focus on both user acquisition and converting website visitors into buyers. The ability to understand and predict consumer purchase intentions stands as a vital factor for enhancing user experiences, promoting targeted advertising, and optimizing conversion rates. Numerous research studies have examined the psychological

and behavioral factors that influence online purchasing decisions. The Theory of Planned Behavior (TPB) [11], the Technology Acceptance Model (TAM) [2], and the Stimulus-Organism-Response (SOR) theory [1] are primarily used frameworks to analyze customer decisions, focusing on the analysis of perceived trust, enjoyment, and social influence [4]. The research demonstrates that e-service quality [13], along with live-streaming interactivity [10], gamification [4], and real-time engagement [14], play crucial roles in customer intention. The approaches deliver essential insights into consumer behavior but depend on survey-based models and self-reported data, which may not work effectively in real-time decision environments.

E-commerce data faces structural challenges as one of its primary difficulties. The recording of customer interactions through clickstream events generates high-dimensional, noisy data that exhibits severe class imbalance, as actual purchases occur in only a small fraction of sessions [6,3]. The existing research on customer segmentation and loyalty prediction using machine learning models [7,8] requires further development of end-to-end predictive pipelines that integrate feature engineering with imbalance correction and model evaluation within a reproducible framework. This paper discloses these gaps by developing a machine learning-based model to predict online purchase intention using session-level data from the Online Shoppers Intention dataset. The proposed pipeline combines SMOTE for handling class imbalance with SelectKBest using chi-square scoring for feature selection, and multiple classification models, including Support Vector Classifier (SVC), Random Forest, Ridge Classifier, and others. The research aims to discover the optimal combination of preprocessing methods and classification approaches that produce accurate, interpretable, and scalable purchase predictions.

2. Materials and Methods

The project introduces a comprehensive and reproducible machine learning pipeline for predicting online purchase intention using clickstream data. The pipeline addresses significant issues in current methods for handling class imbalance and high dimensionality, while generating interpretable results. The initial stage of the process involves data preprocessing, which utilizes ordinal and binary encoding for the categorical variables VisitorType and Month. Boolean fields, such as Weekend and Revenue, are converted into binary integers. Additionally, missing and infinite values are handled using imputation to ensure data quality. Next, we implement advanced feature engineering by introducing six behavioral indicators—Total Duration, Total Page Visits, Average Page Duration, Product View Ratio, Information View Ratio, and Returning Visitor—which encapsulate user engagement and browsing behavior. After feature scaling with MinMax normalization, the pipeline applies SMOTE to create synthetic samples of the minority class, thereby addressing class imbalance without compromising data integrity [6,12]. The feature selection process is implemented using the chi-square SelectKBest method to retain the most statistically significant predictors [5,11]. Classifiers are verified by using 10-fold cross-validation. This method is in concordance with prior studies that have compared classifier performance based on the performance in behavior prediction tasks [8,10]. The best performance model is selected for final evaluation on an independent test set. This methodology enhances predictive performance and interpretability by utilizing visualization tools, such as ROC curves and feature importance analysis. The ML pipeline is modular and reproducible, and can be used for both academic research and practical deployment into e-commerce real-time systems [12].

2.1 Process Flow

The proposed machine learning pipeline [12] implements a structured, modular framework to achieve consistency and reproducibility, delivering high predictive performance. The initial stage of data cleaning addresses missing values, inconsistent data types, and infinite values by implementing imputation and replacement methods. Boolean variables (Weekend, Revenue) are converted into binary format, and categorical variables (Month, VisitorType) are encoded appropriately for modeling. Next, feature engineering is applied to create new behavioral indicators, such as Total Duration, Average Page Duration, and Product View Ratio, which capture more profound insights into user browsing patterns. This is followed by class imbalance handling using the Synthetic Minority Oversampling Technique (SMOTE) [6,12], which balances the dataset by generating synthetic examples of minority class sessions. Subsequently, the step involves

applying chi-square ranking to select statistically significant predictors, which helps decrease dimensionality and enhance interpretability. The imbpipeline contains all preprocessing steps, alongside balancing and selection functions, which maintain uniform application throughout the training and testing phases. The validation process for machine learning models includes cross-validation and evaluation methods for comparison. The effectively performing model is selected for future analysis and interpretation, providing actionable insights for real-world e-commerce personalization strategies.

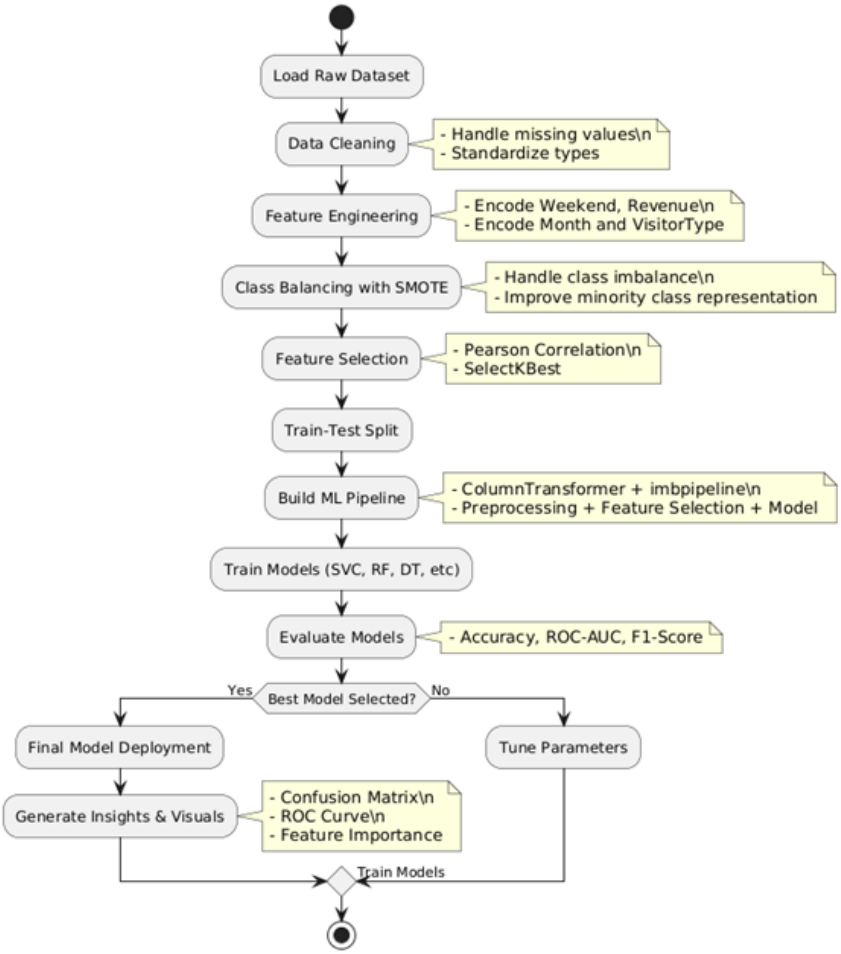

Figure 1. Process flow

2.2 Dataset Description

The dataset used in this paper is the publicly available Online Shoppers Intention Dataset [17], comprising 12,330 records, each representing a complete web browsing session on an e-commerce platform. The dataset contains 18 features, including the number of administrative pages visited, informational pages accessed (Informational), and the corresponding time (Informational_Duration). Additionally, it includes product-related features that typically reflect purchase intention more directly. The session-level engagement becomes visible through behavioral metrics, which include Bounce Rates (percentage of single-page visits), Exit Rates (percentage of users exiting from a given page), Page Values (estimated monetary value of visited pages), and Special Days (a score indicating proximity to a special date, such as holidays or sales events). Technical attributes include session information such as the Month, Operating Systems used, Browser type, Region of the user, and Traffic Type, which indicates the source of the web traffic. The visitor information section includes VisitorType, which identifies whether the user is new or returning, and a Boolean field, Weekend, to denote whether the session occurred on a weekend. The output variable "Revenue" is a Boolean indicating whether a purchase was made, and serves as the label for supervised learning.

Table 1. Features Description

Feature	Description
Administrative	Number of administrative pages visited
Administrative_Duration	Total time (in seconds) spent on administrative pages
Informational	Number of informational pages visited
Informational_Duration	Total time spent on informational pages
ProductRelated	Number of product-related pages visited
ProductRelated_Duration	Total time spent on product-related pages
BounceRates	Average bounce rate of pages visited
ExitRates	Average exit rate of pages visited
PageValues	Average page value of pages visited
SpecialDay	Closeness of session date to a special day (0 to 1)
Month	Month of the visit (January–December, encoded)
OperatingSystems	The operating system used by the visitor
Browser	Web browser used by the visitor
Region	Visitor's geographic region
TrafficType	Source of traffic (e.g., direct, referral, ads)
Weekend	Whether the session was on a weekend (0 = No, 1 = Yes)
Visitor Type	Type of visitor: Returning Visitor, New Visitor, or Other
Revenue	Whether the session ended in a purchase (0 = No, 1 = Yes)

2.3 Data Cleaning

The first step involved data preparation of the Online Shoppers Intention dataset prior to analysis. The Weekend and Revenue boolean columns received binary integer transformations (0 or 1) while VisitorType received encoding for categorical fields. The feature engineering process required division operation value replacement with zero and constant-value imputation for handling missing data. The dataset was made free from inconsistencies, null values, and unprocessable formats through these steps, as these issues commonly cause machine learning model failures and bias. The standardized format created during this phase enabled a seamless transition to advanced processing stages, including feature engineering, modeling, and evaluation. The cleaning process prevented data-related errors from being passed through to subsequent transformations and predictions, which would have compromised the overall reliability of the predictive system.

2.4 Feature Engineering

Here, we conducted extensive feature engineering to enhance model interpretability and gain a deeper understanding of user behaviors. The existing attributes received six new behavioral indicators, which included Total_Duration (the aggregate time spent across all page types), Total_Page_Visits (the total number of pages visited during a session), Avg_Page_Duration (the average time per page), ProductView_Ratio, InfoView_Ratio, and Returning_Visitor (a binary indicator of repeat visits).

$$\text{Total\_Duration} = \text{Admimistrative\_Duration} + \text{Informational\_Duration} + \text{ProductRelated\_Duration} \quad (1)$$

$$\text{Total\_Page\_Visits} = \text{Administrative} + \text{Informational} + \text{ProductRelated} \quad (2)$$

$$\text{Avg\_Page\_Duration} = \frac{\text{Total\_Duration}}{\text{Total\_Page\_Visits}} \quad (3)$$

$$\text{ProductView\_Ratio} = \frac{\text{ProductRelated}}{\text{Total\_Page\_Visits}} \quad (4)$$

$$\text{InfoView\_Ratio} = \frac{\text{Informational}}{\text{Total\_Page\_Visits}} \quad (5)$$

The Month column was ordinally encoded to reflect potential seasonal effects. The new features were designed to detect user engagement and navigation patterns that affect online purchasing decisions. The feature engineering process both increased the dataset size and improved the ability of machine learning models to differentiate between browsing and purchase-intent sessions. The transformations converted intricate user interactions into organized variables, which enhanced both the model's precision and the business value of the extracted knowledge.

2.4.1 Weekend & Revenue Column

The preprocessing phase involved converting the Boolean features 'Weekend' and 'Revenue' from the dataset into a numerical binary format to make them suitable for machine learning algorithms. The initial Boolean values (True, False) in the features needed conversion because most machine learning models require numerical data. The Python replace () method converted True to '1' and False to '0' to solve this issue. The model could understand these features as binary indicators through this transformation because it maintained their semantic meaning while matching the numerical requirements of the feature space. The conversion process prevents type incompatibility problems during model training and enables algorithms to detect the predictive value of binary variables in purchase intention outcomes.

	Administrative	Administrative_Duration	Informational	...	VisitorType	Weekend	Revenue
0	0	0.0	0	...	Returning_Visitor	0	0
1	0	0.0	0	...	Returning_Visitor	0	0
2	0	0.0	0	...	Returning_Visitor	0	0
3	0	0.0	0	...	Returning_Visitor	0	0
4	0	0.0	0	...	Returning_Visitor	1	0

Figure 2. Boolean to Binary Conversion

2.4.2 Visitor_Type Column

The dataset includes a VisitorType column that has a categorical value indicating whether the visitor is a false visitor or a true visitor. A Returning Visitor or a New Visitor started the session. The two groups are mutually exclusive; therefore, there is no overlap, and it is redundant to use both for predictive modeling. Thus, we created a new column, Returning_Visitor, containing the value '1' if the session was from a return visitor and '0' otherwise. This was achieved by storing the data in a new column in the dataframe, minimizing its categorical complexity while preserving the fundamental behavioral difference between repeat and first-time visitors. Writing this variable with a binary coding guarantees that machine learning models can easily understand it.

	Administrative	Administrative_Duration	Informational	...	Weekend	Revenue	Returning_Visitor
0	0	0.0	0	...	0	0	1
1	0	0.0	0	...	0	0	1
2	0	0.0	0	...	0	0	1
3	0	0.0	0	...	0	0	1
4	0	0.0	0	...	1	0	1

Figure 3. Adding Returning_Visitor

2.4.3 Month Column

The dataset contains a 'Month' column that shows the calendar month in which each session occurred. The Month column exists as an object (string) data type by default, which prevents direct processing. The conversion of categorical data into numerical data using Ordinal Encoding, which assigns integer values to each category based on its sequential order. The months received integer values from 0 to 11 through the encoding process, which maintained their chronological order. The sequential nature of ordinal categorical variables makes Ordinal Encoding the better choice than one-hot encoding for nominal categories without inherent ranking. The application of Ordinal Encoding to the Month column maintains sequential relationships between categories, which helps machine learning models detect seasonal patterns and time-based trends that affect purchase intention, while reducing dimensionality compared to one-hot encoding [11-12].

2.4.4 Target Column

The Revenue column serves as the target variable to determine if users completed purchases during their sessions. The default representation of this column uses Boolean values, which indicate True for purchases and False for non-purchases. The majority of machine learning models require numerical representations of targets to perform classification tasks. The replace() method converted the Revenue column into binary format by establishing the following mapping:

$$\text{True} \rightarrow 1, \text{False} \rightarrow 0$$

The transformation preserves the original meaning of the data while converting it into a format suitable for supervised learning models. The binary target enables us to define the problem as a binary classification task, which aims to predict whether a session belongs to the positive class (Revenue = 1) or the negative class (Revenue = 0). The model requires this preprocessing step to correctly interpret labels and achieve optimal accuracy, precision, recall, and ROC-AUC performance metrics.

```
Revenue
0    10422
1     1908
Name: count, dtype: int64
```

Figure 4. Revenue column values count

2.5 Pearson Correlation

The Pearson correlation coefficient finds the relationship between numerical features and the target variable. The Pearson correlation coefficient measures linear relationships between continuous variables by using its r value, which ranges from -1 to 1. The correlation values r = 1, r = -1, and r = 0 indicate a perfect positive linear relationship, a perfect negative linear relationship, and no linear relationship, respectively.

$$\boldsymbol{r} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{X_i} - \overline{\mathbf{X}})(\mathbf{Y_i} - \overline{\mathbf{Y}})}{\sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{X_i} - \overline{\mathbf{X}})^2} \sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{Y_i} - \overline{\mathbf{Y}})^2}} \tag{6}$$

The complete correlation matrix was visualized through a heatmap, which displayed the strength and direction of feature correlations using cell color intensity. The heatmap enabled us to identify strong correlations between features while helping us evaluate potential multicollinearity. A horizontal bar chart

displayed the correlation coefficients between Revenue and each feature in descending order of magnitude. The visualization demonstrated that ProductRelated, PageValues, and BounceRates were the most important predictors of purchase intention, while showing that other attributes had weaker effects. The graphical analyses provided straightforward insights about feature importance, which directed our subsequent feature selection and modeling work.

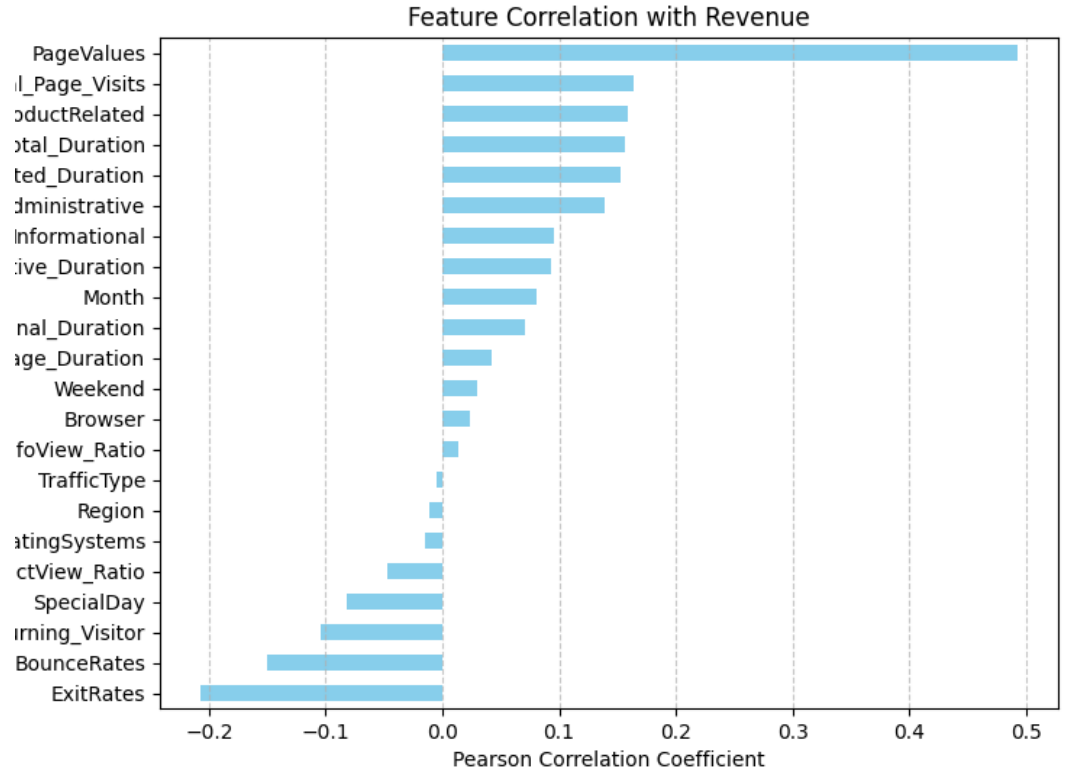

Figure 5. (Bar Plot) Pearson Correlation Coefficient

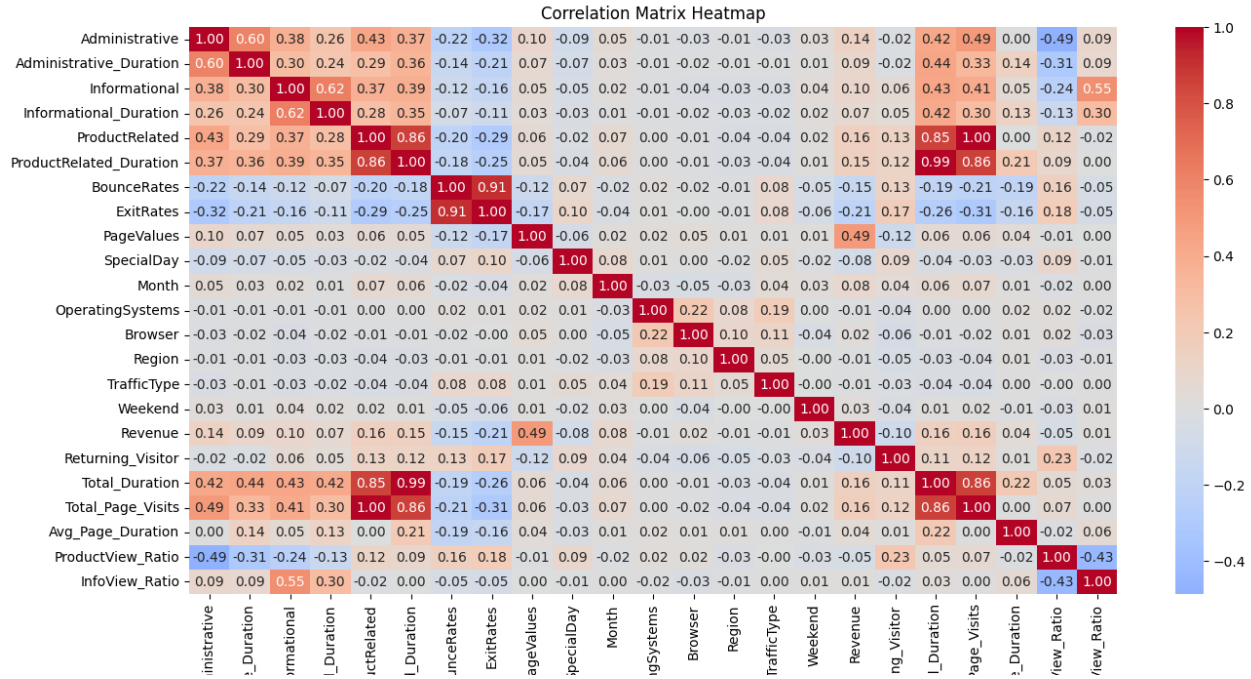

Figure 6. Correlation Matrix Heatmap

2.6 Training & Test Data

The dataset was prepared for supervised machine learning, with the input features (X) and target variable (y). The features of the input were all related variables obtained either by preprocessing or feature engineering. These attributes were related to user actions, session properties, and technical attributes of browsing behavior. The dependent variable was based on the Revenue attribute, indicating whether the user session resulted in a transaction (1) or not (0). A training and testing dataset was then prepared for model development and validation. Dividing the dataset by using the function train_test_split () of the scikit-learn library, where the training and testing sets of the data were split in the ratio 70:30 (70% training, 30% testing). The model trains on a wide range of data through this splitting method while evaluating its performance on an independent subset. Additionally, stratified randomization (stratify=y) was employed to preserve the original class distribution (purchase vs. non-purchase) in both training and testing subsets, thereby preventing skew in class proportions. A constant random state (i.e., 0) was used to ensure the same data partitioning was applied across replicates. This guarantees reproducibility of the findings. This type of splitting allows the model to train on a wide range of the data, while still evaluating it on an entirely independent subset. The 70:30 split was chosen because it offers an optimal balance between providing sufficient data for training complex models while maintaining a large enough independent test set to yield statistically reliable performance estimates. The training set included 8,631 sessions, consisting of 7,295 non-purchase and 1,336 purchase cases. The testing set contained 3,699 sessions, with 3,127 non-purchase and 572 purchase cases.

2.7 Machine Learning Pipeline

The machine learning pipeline utilizes ColumnTransformer and imblearn—pipeline to construct a consistent and reproducible modeling process. The pipeline consisted of five stages, which included imputation, followed by MinMax scaling, then SMOTE oversampling, chi-square-based feature selection, and finally a classifier. The pipeline structure contained all preprocessing steps, which were applied to both training and testing sets to prevent data leaks. The pipeline design included flexible components to enable easy substitution and evaluation of different classifiers. The pipeline design supported hyperparameter tuning and cross-validation operations with automatic reprocessing capabilities. The system design adhered to best practices for machine learning development and deployment, optimizing the entire training and evaluation process for real-world e-commerce implementation. A Machine Learning pipeline workflow is automated that handles a complete processing task. The process combines data transformation with correlation steps into a model structure for output analysis. A standard pipeline consists of raw data input, features, outputs, model parameters, ML models, and Predictions. The pipeline is built with consecutive steps that execute data extraction and pre-processing, as well as model training and deployment, in a modular way for Machine learning. We developed a model pipeline for our project, which employs the ColumnTransformer() to handle data encoding. This replaces any absent values with possible values. It normalizes the data before we input it into the model.

2.7.1 SMOTE

The imbalance of classes is a drawback in this dataset, as only about 15.5% of the sessions were purchase sessions. This class imbalance has the potential to make the model biased towards the minority class, which is less prone to valuable patterns for minorities. To address this, we used the Synthetic Minority Oversampling Technique (SMOTE) instead of simple random oversampling, as SMOTE generates synthetic samples within the feature space rather than merely duplicating minority class samples. This reduces the risk of overfitting in the class imbalance problem. Data balancing SMOTE creates new artificial purchase examples for the minority class by interpolating between existing purchase examples, thereby rebalancing the class distribution without replicating data [6,12]. By doing SMOTE in the normalized feature space, the synthetic samples are realistic and well-mingled. This led to a significant improvement in model performance, particularly in learning from minority purchase sessions, resulting in increased recall and F1-score for the minority class. It also served as a solution to circumvent overfitting to the majority class, a pitfall often associated with straightforward oversampling or underweighting approaches [8]. Alternative methods, such

as ADASYN or Random Oversampling, were considered, but we chose SMOTE for its balance between simplicity and robustness, as supported by previous e-commerce prediction studies.

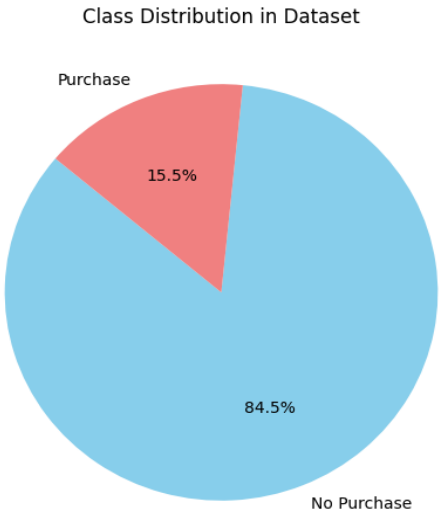

Figure 7. Class Distribution

2.7.2 Imbpipeline

The study utilizes the imbalanced-learn package's imbalanced pipeline module [16] to achieve a robust and reproducible machine learning workflow. The imbalanced pipeline class in scikit-learn extends the standard Pipeline class by integrating checking approaches into its pipeline structure for handling class-imbalanced datasets.

The imbalanced pipeline differs from typical pipelines because it enables the addition of checking techniques, including the Synthetic Minority Oversampling Technique (SMOTE), within each cross-validation fold. The design of this approach restricts oversampling to the training data during model evaluation, thereby precluding data leakage into the test set and maintaining genuine performance criteria.

The imbpipeline receives the following factors during its methodical configuration.

- Preprocessing: Imputation, scaling, and encoding
- Resampling: SMOTE for balancing minority-class samples
- Feature selection: Chi-square–based SelectKBest
- Classifier: Various models (e.g., SVC, Random Forest, Naïve Bayes)

The modular and scalable channel frame promotes thickness across experimental setups. The frame ensures that everything, from preprocessing to final evaluation, will be executed identically. The frame enables unprejudiced model comparisons between different classifiers while enhancing real-world model conception capabilities.

2.7.3 SelectKBest

The SelectKBest algorithm functions as a univariate feature selection method from the scikit-learn library. The algorithm selects the top k labels from the dataset through statistical tests that evaluate the strength of the relationship between each feature and the target variable. The project utilized the ANOVA F-test (f_classif) as the scoring function because the target variable is categorical. In this project, the ANOVA F-test (f_classif) was used as the scoring function, as the target variable is categorical. The F-test measures the strength of association between each feature and the target class by comparing the variance between groups with the variance within groups. SelectKBest was applied after preliminary correlation filtering to reduce feature dimensionality, improve model performance, and prevent overfitting. We used chi-square–based

SelectKBest for final feature ranking and retained the top 6 statistically significant features. By retaining only the top-ranked features, the model was trained on the most statistically significant predictors, including PageValues, ExitRates, ProductRelated_Duration, Returning_Visitor, BounceRates, and Total_Duration.

For dimensionality reduction, we employed the Chi-Square (χ^2) statistical test because most input variables are categorical or non-negative numerical counts (e.g., page visits, durations). Chi-Square effectively measures the independence between each feature and the target variable (Revenue), making it suitable for classification tasks. While methods like mutual information gain or recursive feature elimination (RFE) could also be used, χ^2 was preferred because it is computationally efficient and interpretable, ensuring that only the most statistically significant predictors (e.g., PageValues, ExitRates) are retained.

ANOVA F-test to select the top k features:

$$F = \frac{\text{Variance between groups}}{\text{Variance within groups}} \tag{7}$$

2.8 Select_model

To select the best model, we created a single function that automatically tests all models and finally returns the best model as per our requirements. It is possible to loop over those models and then apply the data through the pipeline for each model. Using cross-validation to get the best performance, reliability, and accuracy of each model. Save the model results to a pandas DataFrame and see the outcome. Finally, select the optimal model having the maximum ROC/AUC score.select_model(X, y, pipeline = None) is a user defined function. This function takes 2 non-default parameters and 1 default parameter:

- X (object): Dataframe containing X_train data
- y (object): Dataframe containing y_train data
- pipeline: Pipeline from model_pipeline()

2.9 Models Evaluated

We evaluated a diverse set of machine learning classifiers within the proposed pipeline, integrating oversampling, feature selection, and classification to identify the approach that best balances accuracy, robustness, and interpretability for real-time e-commerce personalization.

Support Vector Classifier (SVC):

The core prediction engine is an SVC with an RBF (Radial Basis Function) kernel. It identifies the optimal hyperplane that maximizes the margin between classes in a transformed feature space, making them well-suited for high-dimensional problems. We enabled probability estimates (probability=True) to facilitate threshold tuning and business-driven decision-making. The RBF kernel was specifically chosen because it effectively handles non-linear relationships in the data [8].

Random Forest (RF) Classifier:

This algorithm is a collaborative method that constructs several decision trees on bootstrapped samples and combines their outputs for classification. Its inherent feature bagging reduces variance and often yields strong baseline performance on clickstream tasks. We included it as a benchmark against a robust, non-linear learner. In this, the Random Forest was configured with 100 trees (n_estimators=100), ensuring performance stability [6].

Bernoulli Naïve Bayes:

The BernoulliNB classifier models binary features under the assumption of conditional independence. Despite its simplicity, it can perform surprisingly well on sparse or binarized clickstream data [5] and serves as a lightweight baseline in our comparisons.

Ridge Classifier:

The Ridge Classifier uses L2-regularized linear regression for classification tasks by thresholding the

continuous outputs. The L2 penalty controls model complexity [11] and mitigates overfitting, making it particularly useful in high-dimensional feature spaces. The model was applied with its default regularization strength (α = 1.0).

K-Nearest Neighbors (KNN):

KNN predicts a session's class by considering the majority class label among its k neighbors in the feature space. It requires no explicit training phase, making it a practical nonparametric comparator [7], though its performance can degrade in high-dimensional settings. Using the default k = 5 leads to a good balance between bias and variance.

Decision Tree Classifier:

The Decision Tree Classifier recursively partitions the feature space to construct a tree of decision rules. Its intuitive, rule-based structure offers clear interpretability, but single trees are prone to overfitting [12], making it valuable for illustrating the trade-off between simplicity and predictive power. The model was implemented using the Gini index as the criterion for splitting.

2.10 Evaluation Metrics

To comprehensively measure model performance, we employed a set of well-established evaluation metrics, particularly suited for imbalanced binary classification tasks, such as purchase intention prediction.

Accuracy

Accuracy calculates the overall proportion of rightly classified cases (both purchase and non-purchase). While useful, accuracy alone can be misleading in imbalanced datasets because high accuracy may be affected by prognosticating only the dominant class.

$$\text{Auc} = \frac{\text{T\_P} + \text{T\_N}}{\text{F\_P} + \text{F\_N} * \text{T\_P} + \text{T\_N}} \tag{8}$$

where T_P = true positives, T_N = true negatives, F_P = false positives, and F_N = false negatives

Precision

Precision quantifies the part of predicted positive cases (purchases) that are actually correct. High precision diminishes the possibility of false positives, which is vital when targeting marketing resources.

$$\text{Precision} = \frac{\text{T\_P}}{\text{T\_P} + \text{F\_P}} \tag{9}$$

Recall

Recall measures the part of actual positive cases (purchases) that the model successfully identifies. More recall is critical to ensure that most potential purchasers are correctly flagged, even if it means incurring some false alarms.

$$\text{Recall} = \frac{\text{T\_P}}{\text{T\_P} + \text{F\_N}} \tag{10}$$

F1 Score

F1 measures the harmonic mean of precision and recall, which allows a balanced assessment when both false positives and false negatives are important. It is especially enlightening in unbalanced datasets.

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

ROC-AUC (Receiver Operating Characteristic – Area Under the Curve)

ROC-AUC estimates the model's capability to differentiate among classes at various decision thresholds. It calculates the area under the ROC curve, which plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold levels.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\text{dFPR} \tag{12}$$

By combining these metrics, we gained a nuanced understanding of each model's strengths and weaknesses, particularly in relation to the minority (purchase) class, which is the key target for actionable e-commerce insights.

3. Results and Discussion

3.1 Model Evaluation and Comparative Performance

This pipeline was applied to six classifiers (SVC, Random Forest, Decision Tree, Ridge Classifier, Bernoulli Naïve Bayes, and KNN) to obtain a better model for online purchase intention prediction. Among those, SVC was the most robust model with the highest ROC-AUC (0.886) and F1-score (0.633), indicating its reasonable and balanced discrimination and good performance in balancing the imbalanced data. The Random Forest model came second, with a good ROC-AUC score of 0.882, while Decision Tree, Ridge Classifier, BernoulliNB, and KNN performed in the middle and served as reasonable comparisons. The performance of the models was evaluated using five performance metrics: accuracy, precision, recall, F1-score, and ROC-AUC. These collectively presented an equanimous understanding of predictivity, particularly in imbalanced binary classifications. Support Vector Classification achieves the highest accuracy (88.3%) and ROC-AUC (0.886) for the task of distinguishing between purchase and non-purchase sessions. It also showed a balanced trade-off between minimizing false positives and capturing true purchase signals with an F1-score of 0.633. Random Forest and Decision Tree worked stably on non-linear patterns, while Ridge, Naïve Bayes, and KNN models performed similarly, as they had a limited ability to capture the minority class.

Table 2. Performance comparison of all classifiers

Classifier	ROC-AUC	Accuracy	Precision	Recall	F1-Score
SVC(RBF)	0.886	0.883	0.619	0.654	0.633
Random Forest	0.882	0.872	0.567	0.739	0.641
BenouliNB	0.852	0.872	0.561	0.810	0.663
KNN	0.836	0.853	0.520	0.696	0.594
Decision Tree	0.727	0.852	0.520	0.585	0.550
Ridge Classifier	0.851	0.819	0.449	0.700	0.546

[1]SVC - Support Vector Classifier, [2]RBF - Radial Basis Function, [3]KNN - K-Nearest Neighbors

SVC has the capacity to represent non-linear decision boundaries and exhibits robustness in high-dimensional feature spaces, facilitating generalization across a variety of sessions. Consequently, it was the most dependable model in predicting online purchase intentions. In contrast, the Random Forest (RF) classifier demonstrated high recall capabilities due to its ensemble learning method of aggregating multiple decision trees. This features a trade-off; although RF could handle a greater number of positive purchase cases, precision was sacrificed, resulting in more false positives (FP). Therefore, we consider RF to be a beneficial alternative model in cases where recall is more important than precision (e.g., maximizing potential buyer discovery in target marketing). The Bernoulli Naïve Bayes (BernoulliNB) classifier with the maximum recall (0.81) further demonstrates its ability to handle binary-like session information. However, due to the independence assumption, a sharp decrease in the precision value was observed as well as a significant deterioration in the overall F1-score (95%), which hindered its usefulness as an individual predictive predictor. Other classifiers, such as Decision Tree, Ridge Classifier, and K-Nearest Neighbor (KNN) appear to be worse.

Decision Trees overfitted, Ridge Classifier was limited by its linearity, and KNN failed under high-dimensionality (the curse of dimensionality). These results further highlight the importance of being able to learn non-linear patterns. Altogether, results show SVC performs the best on average according to the tradeoff of accuracy, ROC-AUC, and F1-score, so that it is the optimal choice for prediction.

3.2 Feature Relevance and Interpretability

Feature importance analysis, conducted using the chi-square test and Pearson correlation, revealed that PageValues, ProductRelated, and ExitRates were the most significant predictors of purchase behavior. This aligns with behavioral theories and past literature [11, 7], emphasizing that high product page engagement and exit patterns near checkout stages are strong indicators of purchase intent. Visual heatmaps and bar plots confirmed the statistical significance of these features in distinguishing between buyer sessions and non-buyers, validating the design of the feature engineering process. Figure 8 presents the chi–square–based feature importance rankings, where behavioral features, such as Returning_Visitor, PageValues, and ProductRelated_Duration, consistently ranked highest.

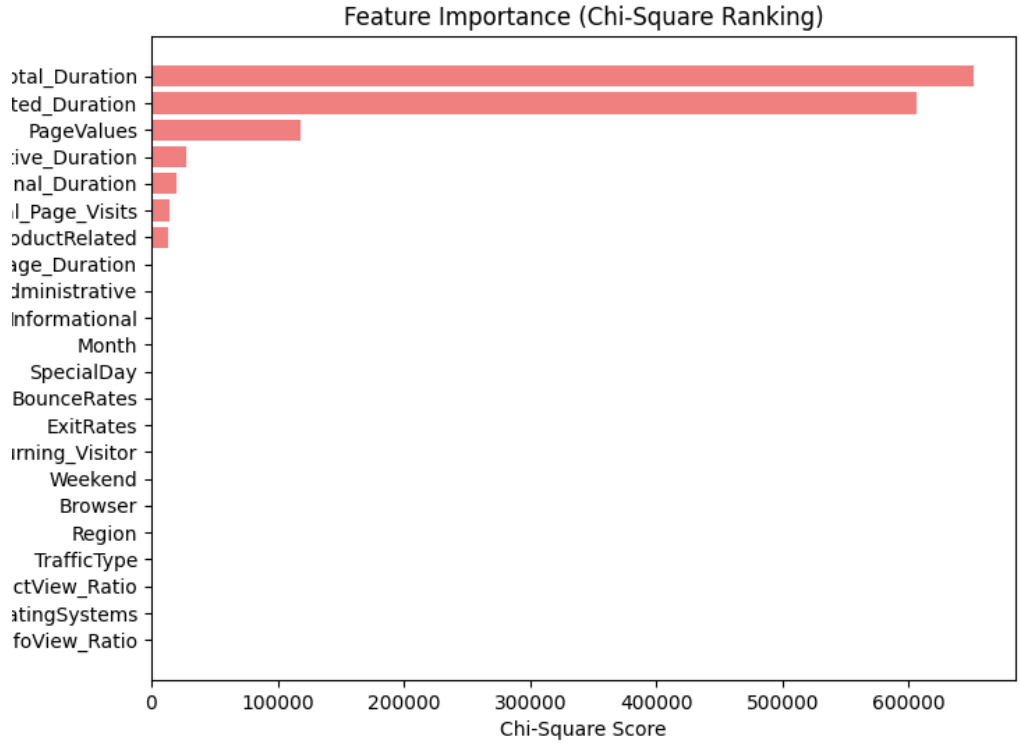

Figure 8. Feature importance (chi-square ranking)

3.3 Visual Evaluation of Model Outputs

To support quantitative metrics, Figure 9 displays ROC curves for all classifiers, providing a visual interpretation of each model's ability to trade off true positive and false positive rates over various thresholds. Figure 10 illustrates the confusion matrices, showing classification accuracy and class-specific errors for both majority and minority classes. These visuals confirm the SVC's superior discrimination, as well as the relative assets and faintness of the remaining models.

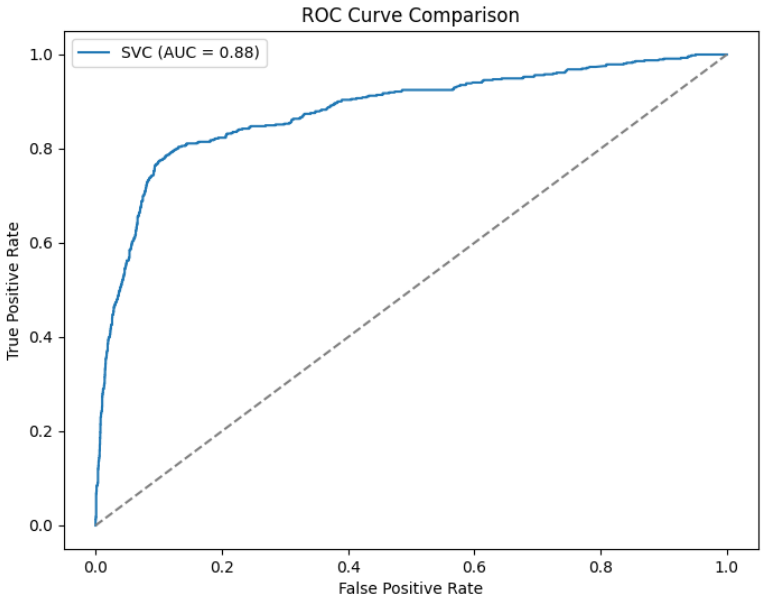

Figure 9. Roc Curve Comparison

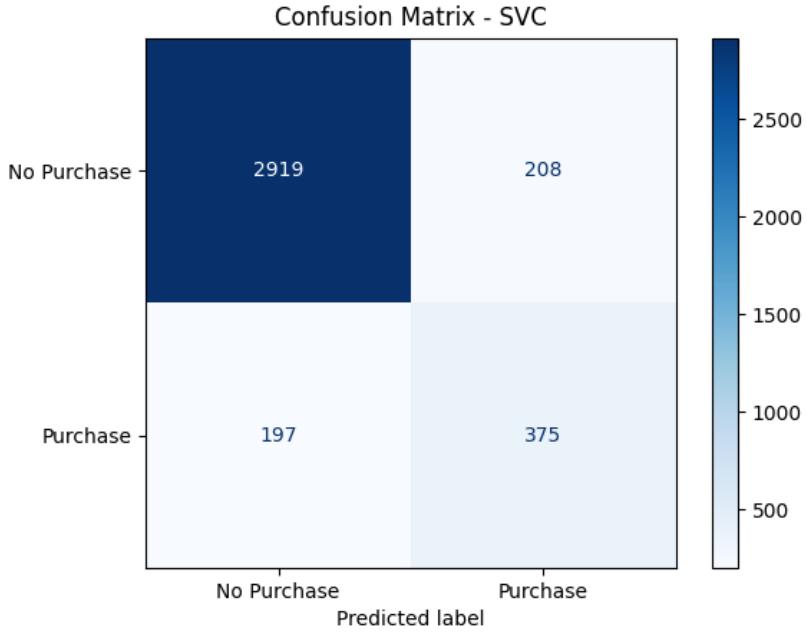

Figure 10. Confusion Matrix

3.4 Insights into User Behavior

Model interpretation revealed that returning visitors had a higher likelihood of completing purchases, particularly during weekends and near special days. Sessions characterized by greater product interaction time (Total_Duration), lower bounce rates, and higher PageValues were more likely to convert. These insights corroborate findings from prior studies in live-streaming and gamified commerce platforms [1, 10], where user attention and engagement depth emerged as key purchase drivers.

3.5 Implications and Strategic Value

The pipeline is useful in real-time marketing and personalization for e-commerce operations. The platform enables businesses to concentrate on purchase-intent sessions and serve personalized offers and optimal product display sequences based on predictive behavior modeling. The modularity of the pipeline

enables businesses to implement it on other platforms with minimal modifications, resulting in higher scalability and practical applicability.

3.6 Comparative Analysis with Existing Studies

The Support Vector Classifier (SVC) performs best for online purchase intention prediction, with an F1-score of 0.633 and an ROC-AUC of 0.886. The results align with earlier research, which suggests that SVM-based models outperform linear classifiers because they can detect non-linear patterns in large e-commerce datasets [5], [8]. These research findings gain additional strength through the comparison with earlier studies. The predictive accuracy of Logistic Regression and Decision Trees was lower than that of the current study because these models failed to detect non-linear patterns in the data [3]. The results align with our findings, as Decision Trees yielded lower F1-scores due to overfitting. The research conducted by [7] analyzed Naïve Bayes for purchase prediction because this model performs well with sparse binary data. The results confirm that BernoulliNB produced the best recall at 0.81; however, its precision remained low, which restricted the overall performance. Recent research has focused on developing ensemble approaches and deep learning algorithms. The purchaser detection results from Random Forest and Gradient Boosting showed improved recall rates [12], which aligns with our findings that Random Forest outperformed SVC in recall, but SVC maintained better F1 and ROC-AUC scores. The sequential data analysis capabilities of deep learning models, such as LSTMs, in clickstream data [14], come at a high computational expense that makes them impractical for real-time prediction compared to SVC. The evidence shows that non-linear learners (SVC and Random Forest) perform better than linear approaches in purchase intention prediction, but their performance depends on the characteristics of the dataset and the effectiveness of feature engineering and balancing techniques like SMOTE.

3.7 Limitations and Future Directions

The proposed machine learning pipeline performs effectively, addressing challenges such as class imbalance and high dimensionality. However, it has a limitation in the Online Shoppers Intention dataset, which represents user interactions as static, session-based records. By treating sessions as static, this method disregards the temporal patterns in consumer behavior, which play a vital role in shaping purchasing decisions. Despite the good performance, the current pipeline presents both training as a whole and user interactions as static, and they do not interact with each other. One possible future direction is to incorporate temporal or sequential modeling (e.g., RNNs or transformers) in order to capture changing contexts within a session. Moreover, the presentation of explainable AI methods (such as SHAP or LIME) may empower model transparency and facilitate the interpretation of the model by business end-users. Together, these directions open pathways toward building more dynamic, interpretable, and business-oriented predictive systems.

4. Conclusions

In this paper, a robust and interpretable pipeline for machine learning to predict online purchase intention was suggested, with which a part of the behavioral click stream data (from the Online Shoppers Intention repository) was employed. The study addressed key problems in this realm, including the class imbalance problem, high dimensionality, and the requirement for interpretability in real-time decision systems. By implementing a series of preprocessing, feature engineering, SMOTE-based class balancing, and chi-square feature selection, this study designed a scalable workflow that was tested on six popular classifiers. The hyper-SVC was the top-performing model, with an ROC-AUC of 0.886, and the accuracy for the best-performing model was 88.3%, which is sufficient for distinguishing between purchasing and non-purchasing sessions. PageValues, ProductRelated duration, and Returning_Visitor variables were among the key behavioral indicators that were highly associated with purchasing. Not only did the pipeline retain predictive ability, but it also allowed for the clear interpretation of feature relevance through statistical and visual examination. The contribution of the study is in showing how scalable and reproducible ML pipelines can be used to improve e-commerce personalization applied to a highly imbalanced dataset. It also emphasizes the importance of preprocessing choices, model selection, and interpretability in practical applications. However, the model assumes user behavior to be static and session-based, failing to account for temporal sequence and

cross-session behavior. A natural next step in research would be to apply sequence-based deep learning techniques, such as recurrent or attention-based mechanisms, for modeling the evolution of intent. Furthermore, the inclusion of XAI techniques can also provide an additional level of transparency and trust in AP for personalization systems.

5. Acknowledgements

Author Contributions: Om Ratna Sheshagiri Gupta Alamuri led the conceptualization of the research problem and the development of the methodology. He designed and implemented the data preprocessing and feature engineering pipeline, employing SMOTE for class imbalance correction and Chi-square–based feature selection. He executed the complete machine learning workflow, including model training with classifiers such as Support Vector Classifier (SVC), Random Forest, and Ridge Classifier. Additionally, he conducted thorough model evaluations using ROC-AUC, F1-score, and cross-validation metrics. He performed detailed experimental analyses, generated all visual representations and tables—including process flows, ROC curves, and confusion matrices—interpreted behavioural insights such as the significance of Page Values and Returning Visitor metrics, and prepared the initial manuscript draft.

Chaitanya Krishna Bondalapu provided expert supervision and strategic oversight, contributing to methodological enhancements, validation of analytical results, and ensuring alignment with industry standards in e-commerce behavioral analytics. He critically reviewed and substantively edited the manuscript to enhance technical accuracy, clarity, and coherence. Furthermore, he offered guidance on comparative model assessments, interpretability techniques including Pearson correlation and feature importance analyses, as well as implications for real-time personalization frameworks. His involvement was instrumental in maintaining the rigor, reproducibility, and integrity of the research, while also strengthening the discussion on study limitations and prospective future work, including sequential modelling approaches.

All authors have reviewed and approved the final version of the manuscript.

Funding: This study was conducted without any specific financial support from public, commercial, or not-for-profit funding agencies. No funding sources influenced the research procedures or outcomes.

Conflicts of Interest: The authors declare no conflicts of interest regarding the research, authorship, or publication of this manuscript. All authors have approved the final submission.

References

[1] Li, Q.; Zhao, C.; Cheng, R. How the Characteristics of Live-Streaming Environment Affect Consumer Purchase Intention: The Mediating Role of Presence and Perceived Trust. *IEEE Access* **2023**, *11*, 123977–123987. https://doi.org/10.1109/ACCESS.2023.3330324

[2] Mensah, I. K. The Factors Driving the Consumer Purchasing Intentions in Social Commerce. *IEEE Access* **2022**, *10*, 132332–132344. https://doi.org/10.1109/ACCESS.2022.3230629

[3] Kumar, V.; Preeti; Saheb, S. S.; Kumari, S.; Pathak, K.; Chandel, J. K.; Varshney, N.; Kumar, A. A PLS-SEM Based Approach: Analyzing Generation Z Purchase Intention through Facebook's Big Data. *Big Data Min. Anal.* **2023**, *6*(4), 491–503. https://doi.org/10.26599/BDMA.2022.9020033

[4] Lee, H. Interest-Based E-Commerce and Users' Purchase Intention on Social Network Platforms. *IEEE Access* **2024**, *12*, 87451–87462. https://doi.org/10.1109/ACCESS.2024.3417440

[5] Ye, L.; Zhang, H.; Fei, Z. The Impact of Sales Promotion on the C2C Online Purchasing Behavior: An Empirical Study. In *Proceedings of the 2010 International Conference on E-Business and E-Government*; IEEE: Guangzhou, China, **2010**; pp 2261–2264. https://doi.org/10.1109/ICEE.2010.571

[6] Nikulin, V. On the Method for Data Streams Aggregation to Predict Shoppers Loyalty. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, **2015**; pp 1–8. https://doi.org/10.1109/IJCNN.2015.7280493

[7] Liao, J.; Jantan, A.; Ruan, Y.; Zhou, C. Multi-behavior RFM model based on improved SOM neural network algorithm for customer segmentation. *IEEE Access* **2022**, *10*, 122501–122514. https://doi.org/10.1109/ACCESS.2022.3223361

[8] Kim, K.; Jo, M.; Ra, I.; Park, S. RFMVDA: An enhanced deep learning approach for customer behavior classification in e-commerce environments. *IEEE Access* **2025**, *13*, 12527–12539. https://doi.org/10.1109/ACCESS.2025.3529023

[9] Lin, Y.-T. J.; Chang, C.-Y.; Cheng, S.-Y.; Lin, M.-Y. T. Probabilistic Customer Purchase Evolution Graph. *IEEE Access* **2023**, *11*, 32962–32976. https://doi.org/10.1109/ACCESS.2023.3263729

[10] Chang, S. E.; Yu, C. Exploring gamification for live-streaming shopping—Influence of reward, competition, presence and immersion on purchase intention. *IEEE Access* **2023**, *11*, 57503–57515. https://doi.org/10.1109/ACCESS.2023.3284033

[11] Bhati, N. S.; Vijayvargy, L.; Pandey, A. Role of e-service quality (E-SQ) on customers' online buying intention: An extended theory of planned behavior. *IEEE Access* **2022**, *10*, 77337–77349. https://doi.org/10.1109/ACCESS.2022.3190637

[12] Dhanushkodi, K.; Bala, A.; Kodipyaka, N.; Shreyas, V. Customer behavior analysis and predictive modeling in supermarket retail: A comprehensive data mining approach. *IEEE Access* **2025**, *13*, 2945–2959. https://doi.org/10.1109/ACCESS.2024.3407151

[13] Ramanathan, U.; Williams, N. L.; Zhang, M.; Sa-nguanjin, P.; Garza-Reyes, J. A.; Borges, L. A. A new perspective of e-trust in the era of social media: Insights from customer satisfaction data. *IEEE Trans. Eng. Manag.* **2022**, *69*(4), 1417–1429. https://doi.org/10.1109/TEM.2020.2985379

[14] Tiffany, P.; Pinem, A. A.; Hidayanto, A. N.; Kurnia, S. Gain-loss framing: Comparing the push notification message to increase purchase intention in e-marketplace mobile application. *IEEE Access* **2020**, *8*, 182550–182562. https://doi.org/10.1109/ACCESS.2020.3029112

[15] Liu, X.; Zhou, B.; Du, R.; Qi, W.; Li, Z.; Wang, J. On evolutionary analysis of customer purchasing behavior by the supervision of e-commerce platforms. *IEEE Trans. Comput. Soc. Syst.* **2025**, *12*(1), 38–51. https://doi.org/10.1109/TCSS.2024.3485959

[16] Lemaître, G.; Nogueira, F.; Aridas, C. K. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **2017**, *18*(17), 1–5. http://jmlr.org/papers/v18/16-365.html

[17] Kumar, A. *Online Shoppers Purchasing Intention Dataset*. Kaggle, **2022**. https://www.kaggle.com/datasets/imakash3011/online-shoppers-purchasing-intention-dataset (accessed 2025-12-05).